Investigating the Use of Distributional Semantic Models for Co-Hyponym Identification in Special Corpora

by

Behrang QasemiZadeh
M.Eng. (AI & Robotics), B.Eng. (Software)

Submitted to the National University of Ireland, Galway
in fulfilment of the requirements for the degree of Doctor of Philosophy

Insight Centre for Data Analytics
National University of Ireland, Galway

July, 2015

This page is intentionally left blank.

Doctoral Dissertation

INSIGHT Centre for Data Analytics, NUI Galway.

This publication has emanated from research conducted with the financial support of the Science Foundation Ireland under Grant Number SFI/12/RC/2289 and SFI/08/CE/I1380.

Typeset by the author using LaTeX.

This page is intentionally left blank.

Declaration

I declare that this thesis is composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification.

Schoenleitnerweg, July 2015
Behrang QasemiZadeh

This page is intentionally left blank.

Investigating the Use of Distributional Semantic Models for Co-Hyponym Identification in Special Corpora

by Behrang QasemiZadeh

Abstract

Knowledge is assumed by cognitive science to consist of concepts that are organised and maintained by complex processes taking place in human minds. These processes are not yet accessible directly. Language is *still* the primary medium for communicating knowledge and presumably linguistic objects and structures are expressions of knowledge and its organisation in mind. Collecting terms (i.e., creating a specialised vocabulary) and capturing their relationships are thus important mechanisms for distilling knowledge from specialised texts and for formalising it for machines. The approach taken in this thesis is to analyse the co-hyponymy relationships between terms as an organisational mechanism.

Co-hyponyms are sets of lexical units sharing a common hypernym; *bank* and *building society*, for example, are co-hyponyms of the hypernym *financial organisation*. Analysing the co-hyponymy relationships between terms is important because it bridges the semantic gap between a) specialised lexical knowledge, b) the quantitative interpretation of meanings in specialised discourse, and c) machine-accessible conceptualisation of knowledge. This thesis proposes the use of a vector-based distributional representation of terms in order to construct a quantitative conceptual model of kinds-sorts in a given field of knowledge.

Among empirical methods for analysing linguistic structures, distributional approaches to semantics encode language data to models that should correspond to the meanings of linguistic entities. The meaning of an entity, such as a word or a phrase, is assumed to be a *function* of its statistical distribution in *contexts*. In order to use these methods we thus need to define (a) the contexts, that is, which statistical information must be collected; and (b) the functions, that is, how this information must be used to correlate with a meaning. This thesis is a study of corpus-based distributional methods for characterising co-hyponymy between terms.

Terms are represented as vectors to form a so-called term-space model. To obviate the curse of dimensionality and to facilitate the construction of models, novel methods employing sparse random projections are proposed. Random Manhattan indexing is used to construct ℓ_1-normed spaces and random indexing for ℓ_2-normed spaces. Following these steps a memory-based classifier exploits the distance between vectors to identify the presence of targeted co-hyponymy relationships. An evaluation is also performed to assess any reciprocal influences of the method's parameters on its performance. User-friendliness, flexibility in updating and maintenance, and an innate capacity to resemble conceptual structures in a domain knowledge are the advantages of this method.

This page is intentionally left blank.

O snail
Climb Mount Fuji,
But slowly, slowly!

Kobayashi Issa (1763–1828).

This page is intentionally left blank.

Acknowledgements

In the last two years that I have been writing this thesis, a few colleagues of mine (who also did a doctoral study) reminded me repeatedly that no one was going to read it anyway. They definitely meant well: Let go of it; do not be obsessed with this document; and, spend time on more *valuable*[1] activities.[2] These reminders provoked a set of questions in my mind.[3] In the beginning, I was thinking of the possible *vanity* of doing a doctoral study: writing a thesis that no one would read![4] What would be the value of a thesis (or any communicative artefact) if no one were going to read (or use) it? I asked myself. Am I seeking to inflate my ego? Or, is this another vanity (see also Figure 1) as a remedy for vanity? Eventually, out of these questions, the process of writing a thesis became a *koan*.[5] In turn, that taught me a big lesson I tend to forget: Enjoy the process, that is, every single moment of what you do. Thereafter was a great comfort as I found myself genuinely enjoying the process of writing, melding words into each other, one at a time, to read it (at least) by myself. This is an attempt to explain the real motive behind writing the acknowledgements.[6]

Here I am, sitting down to finish this document by writing the acknowledgements section, feeling neither empty nor fulfilled but pre-occupied with the fact that life, and every process that it encompasses such as doing a doctoral study, is too short and miraculous. Such being the case, I ask myself who should I thank then? I decided to start giving my thanks to those colleagues of mine who reminded me of the possible vanity of this process on the basis that no one supposedly may ever read this document. Their reminders made me think more often about why I was doing this and how it could possibly change anything in this world except myself.[7]

In turn, I wish to thank Adam Kilgarriff who simply set me on this path. From time to time, I used to get career advice from Adam.[8] It was by his advice and recommendation that I applied for a position at the natural language processing group of Paul Buitelaar at DERI.[9] I remember Paul and Conor Hayes interviewing me one Friday afternoon and Paul offering me a position in DERI afterwards (and, how happy I was). So, next I want

[1] Valuable is perhaps one of those words that aggregates all the complexities of natural language and its relationship to knowledge and our belief system.

[2] In this context, I guess they meant writing grant proposals to acquire *money*.

[3] I have heard almost all doctoral students struggle with these questions at a stage in their study.

[4] 'Vanity of vanities, saith Qoheleth; vanity of vanities; all is vanity (Ecclesiastes)'.

[5] 'A paradoxical anecdote or riddle without a solution, used in Zen Buddhism to demonstrate the inadequacy of logical reasoning and provoke enlightenment (Koan, 2015)'.

[6] Do not blame me writing such things in the acknowledgements. I was afraid I may never find another chance to say these words. There are many *thank you's* to follow.

[7] This does not imply that I do not wish this study cause changes outside myself, such as contributing to the body of knowledge; and, that many people will read this thesis, and they will benefit from it; and, indeed, I use it to boost my ego and feel a kind of 'joy' that comes with it.

[8] Those who knew Adam admire him for delivering generous, wise, and responsive advice; even during his battle with cancer.

[9] As put by Adam, a nice and knowledgeable person to work with and a good place to be in.

Figure 1: 'For the Love of God' (Hirst, 2007): A platinum cast of an 18th-century skull decorated using a number of teeth and a lot of diamonds (very expensive ones).

to thank Paul for letting this journey begin by accepting me in his group and for his trust. Although he was not part of the thesis committee of mine, the many regular discussions that we had (in about a year and half that I worked in his group) influenced the formation of the ideas behind this thesis. Particularly, now that I recall events during my study, I see that he was[1] supportive and patient, and provided me with a number of opportunities such as visiting to Shanghai Jiao Tong University.

The next 'thank you' goes to my supervisor: Siegfried Handschuh. First of all, I would like to thank him for his absolute trust in me. During the time that he supervised my thesis, he never doubted, questioned, or asked anything about my thesis. He was also very supportive and certainly without the opportunity he gave me, I would not be able to be here, where I am now. From him, I learned much about the role of discourse in scientific writing and communication, as well as many essential research skills. In Siegfried's team, I must name Brian Davis. He thought me a lot about the importance of interpersonal relationships and the undeniable role of power and social networking within an organization. Brian is a knowledgeable person and I enjoyed his company many evenings that we talked about what constitutes a *normal* person, and what is the *reality* of our experiences in life? Needless to say, the careful review of a chapter of this thesis he provided proved to be invaluable.

As I started to write this thesis, I understood immediately that I needed to improve my writing skills. Miechelle van Kampen has been a real help since then. She patiently read my writings and provided me with valuable comments and carefully suggestions to improve my English skills.[2] The second person is Carla. Apart from reading my messy

[1] He still is.

[2] I still need to. Writing is like running: practice is essential to develop and maintain it (specially in L2).

drafts, Carla's persistent love and support changed my life forever. She even motivated me to lose some weight and get in shape during the process of writing, which resulted in learning about the concept *one step at a time* or *one word at a time*. I would also like to thank Eric van Lente. He reviewed a chapter of my thesis. I thank him not only for providing me with honest questions about the validity of my research, but also for the fruitful discussions we had about the *illusion of the self*, ego, and the alter ego. I must thank Nick Kennedy for reviewing parts of this document, too. Lastly, I would like to thank Anne Schumann for correcting me in several occasions; I have learned much from her during the short time that I have known her.

I would like to sincerely thank the examination committee of my thesis: Conor Hayes, who chaired the session, Christopher Brewster, the external examiner, and Stefan Decker, the internal examiner. Having Stefan and Christopher on board was a true advantage: Stefan, a logician who can translate everything to logic with an unbelievable performance, and Christopher, whose vast knowledge of linguistic and philosophy still bewilders me, not to say his *humbleness*. Their comments and questions as well as the subsequent discussion we had gave me a whole new perspective on my research. What I have learned from them is going to be a valuable asset in my future research. There is no doubt that meanings (i.e., concepts) and then knowledge are born only as a result of constructive dialogue.[1] I wish I will have another chance to have these two brilliant scientists under the same roof discussing my research. Particularly, I would like to thank Christopher for detailed comments as well as bringing a number of important references to my attention. Apart from examining my thesis, I would like to thank Stefan as the director of DERI. He made DERI the place to be in. During some not so easy days that I had, he was kind and supportive like every other staff in DERI.

Speaking of DERI Galway, which nowadays feels like a second home, I would like to thank every single staff of this place: Brian Wall, Maria Smyth, Carmel Fennell, Michelle Treacy, Hilda Fitzpatrick, Claire Browne, and Gerard Conneely (in order of their offices in the building) specially. They are simply the best 'Irish role models', people who can best present the rich Irish culture. I particularly learned that I can have a bit of *craic*, do small talks, smile, have a cup of tea with my colleagues, be relaxed and not too stiff about timing and structures; but still be productive, deliver everything on time, and fully respect rules and regulations. By all means, that was the common culture in the broader community of NUIG. Here, I would like to express my gratitude to the health unit, disability services, and student counselling unit of NUIG. I congratulate them all for the inspiring micro-society they have created at the far west coast of Europe.

During my study, I have made many friends whose presence in my life is heartening. Conor MaGuire has specially been a true friend and really supportive. He and his family made Galway more like home and I really enjoyed their company. There are many more: Laleh (a.k.a the killer of the black fish), Deidre, Ismael, Simon, Nuno, Vit, Ali,

[1] One with another or himself.

Ramisa, Tobias, Karthik, Pradeep, and Max Fisher. I thank Laleh again for sorting our some of the paperworks for me. Special thank also goes to my friend Wilson McMuller. He was really helpful in a few critical occasions (apart from helping me to move from Belfast to Galway). At this point, I also like to thank my parents who pushed me to leave the safety of my own country to understand how big this world is and, of course, to get to know these wonderful people.

There is also a list of scientists that I have not met yet, but I would like to thank them for their advice. I would like to thank Peter Turney for patient replies to my emails and questions and the generous advice and answers he gave me. The same goes for Marco Baroni and other *distributionalists*, such as Magnus Sahlgren, Katrin Erk, and Alessandro Lenci; also, Marie L'Homme for being approachable and providing me with a generous review of a paper as well as sending me copies of her own publications whenever I asked for it. Thanks go to Thierry Hamon who gave me valuable advice, too.

Apart from their hospitality,[1] I am grateful to the Irish tax payers who enabled the Science Foundation Ireland to provide me with a generous scholarship. This thesis was finalised when I was a research assistant at the University of Passau. I would like to extend my gratitude to the University of Passau, particularly the chair of digital libraries and web information systems.

I have been often warned to get ready for questions such as *what was the major outcome of this study?*. The short answer, that I can give here, is that now I know how little I know about my research topic. Now, I understand how young, ambitious, and naïve I was when I started my study. I am still relatively young and ambitious, and perhaps, still very naïve. However, at the time that I started my study, I was ignorant towards these facts. The major outcome of this study is that now I am more aware of the ignorant nature of my existence. I have learned a) how much I do not know, b) I am more mindful about this fact, and c) not to be afraid of this but happy as it is the beginning of a possible new beginning; and, d) of course, thinking in a more structured way (a) (b) (c) ...!

Finally,

It's been proven by history: all mankind makes mistakes,

as *Captain Sharp* said once (Moonrise Kingdom, 2012). If you find mistakes of any kind (e.g., you spot misspelled words, or think that your name is missing from this section), *pleasae* let me know.

[1] I leaned from the Irish to say 'welcome' and 'welcome back' any time I see a visitor in my town.

To Adam Kilgarriff (1960–2015)

as, in many *senses*, I share his belief about *word senses*.

This page is intentionally left blank.

Table of Contents

List of Figures

This page is intentionally left blank.

List of Tables

Part I

Prologue

Chapter 1

Introduction

In the Beginning, Was the Word ...

1.1 Motivation

Directly accessing human thoughts and transferring the knowledge they possess to machines is still far beyond the reach of technology.[1] Language—and thus text—is still the main vehicle for knowledge dissemination. An ever-increasing amount of text data in our digital era manifests the fluid nature of knowledge and its rapid growth. However, capturing knowledge from text and representing it in a machine-accessible format is a tedious and time-consuming problem. Since the early days of commercial computers, this has resulted in difficulties in developing knowledge-based systems—as is still best described by the term *knowledge acquisition bottleneck* coined by Feigenbaum (1980).

Automated text analysis techniques have thus been developed to facilitate the process of *knowledge acquisition* from text and to improve the productivity of knowledge workers.[2] Evidently, the development of these methods has evolved into several multidisciplinary research areas. In these research, the study of knowledge and its relationship to language is a common theme. *Concepts* are often seen as the constituents of knowledge; disputes about their nature, structure, and relationship to language and linguistic communication, however, have led to different ways of formulating research questions in these studies.[3] Disregarding these differences, the essence of the problem has remained the same: bridging the *semantic gap* between text and machine-accessible knowledge structures (see Brewster, 2008, chap. 2 for a thorough perspective).

In the study of language structure and its relationship with knowledge, much attention has been paid to lexical units known as *terms*. Human knowledge is an expression of a plurality of domains of knowledge. In each domain, terms constitute a *specialised vocabulary* to communicate knowledge.[4] Since concepts are abstract mental objects that cannot be sensed, terms are often seen as labels to access salient concepts in a domain knowledge (L'Homme and Bernier-Colborne, 2012). As a result, identifying terms and constructing terminological resources can be considered as a stepping-stone for constructing domain-specific knowledge bases. For instance, Brewster et al. (2009) suggest that identifying terms is the *key step* for building a *domain ontology*. The discipline of *terminology*, and its sub-discipline computational terminology, has developed as a result of the systematic study of terms (see Chapter 3).

Specialised vocabularies are invented mainly to reduce lexical ambiguity. General language words are inherently vague due to their envisaged function in natural language communication systems—that is, a *finite* set of words are used to communicate *innumer-*

[1] Such as depicted in *Star Trek* by the *Vulcan mind meld* and the *Marijne VII beings* communication ability; however, a similar technology is not yet available to the *computer access and retrieval system* in the 29th century (Roddenberry, n.d.).

[2] Or, breaking the knowledge acquisition bottleneck, as put by the artificial intelligence community.

[3] See Margolis and Laurence (2014), for a gentle philosophical explanation.

[4] This perspective is maintained throughout this thesis. Hence, in this thesis, it is assumed that the interpretation of the meanings of a term is bounded to a particular domain knowledge.

able concepts.[1] To alleviate ambiguity in the process of knowledge dissemination (e.g., technical and scientific writing), special attention is paid to *lexical cohesion* (e.g., as emphasised in *technical writing pedagogy*).[2] In achieving this goal (i.e., lexical cohesion) and to ensure precision in communication, the invention of terms for reducing lexical ambiguity is a dominant mechanism employed in technical writing.[3]

In this process, the collection of documents that represents a domain knowledge, as a whole, constitutes the discourse in which meanings of terms are interpreted.[4] As such, lexical cohesion is established over the corpus and not individual documents or text segments.[5] Empirical studies in natural language processing—particularly, *word sense disambiguation*—support this argument. Results obtained based on generalisations of the so-called *one sense per discourse* (OSD) hypothesis by Gale et al. (1992) are well-known examples.[6] Accordingly, Martinez and Agirre (2000) show that the OSD hypothesis is strongly held in corpora that share a related genre or topic. Similarly, enhances in the performance of word sense disambiguation algorithms as a result of domain-adaptation are also evidence that support the proposed argument (e.g., see Chan and Ng, 2007).

In computational terminology, *automatic term recognition* (ATR) techniques are often at the centre of attention. ATR techniques are developed as an (assistive) tool for extracting terms from text and maintaining up-to-date inventories of specialised vocabularies. ATR algorithms do not specify semantic relationships between terms. The input of ATR is often a *domain-specific corpus*,[7] and the output is an unstructured set of terms. These terms signify a broad spectrum of concepts from the domain knowledge that they represent. However, in many applications (e.g., in *ontology-based information systems*[8]), the extracted terms are required to be organised to meet demands or to enhance performances of information systems. An analogy of this convention is the method employed in the *Princeton WordNet* lexical database (Fellbaum, 1998) for organising words.

WordNet distinguishes between *word* and *concept*: a *word* is a lexical form of a *concept* (or *meaning*). The relationship between words and concepts is assumed to be many-to-many. Hence, *synonymy* is one of the main relationships employed to organise

[1] The ambiguity of words is not limited to *polysemy*; see Murphy (2002, chap. 11, p. 404) for an elaboration of the meaning of the word *vague* in this context.

[2] For example, see Halliday and Hasan (2013, chap. 6).

[3] In general language a similar mechanism is used, too, perhaps using *compounding*: 'The process of forming a word by combining two or more existing words (Trask, 2013)'.

[4] Note that *what constitute this whole* and the *discourse* is a subject of study and a research question in itself (e.g., see Wilks and Brewster, 2009, chap. 4).

[5] Also, see the complementary perspective given based on Zellig Harris's work in Section 1.3.

[6] As cited by Wilks and Tait (2005a), *Karen Spärk Jones* must be acknowledged as the pioneer of introducing ideas of this nature.

[7] For an account of the term domain-specific (or, *special*) corpus see Section 1.3. Also, note that depending on the application and availability of information resources, an ATR algorithm can use additional background knowledge, such as an existing terminological resource—see Chapter 3.

[8] Or, the classic *property assignment* (slot filling) task in Minsky's (1974) frame-based knowledge representation systems.

words.[1] In WordNet, words that refer to the same concept are synonymous and organised as one *synset* (Miller et al., 1990). In turn, the synonym relation between words and constructing synsets can be seen as the mechanism employed to denote concepts.[2] In contrast, Miller et al. define another set of relationships between 'word meanings' (i.e., concepts or synsets in WordNet). Among these relations, the *hyponymy–hypernymy* is a transitive and asymmetrical relationship between synsets employed to organise general English *nouns*. The result is a hierarchical structure (i.e., a taxonomy), in which a hyponym synset is classified below its superordinate.[3]

This thesis suggests an organisation of terms based on *co-hyponymy* relationships between them, in analogy to the role that the synonymy relationship plays for organising words in WordNet. Terms and their corresponding concepts are usually organised into semantic categories; each category characterises a group of terms from 'similar' concepts in a domain knowledge—that is, a *type-of* or *is-a* relationship between a set of terms and their superordinate.[4] Terms organised under a particular hypernym are in a *co-hyponymy* relationship simply because they are hyponym of the same hypernym. For example, in an application, one may consider terms such as *corpus*, *dictionary*, *bilingual lexicon*, and so on as co-hyponyms under the hypernym *language resource* (see Figure 5.1).[5]

Using co-hyponymy as a basis for organising terminologies can be motivated by at least two observations:

a) *Persistency*: that is, many practical applications of the co-hyponymy relationships (which have emerged under various names and for diverse reasons, as is abridged in the following paragraphs); and,

b) *Regularity*: that is, in a specialised vocabulary, the co-hyponymy relationship between terms is more *frequent* than other types of relationships such as synonymy.

The latter is a direct outcome of the deliberate act of reducing lexical ambiguity in domain knowledge dissemination and in adopted perspectives in terminology (see Chapter 3). Although a synonymy relationship between terms exists (mainly as a function of *term variation* such as addressed by Freixa, 2006), to a large extent synonymy is (and to an extent polysemy) less frequent than co-hyponymy in terminological resources. In turn, the

[1] Inarguably, Jones is the originator of the discussion about the relationship between semantic classes and the synonymy relationship between words (see Jones, 1986).

[2] Synonymy and *synset construction* are two sides of the same coin, as Wilks and Tait (2005a) explain.

[3] See also Resnik's (1993) elaboration on the *class-based* approach to lexical relationships.

[4] The study of the nature of this *kinds-sorts* relationship and how it is established (e.g., as examined by Carlson, 1980), unfortunately and although quite relevant, is beyond the scope of this thesis. A recent stimulating discussion on *kind-level* and *object-level* nominals can be found in Acquaviva (2014). Also, an applied perspective in the context of knowledge engineering is given by Cimiano et al. (2013). This thesis deliberately does not distinguish between the delicate difference between form and concept.

[5] This discussion is further extended in Chapter 5. As explained in Section 5.1, in the context of mapping a vocabulary to a domain ontology, terms that are *reified* to same ontological references are considered co-hyponyms.

synset-based mechanism employed in WordNet is not effective for organising entries of a terminological resource, at least as a conceptual denotation (categorisation) mechanism.[1]

The overture proposed in the above paragraphs leads us to an important, though indirect outcome, of the presented study. Organising terms by characterising co-hyponymy relationships can be seen as a step towards bridging the semantic gap between the three elements a) lexical knowledge,[2] b) a conceptual representation of a domain knowledge, and c) a quantitative interpretation of meaning of terms in a specialised discourse. Given this perspective, this thesis is an investigation of vector-based distributional representations of terms in order to form a quantitative model of kinds-sorts that resembles a 'correlate to conceptual representations[3] (as nicely put by McNally, 2015)'.[4]

The proposed co-hyponymy-based mechanism for organising specialised vocabularies, in turn, paves the road towards a *class-based approach* to the manipulation of terms on the basis of their distributions in domain-specific corpora (i.e., in a similar fashion that Resnik (1993) and Brown et al. (1992) suggest for words in general language). The list of literature that motivates the identification of co-hyponym terms is beyond the references listed in this section; the emphasis that *Adrienne Lehrer* puts on the structure of vocabulary and its relationship to meaning is particularly worthwhile mentioning (e.g., see Lehrer, 1978). It is also important to note that co-hyponymy is not sufficient for capturing all the semantics in a specialised vocabulary,[5] but it is an essential relationship for extending the inventory of relationships that address a number of practical problems in knowledge engineering.

Section 1.2 continues this discussion from a computational perspective, followed by the complementary view of natural language processing in Section 1.3. Section 1.4 enumerates the practical research questions investigated in this thesis. A summary of contributions is listed in Section 1.5. Section 1.6 provides readers with information about the structure of this thesis.

1.2 Implied Computational Challenges: A Solution

Although Section 1.1 promotes a novel perspective for organising terminologies based on their distributional similarities in corpora (as with other researchers such as McNally and Herbelo (2015)), the extraction of co-hyponym terms is not a new task by all means. The identification of co-hyponymy relationships as a linguistic phenomenon has been

[1] The recursive nature of hyponym–hypernym relationship can result in a controversy: at a very fine level of conceptual granularity, perhaps, there is no difference between synonymy and co-hyponymy.

[2] If one insists that it is different from the knowledge itself.

[3] Again, if we can conceive such thing without language.

[4] See also Agres et al. (2015) who apply a similar principle to investigate conceptual relationships in the context of music creativity (cognition).

[5] For example, similar to the problems resulted from *is-a overload* (as described by Guarino, 1998) and as implied by the term *tennis problem* in the context of the WordNet organisation (e.g., as explained recently by Nimb et al., 2013).

addressed previously to meet demands in various use-cases—ranging from entity recognition and term classification methods to taxonomy learning tasks (see also the complementary introduction in Chapter 5).

The most established examples of methods that, in fact, extract co-hyponyms are *entity taggers*. Typically, lexical items of a certain *type* are annotated manually in a corpus. In this context, *type* is the hypernym or the superordinate, and annotated lexical items or entities are a group of co-hyponyms. The corpus is then employed to develop an entity tagger often in the form of a sequence classifier. These methods rely on manually annotated data, in which each mention of a term and its concept category (i.e., the hypernym) must be annotated. *Bio-entity taggers* are familiar examples of this type. Provided that enough training data is available, a reasonable performance can be attained in these recognition tasks (e.g., see report in Kim et al., 2004).

Apart from entity taggers that identify co-hyponyms, as described in Chapter 3, the co-hyponymy identification has also been addressed by a number of methods known as *term classification* (e.g., see Nigel et al., 1999). Given a taxonomy, term classification techniques, similar to entity taggers, often employ a supervised learning classification method to label terms with their hypernyms. Apart from delicate differences between previously introduced methods, they lack a number of features. These methods often do not provide a model of terms that can be used as their (intermediate) semantic representation of terms. The output is often a label, often without a degree of similarity between terms and with no built-in mechanism for representation of conceptual structures. In addition, in these methods, the dynamic nature of the co-hyponymy relationship between terms is largely ignored.

In a study, Lamp and Milton (2012) describe that the employed schema for term categorisation (i.e., the co-hyponym groups) not only changes by the dynamic of a domain knowledge, but also by the way that terms are shared and used at a specific given point in time. Hence, in a given categorisation of terms, change is inevitable—not only from a *diachronic* perspective, but also on a *synchronic* level and depending on the parties involved in the communication process. Comparably, it may be required to organise an existing terminological resource in order to address the constantly changing demands of an information system. This problem has been largely overlooked in methods previously proposed for knowledge acquisition from text (and, the identification of co-hyponym terms).

The major research challenges to develop a mechanism to address the problems mentioned above can be summarised as follows:

1) The mechanism must identify co-hyponymy relationships between terms—that is, the association of a term to a particular hypernym or a category of concepts.
2) The mechanism must be capable of capturing the dynamic nature of the co-hyponym groups in a domain knowledge (e.g., as in Lamp and Milton, 2012).
3) The mechanism must be capable of resembling the conceptual structure of a domain knowledge in some sense (see Section 1.1).

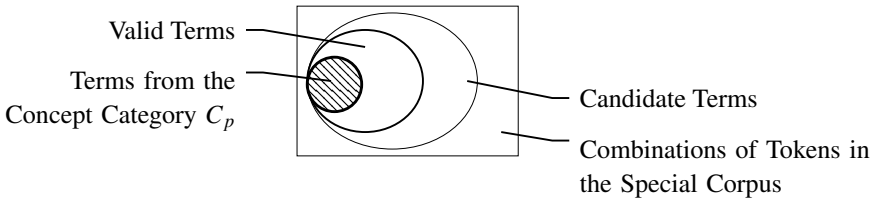


Figure 1.1: Venn diagram that illustrates the relationships among candidate terms, valid terms, and a particular category of terms C_p. ATR targets the extraction of candidate terms and the identification of valid terms. However, the proposed term classification task targets the identification of co-hyponym terms—that is, a subset of valid terms.

The first challenge, in general, is non-trivial since terms cannot be distinguished explicitly from lexical units that are not a term. Co-hyponym terms in particular can not be distinguished from other terms. Devising such a mechanism implies a level of *text understanding*. Therefore, it is an open research question. The second and third challenge listed above rule out the use of previously employed techniques such as entity tagging for finding and encoding co-hyponymy relationships between terms. Entity tagging and other supervised methods are too rigid to be used as an approach to reflect the dynamic of co-hyponym groups and to reflect various co-existing conceptualisation structures (e.g., manual annotations must be revised, the underlying classifiers must be retrained, or even a new classifier must be added to find and represent a new co-hyponym group).

As illustrated in Figure 1.1, identifying a group of co-hyponym terms in a terminological resource is equivalent to charactering a subset of valid terms. Evidently, from a computational perspective, the co-hyponym identification can be boiled down to a classification task. As suggested above, this formulation of the problem has been adopted in a number of previously proposed methods (e.g., see Nigel et al., 1999; Afzal et al., 2008; Kovačević et al., 2012). However, in contrast to these methods and in order to address the research challenges itemised above, this thesis proposes a justification of the co-hyponym identification task in the general framework of *distributional semantics* and using a *similarity-based reasoning* process that employs *memory-based learning*. In turn, the proposed methodology is evaluated systematically.

I assume that the association of a term to a category of concepts (i.e., a co-hyponym group) can be characterised with respect to its co-occurrence relationships in the corpus. Such being the case, I hypothesise that terms from similar concept categories tend to have similar distributional properties. In order to quantify these distributional similarities, I employ vector spaces: a mathematically well-defined framework, which has been widely used in text processing (Turney and Pantel, 2010). In a vector space, candidate terms are represented by vectors in a way that the coordinates of the vector determine the correlation between candidate terms and the collected co-occurrence frequencies. Consequently, the proximity of candidate terms can be used to compare their

distributional similarities. The result, as implied by Schütze (1993) and delineated later by Widdows (2004) and Sahlgren (2006), is a geometric metaphor of meaning: a semantic space that is, accordingly, called a *term-space model*.

In this term-space model, the task is to identify a particular *paradigmatic* relationship between terms—that is, co-hyponymy. It is assumed that each group of co-hyponym terms can be characterised using a set of *reference terms* or examples (shown by R_s)—that is, a small number of terms (e.g., 100) that are annotated with their corresponding hypernym (i.e., concept category). The distance between vectors that represent candidate terms and the vectors that represent R_s is assumed to determine the association of candidate terms to the group of co-hyponyms represented by R_s. This *similarity-based reasoning framework* is then implemented based on the principles of Daelemans and van den Bosch's (2010) memory-based learning—that is, using an instance-based k-nearest neighbours (k-nn) algorithm, as described later in Chapter 5. Notably, k-nn introduces a technique for similarity-based reasoning that can meet the requirements imposed by the dynamic nature of co-hyponym groups (i.e., the ability to update the rational behind the reasoning process at any time during the use of system with minimum effort). To reflect changes in the structure of co-hyponym groups, it is only required to update R_s—that is, to provide a new set of examples.

The use of this proposed method, however, is hampered by two major (related) obstacles:

1. *the curse of dimensionality*: In the proposed term-space model, due to the Zipfian distribution of words in text, vectors that represent candidate terms are usually high dimensional and sparse—that is, most of the elements of the vectors are zero. The high dimensionality of vectors hinders computation and diminishes the method's performance; the sparsity of vectors is likely to diminish the discriminatory power of a constructed term space model (see Chapter 2).
2. *the inflexibility of models to accommodate updates*: In addition, changes in the documents that represent a domain knowledge or adding new candidate terms, inevitably demands changes in the structure of the vector space that represent the domain knowledge. Previous methods employ the so-called *one-dimension-per-context-element* (see Chapter 2). Put simply, in these methods of vector space construction, the structure of vectors is firmly controlled by the input text-data. The basis of vectors (i.e., informally their dimension) is determined by the words that co-occur with terms. An update in a model (i.e., changes in the collection of documents or terms) demands a change in all the vectors since new dimensions must be appended or removed from the model. This is not acceptable considering the fact that models usually are large in size and updates are frequently necessary to reflect the dynamic of a domain knowledge.

In the presented study, special attention is paid to these problems. As a result, so-called *incremental techniques* using *random projections* are proposed to avoid the obstacles listed

above (see Chapters 4 and 5).

As explained thoroughly in the following Section 1.3, in distributional analyses of languages, a major research is the study of co-occurrence relationships with respect to a targeted task (here, co-hyponymy identification). For example, in rule-based information extraction methodologies, the task of a researcher can be to identify and then characterise linguistic patterns in a formal language, such as *regular expressions* or more sophisticated grammar rules. In distributional methods, a similar effort is required; however, in another form and using mathematical tools other than rules. Although a distributional model is built automatically, research is still required to:

a) define the way these models must be constructed;
b) and then to (b) set variable parameters of the envisaged model (e.g., see the proposed research questions in Section 1.4 and the evaluation parameters discussed in Section 5.3, Chapter 5).

Evaluation of distributional models in general, and, in particular, the proposed distributional model for identifying co-hyponym terms, in a way that the interdependencies between parameters are assessed, remains an untouched area of research. Evidently, a distributional model, such as the one proposed in this thesis, is a multi-parameter system in which the interdependence between parameters is not known. In previous research, this fact has often been overlooked; hence, parameters of a model have been mostly evaluated independently of each other. To address this problem, much of the work in this thesis is devoted towards a holistic evaluation of the constructed models.

1.3 The Natural Language Processing Perspective

The motivation for this study can also be described from the perspective of *natural language processing*. Natural languages are certainly the most important vehicles for information creation and dissemination. Consequently, natural language processing has emerged as an important interdisciplinary research field that melds linguistics with computer and information science. The major objective of research in this area has been to establish an abstract system that characterises natural language. The interpretation of this abstract system must enable computers to represent, store, access, process, and unlock information that is encoded in natural languages, for instance as explained in the motivation for this thesis.

In contrast to research topics such as *human language technology*—which pursues the ultimate goal of natural language communication between man and machine similar to man-to-man communication—or, for example, computational cognitive science and psycholinguistics—which study the underlying mechanisms of understanding language in the human mind—natural language processing is modestly concerned with finding a suitable model of language to fulfil a particular task. Although all these areas of research

discern the problem of natural language understanding and the meaning of meanings, in natural language processing the focus is on practical applications. To achieve practicality, then, natural language processing deliberately simplifies aspects of natural language.[1]

The foundation of natural language processing and the method proposed in this thesis can be traced back to as early as the 1950s and the growing availability of commercial computers. On one side, computers facilitated processing language corpora (i.e., a collection of text data); on the other side, using computers for information processing stimulated the need for building computable models of language. The product was the formation of a strong *empiricist*[2] approach towards analysing languages and the development of a set of data-driven techniques for their automatic processing—what are nowadays referred to as statistical natural language processing and corpus-based methods.

Simply put, these methods validate hypotheses about different aspects of natural language—such as, morphology (i.e., the structure of words), syntax (i.e., the structure of sentences), and semantics (i.e., the structure of meanings)—by collecting evidence from corpora (for an overview of these methods and their applications see, e.g., Tognini-Bonelli, 2001; Wilson and McEnery, 1996). The ever-increasing processing power of computers has made these empiricist approaches a dominant technique for realising goals set by natural language processing research.

A number of prominent researchers[3] have contributed towards establishing theoretical frameworks that can be used to explain these corpus-based, data-driven methods—see, for example, the inventory of the names listed in Jones and Kay (1973) and Moskovich (1976). In the context of this thesis, however, theoretical articulations by Zellig Harris (1909–1992) are relied upon, namely, Harris's (1954) *distributional hypothesis* and his idea of *sublanguages* (see, e.g., Harris, 1968, p. 154). As it is best described by Nevin (2002, Foreword, italics are added):

> The consequence *of Harris's theories* is that the work of linguistic analysis can be carried out only in respect to co-occurrence relations in the data of language—what had come to be called distributional analysis.

Harris's (1954) distributional hypothesis is often employed to justify a contemporary research trend in computational semantics that characterises itself by the name *distributional semantics*. As it is described in Chapter 2, distributional semantic methods use a data-driven approach for modelling and interpreting the meanings of linguistic entities such as words, phrases, and sentences. In these methods, the meanings of these entities are a function of their usage in language corpora.

[1] In research literature, terms such as natural language processing and human language technology are often used interchangeably. The aim here is to contrast the objectives of these related areas of research. Also, it is worth mentioning that these research topics are reciprocal in their relationships, that is, research findings in one area are often employed to support claims or stimulate activities in the other. The term *computational linguistics*, perhaps, is the best representative of the aggregation of these research topics.

[2] In the sense that knowledge is elucidated upon 'sense experience' (Markie, 2015).

[3] Conceivably, of an equal importance.

Compared to the distributional hypothesis, Harris's idea of sublanguages is, perhaps, understated. Similar to the notion of substructure in mathematics, Harris argued that a certain subset of sentences in a general natural language can form a sublanguage if and only if it 'is closed under some operations' of the general natural language (the *closure* property):

> A subset of the sentences of a language forms a sublanguage of that language if it is closed under some operations of the language: e.g., if when two members of a subset are operated on, as by and or because, the resultant is also a member of that subset (Harris, 1998, p. 34).

According to Harris, in a sublanguage, information is expressed by the repeated use of limited sentence types and word classes. Therefore, once these types and classes are determined from an *analysis of sample documents*, they can be used to build a structure for the information that will be extracted from the analysis of new sample texts. Despite shortcomings—for example, as stated by Kittredge and Lehrberger (1982), the lack of an adequate definition—and harsh and contradictory critics,[1] Harris's (1968) sublanguages idea provides a theoretical basis for the corpus-based processing of (domain-specific) natural language texts. The notion of sublanguages, particularly, has been employed to justify the generalisation of findings from a *limited* number of observations in a reference corpus to the unseen and unlimited text data that is not the reference corpus.[2]

Since then, Harris's perspective has influenced a substantial amount of research on the automatic analysis of language. Notably, Harris's doctoral student Naomi Sager perfected and applied the idea of sublanguages to real-world applications (see, e.g., Sager, 1975). The influence of the idea of sublanguages can be further traced in the work of Sager's collaborators such as Carol Friedman, Ralph Grishman, and her doctoral student Jerry Hobbs (e.g., see chapters of Grishman and Kittredge, 2014). Through the series of DARPA's founded Message Understanding Conferences,[3] the idea of sublanguages eventually emerged as today's modern information extraction technology (see Hobbs and Riloff, 2010, for an overview of the state of the art in information extraction).

[1] Compare, for example, reviews by Wheeler (1983) and Nevin (1984): Wheeler concluded that

> The work *of Harris* does not help us with semantics, it is not mathematics, and it comes late to the problems of syntax (Wheeler, 1983, italics added).

Nevin (1984), however, suggested that sublanguages 'are essential to an understanding of semantics of natural language'.

[2] As repeatedly stated throughout this thesis, Harris is neither the first nor the only linguist who promotes the structuralist perspective of language through the functional distributional analysis of words. Similar philosophical perspectives are presented in the work of Jost Trier (1894–1970). In many respects, the notion of *word (semantic) fields* as Trier (1934) put forward is similar to Harris's sublanguages (perhaps, only a terminological difference. For example, compare this section with explanations given in Gliozzo and Strapparava, 2009). See also Chapter 2.

[3] See `http://www-nlpir.nist.gov/related_projects/muc/`.

The use of this sublanguages idea is not limited to information extraction. Languages that are used in specialised communicative contexts (which from now on will be called *specialised languages*) and, respectively, the corpora that represent them (which following the suggested guidelines by Sinclair (1996), will be called *special corpora* or *domain-specific corpora*) are the most definite examples of sublanguages (see, e.g., the recent study in Temnikova et al., 2014). For example, as stated by Harris (2002), in order to reflect the information's structure in a specialised knowledge domain, a special language (e.g., the language of science writing) conforms not only to particular structures—for instance, syntactic and discourse structure—but also uses a *specialised vocabulary*.[1]

As discussed in Section 1.1, the entries of this specialised vocabulary (also known as a terminological resource) are often called *terms* and have been the subject of study in the discipline of *terminology*. Whereas traditional terminology investigated terms as self-subsisting linguistic entities, independent of their usage in text, the idea of sublanguages has encouraged the study of *terms* in *context*, as stated by Pearson (1998).[2] Disregarding the theoretical motivations, special corpora and terminological resources have been a vibrant topic in the broad domain of natural language processing and, in particular, the emerging multi disciplinary research field of *computational terminology*.

Accordingly, in this thesis, among research topics in computational terminology, the application of corpus-based methods for extracting co-hyponym terms is revisited using the aforementioned theoretical framework of Harris's distributional hypothesis and sublanguages and the mathematical framework of *real normed vector spaces*. The proposed method is then evaluated in the systematic way that is encouraged by advances in distributional semantics.

1.4 Research Questions

To investigate the hypothesis proposed in this thesis—that is, co-hyponym terms share similar distirbutional properties that can be employed to organise a specialised vocabulary—a number of research questions must be addressed. The first and foremost question—similar to other applications of distributional methods—is:

- *What kind of co-occurrence relationships among relationships must be collected to form a suitable model to characterise the targeted structure?*

As is explained in Chapter 2, previous research in distributional semantics suggests that a *paradigmatic* relationship, such as the one targeted in this thesis, can be distinguished by collecting co-occurrence frequencies from *small* windows of text in the vicinity of candidate terms. This knowledge results in another research question:

[1] The notion of sublanguages can be approached from other perspectives, for example, see the short note and references in Karlgren (1993).

[2] Please note that the study of terms in context has been suggested by several other motivations and theories (e.g., see Faber and L'Homme, 2014).

- *What is the best configuration for this window of text?*

The question above can be broken down into several sub-research questions. However, as explained in Chapter 2 and stated in the previous research (e.g., see Baroni and Lenci, 2010; Sahlgren, 2008), at least three questions can be asked:[1]

RQ 1.1 *In which direction, regarding the position of the candidate terms, must this window of text be stretched?*

1. only to the left side of a candidate term to collect the co-occurrences of the candidate term with preceding words;
2. only to the right side to collect co-occurrences with the succeeding words; or
3. around the candidate term—that is, in both left and right directions?

RQ 1.2 *What is the best size for this window of text—for example, one or two tokens, or bigger sizes, such as six or seven?*

RQ 1.3 *Is the order of words in this window of text important; and, does encoding the sequential order of words improve the discriminatory power of models?*

After collecting the co-occurrences, several other questions arise regarding the use of the suggested similarity-based reasoning framework:

RQ 2.1 *What kind of similarity measure performs better?*

RQ 2.2 *What is the role of neighbourhood-size selection—that is, the value of k in the memory-based learning framework?*

Another question can be asked with respect to the size of corpus, namely:

RQ 3 *Is the size of the corpus used for collecting co-occurrences important? Is bigger, better?*

Last but not least:

RQ 4 *Are the obtained results consistent across concept-categories?*

Apart from the questions listed above, a major research concern that is investigated in this study deals with the *curse of dimensionality* and the design of scalable methods for the construction of vector space models. Whereas a technique such as *truncated singular value decomposition* is mathematically well-defined, its application is limited by the resource required for its computation, particularly when dealing with big text data. In contrast, the alternative scalable technique named *random indexing* lacks adequate mathematical justifications. In this thesis, this argument is formulated by

RQ 5 *What are the mathematical justifications of random indexing in particular, and in general, incremental methods of vector spaces construction?*

The aforementioned research questions result in the scientific contributions that are described in the next section.

[1] See additional questions in Chapter 6.

1.5 Summary of Contributions

Based on the principles of distributional semantics, a method for identifying co-hyponym terms in a terminological resource is proposed. The association of terms to a category of concepts, hence, the co-hyponymy relationship, is modelled as a paradigmatic relationship in a vector space model. The construction of this model is carried out automatically and at a reduced dimensionality using an incremental, thus, scalable methodology. Using minimal supervision and given a small set of examples from the targeted category of concepts, the association of terms to the concept category are computed using an example-based k-nearest neighbour classifier (see Chapter 5).

The methodology is then evaluated in the systematic way that is encouraged by advances in distributional semantics. In order to answer each of the questions asked in the previous section, several experiments are designed and performed. The outcome of these experiments confirms the validity of the proposed hypothesis and method. Each set of experiments targets answering a set of questions that are asked above (i.e., Sections 5.4.1 to 5.4.4 in Chapter 5). In turn, in Section 5.5, the observations from these experiments are discussed and a summary of the findings is provided. Based on these observations, in Chapter 6 a set of guidelines that can be used in similar tasks is proposed.

The *random indexing* technique is studied and the method's incremental procedure is explained mathematically. This study provides a theoretical guideline for setting the method's parameters which has not been previously proposed. To support the theoretical findings, the results from a set of experiments are reported. Using the proposed delineation, the random indexing method is generalised and a novel technique called *random Manhattan integer indexing* is proposed. This method can be employed for the incremental construction of ℓ_1*-normed term-spaces* at a reduced dimensionality (see Chapter 4). The method, therefore, can be used to improve the performance of distributional semantic models when similarities between vectors are measured using the city block (or, the Manhattan) distance.

The contributions listed above are discussed further in Section 6.1.

1.6 Thesis Structure

The remainder of this thesis is organised in three parts: Background II, Core Research III, and Epilogue IV:

The Background II

Chapter 2 is a practical guide that walks the reader through the basics of distributional semantic methods: how they work and how they can be expressed—or formalised—in computers. More precisely, as suggested in Section 1.4, the vector space mathematics will be described and employed. In this framework, the major processes are explained, from

the construction of a model through the distillation of results. The reader who is familiar with these concepts can thus safely skip this chapter. Chapter 3 introduces computational terminology and reviews methods of term extraction and classification. In doing so, the common mechanism of term extraction techniques are discussed using the jargon that is introduced in Chapter 2.

Part III: Core Research

Chapter 4 introduces random projection techniques and their applications in natural language processing. In this chapter, the random indexing technique is revisited and justified mathematically. This justification is employed to provide a set of guidelines for setting the method's parameters. A novel technique called *random Manhattan indexing*, and its enhanced version called *random Manhattan integer indexing*, are then introduced. The discussions in this chapter are accompanied by a series of experiments to support the theoretical discussions.

The main methodology for identifying and scoring co-hyponym terms are then introduced and evaluated in Chapter 5. After introducing the methodology, the evaluation framework is laid out. The section in the remainder of this chapter, targets a particular set of research questions that are proposed earlier. The discussions in this chapter are connected to the explanations in the previous chapters; hence, the reader can start with this chapter and follow the provided pointers for relevant elaboration in other parts of the document. In addition, results from the experiments are connected to the original research questions described in this chapter.

Part IV: Epilogue

Chapter 6 concludes this thesis by providing a summary of findings. The lessons learned are discussed and additional questions that are faced during this study are presented as possible future research.

This page is intentionally left blank.

Part II

Background

Chapter 2

Distributional Semantics and Vector Space Models

Distributional approaches to semantics interpret the meanings of linguistic entities by investigating their distributional similarities in corpora. These empiricist corpus-based methods are often explained using Harris's (1954) *distributional hypothesis*. A vector space is an algebraic structure that can be employed to represent such distributional similarities. This representation of the distributional properties of linguistic entities generates mathematically well-defined models known as *vector space models of semantics*. In a vector space model, a distance formula measures semantic similarities between entities.

This chapter provides an overview of the distributional approaches to semantics. Section 2.1 provides a brief overview of distributional semantic models and the underlying distributional hypothesis. Section 2.2 introduces vector space models and provides mathematical preliminaries. The key processes for the discovery of meaning—that is, the steps from the construction of a vector space model to similarity measurements—are described in Section 2.3. In Section 2.4, the discussions are bound to the statistical learning theory. Finally, Section 2.5 concludes this chapter.

2.1 Distributional Semantics: Introduction

In order to provide a solution to the problems require a minimum level of text understanding, *distributional semantics* is a term that is often used to characterise a set of methods that rely on similarity-based reasoning frameworks. Distributional semantics embraces a number of approaches that employ similarity-based reasoning in an attempt to provide solutions to problems that require a minimum level of text understanding. Disregarding of the type of task and the way similarity-based reasoning is implemented, these methods aim to capture meanings of linguistic entities (e.g., words and phrases) from their usage in corpora. In distributional semantic models, therefore, meaning is a function of the distribution of linguistic entities in a given corpus.

Distributional semantics is motivated by the foundation of *structural linguistics* and the *distributional hypothesis*. The distributional hypothesis, which is often attributed to Harris (1954), presumes a correlation between distributional similarities of linguistic structures and their function in language (e.g., their syntactic role, meanings, and so on). Accordingly, distributional semantic methods suggest that the meanings of linguistic entities are established by the context in which these linguistic entities appear and their relationship to one another. For example, these methods suggest that the way words are distributed in text and co-occur with other linguistic expressions determines their meaning. Consequently, distributional semantics can be viewed as a statistical investigation of the co-occurrences of linguistic entities to capture their semantics from corpora and linguistic data.

Distributional semantics thus provides us with an *empiricist* and *quantitative* model of meaning in natural languages that is *context-dependent*. Compared to distributional semantics—on the other side of the spectrum of the methods that study semantics—*formal semantic* methods are motivated by a *rationalist* approach (e.g, see Partee, 2011). In these methods, the observation of language data is considered to be insufficient for gaining insight into the nature of language.[1] Hence, these methods rely on a priori knowledge that is often expressed in mathematical logic, for example, using the lambda calculus and predicate logic expressions (Blackburn and Bos, 2005). More importantly, compared to a distributional model that exploits an *inductive similarity-based reasoning*, formal semantic techniques rely on *deductive inference*.[2] Formal semantic models provide compelling tools and interesting model-theoretic methods to distil meaning from text. However, these methods can be used only after text is converted into logical expressions and a priori

[1] Put simply, rationalist approach sees the language as an *innate* object, an inherited capability (for a concise comparison see Manning and Schütze, 1999, chap. 1). In contemporary literature, these methods often attributed to Noam Chomsky, who collaborated with Harris as a doctoral student. Contemplating on this matter—although, out of the context of this thesis—will lead to questions such as *can we think without language?*, or *do we think independently of language?*

[2] As stated by Kamp (2002), although these methods are often studied by different communities, they can act as complementary tools for treating different aspects of the meanings in language and, thus, the problem of machine's understating of natural languages.

model of knowledge domain exists, which are a barrier to their use.

Table 2.1 lists several hypotheses that are embraced by the term distributional semantics. Despite the fact distributional semantics correlates differences in the meanings of linguistic entities to the differences in their distributional properties, it does not specify the variety of distributional information that should be taken into account. Moreover, the general idea of distributional semantics does not specify the type of meaning connotation that is attached to distributional differences. In order to establish a model that ties distributional similarity to meaning, therefore, two basic questions must be answered (see Sahlgren, 2006, chap. 3; Lenci, 2008; Baroni and Evert, 2009; Turney and Pantel, 2010):

- Which distributional properties of entities should be taken into account?
- How should different kinds of distributional properties be interpreted?

Different choices of distributional properties and their interpretation correspond to different kinds of models that capture different types of semantic similarities. Finding the appropriate answers for the above questions in a number of semantic computing tasks has formed a major empirical research theme known as distributional semantics.

2.1.1 Why Does Distributional Semantics Work?

In order to answer the question *why distributional semantics works*, I would like to begin with structuralism, an intellectual movement in the 1950s.[1] The essence of structuralism is to interpret human culture as a system of interconnected signs within a framework known as *semiotics* (see Chandler, 2007, for an introduction to the key concepts of semiotics). It was, perhaps, under the influence of the structuralism movement that Harris made his *distributional structure* proposal in order to justify the use of statistical techniques for natural language processing.[2] Particularly, Harris (1954) stated that

> the meaning of entities, and the meaning of grammatical relations among them, is related to the restriction of combinations of these entities relative to other entities.

With a mathematical mindset, Harris elegantly restored the ideas dating back to linguists such as Ferdinand de Saussure (1857-1913). In this school of thought (i.e., structuralism), language is identified as an environment of interconnected elements and as a *functional system*. In simple terms, the elements of language are defined at different levels of abstraction and granularity and connected to each other through various relations. For

[1] Readers, who wish to contemplate the (paradoxical) question asked here, are also invited to seek for answers in light of the *art of science* as explained by Dunbar (1996).

[2] For example, see reports from the *transformations and discourse analysis project* (`http://www.cs.nyu.edu/cs/projects/lsp/pubs/tdap.html`), which includes the development of the first English parsing program. See also Section 1.3 of the first chapter of this thesis.

Reference	Articulation
Harris (1954)	difference of meaning correlates with difference of distribution
Firth (1957)	you shall know a word by the company it keeps
Rubenstein and Goodenough (1965)	words which are similar in meaning occur in similar contexts
Cruse (1986)	the semantic properties of a lexical item are fully reflected in appropriate aspects of the relations it contracts with actual and potential contexts
Miller and Charles (1991, cited in (Charles, 2000))	the semantic similarity of two words is a critical function of their interchangeability, without a loss of plausibility
Morris and Hirst (1991)	word meanings do not exist in isolation. Each word must be interpreted in its context
Schütze and Pedersen (1995)	words with similar meanings will occur with similar neighbours if enough text material is available
Hanks (1996)	the semantics of a verb are determined by the totality of its complementation patterns
Lund and Burgess (1996)	word meanings as a function of keeping track of how words are used in context
Landauer and Dumais (1997)	a representation that captures much of how words are used in natural context will capture much of what we mean by meaning
Lin (1997)	the similarity between A and B, $sim(A, B)$, is a function of their commonality and differences
Lin and Pantel (2001)	if two (dependency) paths tend to occur in similar contexts, the meanings of the paths tend to be similar
Pantel (2005)	words that occur in the same contexts tend to have similar meanings
Sahlgren (2006)	words with similar distributional properties have similar semantic properties
Kilgarriff (2006)	word senses are abstractions from the data
Lenci (2008)	the degree of semantic similarity between two linguistic expressions A and B is a function of the similarity of the linguistic contexts in which A and B can appear
Sinclair et al. (2004, cited in (Stubbs, 2009))	there is a relation "between statistically defined units of lexis and postulated units of meaning"

Table 2.1: Various articulations of the distributional hypothesis

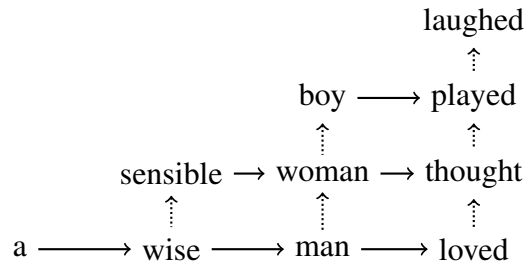


Figure 2.1: An illustration of syntagmatic and paradigmatic relations between words: the dotted lines show paradigmatic relations while solid lines represent syntagmatic relations.

instance, one may abstract language at morphological and phonemic levels, where words, morphemes, and phonemes can be considered as the building elements of language. The proposed relative perception in structuralism, then, allows elements of language to be identified by their relations to each other and not by their perceivable specification.

Structuralists apply the same fundamentals as stated above to lexical semantics. Lexical semantics is the study of the meaning of lexical units (see Paradis, 2012). According to structuralists, the meanings of lexical units (e.g., words) are not substantial and self-subsisting, but a function of relations between them. Structuralists distinguish two types of relations between words: *syntagmatic* and *paradigmatic*. Furthermore, they assume that it is harmonious combinations of these paradigmatic and syntagmatic relations that convey meaning. Given this perspective, distributional semantic methods that model the meaning of lexical units identify significant patterns in this system of interconnected syntagmatic and paradigmatic relationships.

There is a syntagmatic relation between two words if they co-occur more frequently than expected by chance and if they have different grammatical roles in the sentences in which they occur. For instance, a semantic relation in the form of selectional restrictions between a verb and its arguments—such as the relation between *love* and *man* in the sentence *a wise man loved*—is an example of a syntagmatic relation. In contrast, the relationship between two words is paradigmatic if they can substitute one another in a sentence without affecting the grammatical acceptability of the sentence. For instance, for the given sentences *a wise man loved* and *a sensible woman thought*, the pair of words *man* and *woman*, *sensible* and *wise*, as well as *loved* and *thought* have a paradigmatic relationship. Paradigmatic relations may be contrastive associations, in which a group of words might constitute a paradigm. Synonymy and antonymy are examples of such paradigmatic relations. Figure 2.1 provides an illustration of syntagmatic and paradigmatic relations.

As stated by Lenci (2008) and Sahlgren (2008), a distributional semantic model that counts the co-occurrence of words captures a syntagmatic relationship between them. In this category of models, the co-occurring words in a window of text—such as a verse

of a sentence, a sentence, a paragraph, etc.—define the context in which the relationship, thus the meaning, of words are induced. Models that extract multi-word expressions or those that specify syntactic or thematic relations between words are familiar examples in this category of distributional semantic models. In these models, the size of the region in which the co-occurrence frequencies are collected is an essential context parameter to be decided.

In contrast, if a distributional model counts the frequency of shared neighbours between words, then it captures a paradigmatic relation. In this category of models, words—or, in general linguistic entities—that appear surrounding a target word in text units such as a window of text, sentence, and so on, define the context in which the meaning/relationship of these entities are induced. Models that detect synonymy relations or those that associate words to 'semantic categories'—for example, the proposed co-hyponymy identification task as well as the named entity recognition task that organises proper nouns into categories of persons, organisations, etc.—are familiar examples of these models. In this category of models, in addition to the size of text unit in which the co-occurrences are counted, the position of target entities (e.g., words) in relation to the context elements and the direction in which the neighbourhood extends are additional parameters that must be decided.

Let us now return to the question asked in the beginning: why do distributional semantic methods work? As described above, one of the major outcomes of conceptualising language as a functional system is that it can be studied empirically using *the scientific method*. As such, the question stated above is the point in which one of the limits of the scientific method is met. To understand this limitation, one must carefully distinguish between the three elements of fact (or, observation), hypothesis, and theory in the scientific method. Facts are inherently true;[1] in distributional approaches, they are equivalent to the *observations* made about linguistic phenomena that are modelled.[2] Since it is impossible to collect everything that language embraces,[3] conclusions are inevitably based on a number of selected observations. A *hypothesis* is an educated assumption. This assumption is made before designing experiments and collecting facts. If a hypothesis holds against a large number of observations, then the hypothesis is usually formulated as a *theory*. The induced theory is then employed to justify answers to a range of questions.

However, a theory can be rejected if new observations suggest this. Some relevant and unseen observations (or, their characteristics) that are important in the process of making a decision about the *truthfulness* of a hypothesis can be overlooked;[4] in turn, this can result in controversy.[5] Using the scientific method to model language and linguistic

[1] Although very interesting, let us skip questions such as *what is truth?* in their philosophical sense (e.g., as discussed by Russell, 2014, chap. 3 and 4).

[2] Note that a number of prominent linguist object this statement.

[3] Since observations about most (if not all) linguistic phenomena are innumerable and hence it is impossible to record *everything* that is related to them.

[4] Or, observations can be *theory-laden*.

[5] There are well-known examples of this situation in the history of science, such as the *Mendel–Fisher*

phenomena is certainly controversial. For the assessment of distributional hypotheses, given the complexity of natural language as well as its infinite and generative nature, simplifying characteristics of observations and experiments are inevitable. With this prelude, I suggest that, in fact, there is no definite answer to the question asked earlier: why do distributional semantic methods work?[1]

The first answer that seems plausible is that *distributional semantics works* because the underlying theoretical framework (i.e., usually the distributional hypothesis) is sound and effective. As stated above, the *success* of distributional semantics applied to a task depends on a number of parameters, most importantly on the appropriate identification of linguistic elements and their relations within the problem context. As a result, if the *success* stories of distributional semantics are not sufficient to prove the effectiveness of the distributional hypothesis, they may also be insufficient for rejecting it.[2] Situating this discussion in the broader context that is given by Harris' sublanguages idea—as briefly mentioned in Chapter 1—can perhaps open new ways to discuss the question *why do distributional semantic methods work?*, asked here.[3]

By adopting an empiricist approach, the large number of experiments that confirm the ability of distributional methods (to address a range of tasks that require a level of language understanding) can be employed to verify the veracity of the distributional hypothesis.[4] Distributional methods have been successfully applied to information retrieval (e.g., Deerwester et al., 1990), semantic memory (e.g., Lund and Burgess, 1996), and word meaning disambiguation (e.g., Rapp, 2003), among others. These experiments have shown that contextual similarities can be employed to propose a reliable semantic model. However, distributional semantic models come with their own limitations and are still developing. The inability to handle traditional semantic notions such as *negation*, *scope*, *quantification*, and *compositionality* are examples of the distributional semantics limitations. Indeed, a number of these limitations arise from the constraints of similarity-based reasoning. Currently, these limitations are active research topics. Here, it is worth pointing out that the distributional hypothesis has not been employed to only justify distributional semantic methods. For example, a large amount of research in *speech recognition* and *language modelling* is based on the promise of the distributional hypothesis—that is, systemic functional perspective on language (even if it is not mentioned explicitly).

---

controversy (see Fisher, 1936) as well as the *Duhem–Quine* problem (see Stanford, 2013) to name a few.

[1]The short argument given here is discussed (*fairly*) by Eddington (2008) from a broader perspective that analyses the relationship between linguistics and the scientific method.

[2]Here, the notion of success is a source of controversy and ambiguity. While the discussion can be extended by describing the meaning of success, I assume success is defined by a tangible *figure of merit*—whether it is a simple quantitative measure used to evaluate an algorithm (e.g., *recall* in information retrieval tasks), or complex *Turing test-like* performance measures in more sophisticated tasks involved man-machine conversation. In fact, the definition of this performance measure (i.e., the definition of success in the given context) is an overlooked topic and can lead to flaws in the assessment of a hypothesis or unrealistic expectations or constitutions from observations in an experiment.

[3]Perhaps, by formulating and generalising the outcome of experiments more carefully.

[4]Yet, we do not like to curse one of a few tools that is available to us for analysing natural language.

Distributional semantics is often praised for the practical method that it offers for constructing semantic models—that is, building frequency profiles from corpora. Developing a distributional model, therefore, requires minimal supervision; explicit human judgements are not usually required, and no rules need to be handcrafted. Consequently, compared to formal computational semantics, the development and maintenance of a distributional-based model are less time-consuming. More importantly, distributional semantic models equip us with two unique capabilities. As emphasised by Baroni (2013), distributional semantic models offer a systematic method to approximate degrees of similarity. In this framework, in contrast to formal models, semantic similarity is a quantitative prediction (e.g., a distance measure in a vector space). Such quantitative measures allow approximate degrees of similarity to be defined explicitly. This being the case, distributional models of semantics are capable of expressing semantic relatedness in a continuum of shades of *grey* instead of *black* or *white* (Baroni, 2013).

Secondly, distributional semantic methods permit meaning to be captured by arbitrary, heterogeneous, large-scale sets of symbols: from words in a lexicon to visual objects and scenes in images or a combination of these. For example, in order to improve a similarity measurement between words, Bruni et al. (2012) employ co-occurrence counts of words with a set of low-level image-based context elements. This is an exciting area of research considering the advances in wearable computing and the increasing availability of sensory information. As explained later, various techniques, such as random projections, enable distributional models to easily scale as demand requires. Compared to formal semantics, these properties make distributional semantic models a more desirable companion for the current paradigm shift in computing from algorithm-centric to data-driven approaches (e.g., see Zadeh, 2010).

2.1.2 Distributional Semantics and Principles of Interpretation

Distributional profiles and thus distributional semantics can be interpreted in, at least, two different representation frameworks: the probabilistic and vector space frameworks (Erk, 2012).[1] Distributional information consists of the counts of the co-occurrences of linguistic elements that can be stored and viewed in a tabular data format. This tabular data can be analysed either as a contingency table in a probabilistic modelling framework or in a vector space framework. These representation frameworks interpret and measure semantic similarity using different mechanisms.

A probabilistic-based model of distributional semantics employs probability theory and Bayesian mathematics. In this framework, a probabilistic inference indicates semantic similarity. A probabilistic approach associates linguistic entities with probability distributions based on the contexts that they appear in; it also calculates conditional

[1] For instance, information-theoretic framework (e.g., as suggested by Resnik, 1995) and graph-based methodology (e.g., as employed in Navigli and Ponzetto, 2012) can be added to the list of representation frameworks for distributional semantic models.

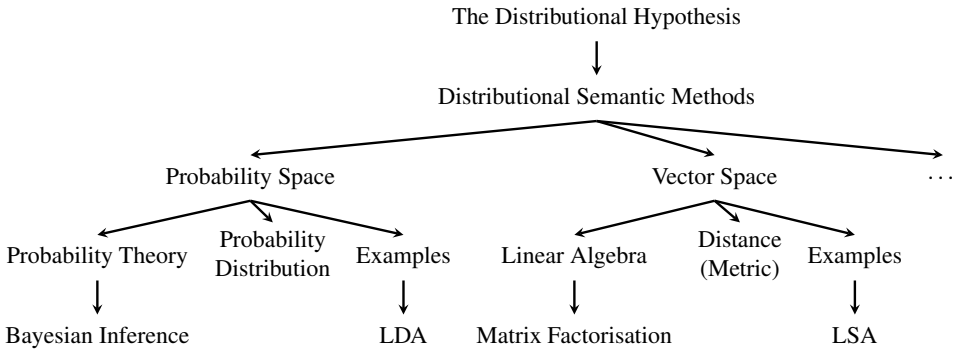


Figure 2.2: A mind map of different representation frameworks that can be employed for the implementation of a distributional semantic model.

and joint probabilities of contexts and elements. Eventually, a parameter estimation technique signifies semantic similarity. Latent Dirichlet Allocation (LDA) is a well-known example of a probabilistic approach to distributional semantics (Blei et al., 2003) .

On the other hand, vector space models construct a *metric* space from the given distributional profiles. Points in this metric space represent linguistic elements under consideration; a notion of distance between elements is defined and it indicates similarity between the elements. A 3-dimensional Euclidean space is probably the most intuitive understanding of such metric space. The vector space models thus results in a "geometrical metaphor" of meaning (Sahlgren, 2006). Landauer and Dumais's (1997) Latent Semantic Analysis (LSA) is a well-known example in this category of distributional semantic models.

Figure 2.2 summarises the discussion in this section. Although probability-based and vector space-based methods propose different conceptualisations of meaning (i.e., distributional probability vs. distance metrics), in essence, they are the same (e.g., see Turtle and Croft, 1992, in the information retrieval context). In both methods, meaning is derived from event frequencies presented by distributional profiles. However, throughout this thesis, vector space models and distance metrics are employed to model semantic similarities. Following many researchers such as Widdows (2004) and Sahlgren (2006), it can be argued that the vector space models and the geometrical interpretation of the meaning are more intuitive than the probabilistic framework—for example, as put by Widdows (2004), *seeing is believing*. However, it is worth mentioning that these representation frameworks must be seen as complementary—such as the comparison of *generative* and *discriminative classifiers* (e.g., see the arguments in Nallapati, 2004, given in the context of information retrieval).

Last but not least, while distributional models of semantics can be presented using representation frameworks other than a vector space, a vector space can also represent

semantic models other than distributional. For instance, Riordan and Jones (2011) use a *feature-based* model of semantics that is represented by a vector space. While distributional models are induced from statistical regularities of entities that appear in particular contexts (c.f., Section 2.2.2 for further details), feature-based models employ a rationalist approach and a set of descriptive features to reflect the meanings. As a result, although feature-based models of semantics can be presented by vector spaces, they are derived from an entirely different perspective on meaning. Therefore, not all the vector spaces necessarily implement distributional models of semantics.

The next section reviews basic mathematical definitions and notations that are used in vector space models.

2.2 Vector Space Models

2.2.1 Vector Space: Mathematical Preliminaries

In mathematics, an algebraic structure is a *set* together with one or more operations in it. Vector space is an algebraic structure that consists of a non-empty set and two binary operations that satisfy certain axioms. A vector space extends an algebraic structure called *field*. Informally, a field is a set of elements called scalars, or numbers, in addition to two binary operations, and certain axioms that implement four familiar arithmetic operations of addition, multiplication, subtraction, and division over the set. The field of real numbers ($\mathbb{R}$) and the field of complex numbers ($\mathbb{C}$) are well-known examples.

A vector space can be denoted by a tuple

$$(V, F, +, \cdot). \tag{2.1}$$

The set V, whose members are called vectors, is defined over a field F of scalars. For example, vectors can simply be a subset of a field such as complex numbers ($F = \mathbb{C}$, $V \subseteq \mathbb{C}$) or real numbers ($F = \mathbb{R}$, $V \subseteq \mathbb{R}$); or they can be an ordered sequence of scalars of a field such as $F = \mathbb{R}$, $V \subseteq \mathbb{R}^n$. The two binary operations are called vector addition ($V \times V \mapsto V : (\vec{v}, \vec{u}) \mapsto \vec{v}+\vec{u}$) and vector multiplication by scalars ($F \times V \mapsto V : (\alpha, \vec{v}) \mapsto \alpha \cdot \vec{v}$). The system $(V, F, +, \cdot)$ is a vector space if, and only if, it satisfies the following axioms:

- The binary operation addition + forms an *Abelian group* over V. This implies the requirements of *Closure*, *Associativity*, and *Commutativity* for the binary operation + over V, as well as the existence of *Identity* and *Inverse* elements in V.
- For the binary operation multiplication by scalars $\cdot$, $\forall \alpha \in F$ and $\vec{v} \in V$, $\alpha \cdot \vec{v} \in V$. In addition, if $\alpha, \beta \in F$ and $\vec{u}, \vec{v} \in V$, then $\alpha \cdot (\vec{u}+\vec{v}) = \alpha \cdot \vec{u} + \alpha \cdot \vec{v}$ and $(\alpha+\beta) \cdot \vec{v} = \alpha \cdot \vec{v} + \beta \cdot \vec{v}$.

Given a vector space V, if $\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_n$ are any vectors in V, and $\alpha_1, \alpha_2, \cdots, \alpha_n$

are any set of scalars in F, then

$$\alpha_1\vec{v}_1 + \alpha_2\vec{v}_2 + \cdots + \alpha_n\vec{v}_n \tag{2.2}$$

is called a *linear combination* of the vectors. From the axioms, it can be shown that a linear combination of vectors in V must belong to V. A set that contains all possible linear combinations of vectors $\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_n$ is called the *span* of $\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_n$.

A set of vectors $\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_n$ from a vector space V are called *linearly independent* if

$$\alpha_1\vec{v}_1 + \alpha_2\vec{v}_2 + \cdots + \alpha_n\vec{v}_n = 0 \iff \forall i, \alpha_i = 0. \tag{2.3}$$

If $B = \{\vec{b}_1, \vec{b}_2, \cdots, \vec{b}_n\}$ is a set of linearly independent vectors in V, and B spans V, then B is called a basis of V. Consequently, vectors $\vec{v} \in V$ can be presented as a linear combination of the vectors $\vec{b}_i \in B$:

$$\vec{v} = \alpha_1\vec{b}_1 + \alpha_2\vec{b}_2 + \cdots + \alpha_n\vec{b}_n. \tag{2.4}$$

It can be proved that there exists at least one basis B for V. The cardinality of B is defined as the *dimension* of V. By limiting the focus to *finite-dimensional* vector spaces, the dimension of V is thus the number of vectors in B[1]. The scalars $\alpha_1, \alpha_2, \cdots, \alpha_n$ in Equation 2.4 are called the *coordinates* of the vector $\vec{v}$ in that basis. It can be proved that the representation of a vector $\vec{v}$ in a basis B is unique. The coordinates of elements of V_n in a basis, subsequently, can be represented as a row or column matrix. Therefore, a collection of m vectors in V_n can be denoted by a matrix $\mathbf{M}_{m\times n}$, where the rows of $\mathbf{M}$ represent the vectors.

In a vector space, additional structures are defined to quantify relationships between vectors. The fundamental concepts of *length* of a vector as well as *distance* and *angle* between vectors are the familiar geometrical interpretation of these structures.

A *norm* is a unary operation that associates a vector in V with a scalar in F (i.e., $V \mapsto F : (\vec{v}) \mapsto \|\vec{v}\|$) and satisfies the following axioms:

- Positivity, that is, $\forall\vec{v} \in V : \|\vec{v}\| \geq 0$;
- Definiteness, that is, $\|\vec{v}\| = 0 \iff \vec{v} = \mathbf{0}$;
- Homogeneity, that is, $\forall\vec{v} \in V$ and $\forall\alpha \in F : \|\alpha\vec{v}\| = |\alpha|\|\vec{v}\|$;
- Triangle inequality, that is, $\forall\vec{u}, \vec{v} \in V : \|\vec{u} + \vec{v}\| \leq \|\vec{u}\| + \|\vec{v}\|$.

A vector space that is endowed with a norm is called a *normed vector space*. The norm of a vector $\vec{v} \in V$ (i.e., $\|\vec{v}\|$) is geometrically interpreted as the *length* of $\vec{v}$. The Euclidean norm—which is also called the ℓ_2 norm—over the field of real numbers (i.e., $F = \mathbb{R}$) is the most familiar structure that satisfies the axioms listed above:

$$\|\vec{v}\|_2 = \sqrt{\Sigma_{i=1}^{n} v_i^2}. \tag{2.5}$$

[1] From now on, an n-dimensional vector space is denoted by V_n.

Given the norm's definition, the distance $d(\vec{u}, \vec{v})$ between the two vectors $\vec{u}, \vec{v} \in V$ is given by

$$d(\vec{u}, \vec{v}) = \|\vec{u} - \vec{v}\|. \tag{2.6}$$

Given the Euclidean norm definition in Equation 2.5, respectively, the Euclidean distance—which is also called the ℓ_2 distance—between the two vectors $\vec{v}$ and $\vec{u}$ in V_n over$F = \mathbb{R}$ is given by

$$d_2(\vec{u}, \vec{v}) = \|\vec{u} - \vec{v}\|_2 = \sqrt{\Sigma_{i=1}^{n}(u_i - v_i)^2}. \tag{2.7}$$

In a similar fashion, an *inner product space* is a vector space that is equipped with an *inner product* structure. An inner product $\langle , \rangle$ is a binary operation that associates a pair of vectors in V to a scalar in F ($V \times V \mapsto F : (\vec{u}, \vec{v}) \mapsto \langle \vec{u}, \vec{v} \rangle$) and satisfies the following axioms:

- Positivity, that is, $\forall \vec{v} \in V : \langle \vec{u}, \vec{v} \rangle \geq 0$;
- Definiteness, that is, $\langle \vec{v}, \vec{v} \rangle = 0 \iff \vec{v} = \mathbf{0}$;
- Additivity for first element, that is, $\forall \vec{u}, \vec{v}, \vec{w} \in V$: $\langle \vec{u} + \vec{w}, \vec{v} \rangle = \langle \vec{u}, \vec{v} \rangle + \langle \vec{w}, \vec{v} \rangle$;
- Homogeneity for first element, that is, $\forall \vec{u}, \vec{v} \in V$ and $\forall \alpha \in F : \langle \alpha \vec{u}, \vec{v} \rangle = \alpha \langle \vec{u}, \vec{v} \rangle$;
- Conjugate interchange, that is, $\forall \vec{u}, \vec{v} \in V : \langle \vec{u}, \vec{v} \rangle = \overline{\langle \vec{v}, \vec{u} \rangle}$.

For $F = \mathbb{R}$ and the two vectors $\vec{u} = (u_1, u_2, \cdots, u_n)$ and $\vec{v} = (v_1, v_2, \cdots, v_n)$, a familiar structure that satisfied the above axioms is given using the standard multiplication of real numbers:

$$\langle \vec{u}, \vec{v} \rangle = \vec{u} \cdot \vec{v} = u_1 v_1 + \cdots + u_n v_n = \sum_{i=1}^{n} u_i v_i. \tag{2.8}$$

A geometric interpretation of the inner product and the norm gives the angle between the two vectors. In $F = \mathbb{R}$, the angle between the two vectors $\vec{u}$ and $\vec{v}$—that is θ—is defined by the cosine inverse function:

$$\theta = \arccos\Big(\frac{\langle \vec{u}, \vec{v} \rangle}{\|\vec{u}\| \cdot \|\vec{v}\|}\Big). \tag{2.9}$$

It is proved that $-1 \leq \frac{\langle \vec{u}, \vec{v} \rangle}{\|\vec{u}\| \cdot \|\vec{v}\|} \leq 1$ and thus θ is always valid—that is, $\theta \in [0, \pi]$.

It is said that the two vectors $\vec{u}, \vec{v} \in V$ are *orthogonal* if $\langle \vec{u}, \vec{v} \rangle = 0$. A basis of V_n is an *orthogonal basis* if the vectors in the basis are mutually orthogonal. Moreover, if the norm of all the vectors in an orthogonal basis is equal to the unit length, then the basis is called an *orthonormal basis*. An orthonormal basis of V_n is called the *standard basis* (i.e., $S = \{\vec{s_1}, \cdots, \vec{s_n}\}$) of V_n if each vector $s_i \in S$ has only one non-zero entry. It is common to represent V_n by the coordinates of vectors in S, which is proven to be unique.

The given definition for vector space is inherently abstract and can be extended to a fairly arbitrary set of objects that forms a field. In addition, there are a number of definitions for the binary operations of addition, multiplication, and norm that satisfy the

proposed axioms in vector spaces. Consequently, alternative structures for comparing vectors can be defined and used by changing the aforementioned components. In the context of distributional semantics, however, the employed vector space structures are usually limited to the *subspaces* of a *finite real space*, particularly, a *finite Euclidean space* $\mathbb{E}^n$.

A subset $W \subset V$ of a vector space is a *subspace* of V if

- for each two vectors $\vec{w}_1$ and $\vec{w}_2$ in W, then $\vec{w}_1 + \vec{w}_2 \in W$;
- for any scalar $\alpha \in F$ and $\vec{w} \in W$, then $\alpha \cdot \vec{w} \in W$.

Given a finite positive integer n, the set of all ordered n-tuples $\vec{u} = (u_1, u_2, \ldots, u_n)$ of real numbers and the binary operations

$$(\vec{u} + \vec{v})_i := u_i + v_i \tag{2.10}$$

and

$$\alpha \cdot \vec{u} = (\alpha u_1, \alpha u_2, \ldots, \alpha u_n) \tag{2.11}$$

that are based on the real numbers' addition and multiplication form a *finite real vector space*, shown by $\mathbb{R}^n$. An $\mathbb{R}^n$ that is equipped with a Euclidean norm (see Equation 2.5), or by analogy with an inner product (Equation 2.8), is called a finite Euclidean space. As it will be discussed in Section 2.3.4, to compute similarities, $\mathbb{R}^n$ can be endowed with a norm structure other than the Euclidean norm.[1]

The vector space-based approaches to distributional semantics use the key concepts introduced in this section to model the meanings of linguistic entities. Given n context elements, each element $\vec{s}_i$ of the standard basis of a vector space V_n is employed to express an i^{th} context element. Given V_n, in order to analyse the meaning of a linguistic entity, it is represented by a vector $\vec{v}$ as a linear combination of $\vec{s}_i$ and scalars α_i, similar to what is shown in Equation 2.4. In this linear combination, the value of α_i is acquired from the frequency of the co-occurrences of the linguistic entity that $\vec{v}$ represents and the context element that $\vec{s}_i$ represents. As a result, the coordinates of $\vec{v}$ show the correlations between the linguistic entity that $\vec{v}$ represents and the employed context elements in the model (see Figure 2.3 as an example).

In this framework, a collection of m linguistic entities whose meaning is being analysed using n context elements builds a subspace of an n-dimensional vector space consisting of m vectors. To compute similarities between the linguistic entities, this vector space is endowed by a structure such as inner product or norm. Subsequently, the angles or distances between vectors indicate the similarities of the linguistic entities that they represent. As stated earlier, often real numbers denote the magnitudes of the correlations between the linguistic entities and the context Respectively, the coordinates of vectors

[1] An elaboration of the discussed topics in this section can be found in William J. Gilbert (2004).

can be denoted by a matrix $\mathbf{M}_{m \times n}$ of real numbers. Each entry of $\mathbf{M}$, thus, represents the intensity of the relationship between a context element and an entity.

In order to distil the meanings of linguistic entities, a vector space will be the subject of several processes. Before introducing these processes in Section 2.3, the discussion continues with an elaboration of choosing the context elements in vector space models of distributional semantics.

2.2.2 Vector Space Models in Distributional Semantics

In natural language processing, vector space models (VSMs) are often identified by the model proposed in Salton et al. (1975). In the context of information retrieval (IR), Salton et al. employed a VSM to measure similarity between documents and queries. In the proposed model, natural language text documents, as well as natural language queries, are represented as vectors in a high-dimensional vector space. In this vector space, vectors that are close to each other are assumed to be semantically similar, while vectors that are far apart are semantically distant.

Given n distinct terms t and a number of documents d, in Salton et al.'s (1975) model, each document d_i is represented by an n-dimensional real vector

$$\vec{d_i} = (w_{i1}, w_{i2}..., w_{in})$$

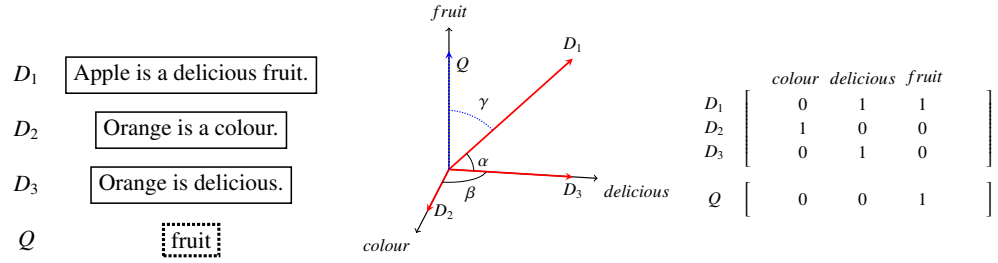
where w_{ij} is a numeric value that associates the term t_j, for $1 < j < n$, to the document d_i. The numeric association between the term t_j and the document d_i may correspond to a *weighted* value, such as the frequency of terms in documents. Alternatively, it can be an un-weighted value restricted to 0 and 1. For a collection of m documents, a *document-by-term* matrix $\mathbf{M}_{m \times n}$ denotes the constructed vector space.

A document-by-term VSM can be equipped by the inner product structure to quantify similarities between documents. Therefore, the similarity between the two documents that are represented by vectors $\mathbf{d}_i$ and $\mathbf{d}_j$ can be given by their *cosine similarity*:

$$sim(d_i, d_j) = \frac{\langle \vec{d_i}, \vec{d_j} \rangle}{||\vec{d_i}||||\vec{d_j}||} = \frac{\sum_{k=1}^{n} w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^{n} w_{ik}^2}}. \tag{2.12}$$

In the above equation, similar to Equation 2.9 in Section 2.2.1, the numerator is the dot product of the vectors and the denominator is the multiplication of the Euclidean length of vectors. The genius of the Salton et al. method is that queries, in a retrieval task, are treated as pseudo-documents and are represented by vectors too. In a vector space constructed from a document collection C, the most similar documents to a query q (such as a *keyword*) are found by computing $sim(q, d)$ for all the documents $d \in C$ (Figure 2.3).

The VSM described above implements a hypothesis known as the *bag of words*.



(a) Document Collection (b) VSM Illustration (c) Document by Term Matrix

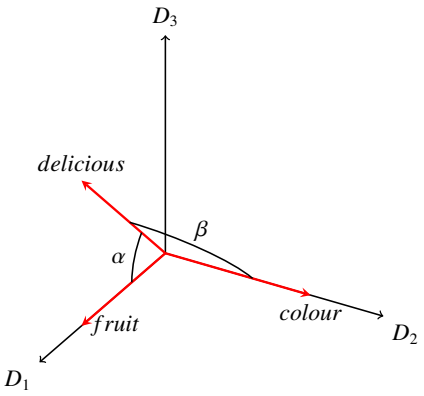
Figure 2.3: The VSM proposed by Salton et al. (1975): (b) shows a vector space that is constructed from the given document collection in (a). Words *fruit*, *delicious*, and *colour* are chosen as the context elements/terms and represented by the standard basis of the VSM. The vectors' elements denote the frequency of the terms in their corresponding documents. As is shown in (b), in this VSM, D_3 is more similar to D_1 than D_2 ($\alpha < \beta$). The given input query $Q = fruit$ is also represented by a vector. Q is closer to D_1 than to other documents ($\gamma < \frac{\pi}{2}$). Figure (c) shows the *document by term* matrix denotation of the constructed VSM.

The BoW hypothesis suggests that the relevance of documents can be assessed by counting words that appear in the documents, independent of their order or syntactic usage patterns. Documents with similar vectors in a document-by-term model, therefore, are assumed to have the same meaning. However, in order to implement a distributional hypothesis other than BoW, a VSM can be generalised to sets of entities other than documents and sets of context elements other than words that appear in documents.

Deerwester et al. (1990) showed that similarity between words can be captured by transposing the *document-by-term* matrix into a *term-by-document* matrix.[1] The proposed model by Deerwester et al. (1990), called latent semantic analysis (LSA), hypothesises that terms that are semantically similar occur in collections of similar documents. In this term-by-document model, the cosine similarity of vectors, which represent terms, can be employed to indicate the semantic relatedness between terms. The same model as the LSA was introduced much earlier by Jones (1972) (cited in Wilks and Tait, 2005b); the novelty of the LSA, however, is the use of *singular value decomposition* (i.e., a matrix factorisation technique) for the arrangement of context elements at a reduced dimensionality (see Section 2.3.3). As described later in Section 2.3.3, singular value decomposition is a matrix factorisation technique, which allows irrelevant context elements to be eliminated from a vector space in order to enhance the similarity measures.

The *term-by-document* model can be further generalised by replacing documents with text of an arbitrary length, such as a word, or window of words of a certain size. For instance, the proposed method in Lund and Burgess (1996) captures the semantic similarity of words using a *word-by-word* vector space. The resulting word-by-word model takes the co-occurrences of words as a measure of similarity. Even lexico-syntactic patterns can

[1] From now on, the terms vector space, context vectors, and context matrix may be used interchangeably.



(a) Illustration of the Vector Space

	D_1	D_2	D_3
delicious	0	1	1
fruit	1	0	0
colour	0	1	0

(b) Term-by-Document Matrix

Figure 2.4: A vector space model that is constructed from the document collection given in Figure 2.3. The three documents D_1, D_2, and D_3 are the context elements. Therefore, the basis of the vector space represents each of them. The vectors represent words/terms, in which the coordinates of the vectors indicate the co-occurrence relationships between the words/term and documents. In the given example, cosine similarities between the vectors suggest that *delicious* is semantically more related to *fruit* than to *colour* (i.e., $\alpha < \beta$ in Figure 2.4a). Figure 2.4b shows a matrix denotation of the constructed *term-by-document* model.

be employed to define context elements. VSMs, thus, can be categorised and studied according to the type of context element that they employ and the linguistic entities that they represent (e.g., as suggested by Turney and Pantel, 2010; Baroni et al., 2010). As discussed, the type of context elements and the linguistic entities in a model is determined by the model's underlying hypothesis and intended application.

2.2.3 Types of Models and Employed Context Elements

Distributional semantic models and the employed context elements for their construction can be categorised and studied from several overlapping perspectives.

First, these models can be categorised by the type of semantic relationship that they target—that is, whether they characterise syntagmatic or paradigmatic relations between the linguistic entities in the model (see also Sahlgren, 2006, chap. 7). As discussed earlier, in Section 2.1.1, the context elements, thus dimensions of a vector space model that captures a syntagmatic relation between linguistic entities, show the magnitude of the frequency of the linguistic entities that co-occur in text. For instance, models that are used to measure lexical semantic relatedness (e.g., as employed in Jurgens et al., 2012) must capture a syntagmatic relation. However, in a model that captures a paradigmatic relation between linguistic entities (e.g., a model that discovers the synonym or the hypernym relationship), the context elements show the neighbourhoods that are shared between the linguistic entities.

As implied in Baroni et al. (2010), distributional semantic models can be also categorised according to the approach that they employ to distil co-occurrence frequencies. A distributional method results in a so-called *flat* or *unstructured* model if the process of collecting co-occurrence frequencies in text is coincident with neglecting linguistic information such as part-of-speech tags or syntactic relations.

To implement a flat model that collects the co-occurrence frequencies of linguistic entities—that is, to capture a syntagmatic relationship—the only parameter that needs to be verified is the size of the text region in which the co-occurrence is regarded. Deerwester et al.'s (1990) LSA is an example of a flat model that captures syntagmatic relations between linguistic entities. In LSA, the text region is of the size of logical documents. Lund and Burgess (1996) present another example of a flat model that captures a syntagmatic relation between words, however, it uses a narrow text region (i.e., a text window of n words for $n = 10$ in the reported experiment). As a rule of thumb, Sahlgren (2006, chap. 9) suggests that a wide text region tends to show a better performance than narrow text region if syntagmatic relations are approximated; inversely, the use of narrow text regions for collecting co-occurrences of the neighbourhoods that are shared between linguistic entities has a better performance than using wide text regions when paradigmatic relations are approximated.

When a flat model collects the co-occurrence frequencies of the neighbourhoods that are shared between linguistic entities (i.e., to capture a paradigmatic relationship), however, the *direction* in which text region are extended is also important. Text regions can be stretched (a) only to the left side of a linguistic entity to collect the co-occurrences of the linguistic entity with its preceding words, (b) only to the right side to collect co-occurrences with the succeeding words or (c) around the linguistic entity (i.e., in both left and right directions). If text regions are extended around linguistic entities, then the position of the linguistic entities in the text region (*symmetry*) is an additional parameters that can be changed.

The *order* of words in the text regions can be also important. To capture the word order information in a model, the appearance of distinct words in distinct positions in text regions must be distinguished—for example, by appending additional dimensions to the model. The words' order information may be also encapsulated implicitly using n-gram sequences, or using an additional vector structure—for instance, as suggested in Jones and Mewhort (2007). Section 5.3.2.3 of Chapter 5 will describe the *permutation technique* and justify it mathematically, which will be employed later in this thesis. This method is first suggested by Sahlgren et al. (2008) for the incorporation of word order information in the vector space models that are built using the random indexing technique.

Curran (2004, chap. 3) distinguishes flat models by the way that they treat logical text boundaries such as sentence and paragraph boundaries. The width of text regions may be fixed irrespective of logical text segment boundaries, or it may be restricted by them. In the first case, text regions can be expanded to two or more logical text segments. Last

but not least, words in flat contexts can be presented in their *stemmed/lemmatised* form to build *stemmed* models (as named by Murphy et al., 2012). The reported experimental results are contradicting with respect to the significance of the inclusion of word order information as well as lemmatisation in the performance of distributional models (e.g., see Bullinaria and Levy, 2012).

Linguistically aware models, which are also called *structured models*, are the second category of the models that are proposed in Baroni et al. (2010). In this models, text regions are first annotated with linguistic information such as part-of-speech tags or syntactic relations. These linguistic annotations may be captured by the model, or it may be used to filter a number of co-occurrences. Linguistically aware models are used based on the intuition that linguistic information provides a stronger cue of semantic similarity than flat models. For instance, a window of words with particular part-of-speech categories, namely nouns, adjectives, and verbs, form the context proposed in Baroni et al. (2010). Widdows (2003) and Jonnalagadda et al. (2012) are other examples that employ part-of-speech tags in order to filter co-occurrences.

Pioneered by Grefenstette (1994), a sub-category of linguistically aware models is defined by the use of syntactic relations. In its simplest form, pairs of dependency relations Dep_r and words in text regions C_w (i.e., (Dep_r, C_w)) form syntactic contexts. In this model, the co-occurrence frequencies are induced by observing target words/entities that are in particular Dep_r relationships with C_w. Syntactic contexts, however, may correspond to more complex syntactic *paths* than that described here. Padó and Lapata (2007) argue that syntactic structure in general and argument structure in particular are close reflections of the lexical meanings. Several experiments suggest that syntactic-based models can outperform flat models (e.g., see Erk and Padó, 2008; Jurgens and Stevens, 2010; Thater et al., 2010; Séaghdha and Korhonen, 2011; Weeds et al., 2014b).

The third group of models, which can be called *attribute-value*-based models, are those that collects the co-occurrences of linguistic entities and particular *lexico-syntactic patterns*. As mentioned by Baroni et al. (2010), lexico-syntactic patterns are often handcrafted and used to capture concept associations, in particular semantic analysis task such as detecting an entailment relation. For instance, a context may be defined as the presence of the lexical pattern "X *such as* Y" between the two entities X and Y in order to indicate a subordinate relation between them. The main assumption here is that a surface pattern can be an indication of the presence of semantic relations. An example of this type of model is suggested by Hartung and Frank (2010).

The types of models that are listed above can be populated by the text kernel methods that are often used in text classification task. A well-known example is a string kernel (Lodhi et al., 2002). Models that are built using text kernels can be placed in one of the categories listed above, depending on the type of the employed kernel. For example, the resulting model from the application of a string kernel is often a flat model. Using a tree kernel such as the one proposed in Collins and Duffy (2002), however, results in a

structured model. Other types of kernels in applications other than text classification are also conceivable (e.g., see Plank and Moschitti, 2013; Mehdad et al., 2010).

The methods that are employed for collecting co-occurrences are not restricted to the above-listed categories. A number of recently employed methods for the construction of distributional semantic models can be categorised as those that use *extra-linguistic* context elements. As explained earlier in Section 2.1.1, the notion of the context element can be extended to sets of objects other than text. For example, in Bruni et al. (2012), low-level visual features enrich a VSM that measures semantic similarities between words (see also Bruni et al., 2014). Similar *extra-linguistic*-based models are employed in Chen et al. (2012); Roller and Schulte im Walde (2013); Silberer et al. (2013). As suggested in Anderson et al. (2012), recent research results (e.g., Mostow et al., 2011; Mitchell et al., 2008) further validate the suitability of *extra-linguistic*-based models for semantic modelling from the cognitive point of view.

Other trending usage examples of *extra-linguistic* context elements, although less exciting than the above list, are found in the context of the Web. Openly available knowledge bases on the Web are rich sources of *extra-linguistic* information and have served an increasing number of distributional models. For instance, the explicit semantic analysis (ESA) technique builds a *term-by-document* model with *extra-linguistic* context elements that are derived from the topical structure of a knowledge base such as Wikipedia (Gabrilovich and Markovitch, 2007). Reversely, Angeli and Manning (2014) employ a distributional model and the structured data in open-domain knowledge bases to enable common sense reasoning, however, for new and unseen entities. In a similar line of research, Gardner et al. (2014) use similarities in a vector space model to enhance reasoning over knowledge-bases.

The list presented here is endless. Table 2.2 lists a number of distributional models and their applications. The type of model and the employed method for collecting co-occurrences is determined by the underlying hypothesis and the task in hand. A new task implies a new hypothesis, and a new hypothesis often demands a new method for collecting co-occurrences and thus a new type of model. In short, the construction of flat models is computationally less expensive. However, flat models are often high-dimensional, which in return may result in a high computational cost for similarity measurement. Such VSMs may include uninformative, and sometimes irrelevant, context elements, which can reduce the performance of the model. The use of linguistic information may prevent the problems mentioned above, however, at the expense of higher computational costs for VSM construction. However, the use of linguistic information may introduce a level of noise that is originated from the use of linguistic analysis tools. If the co-occurrences are filtered by linguistic information or lexico-syntactic patterns, then a larger amounts of text data might be required to avoid the *sparsity* in the constructed models. Depending on the anticipated application for the constructed model, the use of a structured model may not

Reference	Model/Type/Application Domain
Salton et al. (1975)	document-by-term model, flat in *information retrieval*
Deerwester et al. (1990)	term-by-document model, flat in *information retrieval*
Lund and Burgess (1996)	word-by-word model, flat a text window of 2 words to the left and right of each target word as *a representational model of semantic memory*
Lin (1998a)	word-by-word model, linguistically aware words in syntactic relations with target words in *thesaurus construction*, *automatic detection of similar words*
Lin and Pantel (2001)	"path"-by-word, linguistically aware words in syntactic relations with automatically induced lexico-syntactic patterns (path) entitites are constrained paths in dependency tree in *unsupervised inference rules discovery*
Kanejiya et al. (2003)	word-by-word model, linguistically aware part of speech (PoS) tagged words, blocks of POS tag information around a target word in *automated essay scoring*
Widdows (2003)	word-by-word model, lingustically-aware words surrounding a target word target words discriminated by PoS tags in *taxonomy extraction*
Padó and Lapata (2007)	word-by-word model, linguistically aware pair of words and dependency relations (anchored paths) in *synonym detection*, *semantic priming*, and *sense disambiguation*
Gabrilovich and Markovitch (2007)	term-by-document model, extra-linguistic-based concepts that are derived from the Wikipedia' articles in *information retrieval*, *document similarity*, and *word relatedness*
Baroni et al. (2010)	concept-by-attribute-value model, attribute-value-based model lexico-syntactic patterns using PoS tags and dependency structures in *concept description extraction*
Jonnalagadda et al. (2010)	word-by-word model, lingusitcally-aware symmetric text window, PoS tags, encoded words' order in *named entity recognition*
Séaghdha and Korhonen (2011)	word-by-word model, linguistically aware context elements derived from dependency structure in *lexical substitution ranking*
Hartung and Frank (2011)	word-by-attribute model, attribute-value-based adjectives and nouns with context elements that are induced using an LDA topic model algorithm in *attribute selection for Adjective-Noun*
Lops et al. (2013)	term-by-meta-document model, extra-linguistic-based textual metadata derived from web resources, URLs, HTML meta-tags, social bookmarks in *tag recommender systems*
Anderson et al. (2013)	word-by-bag-of-visual-words model, extra-linguistic-based textual models, verbs and textual windows of fixed size, augmented with image-based features, to *study the correlation between fMRI-based neural patterns and distributional semantic measures*

Table 2.2: Examples of the employed context elements in vector space models of semantics in different application domains

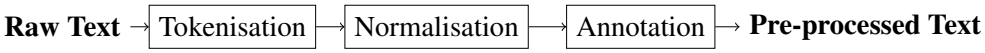


Figure 2.5: Pre-processes to vector space construction

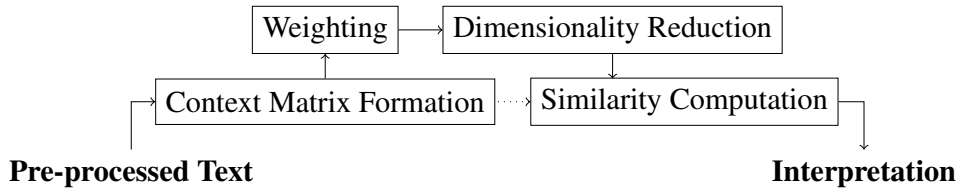


Figure 2.6: From frequency to meaning: a common four-step process flow in vector space models

necessarily enhance the results (e.g., as reported in Zeng et al., 2014).

2.3 Processes in Vector Space Models

The construction of vector space models of semantics and the task of meaning discovery involve a set of processes. These processes vary from one application and model type to another. However, a general pattern of processes can be identified in most of the applications of VSMs: a three-step pre-process followed by a four-step process (Turney and Pantel, 2010).

As shown in Figure 2.5, pre-processing starts with a text *segmentation* and *tokenisation* process in order to detect linguistically well-defined text boundaries such as words and sentences from an input text collection (see Palmer, 2010). The successive normalisation process may organise similar entities or filter some of them. For example, a simple normalisation process may convert all characters to lowercase, convert words to their lemmatised form, or remove some of the tokens such as stop words. Finally, an annotation process augments text units with additional information. For example, PoS tagging and syntactic parsing are common annotation processes.

Pre-processed data usually undergoes a four-step process that start with the collection of co-occurrences and the calculation of event frequencies and ends with an interpretation of the calculated similarity measures (Figure 2.6). In the first step, the frequency of the co-occurrences of linguistic entities and context elements is calculated, and vectors that represent linguistic entities are built. Non-compulsory processes of *weighting* and *dimensionality reduction* may follow the construction of context vectors. The process is finished by a method that measures similarity between the constructed vectors. Although these steps are listed back-to-back, in practice, they may be combined or skipped, as discussed in the following sections.

2.3.1 Context Matrix Formation: Collecting Co-Occurrences

Context matrix formation determines numeric associations between linguistic entities and context elements. In its simplest form, this association is an un-weighted binary value restricted to 0 and 1,[1] and it shows the absence or presence of the occurrences of a linguistic entity with a context element. In a typical term-by-document model, for instance, un-weighted associations indicate the presence of a term (linguistic entity) in a document (context element) using value 1. However, the association between linguistic entities and context elements can be a weighted value. The weighted associations usually correspond to the frequency of the observation of the co-occurrences of linguistic entities and context elements. For example, in a term-by-document model, the frequency of the occurrences of terms in documents can specify a weighted value.

Context matrix is often instated using a sequential scan of input text-data, for example, by collecting the co-occurrence frequencies in a hash table or database. Alternatively, a search engine that keeps an inverted index of context elements and linguistic entities can be used (Turney and Pantel, 2010). The collected frequencies in tabular presentations are then converted to an efficient data structure—for example, a dictionary of keys, list of lists, and so on—that are often used for sparse matrix representation and manipulation (for an introduction to such data structures see Barrett et al., 1993, chap. 4). However, further complications may be imposed by the adapted approach for collecting co-occurrence frequencies. For instance, Schütze (1998) employs a method called *context-group discrimination* that goes beyond counting the co-occurrence frequencies and building context vectors at once.

An alternative set of vector space construction methods may not directly count the co-occurrence events and build a co-occurrence frequency matrix. For example, Gallant (2000) suggests a three-stage process for the construction of a vector space model. In the first step, each word, which is assumed to be an irreducible context element that captures meaning, is assigned to a normalised random vector. In the second step, using an iterative process similar to the training in Kohonen's self-organising maps, vectors of adjacent words are altered in an attempt to preserve and show the neighbourhood relationships. Finally, the vector space is generated using a combination of these vectors such as their weighted sum (see also Gallant, 1991, 1994, for more details).

Kanerva et al. (2000) propose a similar method for vector space construction, which is called *Random Indexing* (RI). The RI technique constructs a vector space using a similar a two-step process and in a fashion to Gallant's (2000) method. In the RI technique, the process of vector space construction is carried out by the accumulation of a set of *randomly*[2] generated sparse vectors, called *index vectors*. Each index vector represents a context element in the model. To collect the co-occurrences, a linguistic entity is first assigned to an empty vector that has the same dimension of index vectors. The

[1] That is, $F = \{0, 1\}$, in the given Tuple 2.1.

[2] See Chapter 4 for an explanation of the meaning of *random* in this context.

co-occurrence of a linguistic entity and a context element is then captured by accumulating the index vector that represents the context element to the vector that represents the linguistic entity. A similar technique, named *TopSig*, is proposed by Geva and De Vries (2011). In these methods, context matrix formation merges with the dimensionality reduction step, often to address scalability issues that are associated with processing large corpora. These methods are studied in depth in Chapter 4.

2.3.2 Weighting

The construction of a context matrix is usually accompanied by a weighting process in order to minimise the effect of the bias that may result from simple co-occurrence counting. The major sources of bias are frequent context elements and entities. Frequent context elements that are associated with greater numeric values can dominate those context elements with smaller numeric values. In a similar way, more frequent linguistic entities may be associated with a larger number of context elements. Both of the above scenarios cause bias. The amount and effect of this bias is dependent on the employed method for the similarity measurement.

The above reasons for weighting can be viewed, by some analogy, in conjunction with *feature selection* in machine learning community (e.g., see Turney and Pantel, 2010, take on the topic).[1] First, it is desirable to give higher weights to more discriminative but less frequent context elements. For example, in an information retrieval (IR) framework that employs a document-by-term model, using the *raw term frequencies* (tf) implies the same significance of terms when measuring the similarity between documents. However, the *term frequency–inverse document frequency* (tf-idf) measure can substitute the raw term frequencies in order to give higher value to more discriminative terms. The tf-idf measure normalises raw term frequency weights by the inverse document frequency of terms (idf): tf-idf = tf $\times$ idf. The idf of a rare term, which assumes to be a discriminative context, is high, while the idf of a frequent term is expected low (for more details on tf-idf weighting in IR context, see Manning et al., 2009, chap. 6).

For types of models other than document-by-term, tf-idf can be replaced by a *measure of association* that indicates the strength of relationships between entities and contexts. As verified in Curran (2004, chap. 4), context elements with stronger correlations to linguistic entities are more informative than contexts with weaker correlations. A weak association between a context element and a linguistic entity implies their independence from each other. However, a strong association suggests that changes in a context element are likely to occur with changes in linguistic entities, thus, the context element discriminates between the linguistic entities well. For instance, in a term-by-term model, the *point-wise mutual information* measure can be replaced by the simple term co-occurrence counts (see, e.g., Bullinaria and Levy, 2007, for further explanation

[1] In this context, the weighting process is often called *feature scaling*.

and experimental comparison). Subsequently, the calculated associations can be used to sort the context elements by their importance, and if desirable to filter a number of them.

Second, the weighting process is leveraged by a method often called *length normalisation* to cancel bias that results from highly frequent linguistic entities. For example, in an IR document-by-term model, length normalisation corresponds to techniques that cancel the advantage of long over short documents in retrieval tasks. Long documents tend to appear with many terms; additionally, long documents are likely to have large term frequencies (Singhal et al., 1996). In this setting, length normalisation adjusts the term weights in conformity with the length of documents. The length normalisation, however, can be widened to any set of linguistic entities. In this generalisation, the frequency of entities is replaced by the exemplified document length. In line with this reasoning, highly frequent linguistic entities are likely to appear with more context elements than less frequent ones. Moreover, the context elements that are occur with highly frequent entities are probably associated with greater weights.

Among techniques that can be used for length normalisation, *unit-length normalisation* is a common approach. First, the length of a vector—that is, its norm—is computed. Then the collected frequencies for context elements in the vector are divided by the its computed. For instance, in an ℓ_2-normed space, the length of vector $\vec{v}$, which represents a linguistic entity, is given by $\|\vec{v}\|_2 = \sum_i^{|\vec{v}|} \sqrt{v_i^2}$. To perform the unit length normalisation, each element v_i of $\vec{v}$ (which represents a context element) is divided by $\|\vec{v}\|_2$. Thus, the element $v_i{}'$ of the new normalised vector $\vec{v'}$ is given by vector $\vec{v}'$ (i.e., $v_i{}' = \frac{v_i}{\|\vec{v}\|_2}$). The impact of unit length normalisation varies from one task to another, and it depends on a number of additional factors, namely, the size of corpus and the distribution of entities and context elements such as suggested by Périnet and Hamon (2014b); Gorman and Curran (2006a), and the employed metric for similarity measurement (see also Clark, 2015). These two factors are inspected later.

Contrariwise, weighting may be used to introduce intentional bias toward the co-occurrences of linguistic entities and certain context elements. For example, in a term-by-term model that counts the co-occurrences of words, Lund and Burgess (1996) assume that context words in closer vicinity to a target word represent more of its semantics than distant words. Therefore, the co-occurrence of words are weighted according to their distance in an inverse relation. For a context window of n words on each side of the target words, the number of intervening words between the target and context words is defined as their distance d, and the frequency of occurrences are weighted with respect to their position in context windows by the magnitude of $n - d$ (Burgess, 2001). By the same token, Sahlgren et al. (2003) employ the function 2^{1-d} for the weighing of a context window.

Baroni et al. (2007) employ a weighting procedure to encode distributional histories of context words in a term-by-term model. The vectors are weighted using a ratio of the encountered frequencies of context words. Baroni et al. (2007) suggest that fre-

quent words tend to co-occur with other words by chance. As a result, more frequent context words have less informative distributional history than rare context words. The employed weighting function, therefore, defines the influence of context words in an inverse proportion to their frequencies. Mathematically speaking, this method implements a Laplace smoothing of the collected co-occurrences, which can be also found in Turney and Littman (2003).

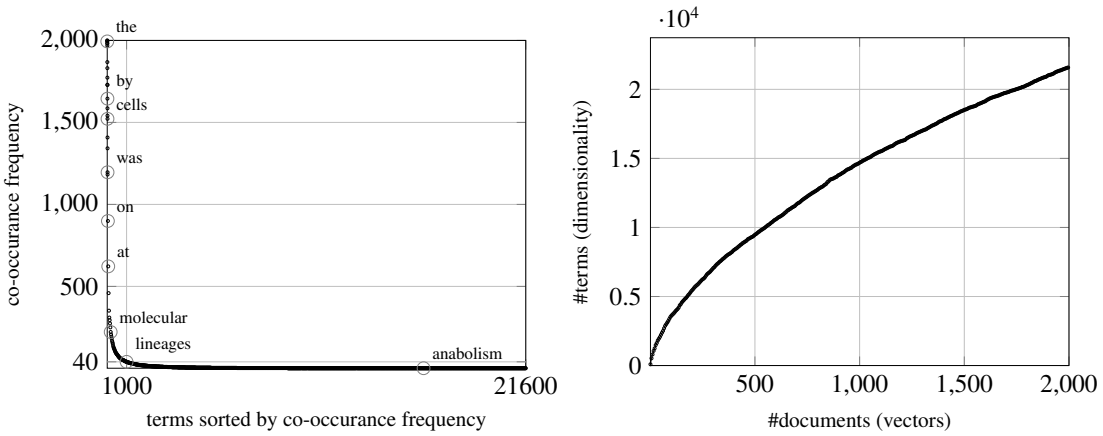
Zhitomirsky-Geffet and Dagan (2009) suggest that semantically similar words are best described by the contexts that are common between them. Therefore, they employ weighting to promote such contexts using a three-step *bootstrapping* process, similar to the proposed method in Bins and Draper (2001). At first, similarity values between words are calculated using contexts that are weighted by a mutual information measure. Next, the common contexts between the obtained set of similar words are promoted by increasing their weights. Yamamoto and Asakura (2010) propose a techniques that is bases on a similar idea. Finally, the similarities are recomputed using the updated weights. These methods can be criticised for their computational complexity, which is imposed by repetitive calculation of similarity measures, and then finding and sorting the common context elements. This procedure of weighting is also the fundamental idea behind the *learning process* in methods that employ neural networks such as Mikolov et al. (2013); Zeng et al. (2014); Irsoy and Cardie (2014).[1]

2.3.3 Dimensionality Reduction

As discussed earlier, in distributional semantics, the distributional properties of linguistic entities—that is, their co-occurrences with various context elements—are compared to quantify some sort of semantic similarities. When a vector space is used to represent and analyse these distributional properties, each element of the standard basis of the vector space—that is, informally, each dimension of the vector space—represents a context element. Consequently, given n context elements in a model, each linguistic entity in the model is expressed by an n-dimensional vector.

As the number of linguistic entities that are being modelled in the vector space increases, the number of context elements that are required to be utilised to capture and represent their meaning escalates (see the example in Figure 2.7). However, the proportional impact of context elements on semantic similarities lessens when their number increases. In a high-dimensional model, unless most coordinates of vectors are significantly different, it becomes difficult to distinguish semantic similarities. For instance, under certain broad conditions, it is likely that most entities are located at almost equal distances from each other (Beyer et al., 1999). Consequently, the proximity of linguistic

[1] Although, advances in technology, such as the availability of graphics processing unit accelerated technology, may remove this critique.



(a) The long-tailed distribution of words (b) Changes in the dimensionality

Figure 2.7: Zipfian distribution of the co-occurrences of linguistic entities and context elements: the distribution of word occurrences in documents a document-by-word model constructed using the GENIA corpus. In (a), the vocabulary is ranked by the frequency of the words' occurrences in the documents. As is shown, most of the words are rare, which results in a long-tailed distribution. Figure (b) shows the increase in the dimensionality of the model when new documents are.

entities may not express their semantic similarities.

For instance, in a word-by-document model that consists of a large number of documents, a word appears only in a few documents, and the rest of the documents are irrelevant to the meaning of the word. Few common documents between words results in sparsity of the vectors; and the presence of irrelevant documents introduces noise. These setbacks, which are caused by the high dimensionality of the vectors, are colloquially known as *the curse of dimensionality*.

This curse of dimensionality is often explained using power-law distributions of linguistic entities and context elements—for example, the familiar Zipfian distribution of words (see Yang, 2013, for further description of power-law distributions). Zipf's law states that most words are rare while few words are used frequently. As a result, irrespective of the input data size, extremely high-dimensional vectors, which are also *sparse*—that is, most of the elements of the vectors are zero—represent linguistic entities.[1] For example, Sahlgren (2005) suggests that 99% of the elements of a vector in a typical word-by-word model are zero (see also Sahlgren, 2006, chap. 4).

A *dimensionality reduction* process lessens noise and improves the performance of the similarity measurement by reducing the number of context elements employed

[1] Turney and Pantel (2010) also suggest that decreasing the sparsity will increase performance. However, they propose insufficient data as the major cause of the sparsity of vectors. Although insufficient data can contribute to the sparsity problem, one can hypothesise that the power-law distributions of contexts and entities play a more significant role in the sparsity of vectors than the data insufficiency. Further analysis is required to investigate the degree of the dimension expansion of a vector space against its sparsity reduction when the size of data increases.

for the construction of a vector space. Dimensionality reduction can be performed by choosing a subset of context elements and eliminating the rest using a *selection process*. To resolve the curse of dimensionality and reduce the sparsity of a vector space, a selection process chooses a number of context elements that account for the most discriminative information in the vector space. Consequently, the selection process results in a vector space of lower dimension constructed by a subset of the original employed contexts.

In its simple form, a selection process filters *irrelevant* contexts using a heuristic based on a threshold. After the construction of a vector space and weighting, context elements that are associated with a weight or a frequency lower than a threshold are omitted from the vector space. The main assumption is that rare low-frequency context elements are uninformative and, therefore, do not influence the impending similarity assessments. For instance, in a text categorisation task that employs a document-by-term model, Yang and Pedersen (1997) show that statistical weight thresholding can be used reliably to halve the dimension of the vector space.

In a linguistic-entity-by-word model, a common selection process is to eliminate context words that belong to a *stop word list*. A stop word list is a fixed set of high-frequency words that are clearly not related to the devised semantic similarity application. Likewise, stemming and lemmatisation can be employed to reduce inflectional, and sometimes derivational, forms of words to a common base form. The experiments performed by Bullinaria and Levy (2012) suggest that although these techniques speed up the similarity computation by reducing the dimension of the vector space, they do not necessarily enhance the observed results. As described earlier in Section 2.2.3, linguistic information, such as syntactic relations, can also replace, or be combined with, statistical measures to select and filter contexts.

A selection process may also be used to rank and filter *redundant* contexts using an information theoretic/statistical measure. Information gain, mutual information, and χ^2 test are examples of measures that can be used to check the correlations between context elements. If the correlation between context elements exceeds a certain threshold, one of them is considered to be redundant and can be eliminated from the list of employed contexts (see Hall, 1999, chap. 4 for further explanation). However, for a very high-dimensional vector space model that consists of hundreds of thousands of context elements, such methods are computationally inefficient.

In a more sophisticated approach, instead of a selection process, heuristics are used to implement a method of context generalisation. In Périnet and Hamon (2014b), context elements are generalised by finding synonym and hypernym-hyponym relationships between them. In the proposed, words in a sliding window constitute the context elements. To reduce dimensionality and sparseness of vectors, the context words are arranged into sets of words that are in a synonym or hypernym-hyponym relationship. To achieve the dimensionality reduction, the obtained sets replace context words (see also Périnet and Hamon, 2014a). Baker and McCallum (1998) uses a similar idea for dimen-

sionality reduction in a document-by-term model in a text classification task. Baker and McCallum (1998) state that while this method enhances the result of the classification task in one corpus, it does not boost the performance in two other corpora. They conclude that the structure of data (e.g., the diversity of vocabulary, the distribution of words and the size of documents) plays a significant role in the performance of these methods of context generalisation.

The process described above leads to an alternative set of dimension reduction techniques known as *transformation* methods. A transformation method maps a constructed vector space $\mathbb{R}^n$ to $\mathbb{R}^m$ of lower dimensions—that is, $\tau : \mathbb{R}^n \mapsto \mathbb{R}^m, m \ll n$. The vector space at the reduced dimension $\mathbb{R}^m$ is the best approximation of the original model $\mathbb{R}^n$ in a sense. The approximation is evaluated by a criterion such as variance, gradient descent, or distance between context elements. The interpretation of these method using the distance between context elements in the transposed entity-context model is, perhaps, more compatible with the suggested mathematical perspective in this thesis. Based on the employed evaluation criteria, transformations are categorised as either *linear*, for example, truncated singular value decomposition, or *nonlinear*, for example, self-organising map.[1]

Truncated singular value decomposition (SVD) is the most familiar transformation-based dimensionality reduction technique in the vector space models of semantics (e.g., see Deerwester et al., 1990, the latent semantic analysis model (LSA)). Truncated SVD is a linear transformation method that exploits *the Euclidean norm* of context elements, or variance,[2] to compare a vector space with its projections in reduced dimensions. Given a vector space $\mathbb{R}^n$ consists of p vectors, which is represented by a matrix $\mathbf{M}_{p\times n}$, the goal is to construct an m-dimensional vector space, represented by a matrix $\mathbf{M'}_{p\times m}$, $m \ll n$, that preserves most of the variance—thus, the Euclidean distances—in $\mathbf{M}$.

SVD factorises the matrix $\mathbf{M}_{p\times n}$ into the product of three matrices: $\mathbf{U}$, a $p \times p$ normalised orthogonal matrix (i.e., $\mathbf{U}\mathbf{U}^T = \mathbf{I}$); $\mathbf{\Sigma} = \{\sigma_1, \sigma_2, \cdots, \sigma_n\}$, a $p \times n$ diagonal matrix; and the transpose of an $n \times n$ normalised orthogonal matrix $\mathbf{V}$ (i.e., $\mathbf{V}\mathbf{V}^T = \mathbf{I}$):

$$\mathbf{M}_{p\times n} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \Big(\sum_{i=1}^{n} u_i \sigma_i v_i^T\Big)_{p\times n}. \tag{2.13}$$

The diagonal elements $\{\sigma_i\}$ of $\mathbf{\Sigma}$ are called the singular values of $\mathbf{M}$, and they are ordered such that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_n = 0$.[3] For a chosen m, $r \leq m \ll n$, the

[1] Mathematically speaking, a selection process is a kind of linear transformation process.

[2] For a matrix $M_{p\times n}$, the Euclidean norm, also called the Frobenius norm, is defined as $\|M\|_F = \sqrt{\sum_{i=1}^{p}\sum_{j=1}^{n}|m_{ij}|^2}$.

[3] $\mathbf{Z} = \mathbf{U}\mathbf{\Sigma}$ are called the principal components of $\mathbf{M}$.

SVD truncation of $\mathbf{M}$ with rank m is given by

$$\mathbf{M}'_{p\times m} = \mathbf{U}_{p\times m}\mathbf{\Sigma}_{m\times m}\mathbf{V}^T_{m\times m} = \Big(\sum_{i=1}^{m} u_i\sigma_i v_i^T\Big)_{p\times m}. \tag{2.14}$$

The basis elements of $\mathbf{M}'$ are orthogonal because the data is decorrelated in the ℓ_2-norm (i.e., second-order) sense and thus their inner product is zero. According to Eckart and Young's (1936) theorem, $\mathbf{M}'$ represents the best approximation of $\mathbf{M}$ in $\mathbb{R}^m$, in which $\|\mathbf{M} - \mathbf{M}'\| = \sigma_{m+1}$ (see Martin and Porter, 2012, for references and elaboration).

The basis elements of the truncated vector space (VSM_t) that $\mathbf{M}'$ in Equation 2.14 represents express linear combinations of the correlated contexts in the original vector space (VSM_o) that $\mathbf{M}$ in Equation 2.13 represents. Therefore, in contrast to a selection process, the basis elements of VSM_t cannot be directly labelled using the contexts employed in the VSM_o. Instead, they show *latent concepts* that express weighted combinations of contexts. Latent concepts may capture certain paradigmatic similarities, often called *high-order* structures, between the context elements employed in VSM_o (see Leopold, 2005, for further mathematical explanation).[1]

Interpretation of the attached variances to context elements justifies different applications of truncated SVD. Turney and Pantel (2010) enumerate *latent meaning*, *high-order co-occurrence*, *sparsity reduction*, and *noise reduction* and leave the door open for further innovative applications. Under the assumption that the covariance of context elements indicates their similarity,[2] truncated SVD can be seen as a technique that exploits the Euclidean norm to measure similarity between context elements. Truncated SVD groups contexts into latent concepts such that it captures *latent meaning* and *high-order co-occurrences*; consequently, SVD truncation results in a vector space VSM_t that expresses entities in a *latent semantic space*.

For instance, in the LSA model, a truncated SVD model represents the semantic relationships between documents using latent concepts that are derived from a document-by-word model. The latent concepts, also called *latent topics*, may capture synonymy relationships between words and enhance similarity measurements (Martin and Berry, 2011).[3] Consequently, the introduction of the latent concepts, which are more general than the contexts employed originally, results in the *sparsity reduction*. SVD truncation, however, does not guarantee generation of most suitable combinations of contexts for an intended application. For instance, in a cross-language information retrieval task performed on Wikipedia articles, Cimiano et al. (2009) report that truncated SVD does not enhance the obtained results.

[1] See also Sahlgren (2006, chap. 7) who suggests that the enhancements in TOEFL experiments with the LSA model are the result of encoding paradigmatic relations between context words using the truncated SVD.

[2] That is, the Euclidean distance between context elements in the transposed model.

[3] This argument can be generalised if synonymy relationship replaces *a paradigmatic relationship.*

Dimension reduction by truncated SVD implies that contexts associated with large variance express discriminative information. By the same token, under the Gaussian assumption of noise, the low variance contexts are supposed to be unimportant and noisy. Therefore, the truncation of SVD using highest singular values, as suggested in Equation 2.14, can be viewed as a filtering procedure that eliminates noise. The performance of noise reduction using SVD, however, depends on the distribution of the co-occurrences of linguistic entities and the context elements. While SVD truncation can be applied to remove Gaussian noise from data (e.g., white noise from sinusoidal signals), it fails with noise of a non-Gaussian nature. For instance, observations such as Figure 2.7a indicate that the co-occurrences of words in documents follow a non-Gaussian distribution (see also Sichel, 1975). Therefore, the use of SVD truncation for noise reduction is not effective in models that are based on the co-occurrences of words.

SVD is sensitive to the measurement scales of the context elements being analysed. Because a truncated SVD model retains linear combinations of the context elements that maximise the magnitude of variance, it is biased towards context elements that have larger variation values. If contexts are presented using values of different scales, then SVD truncation will be in the favour of context elements that are presented in scales of larger magnitude. Therefore, a scaling process is necessary before performing the SVD computation (see Jackson, 2004, for further information on methods of scaling).

In dimensionality reduction using the SVD truncation, the degree of dimension reduction should be decided by choosing a value for m in Equation 2.14. An optimum value for m is determined by the structure of the underlying data as well as the intended application. Direct selection of an optimum m, however, remains an open question (Martin and Berry, 2011). Therefore, the value of m is often found by an exhaustive evaluation. In order to find the most satisfactory m, a performance measure suitable for the intended application is defined to compare several values of m. For example, in an information retrieval task, the estimated precision per m in retrieval tasks decides the best degree of dimension reduction.

The computation of SVD for dimension reduction entails solving a linear equation that finds eigenvectors. For a given n-dimensional vector space, direct solution to this equation, known as the Gram–Schmidt process, is computationally trivial and of $O(n^2)$ complexity. Accordingly, the direct computation of truncated SVD for mapping $\mathbb{R}^n$ to $\mathbb{R}^m$, $m \ll n$, demands computational complexity proportional to $O(n^2 m)$. In practice, the singular values are approximated using iterative techniques such as the Lanczos method and its variations that take advantage of the sparseness of vector spaces (see Saad, 2003, chap. 7). In k iterations, the m largest singular values of a vector space are calculated directly and therefore the computational complexity of the transformation process is decreased to $O(nkm)$.

Truncated SVD requires the vector space of higher dimension than the targeted reduced dimension—that is, $\mathbf{M}$ in Equation 2.13—to be constructed prior to the process

of dimension reduction. However, this may not be desirable when dealing with large corpora. The size of a vector space that is built using a regular method of context matrix formation is a function of the size of the corpus. A regular context matrix formation associates entities to context elements, often using normalised values induced from the observed co-occurrence frequencies across the corpus. Such that context matrix becomes computationally intractable when the corpus size increases (e.g., Figure 2.7b). In a term-by-document model, for instance, the dimension of the vector space *dim* before the dimensionality reduction process is equal to the number of documents in the corpus $|c|$; appending n new documents to the corpus corresponds to an increase in the dimension of the vector space—that is, $dim = |c| + n$. This is a non-trivial task when the corpus is big or its size increases at a sharp rate such as Web-scale information extraction tasks.

In addition to the aforementioned problems, the basis of the vector space with reduced dimensionality, which the data is projected onto, is also required to be devised prior to the projection task. If the structure of the data that is being analysed changes, the basis of the projected vector space also changes. Therefore, every time data is updated (i.e., a new context element or linguistic entity is added to the vector space), SVD should be recalculated in order to generate a suitable projection. This limitation is also generalised to dimensionality reduction techniques that are based on matrix factorisation techniques other than SVD, such as QR and ULV decomposition. *Random indexing* is an alternative dimension reduction technique that alleviates these issues.

The random indexing (RI) method, which is first introduced by Kanerva et al. (2000) for the construction of a word-by-document model and further delineated by Sahlgren (e.g., see Sahlgren, 2005, 2006), utilises all the advantages listed above to create a vector space model of semantics at reduced dimension. Sahlgren (2005) delineates the RI method in the form of a two-step procedure that consists of the construction of *a*) *index vectors* and *b*) *context vectors*. In the first step, each context is assigned to exactly one index vector $\vec{r}_{c_k}$. Sahlgren (2005) indicates that an index vector is a randomly generated high-dimensional vector, in which most of the elements are set to 0 and only a few to 1 and -1. In the second step, the construction of context vectors, each target entity is assigned to a vector of which all elements are zero and that has the same dimension as the index vectors. For each occurrence of an entity, which is represented by $\vec{v}_{e_i}$, in a context, which is represented by $\vec{r}_{c_k}$, the context vector for the entity is accumulated by the index vector of the context—that is, $\vec{v}_{e_i} = \vec{v}_{e_i} + \vec{r}_{c_k}$. The result is a vector space model, which is constructed directly at reduced dimension.

The procedure in the RI technique can be better explained by an example of a word-by-document model. In the first step of the process, each document in the corpus—that is, a context element—is assigned to an index vector $\vec{r}_{d_i}$ of dimension m much smaller than n. Each word in the corpus is then assigned to an empty context vector $\vec{v}_{e_w}$—that is, all the elements of the vector are set to zero—and dimension m. The context vectors assigned to words can then be updated through a sequential scan of the corpus.

For each occurrence of a word in a document d_i, its context vector $\vec{v}_{e_w}$ is updated such that $\vec{v}_{e_w} = \vec{v}_{e_w} + \vec{v}_{d_i}$. Given n documents and p words in the corpus, instead of a matrix $\mathbf{M}_{p\times n}$, the RI procedure results in a matrix $\mathbf{M}'_{p\times m}$ that represents the vector space model at reduced dimension by the factor $\frac{n}{m}$.

The random indexing method, thus, can be used to address a number of issues that are faced when using SVD truncation. For instance, in RI method, adding new context elements to the model is realised by adding new index vectors, without demanding a recalculation of the projection. Chapter 4 provides a comprehensive description and mathematical justification of the RI method. As is shown in Chapter 4, the RI method belongs to a category of dimensionality reduction techniques that are based on *random projections*.

The linear methods, such as SVD truncation and the RI method, have often been criticised for their inability to capture nonlinear structure of data beyond the ℓ_2-norm (or, the second-order statistics). In contrast to linear techniques that assume the text data lies on a linear sub-space of a high-dimensional space, a number of dimensionality reduction techniques go beyond linearity assumption and explicitly reconstruct the data in an *embedded manifold*. These methods, known as nonlinear dimension reduction techniques, are further categorised by their underlying theory (e.g., see Van der Maaten et al., 2009, for a survey). In the context of natural language processing, Kohonen's (1990) self-organising maps is, perhaps, the most familiar example of a nonlinear dimensionality reduction technique (see also chapters of Honkela, 1997). Some experiments suggest that nonlinear methods do not necessarily outperform linear techniques, specially on real-world datasets containing noise or having discontinuous or multiple sub-manifolds (Huang and Yin, 2012).

While the use of neural networks and non-linear transformations are gaining popularity in several domains of study in distributional semantics, the study of these methods is left for another occasion. This section has only scratched the surface of the dimensionality reduction techniques that are most commonly applied in the distributional models of semantics. In the context of distributional models of semantics, dimension reduction techniques are still maturing with respect to several factors such as their performance, efficiency and underlying theories, as well as the data and intended applications of models. Figure 2.8 provides readers with a summary of the discussions in this section.

2.3.4 Similarity Measurement

The computation of vector similarities, which serves as a quantitative approximation of semantic relatedness between entities, is often the last step of the processes. As discussed in Section 2.2.1, a vector space model of semantics is endowed with structures called inner product, norm, and distance that are employed to define similarity measures between

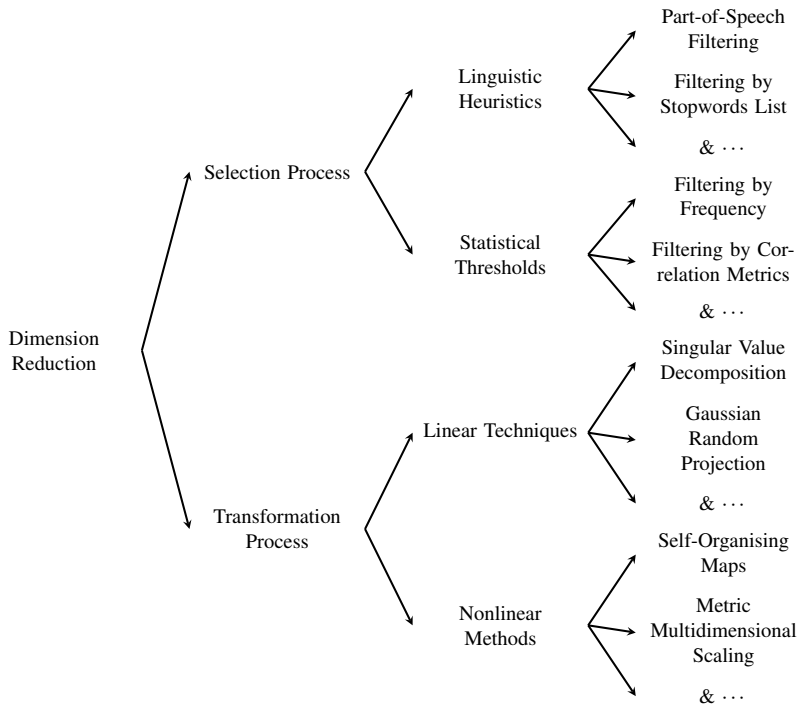


Figure 2.8: A map of dimensionality reduction techniques. Although not all methods neatly fall into the provided categorisation, it provides readers with a summary.

vectors. The cosine similarity and the Euclidean distance[1] are the familiar examples of such measures in the $\mathbb{E}^n$. Given the definition of inner product in $\mathbb{E}^n$ by Equation 2.8 and vectors $\vec{v}_i = \langle v_{i1}, v_{i2}, \cdots, v_{in} \rangle$ and $\vec{v}_j = \left\langle v_{j1}, v_{j2}, \cdots, v_{jn} \right\rangle$, the cosine similarity of $\vec{v}_i$ and $\vec{v}_j$ is given by the inner product of vectors when their length is normalised:

$$cos(\vec{v}_i, \vec{v}_j) = \frac{\langle \vec{v}_i, \vec{v}_j \rangle}{\|\vec{v}_i\|_2 \|\vec{v}_j\|_2} = \frac{\sum_{k=1}^{n} v_{ik} v_{jk}}{\sqrt{\sum_{k=1}^{n} v_{ik}^2 \sum_{k=1}^{n} v_{jk}^2}}. \tag{2.15}$$

Likewise, the Euclidean distance is defined as:

$$d(\vec{v}_i, \vec{v}_j) = \|\vec{v}_i - \vec{v}_j\|_2 = \sqrt{\sum_{k=1}^{n} (v_{ik} - v_{jk})^2}. \tag{2.16}$$

As indicated by the numerator of Equation 2.15, the cosine similarity calculates the overlap between the vectors and thus it is a measure of the shared context elements between linguistic entities. In contrast, the Euclidean distance conveys the differences between corresponding context elements and thus it is a measure of discrepancy between linguistic entities.

[1] Also called 2-norm or ℓ_2.

Name	Formula
Dice	$s_{\text{Dice}}(\vec{v}_i, \vec{v}_j) = \frac{2\sum_{k=1}^{n} v_{ik} v_{jk}}{\sum_{k=1}^{n} v_{ik}^2 + \sum_{k=1}^{n} v_{jk}^2}$
The harmonic mean	$s_{\text{HM}}(\vec{v}_i, \vec{v}_j) = 2\sum_{k=1}^{n} \frac{v_{ik} v_{jk}}{v_{ik} + v_{jk}}$
Jaccard	$s_{\text{Jaccard}}(\vec{v}_i, \vec{v}_j) = \frac{\sum_{k=1}^{n} v_{ik} v_{jk}}{\sum_{k=1}^{n} v_{ik}^2 + \sum_{k=1}^{n} v_{jk}^2 - \sum_{k=1}^{n} v_{ik} v_{jk}}$

Table 2.3: Examples of similarity measures in the inner product family. In these equations, similar to the cosine similarity in Equation 2.15, the inner product of vectors in the denominators of the formulas is normalised using different values. These measures show the commonality between vectors.

The familiar Euclidean norm in a real vector space $\mathbb{R}^n$ can be replaced by other p-norms, $1 \leq p < \infty$,[1] in order to calculate similarity between vectors in ℓ_p-normed spaces—that is, a vector space that is endowed with the ℓ_p norm.[2] For a given vector $\vec{v}$ in an ℓ_p-normed space, the Euclidean norm $\|\vec{v}\|_2$ in Equation 2.8 is generalised to

$$\|\vec{v}\|_p = \Big(\sum_{i=1}^{n} |v_i|^p\Big)^{\frac{1}{p}}. \tag{2.17}$$

Hence, the distance between the two vectors $\vec{v}_i$ and $\vec{v}_j$ in a ℓ_p-normed space—also known as the Minkowski distance—is given by

$$d_p(\vec{v}_i, \vec{v}_j) = \|\vec{v}_i - \vec{v}_j\|_p = \sqrt[p]{\sum_{k=1}^{n} |v_{ik} - v_{jk}|^p}. \tag{2.18}$$

Amongst the d_p distances, besides the Euclidean distance, the ℓ_1 distance, also known as the Manhattan distance or city block distance, has been employed for semantic similarity measurement.

As discussed earlier, the collected frequencies of the co-occurrences of linguistic entities and context elements can be interpreted in mathematical frameworks other than the vector space model. Therefore, it is common to employ probabilistic and information-

[1] For $0 < p < 1$, the p-norm is called a quasi-norm, as it does not satisfy the triangle inequality in the definition of a norm. However, ℓ_0—that is, $p = 0$—does not satisfy the homogeneity condition, and it is thus not a norm. From ℓ_0 one can arrive at the definition of the Hamming distance. While Hamming spaces have been also used for similarity measurement in distributional semantics, their study goes beyond the scope of this thesis. A comprehensive study on the use of Hamming spaces in distributional semantics can be found De Vine (2013) and De Vries (2014) (see also Gionis et al., 1999).

[2] Remember from Section 2.2.1 that $\mathbb{B}^n$ is a $\mathbb{R}^n$ that is endowed with the Euclidean norm (i.e., the ℓ_2-norm); it is thus an *ℓ_2-normed space*.

Name	Formula
Bray-Curtis	$s_{\text{BC}}(\vec{v}_i, \vec{v}_j) = \frac{\sum_{k=1}^{n} \lvert v_{ik} - v_{jk} \rvert}{\sum_{k=1}^{n} v_{ik} + v_{jk}}$
Canberra	$s_{\text{Can}}(\vec{v}_i, \vec{v}_j) = \sum_{k=1}^{n} \frac{\lvert v_{ik} - v_{jk} \rvert}{\lvert v_{ik} \rvert + \lvert v_{jk} \rvert}$
Gower (see Pavoine et al., 2009, for description)	$s_{\text{Gower}}(\vec{v}_i, \vec{v}_j) = \frac{1}{k} \sum_{k=1}^{n} \frac{\lvert v_{ik} - v_{jk} \rvert}{w_k}$
Soergel	$s_{\text{Soe}}(\vec{v}_i, \vec{v}_j) = \frac{\sum_{k=1}^{n} \lvert v_{ik} - v_{jk} \rvert}{\sum_{k=1}^{n} \max(v_{ik}, v_{jk})}$

Table 2.4: Examples of (dis)similarity measures in the ℓ_1 distance family. In the definition given for s_{Gower}, w_k indicates the range of the values for the kth element of vectors.

theoretic measures for similarity calculation. Many of these measures satisfy the axioms listed in the definition of distance (norm)[1] and therefore can be categorised in an ℓ_p distance family. From this perspective, a d_p distance can be normalised in different ways to design new distance measures. However, there are many other measures that do not satisfy the required axioms for a distance metric. An example of this categorisation is given by Cha (2007).

Cha provides a survey of similarity measures and their properties. He enumerates dozens of similarity measures and groups them according to their syntactic characteristics (i.e., the homogeneity of their formulas), the correlation between their generated results in a clustering task, and the caveats in their implementations. Following his survey, Tables 2.3, 2.4, and 2.5 provide a list of similarity measures analogous to ℓ_1, ℓ_2, and inner product formula, respectively (to verify the given definitions, see Deza and Deza, 2006, 2014). Examples of information-theoretic similarity measures are given in Table 2.6.

Amongst his observations, Cha suggests that the family of inner product measures, such as cosine, generates results closely related to ℓ_2 distance. In addition, the results generated by the two distance metrics d_a and d_b are highly correlated if $d_a = cd_b$ or $d_a = 1 - d_b$. Particularly, in distributional semantics, because of the sparseness of vectors, a method of *smoothing* is required to alleviate these problems, which is a major research problem on its own (e.g., see Chen and Goodman, 1999). For example, in these cases, one solution is to replace zero with a very small value—that is, the additive smoothing technique.

There is an extensive body of research on learning distance metrics, with detailed studies that go beyond the scope of the discussion in this section. In these methods, a distance metric is altered, often using a weight normalisation mechanism in order to reflect

[1] See on page 31.

Name	Formula
Clark	$s_{\text{Clark}}(\vec{v}_i, \vec{v}_j) = \sqrt{\sum_{k=1}^{n} \left(\frac{v_{ik}-v_{jk}}{v_{ik}+v_{jk}}\right)^2}$
Symmetric χ^2	$s_{Sym_{\chi^2}}(\vec{v}_i, \vec{v}_j) = \sum_{k=1}^{n} \frac{(v_{ik}-v_{jk})^2}{\max(v_{ik}, v_{jk})}$
Weighted Euclidean	$s_{\text{WE}}(\vec{v}_i, \vec{v}_j) = \sqrt{\sum_{k=1}^{n} \frac{(v_{ik}-v_{jk})^2}{w_k}}$

Table 2.5: Examples of (dis)similarity measures in the ℓ_2 distance family. In the definition given for s_{WE}, w_j denotes a weighting value.

Name	Formula
Bhattacharyya	$s_{\text{B}}(\vec{v}_i, \vec{v}_j) = -\ln \sum_{k=1}^{n} \sqrt{v_{ik} v_{jk}}$
Hellinger	$s_{\text{H}}(\vec{v}_i, \vec{v}_j) = \sum_{k=1}^{n} (\sqrt{v_{ik}} - \sqrt{v_{jk}})^2$
K-Divergence	$s_{\text{KD}}(\vec{v}_i, \vec{v}_j) = \sum_{k=1}^{n} v_{ik} \ln(\frac{2v_{ik}}{v_{ik}+v_{jk}})$
Kullback-Leibler	$s_{\text{KL}}(\vec{v}_i, \vec{v}_j) = \sum_{k=1}^{n} v_{ik} \ln(\frac{v_{ik}}{v_{jk}})$

Table 2.6: Examples of information theoretic similarity measures adopted in the vector space models, assuming vectors represent probabilities.

a set of given constraints on similarities (e.g., w_k in the definition of s_{WE}[1] in Table 2.5 and s_{Gower} in Table 2.4). The weight normalisation problem is usually modelled as a learning task in the framework of an optimisation problem. For instance, given constraints in the form of 'x is close to y' for a set of pairs of vectors x and y, Xing et al. (2002) suggest a method that learns a distance metric. Schultz and Joachims (2004) suggest a similar technique, however, when the constraints are given in the form of a set of triplets such as 'x is closer to y than it is to z'. In the machine learning literature, metric learning is often studied as a learning scheme for feature weighting (see Kulis, 2013, for survey and references). These techniques, thus, can be perceived in combination with the preceding weighting step in which more indicative contexts are assigned to higher weights in order to increase their impact on the similarity measure. Alternatively, given a known set of related vectors, it is possible to compare distance metrics in order to choose the most suitable one.

Bullinaria and Levy (2007) provide a comparison between several similarity measures. The comparison is carried out by studying the results of four different experiments

[1] In this context often called the Mahalanobis distance.

Similarity Measure	Rank	Experiment			
		(TOEFL)	(Distance)	(Synt. Cluster)	(Sem. Cluster)
	1	Hellinger	Kullback-Leibler	City Block	Kullback-Leibler
	2	Bhattacharya	City Block	Hellinger	Hellinger
	3	City Block	Hellinger	Bhattacharya	Bhattacharya
	4	Kullback-Leibler	Bhattacharya	Cosine	City Block
	5	Cosine	Cosine	Kullback-Leibler	Cosine
	6	Euclidean	Euclidean	Euclidean	Euclidean

Table 2.7: Performance of similarity measurements with respect to each other in Bullinaria and Levy's (2007) experiments; rank 1 shows the best-performing similarity measure.

that employ word-by-word models:

TOEFL : From four given choices, a word is selected that has the closest meaning to a target word in a dataset consisting of 80 questions.

Distance : Similar to the TOEFL test, but the distance between a pair of semantically related words (e.g., lettuce and cabbage) is compared with the distances between 10 randomly chosen pairs of words from a set of 200 words in order to assess the structure of the model at a larger scale than the TOEFL test.

Syntactic Clustering : The distance between a target word's vector and the centre of a cluster that represents its syntactic category is measured and the ratio of words that are closer to their real syntactic category than another is defined as the performance measure. The test is limited to 100 words from 12 different syntactic categories.

Semantic Clustering : The same test as above, however, for semantic categories. The performance measure is defined as the ratio of words that are closer to their own semantic category than others. The experiment is limited to 530 words in 53 semantic categories.

Table 2.7 represents the performance of the similarity measures in the tasks explained above. The results shown in the table are limited to when vectors are weighted such that they represent the conditional probabilities $p(w_c|w_t)$, where w_t and w_c are the target and context word, respectively. As is shown in the table, the best performing measure varies from one experiment to another. While a similarity measure such as city block has a constant superior performance with respect to measures such as the Euclidean and the Cosine, this relationship does not hold for other metrics such as the Kullback-Leibler and Hellinger. An approximate difference between the best and worst performing measures is shown in Table 2.8. As suggested in Cha (2007), the Kullback-Leibler and Hellinger, and the Cosine and Euclidean show similar behaviours in Bullinaria and Levy's (2007) experiments.

	Experiment			
	(TOEFL)	**(Distance)**	**(Synt. Cluster)**	**(Sem. Cluster)**
Best ≈ %	75	90	92	71
Worst ≈ %	65	85	82	58

Table 2.8: Approximate values for the best and the worst performances of similarity measurements in Bullinaria and Levy's (2007) experiments.

In an earlier experiment similar to Bullinaria and Levy's (2007) *Distance* test, Lee (2001) provides a report of the performance of similarity measures which is analogous to the result shown in Table 2.7. She also reports that the city block outperforms the cosine, and the cosine outperforms the Euclidean distance, whereas a weighted Kullback-Leibler method, called *skew divergence*, gives the best performance. However, Bullinaria and Levy (2007) show that the cosine similarity can outperform all the similarity measures in every one of the above tasks when a suitable weighting process, such as *pointwise mutual information*, substitutes the probability weighting. In another experimental setup, Curran (2004, chap. 4) suggests that the Dice and Jaccard outperform the cosine similarity measure. In an automatic synonym acquisition task, Shimizu et al. (2008) report that a weighted Euclidean measure, which obtains weights through a supervised learning method, outperforms all other metrics in their experiment. In their reported experiment, the cosine and the Jaccard are the next-best-performing measures and are listed above the Euclidean and, contrary to the above reported-experiments, the city block measure.

In an alternative approach, instead of mere performance comparison, Weeds et al. (2004) suggest an attributive comparison of similarity measures. In a synonym detection task, Weeds et al. (2004) compare 10 various similarity measures by investigating the frequency characteristics of target words and their closest neighbour words given by a similarity measure. They correlate the frequency of the obtained neighbour words to their *distributional* and *semantic generality* and accordingly classify similarity measures into three groups. The first group of measures are those that are biased towards selecting high-frequency, and thus more general, words. The second group of measures are those that are more sensitive to to low-frequency, thus more specific, words. The third group consists of those measures that are in favour of with a similar frequency to target words. In their experiment, the cosine and the skew divergence are categorised in the first group, whereas the Jaccard and the harmonic mean are classified in the third group. A similar study of similarity measures in an information retrieval context is given in Jones and Furnas (1987).

Mathematically speaking, the distribution pattern of entities in a vector space determines the performance of similarity measures. In the absence of *a priori* knowledge of the distribution of data, similarity measures are often evaluated empirically. An ap-

proach, such as that described above or proposed by Lin (1998b), is employed to interpret similarity measures' performance and elucidate their differences. With such intuition, as an example, Lee (1999) suggests that for sparse models, commonality-based similarity measures—such as the Dice and the cosine—are expected to outperform those that are based on differences such as the Euclidean distance. In information retrieval, Jones and Furnas (1987) compare the sensitivity of several similarity measures to *within-object* and *between-object* differences and conclude in favour of the cosine measure.

The literature reviewed unanimously agrees that various similarity measures exhibit different behaviours in different tasks and thus there is no single superior measure for all applications. In a given application, therefore, the choice of a similarity metric is likely to affect the quality of the observed result.

2.3.5 Orchestrating the Processes

This section concludes our discussion on the processes in vector space models of semantics by emphasising the importance of a holistic approach to their design and implementation. As described, the goal of the chain of processes introduced in this section is to simulate a sense of semantic relatedness between vectors that represent the linguistic entities being modelled. As is explained, the semantic relatedness is ultimately translated into the proximity of vectors, which is transpired by a notion of similarity measure. The efficacy of measures is predominated by the distribution of vectors, which, in turn, is a function of the answer to the earlier question of 'what the context elements are'. A change in context elements results in the transformation of the vectors' distribution in the model and thus it is highly likely to cause redesign in the subsequent processes, amongst them the similarity measurement.

Usually, the use of one specific method in one of the processes introduced in this section limits the choice of methods that are available to be applied in the remaining processes. For instance, the choice of a random projection with Gaussian random matrix for the dimensionality reduction limits the options for the similarity measurement. Similarly, the choice of random indexing limits the options for the weighting process. As discussed in Chapter 4, using random indexing for collecting co-occurrences results in a Euclidean vector space model; therefore, the use of similarity measures other than the ℓ_2 distance family cannot be justified, at least mathematically. In other words if for any reason, the use of norms other than ℓ_2 is preferable, then a Gaussian random projection technique such as random indexing cannot be employed. With the same rationale, using SVD truncation is not justified when similarities are measured using a metric other than the ℓ_2 distance family.

Moreover, one method can neutralise the advantages of another method. For example, normalising the Euclidean distance by the inverse of the variance of contexts in a vector space model that is induced by SVD truncation has no effect on the obtained similarities. In the same way, if SVD truncation is used for the dimensionality reduc-

tion, a weight scaling is recommended as a pre-processing step. Nonetheless to say that Likewise, a number of similarity measures, such as the familiar Euclidean and cosine similarity, are equivalent if the vectors are normalised to unit length. In contrast, as the experiment shows, the right combination of methods in the above processes can enhance the observed results dramatically.

Last but not least, the suggested cascaded architecture for processes, in which one process is applied after the other in a pipeline, may not be applicable or desirable in a real-world application. The suggested arrangement of the processes and the clear-cut boundaries between them are given solely for clarity in the presentation. The software architecture of an implemented distributional semantic method may require a complex sequence of interactions far beyond what is described in this section.

2.4 Classification in Vector Spaces

In a vector space model of semantics (VSM), a variety of machine learning algorithms can be employed to address a range of classification and clustering problems. A class is a set of entities that can be identified by characteristics that all its members share. The classification problem is the task of automatic assignment of entities to classes. However, if the classes are not known prior to the assignment task, then the task is called clustering. Clustering thus is the task of grouping entities by their mutual characteristics in such a way that the members of a group, called a cluster, are more similar to each other than to the members of other clusters in a sense. The classification task is usually referred to as *supervised learning*, whereas the clustering task is known as *unsupervised learning*.

Familiar examples of such tasks are document classification and clustering. In a document-by-term model, instead of measuring similarities between a pair of documents, or a query and a document, the documents are categorised by certain criteria, for instance, their subject areas. In this example, if the subject areas are known beforehand—for example, the subject areas are limited to science and art—the task is called document classification. However, if the subject areas are not known beforehand, then the task is called document clustering and it organises the documents, for this given example, into groups that give a sense of the subject areas. Using different context types, documents can be classified, instead of by subjects area, by their relatedness, style, theme, sentiment, author characteristics, etc.

In the combination of a learning technique with a vector space model, the learning algorithm compares the vectors by its own implemented logic of similarity. In a vector space model, which interprets the meaning of linguistic entities such as documents using the geometry of vectors, a class or a cluster refers to a collection of vectors that form a region. A learning algorithm consequently identifies these regions. This perception implies the assumption that entities of the same class or cluster form a *contiguous* region

and regions of different classes do not overlap.[1] Violations of these assumptions are the main causes of inaccuracy in classification and clustering tasks.[2]

A classification task—that is, supervised learning—can be formalised by a mapping function f. For a vector space V and an output space L, which consists of a finite set of category labels l, the classification process is given by $f : V \mapsto L$. The mapping function f is *learned* by a machine learning algorithm during a process called *training*. The training process chooses a function that *best* estimates the relationship between the input vectors and the output labels from a given set of instances $T \in V \times L$, which is called the *training dataset*. If $L = \mathbb{R}$, then the classification task is called regression. For $|L| = 2$, the task is called *binary* classification. If $|L| > 2$, then the task is called *multi-class* or *multi-way* classification. In a clustering task—that is, unsupervised learning—the T and L are not presented explicitly. Instead, criteria—such as the cardinality of L, the way similarities are compared, and a relationship between members of clusters—are given.

These learning algorithms are the subject of vibrant scientific research in a framework known as *statistical learning theory*. The comprehensive study of these methods, therefore, requires dedicated research. In this section, however, the surface of topics in statistical learning theory are scratched and only learning methods that take a geometric approach to a classification task are introduced. These methods classify data in a normed space and, thus, are compatible with the interpretation principles of vector space models, which are introduced earlier in Section 2.2. The methods introduced in this section are used later in this thesis.

In statistical learning theory, learning procedure is formalised using a mapping function $(V \times L)^n \mapsto \mathcal{F}$. In this definition, $\mathcal{F}$, which is called the hypothesis space, is a space of functions $f_m : V \mapsto L$, where V and L are the input vector space and the output label space, respectively. The learning algorithm searches in $\mathcal{F}$ for a function that best approximates the relationship implied between the vectors and the labels by the set of n samples from $(V \times L)^n$. This formalisation is based on two assumptions. First, it is assumed that the data is being classified, that is, the set of n tuples $\langle \vec{v}, l \rangle$, are drawn independently and identically from a fixed but unknown joint probability distribution $p(\vec{v}, l)$. Second, in order to assess the quality of learning, it is assumed that there is a notion of *loss* or error that can determine, for a given input vector, the discrepancy between the expected label and the label predicted by a f_m. This is indicated by a *Loss function* $loss : L \times L \mapsto \mathbb{R}$. For a given vector $\vec{v}$ and the expected label l, $Loss(l, f_m(\vec{v}))$ gives the error of f_m.

By these assumptions, the goal of the learning process is to find a $f_o \in \mathcal{F}$ that minimises the average error. For $f \in \mathcal{F}$, the average error, which is also called the *risk* of f $R(f)$is given by:

$$R(f) = \int_{V \times L} Loss(l, f(\vec{v})) \mathrm{d}p(\vec{v}, l). \tag{2.19}$$

[1] Evidently, it can be also interpreted as a corollary to the distributional hypothesis.

[2] Alternatively, in a probabilistic framework, classes are interpreted as hidden properties of entities, often named latent variables.

However, $R(f)$ cannot be computed because the probability distribution $p(\vec{v}, l)$ is unknown. The learning problem formalised above can be solved using a variety of approaches. From one perspective, similar to the proposed taxonomy of the distributional methods in Section 2.1.2,[1] the learning techniques can be categorised into methods that provide a solution using probability estimation techniques or methods that interpret the learning problem in a metric space.[2] As cited by Jain et al. (2000), however, under certain assumptions on the probability distributions, the two approaches are equivalent.

In the probability-based category, two major approaches to approximate $R(f)$ can be recognised. In the first group of methods, it is assumed that the type of the distribution of data is known; thus, a probability model with a number of fixed parameters can be used to estimate $p(\vec{v}, l)$. Consequently, the training dataset T is used to estimate the value of the model's parameters. For instance, assuming the data has a Gaussian distribution, the joint probability is estimated using the mean and variance of the data samples in T. The familiar algorithm in this group is the naïve Bayes classifier.

The second group of probability-based methods, in contrast to the former methods, do not assume prior knowledge of the type of data distribution. These techniques estimate $p(\vec{v}, l)$ by the observation of the data samples provided in T. In distributional semantics, the Blei et al.'s (2003) latent Dirichlet allocation for uncovering *topic models* is a well-known example of these methods. Both category of methods listed above can exploit the learned joint distribution in a reverse fashion; that is, given a class label l, they can synthesise examples of context elements related to l. Hence, the probability-based methods are often known as *generative* approaches.

On the other side, one category of learning techniques—often named as *discriminative* methods—bypasses the probability estimation and approximates $R(f)$ directly. A subcategory of these methods adopt a *geometric approach* in the sense that they reformulate a learning task as the construction of decision boundaries in a metric space. The support vector machine algorithm and the k-nearest-neighbours technique are the familiar examples in this category. These methods approximate $R(f)$ from the training set T using an *induction principle* such as *empirical risk minimisation* (ERM). Given n samples $\langle \vec{v}_i, l_i \rangle$ in T, the *empirical risk of function* f over T is given by:

$$R_{\text{emp}}(f) = \frac{1}{n} \sum_{i=1}^{n} Loss(f(\mathbf{v}_i), l_i). \tag{2.20}$$

It is expected that the function f that has a small empirical risk (i.e., $R_{\text{emp}}(f)$) will also have a small risk (i.e., $R(f)$). It is proved that for f of *finite complexity*, $R_{\text{emp}}(f)$ converges to $R(f)$ when $n \to \infty$ (see Evgeniou et al., 1999, for further explanation). Therefore, it is assumed that the goal of a learning task can be achieved—that is, finding the $f_o \in \mathcal{F}$ that

[1] See Figure 2.2.

[2] This inventory can be expanded, for example, by adding information-theoretic-based approaches, etc.

minimises the risk $R(f)$—by finding the f_o that minimises the empirical risk $R_{\text{emp}}(f)$:

$$f_o = \underset{f \in \mathcal{F}}{\operatorname{argmin}} R_{\text{emp}}(f) = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \left(\frac{1}{n} \sum_{i=1}^{n} \ell(f(\mathbf{v}_i), l_i)\right). \tag{2.21}$$

Accordingly, $R_{\text{emp}}(f)$ is employed as a quantifiable method for the assessment of the *generalisation* ability of f_o—that is, it is assumed that if f_o has a small $R_{\text{emp}}(f)$, then it also has a high generalisation ability.[1] Whereas research in machine learning investigates developing algorithms by suggesting induction principles other than ERM,[2] and imposing restriction on the complexity of $\mathcal{F}$[3], in this thesis, the scope is limited to the use of memory-based k-nearest neighbours (k-nn) algorithms. The k-nn algorithm implies that the f_o that determines class labels by taking an average of the class labels of instances in T that are close to input $\vec{v}$ has the lowest R_{emp}.[4]

2.4.1 The *k*-Nearest Neighbours Algorithm

The k-nearest neighbours (k-nn) algorithm is a learning technique that is explained by the geometry of vectors in space (Cover and Hart, 1967).[5] In k-nn, instances of data—that is, vectors—are classified based on the class of their nearest neighbours. It is a two-step process: in the first step, the k closest vectors to the data item being classified are located; in the second step, the class label of the data item is determined using the class label of these nearest neighbours.

Given a vector space V and a training dataset $T \in V \times L$, where L is a finite set of class labels, it is assumed that there exists a distance function $d : V \times V \to \mathbb{R}$, such as that given in Section 2.3.4, that assigns a distance value $d(\vec{v}, \vec{t})$ to each pair of vectors $\vec{v} \in V$ and $\vec{t} \in T$. In its simplest form, when $k = 1$, for an input vector $\vec{v} \in V$, T is searched for the $\vec{t}$ that has the least distance to the $\vec{v}$ and its class label is assigned to the $\vec{v}$. This classification task can be formalised by the mapping function nn that returns corresponding label $l \in L$ of vector $\vec{t}$ such that:

$$nn(\vec{v}) = l_{\vec{t}}, \text{ where } \vec{t} = \underset{\vec{y} \in T}{\operatorname{argmin}}\, d(\vec{v}, \vec{y}). \tag{2.22}$$

[1] Although in real-world applications, this assumption does not hold. If the training dataset is small or the hypothesis space $\mathcal{F}$ is large, then there are many functions that can satisfy Equation 2.21. Under these conditions, however, using ERM may not necessarily result in a function that has a high generalisation ability. Under such circumstances, a function f_o that shows a high performance during the learning procedure shows a poor performance when dealing with data samples other than T. This is often called *overfitting*.

[2] Which its study goes beyond the scope of this thesis.

[3] For example, using the assumption that the target function f_o is in the form of a *linear discriminant function*.

[4] Also, see Kulkarni and Harman (2011), for further elaboration of statistical learning theory and stimulating questions.

[5] Perhaps more intuitive than SVM.

By the same token, the $nn(\vec{v})$ can be generalised to k neighbours. After finding the k closest instances in T to $\vec{v}$, that is $\{t_1 \cdots t_k\}$, the most straightforward approach—known as *unweighted voting*—is to assign the majority class label among the k nearest neighbours to the data item being classified:

$$k\text{-}nn(\vec{v}) = l_y, \text{ where } l_y = \underset{l \in L}{\operatorname{argmax}} \sum_{i=1}^{k} \delta(l, f(\vec{t_i})), \tag{2.23}$$

where $f(\vec{t_i})$ denotes the class label of $\vec{t_i} \in T$, and $\delta(x, y)$ is a function that compares the two class labels x and y, that is:

$$\delta(x, y) = \begin{cases} 1 & x = y \\ 0 & x \neq y \end{cases}. \tag{2.24}$$

However, a *distance weighted* method can replace the unweighted sum of labels:

$$k\text{-}nn(\vec{v}) = l_y, \text{ where } l_y = \underset{l \in L}{\operatorname{argmax}} \sum_{i}^{k} w_i \delta(l, f(\vec{t_i})), \tag{2.25}$$

where w_i is real valued function on the distance between $\vec{v}$ and instances from the training set. For example, the weight function can be defined as an inverse of the distances between $\vec{v}$ and $\vec{t_i} \in T$, that is:

$$w_i = \begin{cases} 1 & x = y \\ \frac{1}{d(\vec{v}, \vec{t_i})} & x \neq y \end{cases}. \tag{2.26}$$

Similarly, as suggested by Daelemans et al. (2009) and Cunningham and Delany (2007), w_i can be defined using an exponential function based on Shepard's (1987) justification, that is:

$$w_i = e^{-\alpha d(\vec{v}, \vec{t_i})^{\beta}}, \tag{2.27}$$

where α and β are constant, often $\alpha, \beta = 1$, that are used to control the power of exponential decay factor. The k-nn algorithm, thus, can be alternated by adopting different approaches for assigning class labels through definitions of δ and w.

The k-nn algorithm is known to be a *lazy-learning* technique, which means that it does not require a training procedure prior to the classification task. The induction takes place during run-time and using training data samples that are presented explicitly. The main computation in the learning and classification task is the scoring of training vectors against an input vector in order to find the k nearest neighbours. The k-nn, therefore, is also known as an *example-based* or *case-based* learning technique. It is a simple yet effective method of classification that has been widely used in many applications.

However, the application of k-nn requires selecting the k value where it is dependent on the distribution of the data is being classified, the distribution of training samples,

and the metric that is used to find the nearest neighbours. The value for k is usually selected by a heuristic technique such as cross-validation. In general, larger values of k are believed to reduce the effect of noise; however, this makes class boundaries less distinct. For small values of k, the k-nn method is also known to be sensitive to the presence of noisy or irrelevant data (Yang, 1999). In addition, when the number of training instances increases, the performance of k-nn reduces. However, these limitations have been actively addressed by a large number of research.

Besides the mathematical account given above, based on the application's context, there are several interpretations of the k-nn algorithm. In its simplest form, k-nn can be seen as a ranking system in which a threshold is used for assigning a class label to an input vector (e.g., Bustos and Navarro, 2004). In the context of distributional semantics, however, the k-nn algorithm can be best explained by the substantial research efforts that are often flagged by the term *memory-based language processing* (Daelemans and van den Bosch, 2005)—that is, as described by Daelemans (1999), a union of the two tradition of analogy-based language models in linguistics, and k-nn learning technique in artificial intelligence.

As summarised in Daelemans and van den Bosch (2010), k-nn can be seen as a *similarity-based reasoning* process in which the learning process is analogous to memorising (i.e., storing) a set of examples. Whereas a number of learning techniques employ a meta-language such as rules to construct an abstract representation of text data (known as *eager* learning methods), k-nn relies directly on the text data to perform the classification task. Hence, similar to the discussion in Chapter 1, k-nn offers an empiricist method of classification. Training text samples are, thus, can be kept in their original format with no alteration. As a result, it can be suggested that:

- the process of classification in k-nn is more intuitive than methods that use an abstract representation of the training data;
- language exceptions and less frequent patterns, which are often ignored by a generalised representation of the training data, can be handled effectively;
- even using a very small set of training examples, k-nn shows a reasonable generalisation ability.

In the context of this thesis, the k-nn method is employed for two of its particular characteristics:

- its plausible compatibility with the distributional hypothesis and its intuitive explanation of the classification task;
- its memory-based learning strategy.

As explained above, the former characteristic introduces k-nn as a cognitively plausible data-driven approach for similarity-based reasoning, whereas the second characteristics make it exceptionally flexible and suitable for implementing an interactive learning algorithm. No training process is required to develop a model and the examples can be

added or removed at anytime during the deployment of the method. Hence, the memory-based learning is a simple yet effective approach for the iterative development of terminological resources—in which the model can be updated as a user annotates and organises terms. Lastly, the example-based classification method can be easily scaled out, for example, with the help of *MapReduce programming model*—which is an important feature in big text data analytics.

2.5 Chapter Summary

The discussion in this chapter started by giving an overview of the distributional hypothesis and the vector space models of semantics, which form the theoretical basis for the proposed methods in this thesis (i.e., Section 2.1). Vector spaces as an algebraic structure are described in Section 2.2.1; Section 2.2.2 explained how these algebraic structures are employed to model and interpret distributional properties of linguistic entities in various contexts in order to capture meanings. In Section 2.2.3, this discussion was accompanied by a survey of the employed context elements and types of semantic models that have been employed in different text processing tasks; for example, to address problems in applications such as information extraction and retrieval.

Processes in vector space models of semantics were a major part of the discussion in this chapter (i.e., Section 2.3). The steps that are necessary to build a vector space model are reviewed. These processes, from the vector space construction to the similarity measurement process, were discussed in detail. Accordingly, Section 2.4 explained the use of learning techniques in distributional semantic models, in which an emphasis was put on the methods that employ the geometry of vectors in order to perform a classification task. Particularly, in Section 2.4.1, the k-nearest neighbours algorithm, which will be employed later in this thesis, was introduced.

Chapter 3

Computational Terminology: Term Extraction and Classification

Systematic terminology collection, management, and maintenance are significant tasks in any application that deals with knowledge. These processes are the subjects of study in terminology and subsequently computational terminology. Apart from established applications in knowledge management systems, recent endeavours such as information retrieval, machine translation, ontology learning and semantic search have stimulated research in terminology mining. With a focus on term extraction, this chapter provides an overview of the basic definitions and tasks in computational terminology.

Section 3.1 provides an overview of terminology mining methods. Sections 3.2 describes the common employed mechanism in these methods. Section 3.3, and 3.4 details the processes of candidate term extraction and scoring, respectively. Section 3.5 touches the subject of term organisation. Section 3.6 briefly discusses the use of machine learning techniques in terminology mining. Finally, the chapter concludes with a brief discussion on the evaluation in Section 3.7.

3.1 Introduction to Computational Terminology

Computational terminology embraces a set of algorithms that extract terms from *special corpora* and arrange them in domain-specific knowledge structures such as a vocabulary, thesaurus or ontology. As defined by Sinclair (1996), special corpora contain sublanguage material. Hence, according to this definition, computational terminology is concerned with the automatic analysis of *languages for special purposes*, for example, in order to facilitate interoperability when communicating specialised knowledge.

Computational terminology inherits its complexities from difficulties in the interpretation of meaning in language. In terminology, these complexities are often summarised by the question what counts as a *term*? The Oxford Dictionary defines a term as:

> a word or phrase used to describe a thing or to express a concept, specially in a particular kind of language or branch of study (Term[Def. 1], 2014).

According to the International Organisation for Standardisation (ISO), a term is

> a verbal designation of a general concept in a specific subject field (ISO 1087-1, 2000).

As stated by Cabré (2010), linguistically, terms are *lexical units* and carry a special *meaning* in particular *contexts*. A lexical unit is often considered as a *lexical form*—a single token, part of a word, a word or a combination of these—that is paired with a single meaning and serves as the basic element of a language's vocabulary. Similarly, as suggested by L'Homme (2014), terms are the denomination of items of knowledge—that is, concepts.

According to their lexical forms, terms are usually classified as *simple* or *complex*. Simple terms consist of one token; complex terms are composed of more than one token or word. For instance, 'lexicography' and 'multilingual terminology management' are, respectively, examples of a simple and a complex term in the domain of computational linguistics. The extracted lexical units constitute a *terminological resource*, also known as *terminology*: a specialised vocabulary of knowledge in a domain. Terms and their use are studied in a relatively young discipline, which is also called *terminology* (Cabré, 2003; Kageura, 1999):

> the field of activity concerned with the collection, description, processing and presentation of terms (Sager, 1990).

While terminology can be approached from several perspectives—for example, as a branch of philosophy, sociology, or cognitive science—terminology is dominantly considered a linguistic and cognitive activity. Modern terminology is therefore pursued within a linguistic framework and as the study of specialised languages—that is, languages for special purposes (Faber and Rodríguez, 2012).

(a) The General Theory of Terminology (b) Modern Terminology

Figure 3.1: Association of meaning in the GTT compared to recent theories of terminology: the GTT starts with concepts. Terms are only labels and denote concepts existing a priori. In recent theories of terminology such as the CTT, however, terms are treated like other linguistic units. They signify concepts in a communicative context. In the figures above, the dashed lines indicate the direction in which the meaning of a term is elaborated according to these theories. The indicated communicative context (the dotted triangle in Figure b) can be extended in a number of ways, for example, by considering the application of terms.

In terminology, the meanings of terms and the process of concept denomination are studied within the framework of a *theory of terminology*. As stated in Cabré (2003), a theory of terminology elaborates the fundamental problem of interpretation of meaning into a set of questions in which the definition of a terminological unit—and its characteristics—is often the nucleus, for example:[1]

- What are the basic units of terminological knowledge?
- How are they defined and acquired?
- Where are they observed?
- How are they recognised and what are their characteristics?

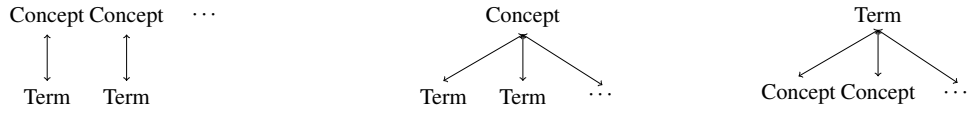
The general theory of terminology (GTT) by Wüster (1974, as cited in Campo (2013, chap. 2)) is widely recognised as the first theory of terminology. The GTT, which is also known as traditional terminology, puts concepts first; terms are merely unambiguous labels for concepts that exist a priori (Faber and L'Homme, 2014) (Figure 3.1a). Put simply, in the GTT, knowledge is gained independently of the language, and thus the usage of terms. As implied by the given definition in ISO 1087-1(2000), the GTT has been one of the major adopted theories amongst terminologists.[2] The sequel to the GTT can also be found in early computational terminology research (e.g., see Ananiadou, 1994). Consequently, the GTT regards terms and concepts as having mono-referential relationships (Figure 3.2a). The objective behind the GTT, understandably, is to eliminate ambiguity in natural language in order to improve clarity in technical communication.

In an authoritative institutional organisation[3] that promotes or enforces standards, terms can be *made* and shared in a top-down manner; hence, the meaning of terms can be

[1] For a comprehensive list of questions and possible answers, see Cabré (2003).

[2] Accordingly, Felber (1982) defines terminology as 'the combined action of groups of subject specialists (terminology commissions) of specialised organisations'.

[3] Here, the *organisation* can be a scientific discipline, a technical domain, a company, etc., that requires a specialised language for effective communication.



(a) One-to-One Relation (b) Synonymous Relation (c) Polysemous Relation

Figure 3.2: Relationships between terms and the concepts they signify: Figure 3.2a illustrates a mono-referential, unambiguous relationship between terms and concepts. Figure 3.2b shows an ambiguity that may arise when several terms denote the same concept in a synonymous relation. Figure 3.2c illustrates an ambiguous term-concept relation, a polysemous relationship where a term may denote several concepts.

interpreted by the mechanism described in the GTT.[1] However, in practice and in many organisations, new terms are introduced in a bottom-up *synthesis* process. A lexical form (which may or may not be newly invented) in contexts that bear a concept (which may or may not be newly invented) is used frequently inasmuch as it becomes a term[2] in the organisation. In practice, therefore, terms can be ambiguous: a term can refer to several concepts—similar to polysemy–homonymy in lexical semantics (Figure 3.2c); or, contrariwise, a particular concept can be denoted by several terms (Figure 3.2b). Heid and Gojun (2012) suggest that the rapid evolution of organisations as well as multi-players that are involved in an uncoordinated way, specifically in multidisciplinary domains, reinforces this situation and thus contributes to term ambiguity.

In contrast to the GTT, recent theories of terminology—for example, the communicative theory of terminology (CTT) by Cabré (1999, chap. 3) and the lexical-semantic approach that is promoted by Faber and L'Homme (2014)—acknowledge the situation stated above and take an empiricist approach to terminology in the sense that the meanings of terms, and as a result the elements of domain knowledge, are not preconceived. Simply put, in modern theories of terminology, knowledge is a posteriori that is dependent upon the language. Hence, terms are understood differently with regards to the communicative context, for example, by the text surrounding them, the application they are used in and so on.

Putting this discussion into the structuralist framework of distributional semantics, terms are *linguistic units* that signify concepts by syntagmatic and paradigmatic relations that they hold in a specialised communicative discourse (Figure 3.1b).[3]

The importance of a theory of terminology lies in the fact that it outlines practical issues that must be addressed in terminology. According to the adopted theory of termino-

[1] it is, perhaps, best demonstrated in the applications of controlled natural languages.

[2] That is, a norm.

[3] It becomes evident that the main difference between the GTT and modern terminology theories is the interpretation of the process of pairing concepts and lexical units—that is, as suggested in Chapater 1, the result of the GTT's rationalist vs. the CTT's empiricist approach to comprehend the process of gaining knowledge and communicating meanings.

logy, computational terminology tasks are formulated differently and are thus approached from alternative perspectives. Consequently, the perspective presented by a theory of terminology establishes boundaries for the definition and classification of the tasks that are currently addressed in computational terminology. However, as indicated by Cabré (2003) in her *theory of doors*[1], the mere fact of the existence of these issues is not affected by the way they are formulated. Research in computational terminology addresses these practical issues. Inevitably, although computational terminology is often associated with the task of automatic term recognition, it goes beyond that and embraces a number of research tasks.

In computational terminology, the task of automatic term recognition (ATR) has been at the centre of discussion as an essential component of modern information systems. In ATR, the input is a large collection of documents, that is, a special corpus, and the output is a terminological resource. In ATR, the meaning of the generated terms is interpreted in a wide spectrum of concepts in the domain that is being investigated and represented by the input domain-specific corpus. Since ATR facilitates the automatic construction of terminological resources, it is a significant processing resource in knowledge engineering tasks for a multitude of applications such as information retrieval and machine translation.

As articulated by Kageura and Umino (1996), ATR deals with the computation of measures known as *unithood* and *termhood*. It is believed that the majority of terms in a domain are complex terms. Unithood indicates the degree to which a sequence of tokens can be combined to form a complex term. It is, thus, a measure of the *syntagmatic* relation between the constituents of complex terms: a lexical association measure to identify collocations. In the absence of explicit linguistic criteria to distinguish complex terms from other natural language text phrases, a unithood measure construes the problem of complex term identification as the identification of *stable* lexical units (Sager, 1990).[2]

Termhood, on the other hand, 'is the degree that a linguistic unit is related to $\cdots$ some domain-specific concepts' (Kageura and Umino, 1996). It characterises a *paradigmatic* relation between lexical units—either simple or complex terms—and the communicative context that verbalises domain-concepts. Termhood, thus, conveys the measurement of meaning. In the absence of a formal answer to the question 'what are domain-specific concepts?'—for instance, see the discussions in Laurence and Margolis (1999); Fodor and Lepore (2012)—devising a termhood measure for distinguishing terms and non-terms is a difficult and often conflictual task.

[1] In the theory of doors, Cabré (2003) elaborates on her position as follows:

> This theory is suitably represented by the image of a house; let us assume a house with several entrance doors. We can enter any one of its rooms through a different door, but the choice of the door conditions the way to the inside of the house. The internal arrangement of rooms is not altered, what does change is the way one chooses to get there.

[2] See Evert (2004) on applications of lexical association measures for the identification of lexical units.

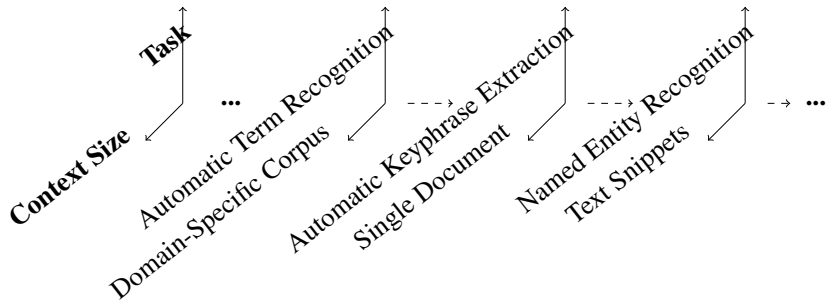


Figure 3.3: Lexical unit extraction tasks and the granularity in which they interpret the of the meanings of a lexical item. Although all the tasks listed in this figure extract lexical items that denote salient domain concept, the scope and the granularity in which they interpret the meanings of lexical units is different. At the highest level of granularity, automatic term recognition tasks investigate the meanings of lexical units across the set of documents that constitute a domain-specific corpus. At the least level of granularity, entity recognition tasks decide about the meanings of lexical units in a given snippet of text. The diagram can be extended by adding new dimensions that take into consideration characteristics of the communicative context other than the size of the input text. This diagram can form a basis to suggest taxonomies of tasks that extract lexical units from text.

Computational terminology, however, embraces a set of techniques other than ATR, which also aim to extract stable lexical units. In ATR, the communicative context is a domain-specific corpus. Therefore, ATR should not be confused with tasks such as keyword extraction and entity recognition that bear a close resemblance to it. These tasks are similar to ATR in the sense that they extract stable lexical units from natural language text. However, they are different from ATR, because the meaning of the extracted lexical units—thus the termhood measure—is interpreted in a context other than a special corpus (Figure 3.3). For example, an automatic keyphrase extraction algorithm pulls out lexical units from a single document that best describe the content of this document. Both unithood and termhood must be also measured in automatic keyphrase extraction. However, the criterion for their definition and the information available for their computation are different from ATR.

Categorisation of term extraction tasks can be extended by considering characteristics of communicative contexts other than the size of the input text. Cabré et al. (2007) classify term extraction tasks as *intermediary* and *terminal* with respect to the end-users' interaction with the extracted terminological resources. An intermediary application constructs a terminological resource—for example, a domain-specific ontology—that will be exploited as a component of other information systems; for example, to address problems such as information extraction and retrieval. Hence, in an intermediary application, end-users do not interact directly with the constructed terminological resource. However, in terminal applications, a terminological resource is constructed to be accessed and used directly by a particular user.

Besides the communicative context, the term extraction techniques are often classified by the linguistic characteristics of the extracted terms. For instance, Yangarber et al. (2002) distinguish tasks that address the extraction of *proper* names from those that focus on the extraction of *generalised* names. Accordingly, Yangarber et al. (2002) relate named entity recognition tasks to the former category of term extraction methods since their output is limited to the names of people, organisations, locations, and so on. For the latter category, they enumerate methods that extract mentions of concepts such as the name of biological agents based on the rationale that these terms are not proper names. Similarly, one may place keyphrase extraction methods in this category.

Tasks that are addressed in computational terminology can be further distinguished by the direction in which they bridge the gap between terminological resources and text. Recent developments of ontological resources have stimulated a research strand that targets the reverse of intermediary term extraction tasks. The goal of these applications is to fill the gap between an available knowledge base—for example, an ontology—and natural language text. In these tasks, given a particular concept in a knowledge base (e.g., a class and its instances in an ontology), a method—which is called *term mapping* by Krauthammer and Nenadic (2004)—decides if this concept or its instances have been mentioned in a given text snippet. Entity linking, which has been promoted by the series of Text Analysis Conferences,[1] is another term that characterises these research efforts (see also Rao et al., 2013).

In contrast to term mapping techniques, there are methods that organise constituent terms of a terminological resource into a variety of classes. Given a terminological resource, in these methods, the usage of terms in a corpus is assessed to decide their membership in concept classes. If the classes are known prior to the assignment task, then the task is known as term classification (e.g., see Nigel et al., 1999). Otherwise, if the classes are *unknown*, the task is called term clustering (e.g., see Dupuch et al., 2014). As described in Chapter 5, from a linguistic point of view, these methods address the identification of *hypernym/hyponym* relationships between the entries of a terminological resource. Krauthammer and Nenadic suggest that these three tasks—that is, term recognition, term classification, and term mapping—are essential to form a closed loop between terminology and natural language text, for the facilitation of automatic construction and maintenance of terminological resources (Figure 3.4).

A more elaborate taxonomy of techniques in computational terminology can be obtained by discerning elements and characteristics of the communicative context other than what is discussed here. As implied in the discussions, besides the methods that are named above, the outlook of 'terms as units of language'—as named by L'Homme (2014)—underlines the requirements for addressing a number of challenges such as *term variation* and *acquisition of semantic relations* for systematic management of termino-

[1] See `http://www.nist.gov/tac/about/`.

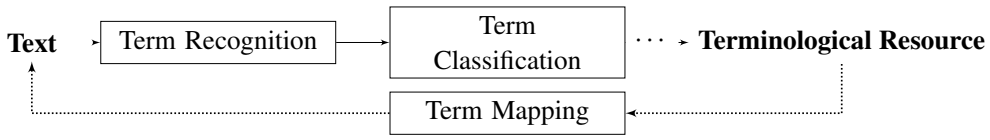


Figure 3.4: Significant processes in computational terminology. Whereas term extraction and classification techniques distil a terminological resource from text, a set of techniques in computational terminology try to bridge the gap between terminological resources—such as domain ontologies—to natural language text.

logical resources. Each of these problems is an active research topic in computational terminology, beyond the scope of this thesis.

In the remaining sections, the common mechanism of term extraction methods is discussed in Sections 3.2. The involved processes, that is, candidate term extraction and the scoring procedure are explained in Sections 3.3, and 3.4, respectively. In Section 3.5, organising terminologies is discussed briefly. The use of machine learning methods and a number of term classification techniques are explained in Section 3.6. Section 3.7 concludes this chapter by explaining the evaluation of theses methods.

3.2 Prevalent Mechanism in Term Extraction Tasks

As suggested in Nakagawa (2001a), the algorithms for term recognition are usually in the form of a two-step procedure: candidate term extraction followed by a term scoring and ranking process (Figure 3.5).

Candidate term extraction deals with the term formation and the extraction of candidate terms. The latter is not a trivial task since usually there are no clear differences between a term and general words and phrases in the language at the text surface level. In particular domains such as molecular biology, a share of new terms—for example, the name of new genes—are single-token simple terms. These terms are usually formed and invented using a set of common predefined morphological patterns. The identification of these patterns, for example, as suggested in Ananiadou (1994) and in Zweigenbaum and Grabar (1999), can be helpful in the process of candidate term extraction. However, this kind of term formation is not employed in a large number of domains. Therefore, solutions such as morphological pattern analysis may not always be useful for identifying simple terms. Furthermore, as suggested by Nakagawa (2001a), multitudes of terms are complex terms in the form of uninterrupted collocations. Similar to other types of multi-word expressions, distinguishing these complex terms from phrasal structures in the language has remained a research challenge. Several methods for the extraction of candidate terms are suggested, which will be reviewed in the next section.[1]

[1] As can be inferred, this processing pattern is very similar to the extraction of multi-word expressions.

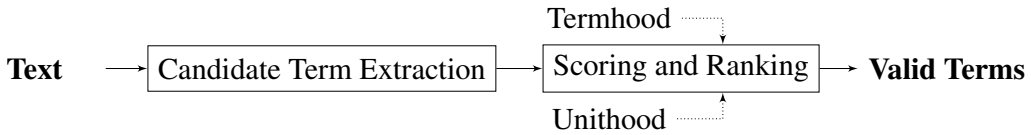


Figure 3.5: Prevalent architecture of terminology mining methods.

Although several Categorisations of the scoring and ranking methods can be given from a methodological point of view (e.g., statistics-based, machine learning-based, rule-based, etc.) or by the kind of information that is exploited for weighting (e.g., linguistic-based, statistical-based, hybrid), as stated earlier, all these techniques rely on the text and take a corpus-based distributional approach to score and rank terms. The usage of candidate terms in a communicative context (e.g., domain-corpus) is formulated in a machine-tractable format—for example, in the form of a contingency table or a vector space model. To compute a score for each candidate term, the collected data is then assessed using statistical measures, similarity metrics, language models or a set of rules. The scoring methodology is determined by the metric employed for scoring candidate terms (e.g., only termhood, only unithood, or a combination of both) as well as the objective of the task in hand, which often decides the type of paradigmatic relation that the termhood measure characterises.

This two-step term extraction procedure can be followed by a number of additional processes. For instance, following the two-step procedure, a term selection process may discard a number of extracted terms that have a score below a particular threshold. The strategy for designing this kind of post-processing technique is determined by the intended application for the extracted terms and therefore is not considered as a core process in a term extraction task. Similarly, depending on the employed methodology, a number of pre-processings—for example, part-of-speech tagging, syntactic analysis, etc.—might be required prior to the two-step term extraction procedure.

3.3 Candidate Term Extraction

The first step in most term extraction tasks is to extract candidate terms from text. As suggested earlier, candidate term extraction is a non-trivial task. Terms' boundaries cannot be distinguished easily from other words and phrases in the text surface. Whereas earlier research in term extraction suggested that terms show particular morphological or syntactic behaviours, recent research in terminology indicates that terms show a sim-

However, aside from the difference in scope of research, one notable difference between the research in multi-word expressions and terminology extraction is the scoring procedure in these areas. In term extraction, both unithood and termhood are employed to weight terms, whereas multi-word expression research leans towards unithood measurement (see Baldwin and Kim, 2010, for an overview of research in multi-word expressions).

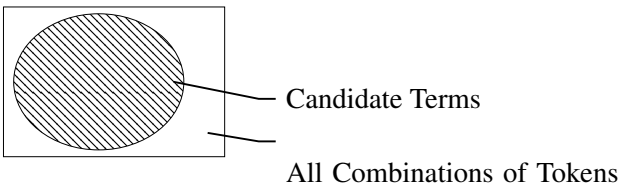


Figure 3.6: Output of the candidate term extraction process: a subset of all combinations of tokens in input text corpus.

ilar linguistic behaviour as general words and phrases in a language. From a radical perspective, in a given text, any combination of tokens and words can be a term. Consequently, choosing candidate terms can be seen as the problem of finding a subset of tokens' sequences (which are likely to be terms) in an exponentially large search space, thus resulting in an *NP-hard problem*. Luckily, a number of linguistic observations suggest particular criteria for the terms' linguistic behaviours—for example, the frequency and the length of terms—which are utilised to define a set of heuristics to limit this search space (Figure 3.6).

In a limited number of domains, knowledge workers may have a guideline for introducing new terms, particularly simple terms. For example, in molecular biology the names of genes are often a combination of letters and numbers. Similar regulations can be found in automotive engine technologies. As suggested earlier, these observations, coupled with the traditional terminology's outlook, led to a number of research methods that assume term formation is a planned, conscious, and well-structured process (Ananiadou, 1994). Hence, in order to extract candidate terms, these methods pay extra attention to the internal morphosyntactic structure of terms and often ignore the context in which they appear (Accordingly, Maynard and Ananiadou (2001) classify these techniques as *intrinsic* approaches). In these methods, a terminological resource is often available prior to the extraction task and it is employed to identify new candidate terms.

While the above-mentioned morphosynatic-based methods have been employed in a few domains, they are not applicable in a large number of sublanguages; for example, creation of new terms may not follow particular morphosyntactic patterns and a terminological resource may not be available prior to the extraction task. Besides, a simple search in a terminological resource shows that the majority of terms are multi-word complex terms. The extraction of these terms introduces additional complexity to the process of candidate term extraction.

Hence, apart from the aforementioned morphosyntactic-based methods that focus on the terms' internal structure, several other techniques have been introduced to address the problem of candidate term extraction. Five major methods can be identified for the extraction of candidate terms:

- the *n*-gram-based techniques;

- linguistic filtering using part-of-speech tag sequence patterns;
- linguistic filtering using syntactic relation patterns;
- techniques that rely on the presence of particular markers in text;
- contrastive approaches.

A combination of these techniques can also be employed to improve the results (e.g., see Aubin and Hamon, 2006). In the following section, each of these methods are described.

3.3.1 The *N*-Gram-Based Methods

In the context of candidate extraction, an *n-gram* is a contiguous sequence of n tokens from text. In n-gram-based methods, the n-gram is usually bound to a text window of a particular size (often, $1 \leq n \leq 6$). The most common size for n is two in which two-word collocations (bigrams) are considered as the potential candidate terms. In order to reduce the number of undesirable sequences of tokens and restrict the size of the set of the extracted candidate terms, a number of heuristics are employed to filter the extracted n-grams. For instance, n-grams that contain *stop words*—such as articles, particular prepositions, auxiliary verbs, etc.—are discarded. A major advantage of the n-gram-based techniques is that they can be employed in the absence of linguistic analysis tools. Hence, they allow the terminology extraction task to be carried out with purely statistical approaches. Therefore, n-gram-based techniques are desirable when dealing with the under-resourced languages where the linguistic analysis tools are usually not available (e.g., see Pinnis et al., 2012).

Compared to other techniques of candidate term extraction, the use of n-gram-based methods often results in lower precision. The n-gram-based methods generate a large set of candidate terms of which the number of correct terms compared to incorrect terms is expectedly very low. For example, in the context of a keyphrase extraction application, Hulth (2003) investigates the performance of a few candidate term extraction methods including an n-gram-based technique. In her methodology, the extracted candidate terms using different techniques are classified as valid or invalid keyphrase using the same supervised machine learning technique. Subsequently, she compares the keywords assigned by the classifier with a list of the author's provided keywords in order to estimate the performance of the candidate term extraction techniques. In these experiments, the employed n-gram-based method shows one of the worst performances. Similar results can be found for an automatic term extraction task in Zadeh and Handschuh (2014a).

3.3.2 Part-of-Speech-Based Methods

Linguistic filters in the form of part-of-speech (PoS) tag sequence patterns have been widely employed for the extraction of candidate terms (Justeson and Katz, 1995). These methods are often affiliated by the linguistic approaches to term recognition. In this category of techniques, patterns of particular PoS tag sequences are employed to extract

candidate terms. These patterns are often represented by regular expressions. The use of these patterns yields to the assumption that the construct of terms is restricted to grammatical structures of particular PoS sequences. For example, by observing the target domain's terms, Justeson and Katz (1995) only consider candidate terms that are composed of a combination of nouns (W_N), adjectives (W_A) and prepositions (W_P) and satisfy the following PoS pattern:

$$((W_A|W_N)^+|(W_A|W_N)^*(W_N W_P)^?(W_A|W_N)^*)W_N$$

Bourigault's (1992) LEXTER is another system that employs PoS-based linguistic filtering for the extraction of candidate terms. However, instead of defining desirable PoS patterns, LEXTER employs *negative* knowledge about the form of terminological units, by identifying patterns that do not meet the requirements for forming candidate terms. In the proposed approach, similar to noun phrase chunking, punctuations and particular PoS tags such as verbs and conjunctions—which Bourigault calls frontier markers—are used for determining the boundaries of sequences of tokens that can form candidate terms.[1] A recent example of this methodology can be found in Meyers et al. (2014).

Park et al.'s (2002) GlossEx is another example of a term extractor system that employs PoS tag sequence patterns to extract words and phrases in order to construct domain-specific glossaries. The automatic extraction of candidate terms in GlossEx is limited to the $P_{\text{Noun Phrase}}$ structure that is defined by the following regular expressions:

$$P_{\text{Noun Phrase}} = W^?_{\text{DT}}(W_{\text{VBG}}|W_{\text{VBN}})^? P^*_{\text{Modifier}}(W_{\text{NN}}|W_{\text{NP}}|W_{\text{NPS}}),$$

in which P_{Modifier} is defined as:

$$P_{\text{Modifier}} = ((W_{\text{JJ}}(W_{\text{CC}}W_{\text{JJ}})^*)|(W_{\text{NN}}|W_{\text{NP}}|W_{\text{NPS}})^?).$$

In these patterns, W_X denotes a word of the particular PoS category X in which X is a PoS tag from the inventory of the tags employed in the Penn Treebank Project. Table 3.1 shows the Penn Treebank PoS tags and their corresponding definitions (Taylor et al., 2003).

In contrast to the above-mentioned methods that define PoS sequence patterns—thus candidate terms—of arbitrary length, a number of research restrain the length of candidate terms. For instance, Daille (1995) limits the length of their employed patterns to four words, whereas Frantzi (1997) employs patterns that are only two words long. Empirical evidences show that the length of terms is often limited to a few words/tokens. For instance, Maynard (2000) states that in most applications the length of term is usually up to 4 words and it is extremely rare for a term to exceed 8 words in length. Hence, limiting the length of candidate terms may enhance the accuracy of the candidate term extraction

[1] The idea behind the method is best described in Abney (1992).

CC	Coordinating conj.	RB	Adverb
CD	Cardinal number	RBR	Adverb, comparative
DT	Determiner	RBS	Adverb, superlative
EX	Existential there	RP	Particle
FW	Foreign word	SYM	Symbol
IN	Preposition	TO	infinitival to
JJ	Adjective	UH	Interjection
JJR	Adjective, comparative	VB	Verb, base form
JJS	Adjective, superlative	VBD	Verb, past tense
LS	List item marker	VBG	Verb, gerund/present participle
MD	Modal	VBN	Verb, past participle
NN	Noun, singular or mass	VBP	Verb, non-3rd ps. sg. Present
NNS	Noun, plural	VBZ	Verb, 3rd ps. sg. present
NNP	Proper noun, singular	WDT	Wh-determiner
NNPS	Proper noun, plural	WP	Wh-pronoun
PDT	Predeterminer	WP$	Possessive wh-pronoun
POS	Possessive ending	WRB	Wh-adverb
PRP	Personal pronoun	LRB	Left bracket character
PP$	Possessive pronoun	RRB	Right bracket character

Table 3.1: The list of part-of-speech tags employed in the Penn Treebank Project: *ps.* and sg. denote *person* and *singular*, respectively.

process without necessarily decreasing its recall.

Using PoS-based filters implies the need for autoamtic PoS tagging prior to the process of candidate term extraction. Ittoo et al. (2010) highlight problems that can arise due to the presence of noise in the output of this automatic PoS tagging process, particularly when dealing with irregular texts with subtle language patterns and malformed sentences. For instance, in the reported experiment by Ittoo et al., authors noticed that many nouns in their evaluation corpus are tagged incorrectly as progressive-verbs, and therefore resulting in misleading and inaccurate detection of candidate terms. To make the employed PoS patterns tolerant to these errors and solve the problem, Ittoo et al. refer to the actual output of their employed PoS tagger and define patterns that encompass progressive-verbs:

$$(W^{?}_{\mathrm{VBG}})(W^{*}_{A})(W^{+}_{N})$$

where W_{VBG}, W_A, and W_N respectively denote progressive verb, adjectives, and nouns.

Dorji et al. (2011) use PoS patterns for the automatic extraction of candidate terms that are used as index terms in a document classification task. By observing appropriate terms in their application, Dorji et al. have adopted PoS sequence patterns with various lengths of two to ten words. However, instead of specifying the complete PoS sequence patterns, they define seven core patterns of lengths two to four words. These sequences of PoS tags can in turn be followed by an arbitrary number of nouns to form patterns of maximum length ten words. Similarly, Eck et al. (2010) only consider a subset of

Daille (1995)	$(AN\|NN)$
Frantzi (1997)	$(N\|A) + N$
Nakagawa (2001b)	$N^?$ $A(N\|A)^*N$ $NP_{OF}N$ F
Zervanou (2010)	$(A\|N\|V_{BG}\|V_{BN})^+N$ $(A\|N\|V_{BG}\|V_{BN})C(A\|N\|V_{BG}\|V_{BN})N$ $(A\|N\|V_{BG}\|V_{BN})^+NCN$ $NP(A\|N\|V_{BG}\|V_{BN})^*N$ $NP(A\|N\|V_{BG}\|V_{BN})^*NCN$ $NCNP(A\|N\|V_{BG}\|V_{BN})^*N$ $(A\|N\|V_{BG}\|V_{BN})C(A\|N\|V_{BG}\|V_{BN})N$ $(A\|N\|V_{BG}\|V_{BN})^+NCN$
Bonin et al. (2010a)	$N^+(P^+(N\|A)^+\|N\|A)$

Table 3.2: Proposed PoS sequence patterns for Candidate Term Extraction. A denotes adjectives; N denotes nouns; C denotes conjunctions; P denotes prepositions; P_{OF} denotes the preposition *of*; F denotes foreign words; V_{BG} denotes verbs in gerund form; and, V_{BN} denotes verbs in the past participle form.

noun phrases that do not contain any preposition. The use of PoS sequence patterns is not limited to what is reported here and has been widely employed in term extraction tasks (e.g., see Anick et al., 2014; Zervanou, 2010; Hsu, 2010; Bonin et al., 2010a; Barrón-Cedeño et al., 2009).

Apart from algorithmic variances, the coverage of patterns is the major difference between techniques that employ PoS-based patterns for candidate term extraction. The higher coverage of patterns yield a higher recall, but usually at the expense of lower precision. Preference for precision requires a strict filter which permits a limited sequence of words as candidate terms, whereas preference for recall demands a filter with relaxed restrictions on the permitted sequences of words (Frantzi et al., 2000a). In addition, Eck et al. (2010) emphasise that the choice of an appropriate PoS pattern depends on the common structures that are employed by the sublanguage of the corpus. The definition of patterns using PoS sequences, thus, is an open question and no best universal pattern can be found. The reported experiment by Hulth (2003) states that considering PoS tags can result in a dramatic improvement of precision. Moreover, in her evaluation, the highest recall has been reported for the candidate term extraction based on a set of PoS tag patterns (surprisingly even in comparison to the $n-$gram technique). Table 3.2 shows additional examples of the employed PoS sequence patterns in research literature.

3.3.3 Syntactic-Based Methods

Research literature reports the use of linguistic filters that employ syntactic relations for the extraction of candidate terms. The first category of these methods employs syntactic patterns for the identification of term variations rather than the extraction of candidate terms. For example, Jacquemin and Tzoukermann (1999) report the use of a transformational unification-based syntactic parser together with morphosyntactic analysis for the identification of term variants in a controlled vocabulary environment. If a dictionary of terms is available prior to the extraction task, this method can be used for generating candidate terms.

The second category of syntactic-based methods use shallow parsing for the extraction of candidate terms. Instead of the extraction of collocations with specific PoS patterns, noun phrase chunks are extracted as candidate terms (e.g., see Evans and Zhai, 1996; Nakagawa, 2001a; Fan and Chang, 2008)[1]. In the reported results by Hulth's (2003), this technique gives the highest precision amongst PoS-based and *n*-gram techniques. However, in an experiment that I have reported in Zadeh and Handschuh (2014a), whereas noun phrase chunking outperforms an *n*-gram-based technique, it underperforms a PoS-based method.

The third category of syntactic-based filters considers the role of compounding in term formation and employs syntactic relations according to the head-modifier principle (e.g., see Jakubíček et al., 2014; Hippisley et al., 2005). By observing the role of compounding in term formation, Hippisley et al. (2005) apply the head-modifier principle in compounding word formation for the extraction of complex candidate terms. According to the head-modifier principle, in a syntactic construct, one of the constituents acts as the head. The head has a strong association to the core semantics of the construct, and it is modified by the other dependent constituents. In the proposed method in Hippisley et al. (2005), candidate terms are extracted by identifying particular syntactic relations to the left and the right side of the head. The major advantage of these techniques is that the head-modifier principle can additionally be used for deconstructing complex terms. Therefore, the proposed approach by Hippisley et al. is more popular within the context of machine translation applications for multilingual term extraction.

A detailed description of a head-modifier-based technique for candidate term extraction can be found in Wong (2009). Using dependency relations, the proposed method starts with a search for the heads in a sentence. Using the acquired head-modifier information from the dependency parse, the head is then extended to both left and right direction to identify maximal-length noun phrases. In the proposed method, the head-driven filter restricts the PoS tags of modifiers to nouns (except possessive nouns), adjectives, and foreign words. This process is followed by the use of a statistical measure in order to attach terms that appear immediately after one another, or terms that are separated by a

[1] Perhaps, a number of methods that are listed in Section 3.3.2 can also be added under this category.

preposition or coordinating conjunction.

The use of syntactic relations for the extraction of candidate terms is not limited to the above-listed methodologies. For example, Seretan et al. (2004) describe a sophisticated technique for the extraction of multi-word complex terms. In the proposed method, a set of pairs of words that are connected directly through a syntactic relationship are first extracted. Instead of the sequence of tokens in the input corpus, the extracted pairs of words are searched for extracting candidate terms. The set of extracted pairs of words is then utilised for the extraction of compound words, idioms and collocations from French and English parallel corpora.

3.3.4 Methods Based on Particular Structures in Text

An alternative approach to candidate term extraction exploits specific properties of the input text. A growing numbers of research exploits the presence of mark-ups in input text to extract candidate terms. For instance, Brunzel (2008) uses the HTML mark-ups in order to extract candidate terms and Hartmann et al. (2011) and Toral and Munoz (2006) exploit the semi-structured representation of text in Wikipedia's articles in order to form a set of candidate terms. The use of these techniques therefore is limited to domains in which text is annotated by mark-ups.

In the same way, in particular domains, candidate terms can be extracted with the help of specific lexical patterns or the presence of mark-ups in input text. For instance, in biotechnology, Rindflesch et al. (2000) describe a method for the extraction of candidate terms that employs a list of general *binding words*. In the proposed application domain, the presence of *binding words* in a noun phrases qualifies it as a candidate term. Similar method for the extraction of disease risk factors for metabolic syndrome in biomedical text is reported by Fiszman et al. (2007). Fiszman et al. (2007) suggest the use of indicative words including specific lists of verbs and nouns. Similar methods are proposed in Hazen et al. (2011) for the extraction of terms related to imaging observations in radiology and in Gooch and Roudsari (2011) for the extraction of clinical terms.

3.3.5 Contrastive Approaches

Contrastive approaches exploit a reference corpus of general language to identify simple and complex candidate terms from input text (Drouin, 2004, 2003). To form the hypothesis space of likely candidate terms, these methods rely on one of the techniques listed in the previous sections, for example, an *n*-gram-based method. Candidate terms are extracted from both the target special corpus and a general language corpus (e.g., the British National Corpus[1] when processing English text) or a special corpus in knowledge domain other that the target special corpus. The extracted candidate terms and their frequencies in

[1] See `http://www.natcorp.ox.ac.uk/`.

these two corpora are exploited to distil a set of likely candidate terms in the given special corpus. Similar methods can be found in Basili et al. (2001).

3.3.6 A Summary of Methods

To summarise, methods that employ linguistic information such as PoS tags and syntactic relations demand more resources than methods that rely only on the text surface structure. Methods that employ PoS-based sequence patterns require a PoS tagger with an acceptable performance. Similarly, syntactic-based methods demand a form of chunking or a syntactic parsing prior to the extraction task. These methods have been reported to deliver high precision; however, their required resources may not be available for all languages or domains. On the other hand, the *n*-gram-based techniques do not require such resources and are language-independent. However, these methods are reported to have a low precision, which can diminish the performance of the subsequent ranking process. The application of techniques such as the use of text structure, or using lexical indicators may not be applicable to all domains. Lastly, as suggested in Bonin et al. (2010b), the use of contrastive techniques can enhance the results.

In real-world applications, in order to improve the results, a combination of the above-listed methods are employed. For instance, Aubin and Hamon (2006) consider a combination of PoS sequence patterns, head-modifier relationship as well as a contrastive technique to extract a list of candidate terms. In another example, Hulth (2003) reports the highest F-Score in her experiments when candidate term extraction is carried out using a combination of *n*-gram techniques and PoS tag sequence patterns.

3.4 Methods for Scoring Candidate Terms

In automatic term recognition tasks, the scoring and ranking process follows the extraction of candidate terms. It is assumed that the set of extracted candidate terms contains both *valid* and *invalid* terms. Put simply, a candidate term is valid if it denotes a concepts from the knowledge domain that is represented by the input special corpus to the term extractor.[1] Hence, the main goal of term scoring process is to distinguish valid terms from invalid terms. This goal is often achieved by a ranking and filtering mechanism. The scoring process assign a score to each candidate term, ideally according to the significance of the concepts that they represent in the target knowledge domain. After this process, candidate terms with a score below a certain threshold are usually discarded and the rest are ranked and accepted as valid terms for further processes (Figure 3.7).

Traditionally and from a methodological perspective, terminology extraction approaches are often classified as *linguistically-motivated*, *statistically-oriented*, and *hybrid*

[1] As discussed earlier in Section 3.1, there is no straightforward definition of valid terms.

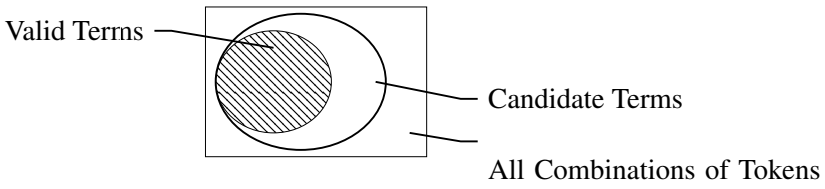


Figure 3.7: It is assumed that the output of the candidate term extraction process—that is, a subset of all combinations of tokens in input special corpus—contains both valid and invalid terms. Hence, a scoring and ranking process is employed to distinguish valid terms—that is, a subset of candidate terms.

methods (e.g., see Kageura and Umino, 1996, description on the topic). In this classification, often the candidate term extraction and scoring procedure are not heeded independently from each other. Hence, linguistically-motivated methods often encompass techniques that employ linguistic filtering for the extraction of candidate terms (although recent methods also use linguistic information as an attribute in statistical models).[1] In this classification, the statistical methods employ a mathematical model such as probabilities to perform the extraction task and ignore linguistic structure of terms and their context. As expected, the methods in this category often use *n*-gram-based methods for the extraction of candidate terms. The third category of methods in this classification, known as hybrid methods, offers solutions that combine both linguistic information and statistical measures. In fact, since the majority of the methods for terminology extraction rely on the text and adopt a corpus-based approach, they use a kind of statistical information derived from the corpus at some stage in the process. Hence, corpus-based methods are classified as statistically-oriented or hybrid technique.

Alternatively, as suggested earlier, the procedure of term extraction can be analysed and classified from a functional perspective: (a) methods that deal with the identification of atomic meaning-bearing lexical units and (b) methods that indicate the desirability of the extracted candidate terms as a unit of meaning in a terminology database. As suggested by Kageura and Umino (1996), in the former group, the focus is on the unithood measurement, thus the extraction of candidate terms that form stable lexical units. However, the focus of the former methods is on the termhood measurement, that is, scoring atomic lexical units by their significance in the target knowledge domain.

In the framework of distributional semantics, the computation of unithood is perceived as the identification of *syntagmatic* relationships between words that constitute a complex term. These relationships are often in the form of collocations. Therefore, the first category of methods deals with lexical association measures. A general account of these methods can be found in Evert (2004); Hoang et al. (2009); and, Pecina (2010). Similarly, in the framework of distributional semantics, the computation of termhood im-

[1]The use of linguistically-motivated approaches can be traced in information retrieval tasks for the problem of index term extraction (e.g., see Baxendale, 1958).

plies the identification of paradigmatic relations. These paradigmatic relations characterise the relevance of the meaning of terms to significant concepts in the knowledge domain and with respect to the communicative context, that is (in its simplest form), the special corpus.[1]

In corpus-based distributional approaches, the text and the statistics that are induced from its analysis are the major source of information to characterise these paradigmatic relations. As detailed in the next few sections, the statistical information about the usage of terms can be modelled and presented in a variety of ways, for example, as simple as computing *tf-idf* of terms to sophisticated learning algorithms. To characterise termhood, techniques other than corpus-based approaches are also feasible. For example, Maynard (2000) draws attention to the incorporation of knowledge-bases and their internal structure for the development of terminology extraction systems. The study of these methods, however, remains out of the scope of this thesis.

As described in the preamble of this section, statistical measures employed in terminology extraction can be classified into two categories: measures that address unithood and those that address termhood. However, drawing such a clear line is sometimes not possible (Kageura and Umino, 1996). According to Kageura and Umino (1996), statistical measures in terminology extraction are employed by relying on the following hypotheses:

- a lexical unit that appears frequently in a special corpus is likely to be a term of the domain knowledge that the special corpus represents;
- a lexical unit that appears only in one special corpus is likely to be a term of the domain knowledge that the special corpus represents;
- a lexical unit that appears more frequently in a special corpus than in a general language corpus is likely to be a term in the domain knowledge that is represented by the special corpus.

As discussed earlier, unithood is only defined for complex terms. The examples of statistical measures that have been used to measure unithood are numerous: Pearson's chi-square test and Log-likelihood, mutual information (e.g., as employed in Church and Hanks, 1990); coefficients for sequential data such as the Ochiai and Kulczynski coefficient suggested by Daille (1995); customised measures such as paradigmatic modifiability by Wermter and Hahn (2005); mutual expectation as suggested in Dias and Kaalep (2003), and so on.

Likewise, a long list of statistical measures have been employed to characterise termhood: inverse document frequency (*idf*) suggested in Jones (1972); term frequency–inverse document frequency (*tf-idf*) as used in Salton (1992) and its modifications such as Feiyu et al.'s (2002) *kfidf*; Frantzi and Ananiadou's (1996) C-*value* and NC-*value*; and, the statistical barrier measure proposed in Nakagawa (2001a) are a few examples.

[1] In fact, the communicative context goes beyond the special corpus. It is a complex system consisting of several elements such as the knowledge the users, the intended application, and so on.

3.4.1 Unithood Measures

Pearson's chi-square test (χ^2 test) is an intuitive statistical measure that can be used for characterising both unithood and termhood. χ^2 is measured by the comparison of the observed and expected frequencies under *the null hypothesis of independence*:

$$\chi^2 = \sum \frac{(f_o - f_e)^2}{f_e}, \tag{3.1}$$

where f_o is the observed frequency and f_e is the expected frequency (see Manning and Schütze, 1999, for further explanations). If f_o and f_e are derived from the observed frequencies in the collocations of constituent words in complex terms—for example, as suggested in Dunning (1993)—then the computed χ^2 value can be interpreted as a measure of unithood. However, if f_o and f_e are derived from the observed occurrences of terms in documents—for example, as suggested in Kilgarriff (1996)—then the computed χ^2 value can be interpreted as a measure of the terms's association to documents, hence termhood (see also Rayson et al., 2004). It is important to note that the chi-squared measure is meaningful only when the collected frequencies are greater than 5.

Log-likelihood ratio test (LL) is another statistical measure that has been used for characterising unithood. According to Dunning (1993), LL shows one of the best performances, particularly when frequencies are collected from small corpora. As described in Daille (1995) and Korkontzelos et al. (2008), LL can be seen as a refinement of the χ^2 test. Instead of relying on the assumption of a normal distribution of words in collocations, LL compares the observed frequency counts in a sub-corpus with the counts that would be expected in a reference corpus to measure the likelihood of co-occurrence. For bigrams $w_i w_j$, LL can be computed as follows:

$$\text{LL} = \log_2 \frac{P_s(w_i, w_j)}{P(w_i, w_j)}, \tag{3.2}$$

where $P(w_i, w_j)$ is the probability of observing w_i and w_j as a bigram in the reference corpus, and $P_s(w_i, w_j)$ is the probability of their occurance as bigram in the subset s of the corpus (i.e., the target domain). Similar to the interpretation of χ^2 test, a high LL means that observed and expected values diverge significantly, and thus indicates that w_i, and w_j do not co-occur by chance. In contrast, a LL value close to 0 indicates that w_i, and w_j do co-occur by chance. LL ratio is highest when w_i, and w_j only appear as bigrams next to each other. However, as mentioned in Korkontzelos et al. (2008), the LL ratio is also high for rare bigrams. Hence, the LL ratio of noisy bigrams such as typographical errors is also high, which consequentially may negatively affect the performance.

Similar to LL and χ^2, point wise mutual information (PMI) can also be used to measure the unithood of complex candidate terms in a corpus. PMI, however, estimates the expected probabilities using the products of the probabilities of the constituent words

of complex terms. For terms that consist of two words w_i and w_j, PMI is defined as:

$$PMI = \log_2 \frac{P(w_i, w_j)}{P(w_i)P(w_j)}, \tag{3.3}$$

where it is assumed that w_i and w_j appear independently. A high PMI value shows a strong association between the constituent words of the candidate terms. Hence, candidate terms that have high PMI value are assumed to be valid complex terms. In contrast to LL, PMI gives a low score to the rare candidate terms. The Dice measure, Z-score, and rank aggregation as suggested in Dinu et al. (2014) are other methods that can be used to evaluate the unithood of complex terms. As stated earlier, any method of sequential data modelling can be used to estimate unithood. Moreover, the use of statistical information other than words occurrence information is also feasible. For example, Tsvetkov and Wintner (2014) construct of a Bayasian network by integrating diverse statistical information to extract multi-word expressions.[1]

3.4.2 Termhood Measures

The tf-idf measure, a term weighting score often used in information retrieval, is perhaps one of the most applied statistical measures for characterising termhood. In automatic term recognition tasks, tf-idf is usually used as a baseline for the comparison of termhood measures (Zhang et al., 2008). The tf-idf score is the product of two statistics: inverse document frequency and term frequency. Inverse document frequency $idf(t_i)$ measures the general importance of a term t_i in a collection of documents D by counting the number of documents that contain t_i, usually but not necessarily in a logarithmic scale:

$$idf(t_i) = \log \frac{|D|}{\left|\left\{d_j \in D : t_i \in d_j\right\}\right|}, \tag{3.4}$$

where $|D|$ denotes the cardinality of D, and the denominator indicates the number of documents that contain t_i. Subsequently, tf-idf for the term t_i over D is give by:

$$tf\text{-}idf(t_i) = tf(t_i) \times idf(t_i), \tag{3.5}$$

where $tf(t_i)$ can be the frequency of the term t_i in the corpus. This definition of the tf-idf score is employed by assuming that important terms occur in particular documents frequently whereas they are relatively rare in the input corpus (i.e., they occur in a small number of documents). This assumption can be refined; hence, alternative definitions of $tf(t_i)$ and $idf(t_i)$ may be used.

Another approach to estimate a termhood score is that of *corpus comparison*—or,

[1] As suggested by Evert (2009) and Kilgarriff (2005), in this context, the assumption of independence is not reasonable and thus can decrease the performance of the method.

contrastive methods as explained earlier for candidate term extraction. In these methods, a corpus is compared against a general language corpus. It is often assumed that the distribution of valid terms and invalid terms varies in corpora of different types (Knoth et al., 2009). One implicit way to implement this logic is the use of statistical hypothesis testing, for example, as described earlier for Equation 3.1 and 3.2 and as employed in Kilgarriff (2001) and Rayson and Garside (2000). Alternatively, a category of contrastive approaches define statistical measures that explicitly exploit the observed frequencies in different corpora (e.g., see Drouin, 2004; Ittoo and Bouma, 2013). Liu and Kit (2008) suggest that these approaches are more desirable than techniques that only utilise a special corpus since they employ intrinsic statistical characteristics of valid terms in different corpora. Ahmad et al.'s (1999) *Weirdness* score is a classic example of this category of techniques that can be used to assign a termhood measure to a candidate terms t in a special corpus:

$$Weirdness(t) = \frac{f_s(t)/n_s}{f_g(t)/n_g}, \tag{3.6}$$

where $f_s(t)$ and $f_g(t)$ are the frequency of t in the special and a general corpus, respectively; similarly, n_s and n_g are the total frequency of terms in the respective corpora.

3.4.3 Hybrid Measures and a Little More of the Context

Amongst the statistical methods for termhood and unithood measurement, Frantzi and Ananiadou's (1996) c-*value* measure has attracted much attention. In contrast to statistics measures introduced previously, the c-*value* score can be seen as a *hybrid termhood-unithood* measure hence its definition considers statistical information that concerns both unithood and termhood of terms. For each candidate term t, the c-*value* score of t, is calculated using four criteria (Frantzi et al., 2000b): the frequency of t in the corpus; the frequency of t when it appears nested in other terms longer than t; the number of those longer terms; and the number of the constituent words of t shown by $|t|$. The c-*value* of t is given by

$$\text{c-}value(t) = \begin{cases} \log_2 |t| f(t) & \text{if } t \notin \text{nested} \\ \log_2 |t| \left(f(t) - \frac{1}{|T_t|} \sum_{b \in T_t} f(b)\right) & \text{otherwise} \end{cases}, \tag{3.7}$$

where T_t denotes the set of all the terms that contain t and are longer than t, and $f(s)$ denotes the frequency of an arbitary term s in the corpus. The greater the c-*value*(t), the more likely t is a valid term.

Following the c-*value* score, Frantzi et al. (2000b) introduce the nc-*value* score. The nc-*value* score is perhaps one of the first widely employed scores that implements the idea of *terms in context* by Pearson (1998). The nc-*value* score improves the c-*value* score by considering the frequency of words surrounding the terms. Frantzi et al. (2000b) hypothesise that valid term appears with a *closed* set of neighbour words. Accordingly, the occurrence of these words around a candidate term is a *positive clue* that can be used

in determining the termhood of the candidate term. This idea is implemented with the help of a function called *context weighting factor*. First, a set of as context words—which consists of nouns, adjectives, and verbs that appear in the vicinity of candidate terms—is extracted. Each word in this set is assigned to a context weight:

$$weight(w) = \frac{t(w)}{n}, \tag{3.8}$$

where $t(w)$ is the number of terms that w co-occur with, and n is the total number of candidate terms considered. The $weight(w)$ is then considered to indicate the important of w as a context word. Subsequently, the NC-*value* for the term t is computed by

$$\text{NC-}value(t) = 0.8\text{C-}value(t) + 0.2 \sum_{b \in C_t} f_t(b) weight(b), \tag{3.9}$$

where C_t is the set of distinct context words that co-occur with term t, and $f_t(b)$ is the frequency of the co-occurrences of the word b and the term t.

Following the NC-*value*, Maynard and Ananiadou (2000) introduce the SNC-*value* score by incorporating further information about the context in which candidate terms appear. To compute SNC-*value*, Maynard and Ananiadou suggest the use of three kinds of contextual information: *syntactic*, *terminological*, and *semantic* information. The syntactic information, as its name suggests, is mostly concerned with the distance between a candidate term and its context words. The terminological information suggests the use of co-occurrence counts of candidate terms and previously known terms (context terms). Finally, semantic information takes similarities of context terms into consideration by computing distances between them in a pre-constructed taxonomy of the context terms—similar to WordNet-based methods such as Wu and Palmer (1994).[1]

By incorporating contextual information in their implementations (e.g., as implied by the last few techniques in this section), statistical techniques can go beyond the simple classic intuitions that are listed in the beginning of this section. Incorporating the contextual information in these models not only enhances the performance of methods that assign unithood and termhood scores to candidate terms, but also enables the design of methods that can model the semantics of terms. Hence, during the past decade, the terminology extraction methods have leaned further towards the implementation of the idea of terms in context, often in the form of *supervised* machine learning techniques. Perhaps, this is partly due to the availability of the language resources that are required for implementing this type of methods.

[1] I would like to draw your attention to the paradigm change in the series of research by Ananiadou in terminology extraction: from a rationalist approach similar to the GTT in Ananiadou (1994) to empiricist *term in context* techniques in Maynard and Ananiadou (2000).

3.5 Organising Terminologies

Modern approaches to terminology encourage perspectives of terminology management similar to the way that lexical items are handled in general language. As discussed in the beginning of this chapter, whereas traditional terminology considers terms as *labels for concept*—untouched by context and detached from linguistic characteristics and interpretations—it has become evident that terms, like other lexical units in general language, are subject to linguistic norms. As suggested by Faber and L'Homme (2014), this latter perspective is perhaps best characterised by the term *lexical-semantic approaches to terminology*, in which conceptual modelling and knowledge representation is one of the major concerns (see also Buitelaar et al., 2009, for a similar discussion in the context of *ontology engineering*).

In order to organise lexical resources, lexical-semantic frameworks identify and employ a set of semantic relations such as synonymy and hyponymy between words. The well-known example of such a general lexical resource is WordNet (Miller, 1995). In WordNet, lexical units are grouped into *synsets*. Each synset contains a set of synonymous words—that is, words that have a similar meaning. Subsequently, these synsets are organised into a hierarchy of lexical concepts by defining a hyponym relationship between them—that is, in simple terms, a *type-of* or is-a relationship. Lexical items can be grouped by mechanisms other than synsets (e.g., see Pustejovsky et al., 1993) and organised by a variety of relationships other than synonym and hyponym relationships between lexical units (e.g., see Khoo and Na, 2006, for a survey on semantic relations).

Driven by demands in information system, in modern terminology, a similar principle is suggested for organising terminological resources. Manual encoding of semantic relationships between terms, however, is a time-consuming and tedious task. Moreover, terminological resources are required to be updated frequently; new terms are often introduced and they must be identified and organised in a terminological resource. More challenges are imminent when other properties of terms, such as their life cycle,[1] is considered (see L'Homme, 2014). Hence, a body of research in terminology mining has paid attention to the automatic organisation of terminological resources and the identification of semantic relationships between terms.

Amongst conceivable semantic relationships between terms, the detection of synonym relationships for the identification of term variations, and hyponym relationships for characterising an organisation of terms in a 'conceptual structure' have been at the centre of attention. The study of research literatures that address the identification of semantic relationships goes beyond the scope of this thesis. However, to provide a complementary view on the term classification task investigated in the later chapters, I briefly review research literature that aim for the identification of *type-of* relationships between terms (see also L'Homme and Bernier-Colborne, 2012; L'Homme, 2014, for an elaboration of the

[1] As discussed in the introduction of the thesis, too.

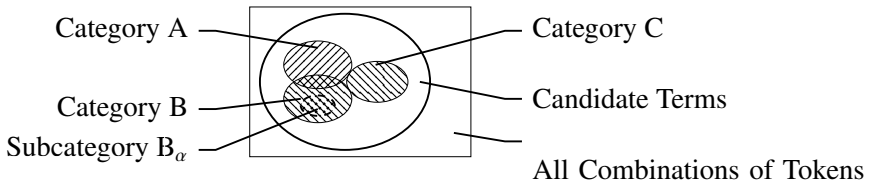


Figure 3.8: A Venn diagram that illustrates organisation of terms with respect to their concept categories. The dashed area shows valid terms. The set of valid terms enfolds several categories of terms, and each characterise a major concept in knowledge-domain. Hence, the identification of terms can be seen as the identification of a number of categories of terms. As discussed earlier, a term may belong to more than one category of concepts. Similarly, a category of concepts may include several subcategories. Entity recognition and term classification tasks are meant to identify particular categories of terms—that is, a subset of valid terms.

use of semantic relationships in terminological resources).

Methods that address automatic organisation of terminological resources by identifying a *type-of* relationship between terms are all similar in the sense that they assume terms can be organised in several categories to form a taxonomy.[1] Each category (taxon) characterises a group of terms from *similar* concepts in the domain of study (see Figure 3.8). For example, in computational linguistics, the terms *lexicon* and *multilingual corpus* can be categorised under the concept category of *language resources*, while *parsing* and *speech recognition* can be categorised under the concept of *methods and technologies*. Scoring techniques discussed in the earlier sections target distinguishing invalid candidate terms from valid terms and thus result in terminological resources that have a flat organisation (as opposed to the structure of taxonomies). To organise terms in an structure, therefore, an additional classification process is employed.

These classification methods can be distinguished with respect to several factors. For example, Weeds et al. suggest that these methods can be grouped by the type of information that they employ. Similar to what is suggested earlier in Section 3.4, Weeds et al. (2005) identify methods that rely on *internal* information (i.e., the lexical properties of the words that constitute terms) or *external* information (i.e., statistical, contextual, or ontological information about terms). As discussed earlier, except early works that rely on internal information, recent methods usually adopt a distributional approach towards modelling the semantics of terms, hence they often rely on external information or a combination of both external and internal information.[2]

From a methodological perspective, Weeds et al. (2005) suggest that the majority

[1] How these categories are defined and observed is a controversial matter (e.g., see Kilgarriff, 1997) that goes beyond the scope of this thesis.

[2] See Chapter 2 of this thesis for an introduction to the distributional methods. Maynard et al. (2008) articulate the basic idea behind these methods through an example: as a person's social life can provide valuable insight into their personality, so we can gather much information about a term by analysing the company that it keeps.

of these classification methods employ machine learning techniques in the form of a *supervised* classification problem. However, other types of methodologies are also possible. For example, Fukuda et al.'s (1998) PROPER system—a bio-entity tagger—employs a rule-based method. The use of rule-based methods, however, is hindered by their requirements for hand-crafted rules. I extend this study by distinguishing the way that the task of organising terminologies and the classification method are formulated.

If a prior knowledge of the concept categories is not available, automatic organisation of terminologies can be carried out using a method of clustering. These clustering methods are *unsupervised* since no manual effort is required prior to the classification (clustering) task. These methods suggest an organisation of terms by automatic identification of a number of concept categories. Recent examples can be found in Bertels and Speelman (2014); Dupuch et al. (2014, 2012). Terms are first grouped by a measure of similarity—usually, with the help of a distributional approach. Depending on the application context, the obtained clusters of terms can be labelled, which may introduce further complications to the process. One of the main applications of these methods is *ontology learning*, where these clustering techniques can be used as an assistive tool in the process of *ontology engineering*.

Concept categories, however, are typically known prior to the extraction of terms (or, at least, a partial knowledge of them exists). In these scenarios, a typical task is to find terms that belong to particular concept categories. The most established example of this kind of task is the identification of terms that correspond to instances of concepts that are of interest to biologists, namely bio-entity recognition (Nigel et al., 1999). These tasks rely heavily on manually annotated corpora: each mention of a term and its category-concept is annotated in a special corpus. The manual annotations are then employed to develop an entity tagger in a supervised fashion and, often, in the form of a sequence classifier—for example, using a machine learning technique such as the conditional random field method, etc. As reported previously, provided that enough training data is available, it is possible to attain a reasonable performance in these recognition tasks (e.g., see Kim et al., 2004).

In an alternative use case, the targeted concept categories—similar to entity recognition tasks—are known. However, no manual annotation is available for the training and development of a term/entity tagger. The lack of language resources is a familiar problem if a terminological resource with a taxonomic structure must be constructed for a new domain and only using text (i.e., from scratch). This is a task with many real-world applications (e.g., see Chakraborty et al., 2014; Anick et al., 2014), which can also be employed to address *ontology population* (e.g., see Tanev and Magnini, 2008; Maynard et al., 2008; Andersson et al., 2014). Lastly, a restored interest in these methods is signalled by the trending task of *cold-start knowledge base population* (see Ellis et al., 2012; Mayfield et al., 2014). As previously stated, one of the common challenge that these methods address is the lack of sufficient language resources for the development of classifiers.

Similar to terminology extraction and in contrast to entity recognition task, in these methods the communicative context is often the special corpus. Hence, these methods do not deal with individual term mentions. However, in contrast to terminology recognition techniques (which extracts terms from diverse concept categories in a specific domain knowledge) and similar to entity recognition, the objective of these methods is to extract a subset of terms from a similar category of concepts in a specific domain knowledge. From a lexical-semantic perspective, given a term in a special corpus, these methods can be used to discover the major *senses* of the term in the corpus. Therefore, the outcome can also be beneficial in ontology-based information systems, in which terms are often used as labels to access concepts. Similarly, these methods can be used for the knowledge base population using the so-called distant supervision technique (e.g., see Dredze et al., 2010). As suggested in the introduction chapter, this thesis investigates the development of a term classification method from this category.

Disregard of the methodology for extracting the term and its concept category, these methods assume terms have non-compositional semantics. The targeted hyponymy/hypernymy relationships are then modelled as a paradigmatic relationship. The same approach is often applied to synonymy identification and addressing the problem of term variation.

3.6 Machine Learning in Terminology Mining

Machine learning techniques have been widely used for extracting terms and constructing organised terminological resources. The extraction of candidate terms—particularly, complex candidate terms—is expectedly the first juncture that learning methods are utilised. In these applications, though implicitly, a learning method is employed to estimate lexical associations and thus unithood. The simplest example is the use of learning techniques for chunking and extracting nominal phrases. More sophisticated examples of this kind can be found in the context of multiword expression extraction in which the extraction of candidate multi-word lexical units often goes beyond extracting nominal collocations (e.g., see Pecina, 2008).[1]

Apart from the use of machine learning techniques for bracketing and candidate term extraction, in the research literature that investigates terminology mining, they are employed in two additional broad applications.

In the first category, a learning technique is employed to combine various scores from different sources of information in order to enhance the computed scores for the extracted candidate terms. Usually, several types of unithood and termhood measures are merged to synthesise a new score. A classic example in this category is Vivaldi et al. (2001) in which a term scoring process is enhanced by combining multiple scores using

[1] Hence, although important in many natural language processing applications, not all the applied methods for extracting multiword expressing are relevant to terminology mining.

a *boosting algorithm*. A more recent research in this line is presented by Hamon et al. (2014). Hamon et al. suggest a parametrised c-*value* scoring technique in which the introduced parameters are learned through an optimisation process based on the principles of *Genetic algorithm*.

In the next category, as suggested in the previous section, learning methods are often employed to organise a terminology by identifying co-hyponym relationships, or comparably, to extract terms that belong to a particular category of concepts (see Figure 3.8). In most applications, as discussed, the learning techniques are often used in the form of a supervised classifier. Based on the reasoning shown in Figure 3.3 and apart from the discussion in the previous section, machine learning-based methods that are employed in terminology mining can be also grouped by the type of communicative context that they model.

In the first group, a snippet of text that contains a mention of a candidate term is assumed to be a sufficient representative of the communicative context. In these applications, the identification of candidate terms and their corresponding concept categories are done simultaneously. In the second group, however, the communicative context is the special corpus. In these methods, the extraction of candidate terms and their Categorisation are usually, but not necessarily, performed in a two-step procedure. The first group, understandably, consists of machine learning-based *entity recognisers*, which aim for the identification of entity mentions in text. The second subcategory, however, encompasses methods that are commonly known as *term classification* methods.

The first group of learning-based methods—that is, entity recognition—is situated at the convergence point of the automatic term extraction and the classic named entity recognition (NER) tasks. The goal of NER is to recognise and classify *proper nouns* and numerical values into particular classes of entities such as location, organisation, time, and date (see Mohit, 2014; Nadeau and Sekine, 2007, for a survey on NER). However, as suggested by Yangarber et al. (2002), these recognition tasks can be generalised to other types of nominal compounds other than proper nouns. Therefore, techniques that have been previously applied to NER, have been widely adopted for the recognition of terms, inasmuch as some research does not differentiate between NER and other term classification methods (e.g., see Spasić and Ananiadou, 2004). The best examples of these tasks can be found in molecular biology domain and the task of *bio-entity recognition*. A bio-entity recogniser aims to identify mentions of a particular class of biological instances in text snippets (e.g., see Kim et al., 2004).

Various learning algorithms and a diverse set of features have been proposed to address the task of bio-entity recognition. For instance, Yamamoto et al. (2003) propose a system that employs a support vector machine to identify protein names from sentences in a set of abstracts from scientific publications—that is, from Kim et al.'s (2003) GENIA corpus. The proposed method relies on several kinds of features: morphological characteristics of candidate terms, the surface form as well as the lemma of the set of words

that co-occur with candidate terms in the training set, part-of-speech tags and syntactic information, and features extracted from available dictionaries in the domain. Many more examples of this kind can be found in biomedical text mining research.

The application of entity recognisers is not limited to the identification of biological instances. Kovačević et al. (2012) suggest a method to identify *methodology mentions* in scientific publications and classify them into four categories: *tasks*, *methods*, *resources*, and *implementations*. The term recognition and classification are merged and formalised as a sequence tagging problem using conditional random fields—a classifier per concept category. In the proposed method, sentences that describe a methodology are identified. The identified sentences are then passed to each of the trained classifiers in order to extract text segments that correspond to the methodology mentions. Similarly, QasemiZadeh et al. (2012) employ *support vector machines* to extract technical terms.

In the second group—that is, term classification—the process of mapping terms to concept categories is often modelled as an ad-hoc process. A classic example of this kind of method is Nigel et al. (1999), in which *decision trees* are employed to classify terms extracted from abstracts in the domain of molecular biology. Similarly, Spasić and Ananiadou (2004) propose another two-step approach for the classification of biomedical terms. In the proposed approach, terms are first extracted using dictionary look-ups and C-*value* and NC-*value* scoring techniques. The extracted terms are then classified by help of *verb selectional patterns* and using a nearest neighbour and genetic algorithm. Likewise, Afzal et al. (2008) propose a two-step method; however, for the identification of terms that signal *bioinformatics services and tools* and using a different set of features and learning technique. A similar method and application can be found in Houngbo and Mercer (2012).

Although in the above-mentioned examples a term classification process follows a term recognition process to select a subset of valid terms, as suggested by Maynard and Ananiadou (2001), the recognition and classification process can be merged. In this way, the scoring process in the term recognition system is replaced by the scoring mechanism that is implemented by the classifier; hence, candidate terms can be directly assessed and classified by the term classifier system(e.g., see Foo and Merkel, 2010; Judea et al., 2014). The type of information that is employed during the classification is what makes these methodologies different from the entity recognisers. These methods are also different due to the type of the output that they generate. The entity recognisers mark the boundaries of terms mentioned in a given sentence or text snippet, whereas the term classifiers are often used to organise terms in a knowledge structure such as ontologies and thesaurus. As a result, term classification methods have been widely employed for learning, populating and extending domain ontologies (e.g., see Wong et al., 2012).

Lastly, a large number of methods proposed for automatic thesaurus construction are comparable to term classification tasks (e.g., see Navigli and Ponzetto, 2012). Whereas automatic thesaurus construction deals with the processing of concept hierarch-

ies in general domain language, terminology classification methods deal with special corpora and sublanguages.

3.7 Evaluation Techniques

Evaluation of the majority of natural language processing systems has posed itself as a research challenge. Several factors can be named as a barrier to an objective evaluation of these systems (see Jones and Galliers, 1995, for a full depiction of these problems):

- disagreements on the basic concepts' definitions—for example, what is *semantics*?
- complexity of the tasks—for example, how to model a communication system? how to model users' background and psychological state? how to measure these factors and study their influences on the performance of a system?
- a large number of interdependent variables that play a role in the performance of a system;
- qualitative nature of the evaluation in a number of applications;
- multi-stage, intermediate, or different representations of the output;
- irreproducible evaluation situations and hence outputs;
- and, the absence of a common baseline on which to establish evaluations.

The most widely adopted framework for the evaluation of natural language processing tasks, including terminology mining methods, is the evaluation approach promoted in the series of message understanding conferences (MUC) for the assessment of information extraction systems. The MUC-style evaluation framework emphasises quantitative evaluations. This evaluation style accommodates a systematic reproducible assessment of the participating methods, which is methodologically clear and understandable. In this framework, the evaluation is carried out by comparing system-generated responses and hand-coded expected outputs (manual annotations) , which is expressed by a quantitative scoring measure. Figure 3.9 illustrates the evaluation's elements and procedure in this framework.

In an MUC-style evaluation, the most important building blocks are the manually annotated reference corpus[1] and the scoring measure. In the past decades, a number of research initiatives[2] and evaluation campaigns[3] have resulted in the development of a number reference corpora and datasets that are successfully employed for the development and evaluation of language processing techniques. Creating corpora for benchmarking terminology extraction techniques has been addressed in several research efforts, too.

[1] As evident, the development of the methods.

[2] For example, the Expert Advisory Group on Language Engineering Standard (The EAGLES Evaluation Working Group, 1996).

[3] For example, the series of automatic content extraction evaluation (see `http://www.itl.nist.gov/`

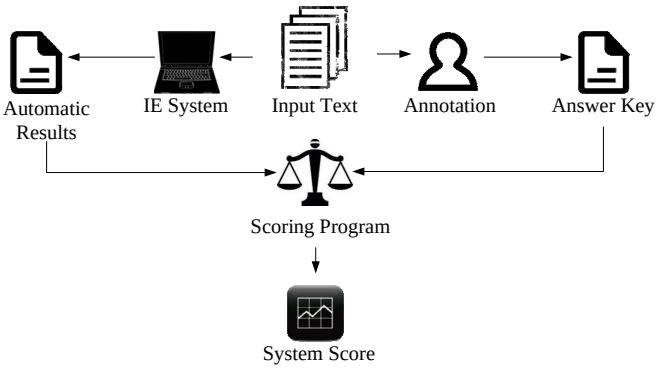


Figure 3.9: MUC-style evaluation for information extraction tasks (Lehnert et al., 1994).

The GENIA corpus is a well-known example of such reference datasets in bio-text mining: a corpus of 2000 abstracts from scientific publications in biological literature that is accompanied by the annotations of 100,000 terms organised in a well-defined ontology (Kim et al., 2003). The Colorado Richly Annotated Full Text Corpus (CRAFT) is another example of a bio-text mining dataset, which consists of 97 articles from the PubMed Central Open Access subset annotated with biomedical concepts such as *mouse genes* (Bada et al., 2012). In a more recent effort, Bernier-Colborne and Drouin (2014) report on creating a corpus for the evaluation of term extraction in the domain of automotive engineering. Similarly, Zadeh and Handschuh (2014a) introduce the ACL RD-TEC, a dataset of manually annotated terms in the domain of computational linguistics.

In quantitative evaluations, *precision* and *recall* are the two most widely-used scoring measures. Precision shows the ratio of the correct automatically generated results against all the information generated by the system. The correct automatically generated results are often those that *match* the answer keys provided through the manual annotation. Recall, however, measures the ratio of correct automatically generated information against all the available information in the reference corpus expected to be generated/extracted by the system. A combination of these measures such as F-score is used for scoring the systems. For an automatic term recognition (ATR) system, precision is the proportion of correct terms in the overall list of extracted candidate terms:

$$Precision = \frac{number\ of\ extracted\ valid\ terms}{number\ of\ candidate\ terms}. \tag{3.10}$$

Recall, on the other hand, is the proportion of extracted terms to the complete set of terms in the corpus:

$$Recall = \frac{number\ of\ extracted\ valid\ terms}{number\ of\ all\ valid\ terms\ in\ the\ corpus}. \tag{3.11}$$

iad/mig/tests/ace/), text analysis conference (http://www.nist.gov/tac/), as well as the series of workshops on semantic evaluation (http://aclanthology.info/venues/semeval).

And, usually but not necessarily, the F-score is given by

$$F_{measure} = \frac{2 * Recall * Precision}{Recall + Precision}. \tag{3.12}$$

The use of precision and recall is limited to the availability of manual annotations. In many real-world applications, manual annotations for all the system generated results are not available. For example, manual annotations are not available for all the candidate terms generated by an ATR system. In this case, precision thus cannot be computed; similarly, the complete set of expected information is not available. For example, the complete set of valid terms in a corpus, which must be extracted by an ATR system, is unknown; hence, recall cannot be computed. Moreover, in a number of use-cases, other quantitative aspects of the generated results are required—for example, the number of valid information items discovered by the system but not annotated/presented in the reference dataset (e.g., see the evaluation in Roark and Charniak, 1998).[1] In these situations, figures of merit other than precision and recall are employed.

In terminology extraction, one popular measure that often replaces precision and recall is *precision at n* (i.e., $P_{@n}$). Given a sorted list of m candidate terms, precision at n, $n \leq m$, measures the precision (i.e., the number of valid terms $|v|$) in the list of top n candidate term that are sorted by the scores assigned by an ATR system:

$$P_{@n} = \frac{|v|}{n}. \tag{3.13}$$

For example, $P_{@n}$ for $n = 10$ is the number of valid terms in the list of top 10 candidate terms sorted by their ATR-computed scores. It becomes evident that if a single number is used to summarise the performance, then the value of n and m can have a large impact on the computed performances. Hence, $P_{@n}$ is often replaced by an averaged precision.

Amongst techniques for obtaining an average of precision, *non-interpolated average precision* for k valid terms (NAP_k) is often used to report the performance of methods as a single number (e.g., see Zhang et al., 2008; Fahmi, 2009, chap. 4). As suggested by Schone and Jurafsky (2001), NAP_k is given by

$$NAP_k = \frac{1}{k} \sum_{i=1}^{k} P^i, \tag{3.14}$$

where k is the number of valid terms that are targeted to be seen in the list of sorted candidate terms, and P^i is the observed precision for pulling out i valid terms. That is, $P^i = \frac{i}{|H_i|}$, in which i is the number of valid terms, and $|H_i|$ is the number of candidate terms that are required to be checked in order to find this i valid terms. Compared to

[1] One controversy here is that while the answer keys cannot be used, how to decide whether an information item is valid.

$P_{@k}$, NAP_k signify the distribution of valid terms in the extracted sorted lists of candidate terms. Depending on the evaluation context, one of these measures is usually used to show a method's performance.

3.7.1 Some Evaluation Caveats and Questions

Even with the availability of language resources, MUC-style quantitative evaluation framework cannot always replace qualitative assessments. For instance, depending on the design principles adopted for the development of reference corpora, quantitative evaluations may not provide proper perspective on the scalability and portability of the systems participating in an evaluation. In addition, as suggested by Lehnert et al. (1994), this quantitative assessment cannot be used to assess the time and effort that is required to develop these systems. Therefore, in a number of occasions, qualitative assessments may still be required for a comprehensive evaluation.[1]

A number of critics draw attention to the way the output of a system matches the provided answer keys in the manual annotations. For example, Esuli and Sebastiani (2010) suggest that the evaluation of an extraction method can be enhanced by permitting the notion of *true negative*, incorporating a measure that is sensitive to the *degree* of overlap between the correct expected answers and the outputs of the extraction system, and allowing for multiple correct output. Other researchers go further and question the basis in which some of the measures such as precision and recall are employed in evaluation scenarios. For instance, Cowie and Wilks (2000) suggest that precision and recall are designed for information retrieval tasks; hence, they are not appropriate for the evaluation of a number of information extraction tasks. For example, in a multi-slot template filling task, counting correct results can produce some paradoxical outcomes and attention should be paid to the details of how performance scores are calculated.

Lavelli et al. (2008) address the evaluation of machine learning-based information extraction systems and the assessment of the ability of these algorithms to *learn*. Besides the factors discussed above, the authors argue that establishing an evaluation methodology and the availability of gold standard corpora do not guarantee a reliable comparison between different approaches and algorithms. Lavelli et al. suggest that considering the influential variables in the overall performance of such systems, for example, the number of *features* and setting of algorithm-specific parameters, is beneficial for a meaningful comparison of learning methods.

To avoid a number of barriers to an objective evaluation of information extraction systems, apart from the *intrinsic* MUC-style evaluations, *extrinsic* or indirect evaluation has been suggested. Extrinsic evaluations measure the quality of the output of a method by assessing the performance of a third system that employs the generated output. For example, a common method of extrinsic evaluation for an information extraction system

[1] This can also be discussed in the context of black box vs. glass box evaluation frameworks.

is to utilise its output in a document classification problem and assess the extraction task by studying the precision and recall of the classification task (Yangarber et al., 2000).

As suggested earlier in this section, as with other information extraction tasks, the evaluation of terminology mining methods is often carried out by comparing the output of a term extractor against a gold standard dataset, manually checking the output of the method with the help of a terminologist/a domain-expert, or an extrinsic evaluation such as the one suggested in Kit et al. (2008).

A number of concerns in the evaluation of terminology mining methods is similar to those that are listed for other information extraction systems. For instance, the evaluation of perfect and imperfect recognition has been one of the concerns in the evaluation of ATR systems (e.g., see Lauriston, 1995). Maynard et al. (2008) suggest that in modern applications, for example, ontology learning, performance metrics such as precision and recall are not sufficient since they provide a binary decision of correctness—that is, a term is either right or wrong and nothing in between. Therefore, they suggest the use of matching techniques that acknowledge partial correctness such as using edit distance as employed in the *balanced distance metric* by Maynard (2005) and the *SOLD* measure by Spasic and Ananiadou (2005).

However, the complexity of the evaluation of terminology mining methods goes beyond the common problems such as partial matching. As is rightly argued by Vivaldi and Rodríguez (2007), in short, the evaluation of these methods inherits its complexity from the definition of terms. In order to have an overall evaluation of terminologies, Vivaldi and Rodríguez suggest that three dimensions of terms' characteristics, namely, unithood, termhood, and their specialised usage, must first be assessed and then combined. This multi-faceted characteristic of terms often makes it hard to find an objective judgement when preparing reference corpora, annotating terms, and preparing an evaluation framework.

Lastly, assuming that all the terms in a corpus are annotated with high confidence, do all these terms have the same importance in domain-knowledge? Is it ever possible to introduce a measure to quantify their importance objectively? These are all questions that still must be addressed in an ideal evaluation framework of terminology mining.

3.8 Summary

In this chapter, terminology extraction methods are reviewed, in the application context of this thesis in which the use of distributional models will be investigated. The discussion started with the definition of the term *term* to highlight the complexity of terminology mining methods; the wide-range of task that it embraces; and, the wide spectrum of problems that it encounters.

In Section 3.2, the general two-step mechanism of a typical terminology mining method is discussed. In Section 3.3, a review of candidate term extraction techniques

was provided, followed by a study of term scoring methods in Section 3.4. Organising terminologies was discussed briefly in Section 3.5. This discussion was followed by an introduction to term classification techniques often used to form co-hyponym groups in Section 3.6. Finally, this chapter concluded with a brief study of the common practices for the evaluation of terminology mining methods.

The presented study in this chapter set the background for the proposed co-hyponym term extraction method in Chapter 5. However, it is worth mentioning that it only scratches the surface of the vast amount of ongoing research in computational terminology.

This page is intentionally left blank.

Part III

Core Research: The Methods

Chapter 4

Random Projections in Distributional Semantic Models

Random projections are mathematical tools that have been widely used in algorithm design. They have had a number of significant contributions in several domains, such as the applications of machine learning techniques to big data. At the expense of negligible loss in the accuracy of the estimated distances between vectors, these methods reduce the size of vectors to enhance the performance of processes. In distributional semantic models, random indexing is one of the widely-used methods that can be understood using the random projections theorems. In this chapter, the principles of random projections are employed in order to reintroduce random indexing and propose new dimensionality reduction methods for the ℓ_1-normed spaces.

This chapter starts with recapping the *curse of dimensionality* problem in distributional semantic models and enumerating a number of motivations for the proposed methods in Section 4.1. In Section 4.2, the random indexing technique is explained and justified mathematically. In Section 4.3, by extending the use of random projections to ℓ_1-normed spaces, a novel technique called random Manhattan indexing (RMI) is introduced. In Section 4.4, RMI and RI are compared, followed by a summary in Seciton 4.5.[1]

[1] Section 4.2 is mainly based on QasemiZadeh (2015) and QasemiZadeh and Handschuh (2015). Section 4.3.1 and 4.3.2 are based on Zadeh and Handschuh (2014d) and Zadeh and Handschuh (2014e), respectively.

4.1 Introduction

In order to model any aspect of the meanings in language, distributional semantic models exploit patterns of co-occurrences. These methods tie the usage context of linguistic entities (e.g., words and phrases) to their meaning. Hence, meanings are assessed by quantification of the distributional similarities of linguistic entities. An intuitive, mathematically well-defined model to represent and process such distributional similarities—amongst other representation frameworks—is vector space.

Recall from Chapter 2, particularly Section 2.2.1, in a vector space model, each element $\vec{s}_i$ of the standard basis (i.e., informally each dimension of the vector space) represents a context element. Given n context elements, a linguistic entity whose meaning is being analysed is expressed by a vector $\vec{v}$ as a linear combination of $\vec{s}_i$ and scalars $\alpha_i \in \mathbb{R}$ such that $\vec{v} = \alpha_1 \vec{s}_1 + \cdots + \alpha_n \vec{s}_n$. The value of α_i is acquired from the frequency of the co-occurrences of the entity that $\vec{v}$ represents and the context element that $\vec{s}_i$ represents. As a result, the values assigned to the coordinates of a vector, that is, α_i, exhibit the correlation of an entity and the context elements in an n-dimensional real vector space $\mathbb{R}^n$.

In this vector space, similarities of vectors are understood to indicate similarities of the meanings of linguistic entities that they represent. In order to assess the similarity between vectors, a vector space V is endowed with a *norm* structure.[1] A norm $\|.\|$ is a function that maps vectors from V to the set of non-negative real numbers, that is, $V \mapsto [0, \infty)$. The pair of $(V, \|.\|)$ is then called a *normed* space. In a normed space, the similarity between vectors is assessed by their distances. The distance between vectors is defined by a function that satisfies certain axioms and assigns a real value to each pair of vectors, that is,

$$dist : V \times V \mapsto \mathbb{R}, \quad d(\vec{v}, \vec{t}) = \|\vec{v} - \vec{u}\|. \tag{4.1}$$

The smaller the distance between two vectors, the more similar they are.

Amongst several choices, an ℓ_2-normed-based metric—such as the Euclidean distance and the cosine similarity—is an innate choice.

Euclidean space is the most familiar example of a normed space. It is a vector space that is endowed by the ℓ_2 norm. In Euclidean space, the ℓ_2 norm—which is also called the Euclidean norm—of a vector $\vec{v} = (v_1, \cdots, v_n)$ is defined as:

$$\|\vec{v}\|_2 = \sqrt{\sum_{i=1}^{n} v_i^2}. \tag{4.2}$$

Using the definition of distance given in Equation 4.1 and the ℓ_2 norm, the Euclidean

[1] Please note other structures than norm can be employed to assess the similarities.

distance is measured as:

$$dist_2(\vec{v}, \vec{u}) = \|\vec{v} - \vec{u}\|_2 = \sqrt{\sum_{i=1}^{n}(v_i - u_i)^2}. \tag{4.3}$$

In ℓ_2-normed vector spaces, various similarity metrics are defined using different normalisation of the Euclidean distance between vectors, for example, the *cosine similarity*.

A classic Salton et al.'s (1975) document-by-term model is, perhaps, the most familiar example of the above-described vector space model (VSM). Given n distinct terms t and a number of documents d, each document d_i is represented by an n-dimensional vector $\vec{d_i} = (w_{i1}, \cdots, w_{in})$, where w_{ij} is a numeric value that associates the document d_i to the term t_j, for $1 < j < n$. For instance, w_{ij} may correspond to the frequency of the terms t_j in the document d_i. For a collection of m documents, a *document-by-term* matrix $\mathbf{M}_{m\times n}$ denotes the constructed vector space. Using the *bag of words* hypothesis, it is assumed that the relevance of documents can be assessed by counting terms that appear in the documents, independent of their order or syntactic usage patterns. Documents with similar vectors are thus assumed to share the same meaning. Using the ℓ_2-norm, the similarity between documents is then calculated by the Euclidean distance or the cosine similarity shown in Figure 4.1.

As discussed in Chapter 2, when the number of entities in a VSM increases, the number of context elements employed for capturing similarities between them surges. As a result, usually high-dimensional vectors, in which most elements are zero, represent entities. However, when the dimension of vectors in a VSM increases, the discriminatory power of the VSM diminishes. This results in setbacks known as the *curse of dimensionality*. Hence, the curse of dimensionality is tackled using a *dimensionality reduction* technique.

Dimensionality reduction can be achieved using a number of methods as an auxiliary process that is followed by the construction of a VSM—ranging from heuristic-based selection process to ad hoc matrix factorisation techniques such as singular value decomposition (see Section 2.3.3). The use of these dimensionality reduction techniques, however, is hampered by a number of factors.

Firstly, a VSM at the original high dimension must be first constructed. Following the construction of the VSM, the dimension of the VSM is reduced in an independent process. The VSM with the reduced dimensionality is thus available for processing only after the whole sequence of these processes. However, construction of the VSM at its original dimension is computationally expensive (e.g., all the co-occurrences must be collected and stored) and the delayed access to the VSM with the reduced dimensionality is not desirable.

Secondly, reducing the dimension of vectors using the methods listed above is of high computational complexity. For instance, mapping $\mathbb{R}^n$ onto $\mathbb{R}^m$ using SVD trun-

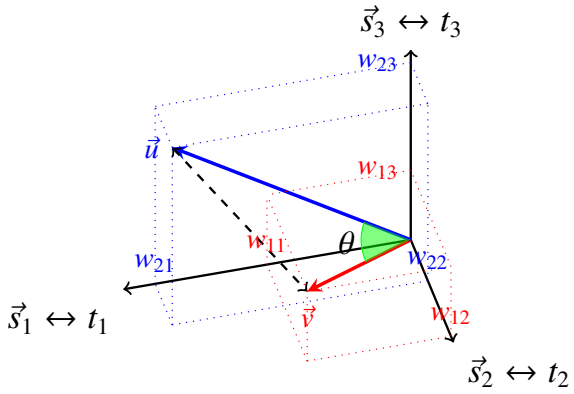


Figure 4.1: Illustration of a *document-by-term* model consisting of 2 documents and 3 terms. Each element of the standard basis s_i (i.e., each dimension), represents one of the 3 terms in the model. The 3-dimensional vectors $\vec{v} = (w_{11}, w_{12}, w_{13})$ and $\vec{u} = (w_{21}, w_{22}, w_{23})$ represent the two documents in the model. The dashed line shows the Euclidean distance between the vectors. Similarly, the cosine of the angel between the vectors, $\cos(\theta)$, defines the cosine similarity between them.

cation demands a process of the time complexity $O(n^2m)$ and space complexity $O(n^2)$.[1] Similarly, in a heuristic-based selection process, the collected frequencies for each of the context elements must be assessed. Depending on the employed heuristic, this process can be resource-intensive, too; for example, the collected frequencies are often required to be sorted by some criteria.

Last but not least, these methods are *data-sensitive*: if the structure of the data being analysed changes—that is, if either linguistic entities or context elements are updated, for example, some are removed or new ones are added—the dimensionality reduction process is required to be repeated and reapplied to the whole VSM in order to reflect these updates. The use of feature selection techniques or truncated SVD, therefore, may not be desirable in several applications, particularly when dealing with frequently updated big text-data.

Random projections are mathematical tools that are employed to implement alternative dimensionality reduction techniques that can alleviate the aforementioned problems. Random projections map high-dimensional vector spaces onto a low-dimension subspace using matrices consisting of randomly generated vectors that guarantee the preservation of distances between vectors. Hence, random projections are used to design dimensionality reduction techniques that (a) bypass a number of computations in the classic dimensionality reduction techniques (e.g., the computation of orthogonal subspaces or selecting context elements), and (b) merge the dimensionality reduction process into the process of vector space construction to suggest an incremental—thus scalable—technique

[1] It is worth mentioning that the use of incremental techniques can relax these requirements to an extent (e.g., see Brand, 2006).

for the construction of VSMs directly at a reduced dimensionality.

In the context of distributional semantic models, the widely-employed random indexing technique can be justified using the mathematical principles of random projections. Random indexing (RI) is an incremental method for the construction of vector spaces at a reduced dimensionality. It was first introduced by Kanerva et al. (2000) and further propounded by Sahlgren (2005). Sahlgren (2005) delineates the RI method as a two-step procedure that consists of the construction of (a) *index vectors* and (b) *context vectors*.

In the first step, each context element is assigned *exactly* to one *index vector*. Sahlgren (2005) indicates that index vectors are high-dimensional, randomly generated vectors, in which most of the elements are set to 0 and only *a few* to 1 and -1. In the second step, during the construction of *context vectors*, each target entity is assigned to a *zero vector* (i.e., all the elements of the vector are zero) that has the same dimension as the index vectors. For each occurrence of an entity, which is represented by $\vec{v}_{e_i}$, and a context element, which is represented by $\vec{r}_{c_k}$, the context vector for the entity is accumulated by the index vector of the context element, that is, $\vec{v}_{e_i} = \vec{v}_{e_i} + \vec{r}_{c_k}$. The result is a vector space model constructed directly at reduced dimension.

Both Sahlgren (2005) and Kanerva et al. (2000) introduce the random indexing method in a mathematical framework other than random projections—that is the sparse distributed memory (SDM).[1] The random indexing method was then developed and justified by Kanerva et al. (2000) as one of the extensions of SDM, without providing a mathematical justification for the suggested two-step procedure and the method's parameters—that is, the dimension of index vectors and the proportion of zero and non-zero elements in them.

In the remainder of this chapter, the random indexing technique is revisited and explained using theorems of random projections, which are refined by advances in statistics. In contrast to the previous delineations of this method, the provided description gives an understanding of the method which can be used for setting the method's parameters, recognising the limits of its use, and extending it to normed spaces other than ℓ_2.

In Section 4.2, random projections in Euclidean spaces—hence random indexing—is refined using mathematical theorems, which are verified by empirical experiments. Accordingly, Section 4.3 describes random projections in ℓ_1-normed spaces, and introduce the random Manhattan indexing technique—that is, a method similar to RI but for estimating city block distances. The differences between RI and RMI are reviewed in Section 4.4. Finally, this chapter concludes with a discussion and summary in Section 4.5.

[1] For a brief introduction to sparse distributed memory see Kanerva (1993)

4.2 Random Projections in Euclidean Spaces

In Euclidean spaces, random projections are elucidated by Johnson and Lindenstrauss's (1984) lemma (JL lemma). Given an ϵ, $0 < \epsilon < 1$, the JL lemma states that for any set of p vectors in a high n-dimensional Euclidean space $\mathbb{E}^n$,[1] there exists a mapping onto an m-dimensional space $\mathbb{E}^m$, for $m \geq m_0 = O(^{\log p}/_{\epsilon^2})$, that does not distort the distances between any pair of vectors, with high probability, by a factor more than $1 \pm \epsilon$. This mapping can be expressed by

$$\mathbf{M}'_{p\times m} = \mathbf{M}_{p\times n}\mathbf{R}_{n\times m}, \quad m \ll p, n, \tag{4.4}$$

where $\mathbf{R}_{n\times m}$ is often called the random projection matrix, and $\mathbf{M}_{p\times n}$ and $\mathbf{M}'_{p\times m}$ denote the p vectors in $\mathbb{E}^n$ and $\mathbb{E}^m$, respectively. According to the JL lemma, if the distance between any pair of vectors $\vec{v}$ and $\vec{u}$ in $\mathbf{M}$ is given by the $dist_{\text{Euc}}(\vec{v}, \vec{u})$, and their distance in $\mathbf{M}'$ is given by $dist'_{\text{Euc}}(\vec{v}, \vec{u})$, then there exists an $\mathbf{R}$ such that

$$(1 - \epsilon)dist'_{\text{Euc}}(\vec{v}, \vec{u}) \leq dist_{\text{Euc}}(\vec{v}, \vec{u}) \leq (1 + \epsilon)dist'_{\text{Euc}}(\vec{v}, \vec{u}).^{2} \tag{4.5}$$

Instead of the original n-dimensional vector space and at the expense of negligible amount of error ϵ, the distance between $\vec{v}$ and $\vec{u}$ can be calculated in the m-dimensional vector space. Accordingly, since $m \ll n$, the time and the space complexity for the computation of distances can be reduced significantly. The random projection matrix $\mathbf{R}$ is stored for later usages, such as adding new entities to the vector space.

The JL lemma does not specify the projection matrix $\mathbf{R}$. Finding $\mathbf{R}$ that satisfy the JL lemma is therefore the most important design decision when using random projections. Originally, Johnson and Lindenstrauss (1984) proved the lemma using an orthogonal projection onto a random m-dimensional subspace of the original vector space. Subsequent studies simplified the original proof and suggested several choices of $\mathbf{R}$ that resulted in projection techniques with enhanced computational efficiency (e.g., see Dasgupta and Gupta, 2003, for references). It is proved that a mapping that satisfies the JL lemma can be obtained, with a *high probability*, using a random projection $\mathbf{R}$ whose entries are independent and identically distributed (i.i.d.) and have zero mean and constant variance.[3]

Recently, Achlioptas (2001) shows that a sparse $\mathbf{R}$ with an *asymptotic Gaussian*

[1] $\mathbb{E}^n$ is an n-dimensional real vector space $\mathbb{R}^n$ endowed by the ℓ_2 norm.

[2] In addition, the lemma states that this mapping can be found in randomised polynomial time.

[3] For the simplicity of theoretical analysis, it is often assumed that entries of $\mathbf{R}$ have the standard Gaussian distribution—that is, for each m-dimensional random vector $\mathbf{r}$ in $\mathbf{R}$, $\mathbf{r} \sim \mathcal{N}_m(0, 1)$. According to the central limit theorem, the probability distribution of i.i.d. variables that have finite variance approaches a Gaussian distribution.

distribution, whose elements r_{ij} are defined as

$$r_{ij} = \sqrt{s} \begin{cases} -1 & \textit{with probability } \frac{1}{2s} \\ 0 & \textit{with probability } 1 - \frac{1}{s} \\ 1 & \textit{with probability } \frac{1}{2s} \end{cases}, \tag{4.6}$$

for $s \in \{1, 3\}$, results in a mapping that also satisfies the JL lemma.[1]

Subsequent research showed that $\mathbf{R}$ can be constructed from even sparser vectors than what is suggested in Achlioptas (2001) (e.g., see Li et al., 2006b; Matoušek, 2008). Specifically, Li et al. (2006b) has proved that in a mapping of an n-dimensional real vector space by a sparse $\mathbf{R}$, the JL lemma holds as long as $s = \mathrm{O}(n)$, for example, $s = \sqrt{n}$ or even $s = {}^{n}/_{\log(n)}$.

Using a sparse $\mathbf{R}$ that is given by Equation 4.6 reduces the number of multiplication operations in Equation 4.4 by the factor $\frac{1}{s}$ and thus speeds up the mapping process—that is, the computation of $\mathbf{M}'$. The larger the value of s, the sparser the random vector is; hence, at the expense of insignificant loss in the accuracy of the estimated distances, it is expected that the succeeding processes will be faster. Moreover, the multiplication of the scaling factor $\sqrt{s}$ can be postponed until after the mapping, or when it is necessary. Floating-point arithmetic operations, therefore, can be avoided during the computation of the mapping, which consequently enhances the computational as well as the memory complexity. Nonetheless, to say that a sparse $\mathbf{R}$ requires less space for its storage.

Apart from the sparse mapping, another major benefit when computing $\mathbf{M}'$ is obtained using the linearity of matrix multiplication. Each vector $\vec{v}_{e_i}$ in the original n-dimensional space, that is, ith row of $\mathbf{M}$, can be represented as a weighted sum of the basis vectors

$$\vec{v}_{e_i} = w_{i1}\vec{s}_{c_1} + w_{i2}\vec{s}_{c_2} + \cdots + w_{in}\vec{s}_{c_n}, \tag{4.7}$$

where w_{ij}, $i \leq p$ and $j \leq n$ are derived from the frequency of the co-occurrences of the entity and context element that $\vec{v}_{e_i}$ and $\vec{s}_{c_k}$ represent, respectively. By the basic properties of the matrix multiplication, the projection of $\vec{v}_{e_i}$ in $\mathbf{M}'$ is given by

$$\vec{v}'_{e_i} = \vec{v}_{e_i}\mathbf{R} = w_{i1}\vec{s}_{c_1}\mathbf{R} + w_{i2}\vec{s}_{c_2}\mathbf{R} + \cdots + w_{in}\vec{s}_{c_n}\mathbf{R}. \tag{4.8}$$

In turn, since, by definition, all the elements of the standard basis $\vec{s}_{c_k}$ are zero except the kth element, which is equal to 1, the statement given in Equation 4.8 can be equally written as

$$\vec{v}'_{e_i} = w_{i1}\vec{r}_1 + w_{i2}\vec{r}_2 + \cdots + w_{in}\vec{r}_n, \tag{4.9}$$

[1] The mapping in Equation 4.6 guarantees that distances are preserved with a probability of at least $1 - p^{-\gamma}$, for some $\gamma > 0$ (see Achlioptas (2001), for proof and explanation.)

where $\vec{r}_j$ is the jth row of $\mathbf{R}$. Equation 4.9 means that row vectors $\vec{v}'_{e_i}$, thus $\mathbf{M}'$, can be computed directly without necessarily constructing the whole matrix $\mathbf{M}$. From one perspective, the jth row of $\mathbf{R}_{n\times m}$ represents a context element in the original vector space that is located at the jth column of $\mathbf{M}_{p\times n}$.[1] Therefore, a vector representation of an entity at a reduced dimension can be computed directly by accumulating the row vectors of $\mathbf{R}$ that represent the context elements that co-occur with the entity.

4.2.1 Improving the RI Algorithm: An Outcome of the Exposition

The RI technique can be reintroduced using the mathematical explanations given in the previous section. As can be understood, the RI technique can be seen as a dimensionality reduction technique for Euclidean spaces. RI implements a random projection that employs a random matrix $\mathbf{R}$ with an asymptotic Gaussian distribution (as it is expressed by Equation 4.4). The construction of index vectors—that is, the first step of RI—is equivalent to the construction of the random projection matrix $\mathbf{R}$, whose elements are given by Equation 4.6. Each index vector is a row of the random projection matrix $\mathbf{R}$. The second step of RI, the construction of context vectors, deals with the computation of $\mathbf{M}'$. Each context vector is a row of $\mathbf{M}'$, which is computed by the iterative process justified in Equation 4.9.

While in previous research the parameters of the RI method are left to be decided entirely through experiments (e.g., see Lupu, 2014; Polajnar and Clark, 2014), the adopted mathematical framework can be leveraged to provide a guideline for setting the RI's parameters. Using the JL lemma, a criterion for choosing the dimension of vector spaces constructed by the RI method at the reduced dimensionality (i.e., m in Equation 4.4) and the number of zero and non-zero elements in index vectors (i.e., s in Equation 4.6) are suggested.

In a VSM constructed using RI at a reduced dimensionality, the degree of preservation of distances between vectors in the original high dimension and at the reduced dimensionality m is determined by the number of vectors in the model and m. If the number of vectors (i.e., the number of entities that are modelled in the VSM) is fixed, then the larger m is, the better the Euclidean distances will be preserved at the reduced dimension m. In other words, the probability of preserving the pairwise distances increases as m increases. However, from the computational perspective, the lower the value of m is, the less computation is required for the construction of the VSM and the calculation of the distances, and therefore the better the efficiency is. From this perspective, the choice of dimensionality in RI-constructed VSMs is a trade-off between efficiency and accuracy. Similarly, the value of m can be seen as the capacity of a RI-constructed VSM for accommodating new entities. Therefore, compared to $m = 4000$ suggested in Kanerva et al. (2000) or $m = 1800$ in Sahlgren (2005), depending on the number of entities that

[1] Informally, the jth dimension of the original n-dimensional vector space.

are modelled in an experiment, m can be set to a smaller value such as $m = 400$.

The discussion above can be approached by investigating the distribution of the pairwise distances in the original high-dimensional vector space and the constructed vector space using RI (see also Stein, 2007). If the pairwise distances in the original space are and relatively small, then in order to be able to distinguish them, the distortion of the pairwise distances at the reduced dimensionality must be small (i.e., ϵ in Equation 4.5). If the number of entities in the model is fixed, then the distortion of the pairwise distances reduces when m increases. Hence, the distribution of the pairwise distances is a factor that can influence the chosen value for m.

Based on the results reported in Li et al. (2006b), when embedding an n-dimensional vector space onto a vector space of a reduced dimensionality m, the JL lemma holds—that is, pairwise Euclidean distances between vectors are preserved—as long as s in Equation 4.6 is $O(n)$. In text processing applications, the number of context elements and thus the dimension of vector spaces (i.e., n) is often very large. When using the random indexing method, therefore, even a careful choice such as $s = \sqrt{n}$ in Equation 4.6 results in very sparse random index vectors. In most text processing applications, therefore, by setting only 2 or 4 non-zero elements in index vectors, distances in the RI-constructed model resemble distances in the high n-dimensional model (for the mathematical proofs, see Li et al., 2006b, Appendix B).

It is worth reminding that if the dimension of index vectors (i.e., m) is fixed, then increasing the number of non-zero elements in index vectors causes additional distortions in the pairwise Euclidean distances. For index vectors of fixed dimensionality m, if the number of non-zero elements increases, then the probability of the orthogonality between index vectors decreases (see examples from a simulation in Figure 4.2). Hence, an increase in the number non-zero elements while m is fixed can stimulate distortions in pairwise distances. However, it is important to note that causing distortions in the pairwise distances can be beneficial; for example, it may reduce the effect of noise and foster assortment of similar context elements. As a result, distortions in the pairwise distances can be favourable in a number of applications.

To verify the theoretical explanations given above, the discussion continues by reporting the observed empirical results from a set of experiments in the next section.

4.2.1.1 Setting the parameters of RI: Empirical observations

Instead of a task-specific evaluation, the ability of RI-constructed vector spaces in preserving pairwise Euclidean distances is shown when the method's parameters are set differently.

In the reported experiments, a subset of Wikipedia articles, which are chosen randomly from the *WaCkypedia_EN* corpus—that is, a 2009 dump of the English Wikipe-

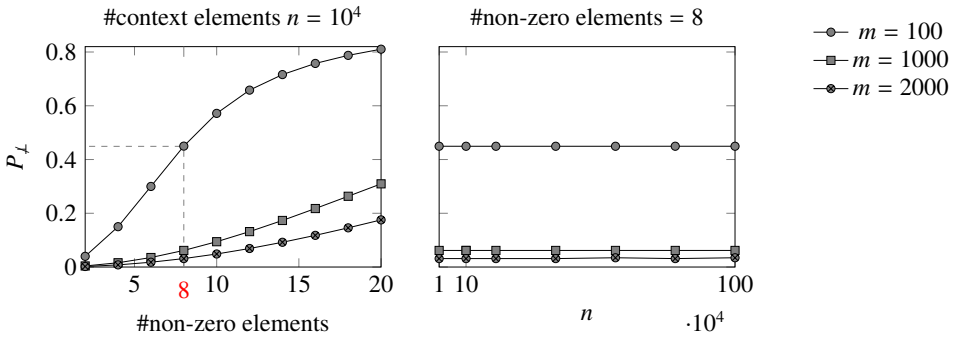


Figure 4.2: Orthogonality of index vectors: the y-axis shows the proportion of non-orthogonal pairs of index vectors (denoted by $P_\perp$) for sets of index vectors of various dimension $m = 100, 1000$, and 2000 obtained in a simulation. For sets of index vectors of a fixed size $n = 10^4$, the left figure shows the changes of $P_\perp$ when the number of non-zero elements increases. The right figure shows $P_\perp$ when the number of non-zero elements is fixed to 8, however, the number of index vector n increases. As shown in the figure, $P_\perp$ remains constant independently of n.

dia (Baroni et al., 2009)—are used.[1] A document-by-term VSM at its original high dimension is first constructed from a set of 10,000 articles (shown by D). A pre-processing of documents in D—that is, white-space tokenisation followed by the elimination of non-alphabetic tokens—results in a vocabulary of 192,117 terms. Each document in D is represented by a high-dimensional vector; each dimension represents an entry from the obtained vocabulary (as illustrated earlier in Figure 4.1). Therefore, the constructed VSM using this classic *one-dimension-per-context-element* method has a dimensionality of $n = 192{,}117$.[2]

To keep the experiments in a manageable size, each document d in D is randomly grouped by another 9 documents from D, which consequently gives 10,000 sets of a set of 10 documents. Using the constructed n-dimensional ($n = 192{,}117$) vector space, for each set of documents, the Euclidean distances between d and the remaining 9 documents in the set are computed. Subsequently, these 9 documents are sorted by their distance from d to obtain an ordered set of documents. The process therefore results in 10,000 ordered sets of 9 documents. The Euclidean distance is replaced with the cosine similarity and repeat the processes mentioned above. Figure 4.3 shows a histogram of the distribution of the distances between documents in these sets of documents. Figure 4.4 shows the distribution of the pairwise distances for all of the 10,000 documents; as shown, the distribution of the sampled distances closely resembles the distribution of all the pairwise distances in the model.

The procedure described above is repeated, however, by calculating distances in VSMs that are constructed using the RI method. Each term in the vocabulary is assigned

[1] The corpus can be obtained from `http://wacky.sslmit.unibo.it/doku.php?id=corpora`.

[2] In all the performed experiments, the frequency of terms in documents is used to indicate weights in corresponding vectors.

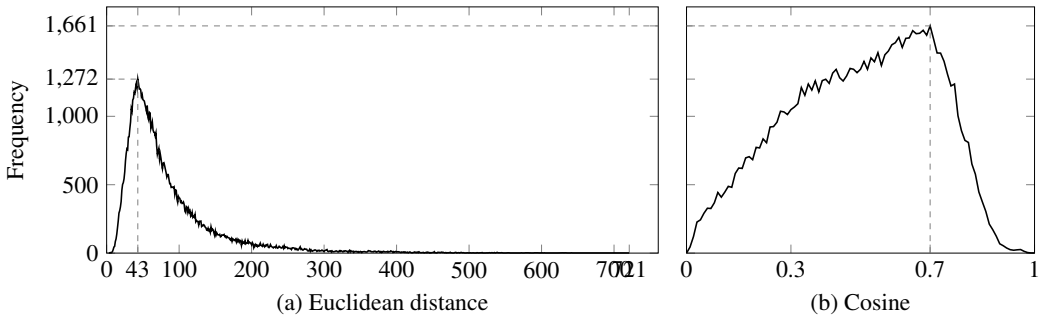


Figure 4.3: A histogram of the distribution of (a) the Euclidean distances and (b) the cosine similarities between pairs of vectors in the VSM of dimension 192,117 that are sampled randomly and employed for the experiments.

to an m-dimensional index vector and each document to a context vector. Context vectors are updated by accumulating index vectors to reflect the co-occurrences of documents and terms. Subsequently, the obtained context vectors are used to estimate the Euclidean distances and the cosine similarities between documents. The estimated distances are then used to create the ordered sets of documents, exactly as explained above. This process is repeated several times when the parameters of RI—that is, the dimension m and the number of non-zero elements in index vectors—are set to different values.

It is expected the relative Euclidean distances as well as the cosine similarities between documents in the RI-constructed VSMs to be the same as in the original high-dimensional VSM.[1] Hence, the ordered sets of documents obtained from estimated distances in the RI-constructed VSMs must be identical to the corresponding sets that are derived using the computed distances in the original high-dimensional VSM. For each VSM constructed using the RI method, therefore, the resulting ordered sets are compared with the obtained ordered sets from the original high-dimensional VSM using the Spearman's rank correlation coefficient measure (ρ).

The Spearman's rank correlation coefficient evaluates the strength of an association between two ranked variables, that is, two lists of sorted documents in our experiments. Given a list of sorted documents obtained from the original high-dimensional VSM (list_o) and its corresponding list obtained from a VSM constructed using the RI method (list_{RI}), Spearman's rank correlation for the two lists is given by

$$\rho = 1 - \frac{6 \sum dif_i^2}{n(n^2 - 1)}, \tag{4.10}$$

[1]The preservation of the cosine similarities can be verified mathematically, for example, see the provided proofs in Kaski (1998). Simply put, the cosine similarity can be expressed using the Euclidean distance when the length of vectors is normalised to unit length. This simple fact can be used to show that the cosine similarities are preserved when using the RI method.

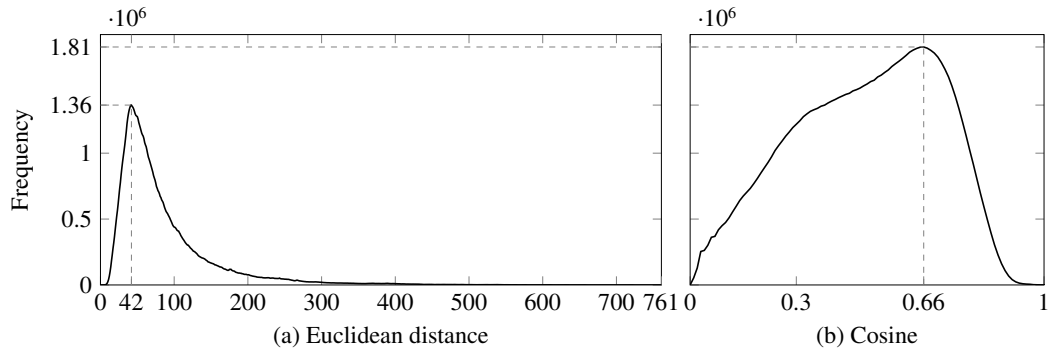


Figure 4.4: A histogram of the distribution of all the pairwise distances in the VSM of dimension 192,117 for (a) the Euclidean distances and (b) the cosine similarities.

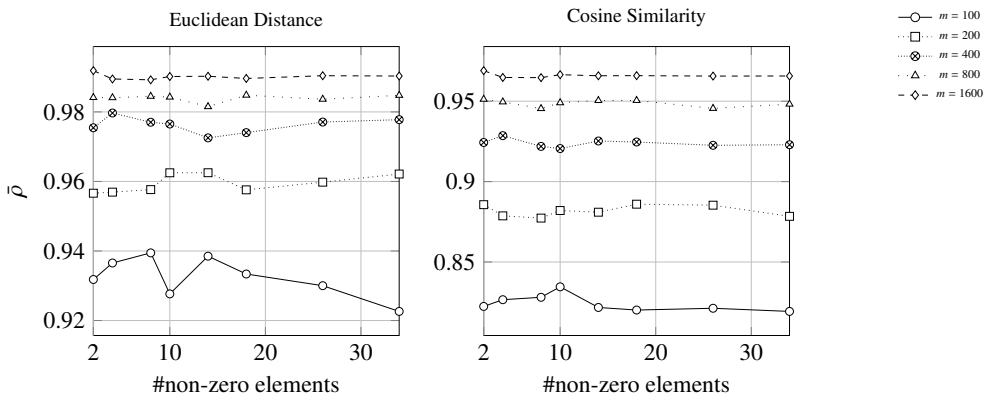


Figure 4.5: Correlation between the estimated Euclidean distances in RI-constructed vectors spaces and the original high-dimensional vector space: $\bar{\rho}$ shows the average of the Spearman's rank correlation coefficient between the ordered sets of documents that are obtained using the RI-constructed vectors spaces and the original high-dimensional vector space. Results are shown for both Euclidean distances as well as the cosine similarities when parameters of the RI method are set to different values.

where dif_i is the difference in paired ranks of documents in list_o and list_{RI}, and $n = 9$ is the number of documents that are sorted in each list. The average of ρ over the obtained sets of ordered set of documents ($\bar{\rho}$) is reported to quantify the performance of RI with respect to its ability to preserve ℓ_2-normed distances, when its parameters are set to different values: the closer $\bar{\rho}$ is to 1, the more similar the order of documents in an RI-constructed and the original high-dimensional VSM.

Figure 4.5 shows the obtained results. Since the dimension of the original vector space is very high, 2 non-zero elements per index vector are sufficient to construct a vector space that resembles relative distances between vectors in the original high-dimensional vector space, even for $m = 1600$. In addition, because only a small number of documents—that is, $p = 10000$—are modelled, even at the reduced dimension of $m = 100$,

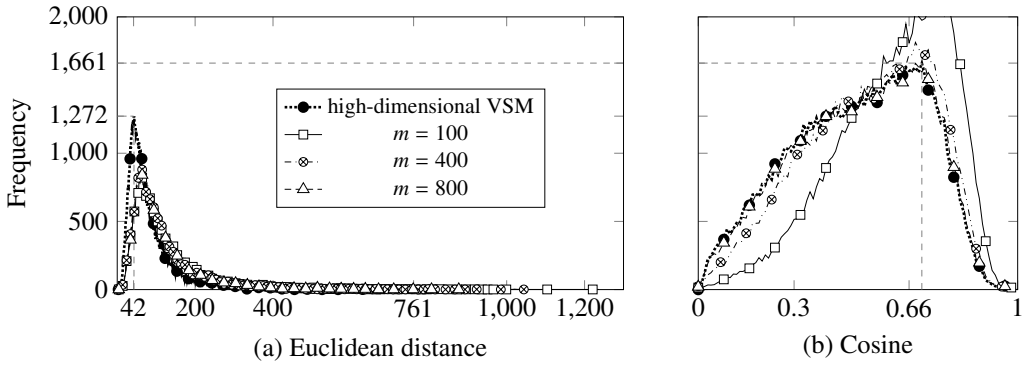


Figure 4.6: Distribution of distances in the RI-constructed VSMs: as m increases, the distribution of the distances in the RI-constructed VSMs are becoming more similar to the distances' distribution in the original high-dimensional VSM.

the estimated distances in the RI-constructed vector space shows a high correlation to the distances in the original vector space (i.e., $\bar{\rho} > 0.92$ for pairwise Euclidean distances and $\bar{\rho} > 0.82$ for the cosine similarity). As expected, the generated random baseline for $\bar{\rho}$ in Figure 4.5 is -0.002, that is, approximately 0. For $m = 1600$, the observed pairwise distances in the RI-constructed vector space are almost identical to the original vector space, that is, $\bar{\rho} > 0.99$ for Euclidean distances and $\bar{\rho} > 0.96$ for the cosine. Figure 4.6 compares the distribution of distances in the original high-dimensional VSM and the RI-constructed VSMs. As expected, when m increases, these distributions are becoming more similar to each other.

4.2.2 Related Work and Other Justifications of RI

As cited by Sahlgren (2005), the RI method was inspired from Kanerva's sparse distributed memory (SDM).[1] SDM, which was initially designed as a model of human long-term memory, is a cognitive-mathematical model. To formalise computation in several applications, it employs a high-dimensional *binary* vector space, the *Hamming* distance, as well as mathematical theorems that are often used in neural networks.[2] The RI method was then developed and justified by Kanerva et al. as an extension of SDM, without providing mathematical details, which are provided here. [3] An impression similar to Kanerva

[1] Perhaps more comprehensible than the JL lemma

[2] Recently, Snaider (2012, Chap. 2) has provided a summary of the SDM's mathematical foundation, and compared it with other mathematical models.

[3] Neither Sahlgren (2005) nor Kanerva et al. (2000) specify the proportion of the zero and non-zero elements in the index vectors, except that most of the elements of the index vectors are zero and only a *few* are 1 and -1. For instance, Kanerva et al. (2000) suggest 10 non-zero elements for a 4000-dimensional index vector without providing further explanation. Although Sahlgren and Karlgren (2005) suggest the following distribution (which can also be found in Sahlgren, 2006, chap. 4) for the elements of the index

et al.'s (2000) RI can also be found in the methods suggested by Gallant (e.g., see Gallant, 1991).[1]

An account of random projection in Euclidean spaces similar to RI can be given following Kohonen's seminal work on self-organising maps (e.g., see Ritter and Kohonen, 1989, Appendix I). For instance, Kaski (1998) introduces *random mapping*, a dimension reduction technique that employs random projections in Euclidean spaces. Instead of the JL lemma, Kaski (1998) relies on the fact that the least distortion in a mapping in a Euclidean space, such as Equation 4.4, is attained when $\mathbf{R}$ is orthogonal. Using reported results in Hecht-Nielsen (1994), Kaski assumes that randomly created vectors are most likely to be orthogonal and suggests mapping by a random matrix constructed by i.i.d. random vectors $\mathbf{r} \sim \mathcal{N}_m(0, 1)$.[2] He then shows that the distortion in the *inner product* of pairs of vectors at reduced dimension is on average zero and its variance is less than $2/m$. Several other theorems and proofs, which give similar results to the JL lemma, can be found to explain the use of random projection for dimension reduction in Euclidean spaces in various applications (e.g., see Linial et al., 1995; Arriaga and Vempala, 2006).[3]

The viability of the random projection techniques in general, and the RI method specifically, have been verified in several research reports. Amongst them, experimental results reported by Bingham and Mannila (2001) admit that the dimension reduction using the suggested sparse random matrix in Achlioptas (2001) provides comparable results to the conventional dimension reduction techniques, such as truncated SVD, in a document similarity measurement application. In addition, a growing number of research in diverse application domains employ the RI technique for dimension reduction (e.e., see Jurgens and Stevens, 2009, 2010; Musto et al., 2012; Yannakoudakis and Briscoe, 2012).

Apart from setting the RI method's parameters, the proposed theorems in Section 4.2 enable us to (a) categorise methods employed for incremental VSM construction at a reduced dimensionality, and (b) provide mathematical justifications for several variations of the RI method proposed in research literature. First and foremost, incremental methods can be categorised based on the type of projections that they employ to con-

vectors:

$$r_{ij} = \begin{cases} +1 & \text{with probability } \frac{\beta/2}{m} \\ 0 & \text{with probability } \frac{m-\beta}{m} \\ -1 & \text{with probability } \frac{\beta/2}{m} \end{cases}, \tag{4.11}$$

they do not provide a criterion for choosing the values of m and β. The given distribution in Equation 4.11 expresses the probability of non-zero elements in terms of the dimension of the index vectors (i.e., m), and the number of non-zero elements (i.e., β). In this way, the degree of the sparsity of index vectors is shown by the probability of the non-zero elements.

[1] For an algorithmic description of these methods in a retrieval task see Caid and Oing (1997) and its references.

[2] Similar conclusion is drawn for the RI technique.

[3] Resulting from the popularity of *connectionist* methodology in late 80s and early 90s, the list of research that propose similar methodologies is very long. Giving a comprehensive view of this research effort is beyond the scope of this thesis. Interested readers can perhaps gain insight by following a citation network, for example, by starting from Pollack (1990) or any of the references listed in this section.

struct VSMs at a reduced dimensionality (hence, the type of similarity metrics that they estimate). Despite that in natural language processing applications, the majority of these methods suggest the use of Gaussian random projections for estimating ℓ_2 norm-based similarities, a few researchers suggest random projections other than Gaussian to estimate similarities in VSMs other than ℓ_2-normed (e.g., see *TopSig* by Geva and De Vries (2011) and the random Manhattan indexing method proposed later in this chapter).

If a method based on random projections is employed to construct ℓ_2-normed VSMs, then its underlying mathematical principles is similar to RI[1]; hence, this method can be categorised in the same group of methods as RI. The major differences between methods in this category often result from (a) the procedure that they employ to construct a VSM at a reduced dimensionality (i.e., the second step of the RI procedure as explained from Equation 4.7 to 4.9) and/or (b) the weighting methodology that they employ in order to smooth collected co-occurrence frequencies.[2] The weighting process can be combined with the context vector construction, too.

As suggested earlier, the context vector construction can be carried out using a sequential scan of a corpus. The sequential scan, however, can be tailored to meet the requirements of a particular application. For example, context vectors can be updated every time the corpus is updated. Similarly, the weighting strategy can be changed to serve a specific purpose. Both of the alterations can take place by an intuitive or cognitive perspective, which may seem different from the RI technique. However, as long as substituted strategies can be interpreted using theorems suggested in Section 4.2, the resulting methods are, in essence, equivalent to the mapping that is given by the RI technique. In this case, the resulting vector space at reduced dimension still conforms to what is stated here for the RI-constructed VSMs.

The incremental semantic analysis (ISA) method, which is proposed by Baroni et al. (2007), and the reflective random indexing method, which is proposed by Cohen et al. (2010), are examples of the techniques discussed in the above paragraph. These methods offer interesting intuitions, other than the RI method, in order to enhance the results obtained for semantic similarity measurements in some applications. However, in both of these methods, the strategy employed for the construction of VSMs at reduced dimensionality can be interpreted as a technique for the adjustment of w_{ij} weights in Equation 4.4. Therefore, both methods are essentially the same as the RI method described here—that is, random projection with a sparse asymptotic Gaussian random matrix. For example, it can be verified that Baroni et al.'s (2007) ISA technique integrates a *Laplacian smoothing* to the RI's two-step procedure.

[1] Or, can be equivalently represented as.

[2] A description of the weighting process in VSMs is given in Chapter 2).

4.2.3 RI's Advantages Versus Limitations

The RI technique reduces the time and the space complexity of the required processes for constructing a VSM with regards to the values of

- n and m in Equation 4.4—that is, the original dimension of VSM and its reduced dimension obtained using RI, respectively;
- s in Equation 4.6—that is, the proportion of zero and non-zero elements in index vectors.

When using a sparse matrix representation, compared to a classic *one-dimension-per-context* VSM construction technique, the RI method imposes an additional $\beta - 1$ *addition* operations, where β is the number of non-zero elements in index vectors. However, this additional computation is insignificant considering the fact that RI combines the construction of a vector space with the dimension reduction processes. RI eliminates the need for a resource-intensive dimension reduction technique, such as the truncated SVD. Evidently, by reducing the dimension of the vector space, RI enhances the time complexity of the process of measuring distances between vectors by an approximate factor of $\frac{n}{m}$. As suggested earlier, the use of sparse projections further enhances the time complexity of the construction of VSM by a factor equal to $\frac{1}{s}$, and, to an extent, the space complexity for storing and manipulating VSMs.

In many dimension reduction techniques other than random projection, the projection subspace is devised by the analysis of data in the original high-dimensional VSM. For instance, in order to employ truncated SVD, a linear equation that finds eigenvectors should be solved. Therefore, in these methods, if the structure of the data being analysed changes, the basis of the projection subspace also changes. Additionally, in such *data-sensitive* dimension reduction techniques, the vector space at the reduced dimension—thus, similarity assessments—is only available after the computation of the transformation and applying it to the data at the original high dimension. Both stipulations impose limitations when using a data-sensitive dimension reduction technique, which the RI method can resolve.

The first limitation is faced when updating a vector space that is followed by a data-sensitive method of dimensionality reduction. In this setup, updating the vector space results in cumbersome processes. The process of dimensionality reduction needs to be repeated in order to reflect the changes in the model. For example, the use of the truncated SVD demands the recalculation of the eigenvectors, and therefore the alternation of the transformation process, which affects all the vectors in the model at reduced dimension. As a result, a process such as distance computation should be repeated for all the vector space entries. However, in the RI technique, the employed subspace for dimension reduction, to a great extent, is independent of the data structure. Updating the vector space is carried out by the accumulation of existing or new index vectors, which affects only

certain vectors. Thus, processes such as distance calculation are only necessary for the affected vectors.

The second limitation of a data-sensitive dimension reduction technique is that vector space at reduced dimension is available for processing only after the computation of the transformation. In contrast, when using the RI method, vector space at reduced dimension is available for processing during the construction of the vector space. As a result, similarity assessment is feasible at any time during the vector space construction, even when all the occurrences of entities in contexts are not observed. This is an extra advantage when processing frequently updated information, such as text streams in social media (e.g., see Sahlgren and Karlgren, 2009; Jurgens and Stevens, 2009; Karlgren et al., 2012).

The dimension of a vector space constructed using the RI method is fixed and, to a great extent, independent of the number of employed contexts and the size of corpus. However, the dimension of the vector space in a *one-dimension-per-context* model increases when new contexts are required to be added to the model. In a distributional model of semantics, due to the power-law distribution of context elements, appending a new entity to a model often requires appending new context elements to the model. The new entity most likely appears in/with context elements that have not yet appeared in the model. Therefore, in order to keep the model updated, its dimension should be increased to encompass new appended context elements. In contrast, in the RI technique, a large number of new context elements can be easily added to a vector space without changing its dimension, but at the expense of an insignificant loss of accuracy, which can be estimated by the JL lemma. A new context is defined and appended to the model simply by defining a new index vector.

The fixed dimensionality of the vector space constructed by RI and advance knowledge of its value are major advantages when dealing with big data, particularly in distributed computing frameworks. As described above, the induced vector space models using a technique such as the RI method scale up linearly with respect to the number of entities and not the number of contexts. In addition, the prior knowledge of the vectors' dimension is advantageous for load balancing in distributed computing frameworks (e.g., see Gufler et al., 2012, for an explanation of the load balancing problem).

The RI technique, however, comes with a number of limitations, which can be inferred from the proposed mathematical understanding of RI. The mathematical justification given in Section 4.2 explicitly states that the RI method, which employs a random matrix $\mathbf{R}$ whose elements are defined using the asymptotic distribution given in Equation 4.6, can only be applied for the approximation of similarity measures in the ℓ_2 normed spaces. That is, RI can be employed if similarity measures are derived from the ℓ_2 norm such as the Euclidean distance and the cosine similarity. For instance, the use of RI-constructed VSMs for estimating the city block distances between vectors—for example,

as suggested in Lapesa and Evert (2013)—is not justified, at least mathematically.[1]

This list of the advantages and disadvantages is not exhaustive and new items can be added or removed according to the application context or the comparison framework.

4.2.4 A Summary of the Exposition's Outcomes

In Section 4.2, the use of Gaussian sparse random projections for dimension reduction in Euclidean spaces is described, which consequently arrives at the well-known random indexing technique. Accordingly, in Section 4.2.1.1, observed results in an empirical experiment are shown to understand the method's behaviour with respect to its ability to preserve pairwise Euclidean distances, or in general ℓ_2-normed-based similarity measures. In addition, several important outcomes from the mathematical description of the RI method are emphasised.

Firstly, whereas the original delineation of the method did not provide a concrete guideline for setting the method's parameters, Section 4.2.1 ameliorates the previous two-step procedure with criteria for choosing the dimensionality as well as proportion of zero and non-zero elements of index vectors.

Secondly, the proposed understanding of the RI method is employed to discern its limitations and application domain. It is proven that the employed random projections by the RI method do not preserve distances other than ℓ_2 (e.g., see Brinkman and Charikar, 2005). Hence, it is important to note that RI-constructed VSMs can only be used for estimating similarity measures that are derived from the ℓ_2 norm—for example, the Euclidean distance and the cosine similarity.

Thirdly, the rationale given in the aforementioned sections provides a framework to justify several variations of the RI technique mathematically. Although these methods are based on plausible intuitions, similar to RI, they lack theoretical justifications. For example, the given mathematical description can be employed to identify the method proposed in Baroni et al. (2007) as a variation of RI that employs *Laplacian* smoothing. Similarly, the same rationale can be used for categorisation of the methods that construct VSMs at a reduced dimensionality. This idea can be generalised to coordinate all other major processes that are often involved when using VSMs.

Lastly, the given understanding of the mechanism of RI can be employed to generalise RI to normed spaces other than ℓ_2. This generalisation can be achieved using random projections with a distribution other than asymptotic Gaussian—for example, as suggested in Indyk (2006); Li et al. (2013)—and altering Equation 4.6. Accordingly, in the next section, the random Manhattan indexing is proposed for constructing ℓ_1-normed VSMs incrementally and directly at a reduced dimensionality.

[1] For example, see proofs in Brinkman and Charikar (2005). Also, see the reported empirical observations in Section 4.4.

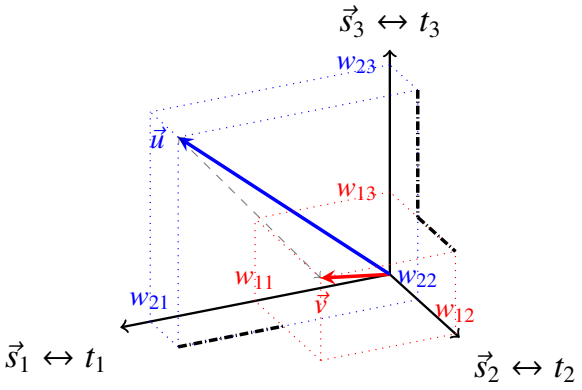


Figure 4.7: The sum of the dash-dotted lines is the Manhattan distance between the two vectors $\vec{v_1} = (w_{11}, w_{12}, w_{13})$ and $\vec{v_2} = (w_{21}, w_{22}, w_{23})$. Whereas the Euclidean distance between the two vectors is the length of the straight line between them (the dashed line), the Manhattan distance between the two vectors is the sum of the absolute differences of their coordinates.

4.3 Random Projections in ℓ_1-Normed Space

As stated earlier, in a vector space, the similarity between vectors can be assessed using a *norm* structure. Besides the ℓ_2 norm, ℓ_1 norm is another *not so common* choice for the similarity measurement. The ℓ_1 norm for $\vec{v}$ is given by:

$$\|\vec{v}\|_1 = \sum_{i=1}^{n} |v_i|, \tag{4.12}$$

where $|.|$ signifies the modulus.[1] Expectedly, a vector space endowed with the ℓ_1 norm is called an ℓ_1-normed space. The distance in an ℓ_1-normed vector space is often called the *Manhattan*, *taxicab*, or the *city block* distance. According to the definition given in Equation 4.1, the Manhattan distance between two vectors $\vec{v}$ and $\vec{u}$ is given by:

$$dist_1(\vec{v}, \vec{u}) = \|\vec{v} - \vec{u}\|_1 = \sum_{k=1}^{n} |v_i - u_j|. \tag{4.13}$$

Shown in Figure 4.7, the collection of the dash-dotted lines is the ℓ_1 distance between the two vectors. Similar to the ℓ_2-normed spaces, various normalisations of the ℓ_1 distance[2] define a family of ℓ_1-normed similarity metrics.

Similar to ℓ_2-normed spaces, the curse of dimensionality can obstruct efficient computation in ℓ_1 normed spaces. Both heuristic-based and transformation-based dimen-

[1] The definition of the norm is generalised to ℓ_p spaces with $\|\vec{v}\|_p = (\sum_i |v_i|^p)^{1/p}$; the discussion about ℓ_p-normed spaces other than $p = 1, 2$ goes beyond the scope of this thesis.

[2] As long as the axioms in the distance definition hold.

sionality reduction techniques can also be employed to alleviate the curse of dimensionality in ℓ_1-normed spaces. For example, similar to SVD truncation in ℓ_2-normed spaces, matrix factorisation techniques that guarantee the least distortion in the ℓ_1 distances can be employed (e.g., see Kwak, 2008). However, as discussed in Section 4.1, these methods are not desirable in a number of applications; for example, due to the resources they demand for computing VSMs at a reduced dimensionality, delays that they may cause in accessing VSMs at a reduced dimensionality, and frequent changes in the structure of data in VSMs. Accordingly, it is stated that random projections can be used to implement alternative dimensionality reduction techniques that can alleviate these problems.

In Euclidean spaces, random projections can be employed to introduce the RI technique. RI solves the problems stated above by combining the construction of a vector space and the dimensionality reduction process. Unlike methods that first construct a VSM at its original high dimension and conduct a dimensionality reduction afterwards, the RI method avoids the construction of the original high-dimensional VSM. Instead, it merges the vector space construction and the dimensionality reduction process. RI, thus, significantly enhances the computational complexity of deriving a VSM from text. However, the application of the RI technique (likewise, the standard truncated SVD in LSA) is limited to ℓ_2-normed spaces, that is, when similarities are assessed using a measure based on the ℓ_2 distance. It is verified that using RI causes large distortions in the ℓ_1 distances between vectors (Brinkman and Charikar, 2005). Hence, the RI technique is not suitable for constructing VSMs if similarities are computed using the ℓ_1 distance.

Depending on the distribution of vectors in a VSM, the performance of similarity measures based on the ℓ_1 and the ℓ_2 norms varies from one task to another. For instance, it is suggested that the ℓ_1 distance is more robust to the presence of outliers and non-Gaussian noise than the ℓ_2 distance (see the problem description in Ke and Kanade, 2003)). Hence, the use of the ℓ_1 distance can be more reliable than the ℓ_2 distance in certain applications. For instance, Weeds et al. (2005) suggest that the ℓ_1 distance outperforms other similarity metrics in a term classification task. In another experiment, Lee (1999) observed that the ℓ_1 distance gives more desirable results than the cosine and the ℓ_2 measures.

In this section, a novel method called *random Manhattan indexing* (RMI) is introduced, which employs random projections in ℓ_1-normed spaces. RMI constructs a VSM directly at a reduced dimension while it preserves the pairwise ℓ_1 distances between vectors in the original high-dimensional VSM. A computationally enhanced version of RMI called *random Manhattan integer indexing* (RMII) is then introduced. RMI and RMII, using the similar principles employed by RI, merge the construction of a VSM and dimension reduction into an incremental—thus, efficient and scalable—process. In Section 4.3.1, the RMI method is explained and evaluated. In Section 4.3.2, the RMII method is explained. RMI and RMII are compared to RI in Section 4.4.

4.3.1 Random Manhattan Indexing

In this section, the Random Manhattan Indexing (RMI) method is proposed: an algorithm that adapts random projections in order to introduce an incremental procedure for constructing ℓ_1-normed vector spaces at a reduced dimensionality. The RMI method employs a two-step procedure: (a) the creation of *index vectors* and (b) the construction of *context vectors*.

In the first step, each context element is assigned exactly to one *index vector* $\vec{r}_i$. Index vectors are high-dimensional and generated randomly such that entries r_j of index vectors have the following distribution:

$$r_i = \begin{cases} \frac{-1}{U_1} & \text{with probability } \frac{s}{2} \\ 0 & \text{with probability } 1 - s \\ \frac{1}{U_2} & \text{with probability } \frac{s}{2} \end{cases}, \tag{4.14}$$

where U_1 and U_2 are independent uniform random variables in $(0, 1)$. In the second step, each target linguistic entity that is being analysed in the model is assigned to a context vector $\vec{v}_c$ in which all the elements are initially set to 0. For each encountered co-occurrence of a linguistic entity and a context element—for example, through a sequential scan of an input corpus—$\vec{v}_c$ that represents the linguistic entity is accumulated by the index vector $\vec{r}_i$ that represents the context element—that is, $\vec{v}_c = \vec{v}_c + \vec{r}_i$. This process results in a VSM of a reduced dimensionality that can be used to estimate the ℓ_1 distances between linguistic entities.

In the constructed VSM by RMI, the ℓ_1 distance between vectors is given by the *sample median* Indyk (2000). For given vectors $\vec{v}$ and $\vec{u}$, the approximate ℓ_1 distance between vectors is estimated by

$$\hat{L}_1(\vec{u}, \vec{v}) = \text{median}\{|v_i - u_i|, i = 1, \cdots, m\}, \tag{4.15}$$

where m is the dimension of the VSM constructed by RMI, and $|.|$ denotes the modulus.

Similar to RI, RMI employs random projections (RPs): a high-dimensional VSM is mapped onto a random subspace of lowered dimension expecting that—with a high probability—relative distances between vectors are approximately preserved. As suggested earlier in Equation 4.4, using the matrix notation, this projection is given by

$$\mathbf{M}'_{p\times m} = \mathbf{M}_{p\times n} \times \mathbf{R}_{n\times m}, \quad m \ll p, n, \tag{4.16}$$

where $\mathbf{R}$ is often called the *random projection matrix*, and $\mathbf{M}$ and $\mathbf{M}'$ denote p vectors in the original n-dimensional and reduced m-dimensional vector spaces, respectively.

In RMI, the stated mapping in Equation 4.16 is given by *Cauchy random projections*. Indyk (2000) suggests that vectors in a high-dimensional space $\mathbb{R}^n$ can be mapped

onto a vector space of lowered dimension $\mathbb{R}^m$ while the relative pairwise ℓ_1 distances between vectors are preserved with a high probability. In Indyk (2000, Theorem 3) and Indyk (2006, Theorem 5), it is shown that for an $m \geq m_0 = \log(1/\delta)^{O(1/\epsilon)}$, where $\delta > 0$ and $\epsilon \leq 1/2$, there exists a mapping from $\mathbb{R}^n$ onto $\mathbb{R}^m$ that guarantees the ℓ_1 distances between any pair of vectors $\vec{u}$ and $\vec{v}$ in $\mathbb{R}^n$ after the mapping does not increase by a factor more than $1 + \epsilon$ with constant probability δ, and it does not decrease by more than $1 - \epsilon$ with probability $1 - \delta$.

In Indyk (2000), this projection is proved to be obtained using a random projection matrix $\mathbf{R}$ that has a *Cauchy distribution*—that is, for r_{ij} in $\mathbf{R}$, $r_{ij} \sim C(0, 1)$. Since $\mathbf{R}$ has a Cauchy distribution, for every two vectors $\vec{u}$ and $\vec{v}$ in the high-dimensional space $\mathbb{R}^n$, the projected differences $x = \hat{\vec{u}} - \hat{\vec{v}}$ also have Cauchy distribution, with the scale parameter being the ℓ_1 distances:

$$x \sim C(0, \sum_{i=1}^{n} |u_i - v_i|). \tag{4.17}$$

As a result, in Cauchy random projections, estimating the ℓ_1 distance between any two vectors $\vec{u}$ and $\vec{v}$ boils down to the estimation of the Cauchy scale parameter from i.i.d. samples x. Because the expectation value of x is infinite,[1] the sample mean cannot be employed to estimate the Cauchy scale parameter. Simply put, this means that $\sum_{i=1}^{n} |u_i - v_i|$ can be used to estimate distances at the reduced dimensionality. Instead, using the 1-stability of Cauchy distribution, Indyk (2000) proves that the median can be employed to estimate the Cauchy scale parameter, and thus the ℓ_1 distances at the projected space $\mathbb{R}^m$.

Subsequent studies simplified the method proposed by Indyk (2000). Particularly, Li (2007) shows that $\mathbf{R}$ with Cauchy distribution can be substituted by a *sparse* $\mathbf{R}$ that has a mixture of symmetric 1-Pareto distribution. A 1-Pareto distribution can be sampled by $1/U$, where U is an independent uniform random variable in $(0, 1)$. This results in a random matrix $\mathbf{R}$ that has the same distribution as described by Equation 4.14.

The RMI's two-step procedure is explained using the basic properties of matrix arithmetic and the descriptions given above. Given the projection in Equation 4.16, the first step of RMI refers to the construction of $\mathbf{R}$: index vectors are the row vectors of $\mathbf{R}$. The second step of the process refers to the construction of $\mathbf{M}'$: context vectors are the row vectors of $\mathbf{M}'$. Using the distributive property of multiplication over addition in matrices,[2] it can be verified that the explicit construction of $\mathbf{M}$ and its multiplication to $\mathbf{R}$ can be substituted by a number of summation operations, exactly as explained from Equation 4.7 to Equation 4.9 for projections in Euclidean spaces. That is, $\mathbf{M}$ can be represented by the sum of unit vectors in which a unit vector corresponds to the co-occurrence of a linguistic entity and a context element. The result of the multiplication of each unit vector and $\mathbf{R}$ is the row vector that represents the context element in $\mathbf{R}$—that is, the index vector.

[1] That is, $E(x) = \infty$, since x has a Cauchy distribution. Cauchy distribution is a heavy tailed distribution, therefore, the expected value does not exist.

[2] That is, $(\mathbf{A} + \mathbf{B})\mathbf{C} = \mathbf{AC} + \mathbf{BC}$.

Therefore, $\mathbf{M}'$ can be computed by the accumulation of the row vectors of $\mathbf{R}$ that represent encountered context elements, as stated in the second step of the RMI procedure.

4.3.1.1 Alternative distance estimators

As stated above, Indyk (2000) suggests using the sample median for the estimation of the ℓ_1 distances. However, Li (2008) argues that sample median estimator can be biased and inaccurate, particularly if the targeted reduced dimensionality (i.e., m) is small. Hence, Li (2008) suggests using the geometric mean estimator instead of the median sample.[1] Accordingly, the ℓ_1 distances at the reduced dimensionality can be estimated by

$$\hat{L}_1(\vec{u}, \vec{v}) = \Big(\prod_{i=1}^{m} |u_i - v_i|\Big)^{\frac{1}{m}}. \tag{4.18}$$

I suggest computing the $\hat{L}_1(\vec{u}, \vec{v})$ in Equation 4.18 using the arithmetic mean of logarithm-transformed values of $|u_i - v_i|$. Therefore, with the help of the logarithmic identities, the multiplications and the exponent power in Equation 4.18 are, respectively, transformed to a sum and a multiplication:

$$\hat{L}_1(\vec{u}, \vec{v}) = \exp\Big(\frac{1}{m}\sum_{i=1}^{m} \ln(|u_i - v_i|)\Big). \tag{4.19}$$

For a computational implementation, Equation 4.19 for estimating $\hat{L}_1$ is more plausible than Equation 4.18—for example, the overflow is less likely to happen during the process. Moreover, calculating the median involves sorting an array of real numbers. Thus, computation of the geometric mean in logarithmic scales can be faster than computation of the median sample, particularly when the value of m is large.

4.3.1.2 RMI's parameters

In order to employ the RMI method for the construction of an ℓ_1-normed VSM at a reduced dimensionality, two model parameters should be decided: (a) the targeted reduced dimensionality of the VSM, which is indicated by m in Equation 4.16 and (b) the number of non-zero elements in index vectors, which is determined by s in Equation 4.14. In contrast to the classic *one-dimension-per-context-element* methods of VSM construction and similar to RI,[2] the value of m in RPs and thus in RMI is chosen independently of the number of context elements in the model (n in Equation 4.16).

In RMI, m determines the probability and the maximum expected amount of distortions ϵ in the pairwise distance between vectors. Based on the proposed refinements of Indyk (2000, Theorem 3) by Li et al. (2007), it is verified that the pairwise ℓ_1 distance

[1] See also Li et al. (2007, Lemma 5–9).

[2] That is, n context elements are modelled in an n-dimensional VSM.

between any p vectors is approximated within a factor $1 \pm \epsilon$, if $m = O(\log p/\epsilon^2)$, with a constant probability. Therefore, the value of ϵ in RMI is subject to the number of vectors p in the model. For a fixed p, a larger m yields to lower bounds on the distortion with a higher probability. Because a small m is desirable from the computational complexity outlook, the choice of m is often a trade-off between accuracy and efficiency. Similar to discussions in Section 4.2.1 for RI, m can be seen as the capacity of the model for accommodating new vectors without causing a large amount of distortion in the distances between vectors.[1] According to my experimental experiences, $m \geq 400$ is suitable for most applications.

The number of non-zero elements in index vectors, however, is decided by the number of context elements (i.e., n) and the sparseness of the VSM at its original dimension (denoted by β). Li (2007) suggests $\frac{1}{O(\sqrt{\beta n})}$ as the value of s in Equation 4.14. As discussed elsewhere, because of the long tail distribution of context elements and linguistic entities (e.g., the Zipfian distribution of words in documents), VSMs employed in distributional semantics—and in general, text analysis—are highly sparse. The sparsity of a VSM in its original dimension (i.e., β) is often considered to be around $10^{-4} \leq \beta \leq 10^{-2}$. However, as the original dimension of VSM n is very large—otherwise there would be no need for dimensionality reduction—the index vectors are often very sparse. Similar to m, larger s produces smaller errors. However, during the construction of a VSM, a large s imposes more processes than a small s.

It is important to note that the influence of s in RI and RMI is different. Whereas in RI, a large s may cause further distortion in the relative estimated distances, in RMI a larger s can help the estimated relative distances converge faster to the relative distances in the original high-dimensional space. Based on the performed experiments and without providing mathematical proofs, for an m-dimensional VSM, I suggest $2\lceil\frac{m}{2\sqrt{\alpha n}}\rceil$ non-zero elements, in which half of them are positive and the other half are negative.

4.3.1.3 Empirical evaluation of RMI

This section reports the performance of the RMI method with respect to its ability to preserve the relative ℓ_1 distance between linguistic entities in a VSM—similar to the observations reported earlier to evaluate RI.[2] Therefore, instead of a task-specific evaluation, it is shown that the relative ℓ_1 distance between a set of words in a high-dimensional *word-by-document* model remains intact when the model is constructed at a reduced dimensionality using the RMI technique. This evaluation is repeated for a *document-by-word* model using the same dataset used in Section 4.2.1.1 for RI, too. The effect of various settings of the RMI's parameters are then explored in the observed results.

The purpose of the reported evaluations is to show the ability of RMI in preserving

[1] Li et al. (2007) details the choice of m using mathematical arguments and observations over synthesised date.

[2] See the experiment in Section 4.2.1.1.

PoS	Words						
Noun	website research	email nhs	support link	software services	students organisations	skills	project
Adjective	online disabled	digital new	mobile current	sustainable fantastic	global innovative	unique	excellent
Verb	use	visit	improve	provided	help	ensure	develop

Table 4.1: Words employed in the experiments. These words are the chosen examples in Ferraresi et al. (2008).

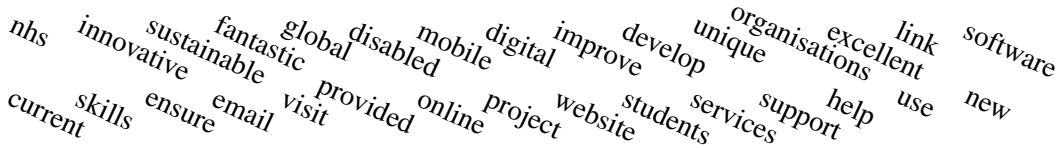


Figure 4.8: List of words sorted by their ℓ_1 distance to the word *research*. The distance increases from left to right and top to bottom.

the relative ℓ_1 distances. Depending on the structure of the data that is being analysed and the objective of the task in hand, the performance of the ℓ_1 distance for similarity measurement can be better or worse than other similarity metrics (e.g., see the experiments in Bullinaria and Levy, 2007). The evaluation designed in this section takes this fact into the consideration. Hence, the purpose of the reported evaluations is not to show the superiority of RMI (thus the ℓ_1 distance) to dimensionality reduction techniques in normed spaces other than ℓ_1 (e.g., RI or truncated SVD in ℓ_2-normed spaces) in a specific task. If, in a task, the ℓ_1 distance shows higher performance than the ℓ_2 distance, then the RMI technique is preferable to the RI technique or truncated SVD. Contrariwise, if the ℓ_2 norm shows higher performance than the ℓ_1 norm, then RI or truncated SVD are more desirable than the RMI method.

In the reported experiment, a word-by-document model is first constructed from ukWaC at its original high dimension. UkWaC is a freely available corpus of 2,692,692 web documents, nearly 2 billion tokens and 4 million types (Baroni et al., 2009).[1] Therefore, a word-by-document model constructed from this corpus using the classic one-dimension-per-context-element method has the maximum dimension of 2.69 million. In order to keep the experiments computationally tractable, the reported results are limited to 31 words from this model, which are listed in Table 4.1. Figure 4.9 shows the increase in the dimensionality of the VSM when a new word from this list is added to the VSM.

In the designed experiment, a word from the list is taken as the reference and its ℓ_1 distance to the remaining 30 words is calculated using the vector representations in

[1] UkWaC can be obtained from `http://wacky.sslmit.unibo.it/doku.php?id=corpora`.

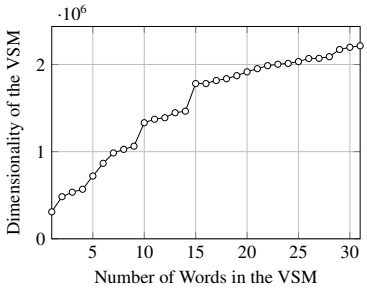


Figure 4.9: The increase in the dimensionality of a word-by-document model constructed from the ukWaC: Adding a new word to the model causes the VSM's dimension to burst when it is constructed using the classic one-document-per-dimension.

the high-dimensional VSM. These 30 words are then sorted in ascending order by the calculated ℓ_1 distance. The procedure is repeated for all of the 31 words in the list, one by one. Therefore, the procedure results in 31 sorted lists, each containing 30 words. Figure 4.8 shows an example of such an obtained sorted list, in which the reference is the word *research*.[1]

The procedure described above is replicated to obtain the lists of sorted words from VSMs that are constructed at reduced dimensionality using the RMI technique, when the method's parameters—that is, the dimension of index vectors as well as the proportion of zero and non-zero elements in them—are set differently. It is expected the obtained relative ℓ_1 distances between each reference word and the 30 other words in an RMI-constructed VSM to be the same as the obtained relative distances in the original high-dimensional VSM. Therefore, for each VSM that is constructed by the RMI technique, the resulting sorted lists of words are compared by the sorted lists that are obtained from the original high-dimensional VSM.

Similar to the other experiments reported in this chapter, the Spearman's rank correlation coefficient (ρ) is employed to compare the sorted lists of words and thus the degree of distance preservation in the RMI-constructed VSMs at reduced dimensionality. Hence, given a list of sorted words obtained from the original high-dimensional VSM (list_o) and its corresponding list obtained from a VSM of reduced dimensionality (list_{RMI}), the Spearman's rank correlation for the two lists is calculated using Equation 4.10 (in which, dif_i is the difference in paired ranks of words in list_o and list_{RMI}, and $n = 30$ is the number of words in each list). The average of ρ over the 31 lists of sorted words, denoted by $\bar{\rho}$, is reported to indicate the performance of RMI with respect to its ability for distance preservation. The closer $\bar{\rho}$ is to 1, the better the performance of RMI with respect to the relative ℓ_1 distance preservation.

[1] Please note that the number of possible arrangements of 30 words without repetition in a list in which the order is important (i.e., all permutations of 30 words) is 30!. As a result, the probability of generating the same sorted list of words when they are arranged by their ℓ_1 distance to another word is $\frac{1}{30!}$.

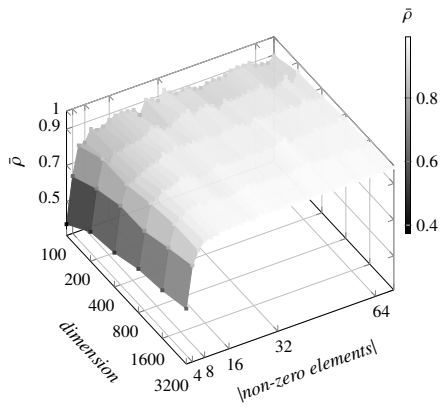


Figure 4.10: The $\bar{\rho}$ axis shows the observed average Spearman' rank correlation between the order of the words in the lists that are sorted by the ℓ_1 distance obtained from the original high-dimensional VSM and the VSMs that are constructed by RMI at reduced dimensionality using index vectors of various numbers of non-zero elements.

Figure 4.10 shows the observed results at a glance when the distances are estimated using the median (Equation 4.15). As shown in the figure, when the dimension of the VSM is above 400 and the number of non-zero elements is more than 12, the obtained relative distances from the VSMs constructed by the RMI technique start to be analogous to the relative distances that are obtained from the original high-dimensional VSM, that is, a high correlation ($\bar{\rho} > 0.90$). For the baseline, the average correlation of $\bar{\rho}_{random} = -0.004$ between the sorted lists of words obtained from the high-dimensional VSM and 31×1000 lists of sorted words that are obtained by randomly assigned distances is reported.

Figure 4.11 shows the same results as Figure 4.10, however, in minute detail and only for VSMs of dimension $m \in \{100, 400, 800, 3200\}$. In these plots, squares (▪) indicate the $\bar{\rho}$ while the error bars show the best and the worst observed ρ amongst all the sorted lists of words. The minimum value of the ρ-axis is set to 0.611, which is the worst observed correlation in the baseline (i.e., randomly generated distances). The dotted line (i.e., $\rho = .591$) shows the best observed correlation in the baseline and the dashed-dotted line shows the average correlation in the baseline ($\rho = -0.004$). As suggested in Section 4.3.1.2, it can be verified that an increase in the dimension of VSMs (i.e., m) increases the stability of the obtained results (i.e., the probability of preserving distances increases). Therefore, for large values of m (i.e., $m > 400$), the difference between the best and the worst observed ρ decreases; average correlation $\bar{\rho} \to 1$, and the relative distances in RMI-constructed VSMs become identical to those in the original high-dimensional VSM.

Figure 4.12 represents the obtained results in the same setting as above, however, when the distances are approximated using the geometric mean (Equation 4.19). The obtained average correlations $\bar{\rho}$ from the geometric mean estimations are almost identical to the median estimations. However, as expected, the geometric mean estimations are more

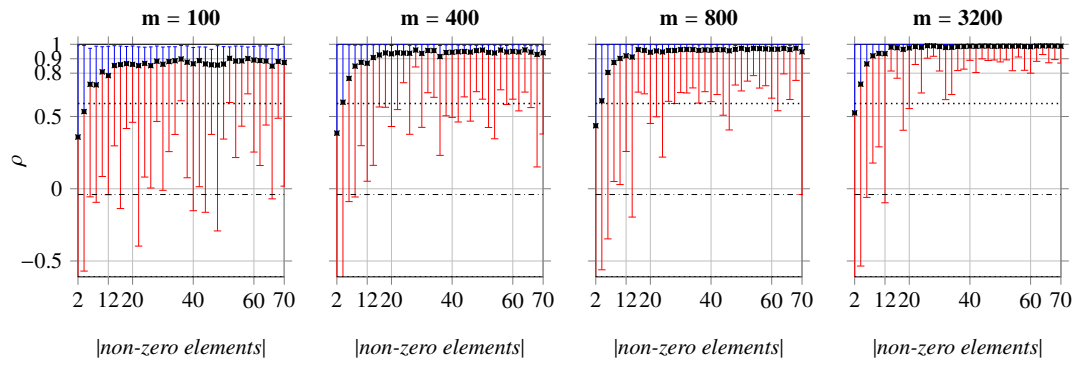


Figure 4.11: Detailed observation of the obtained correlation between relative distances in RMI-constructed VSMs and the original high-dimensional VSM. The ℓ_1 distance is estimated using the median. The squares denote $\bar{\rho}$ and the error bars show the best and the worst observed correlations. The dashed-dotted line shows the random baseline.

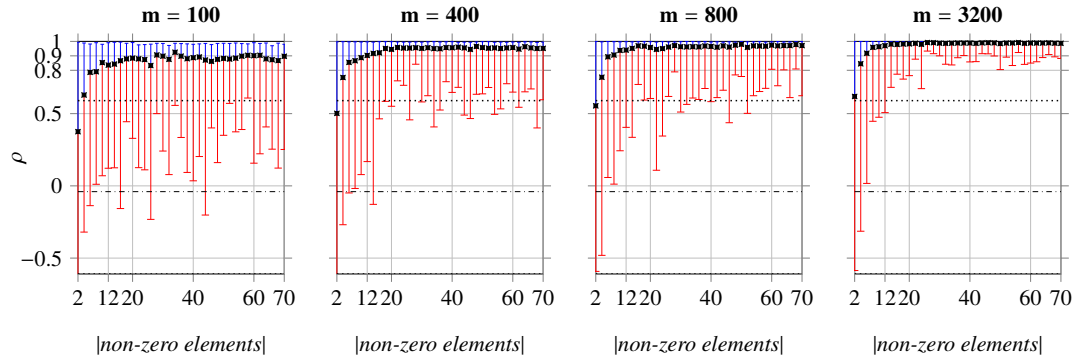


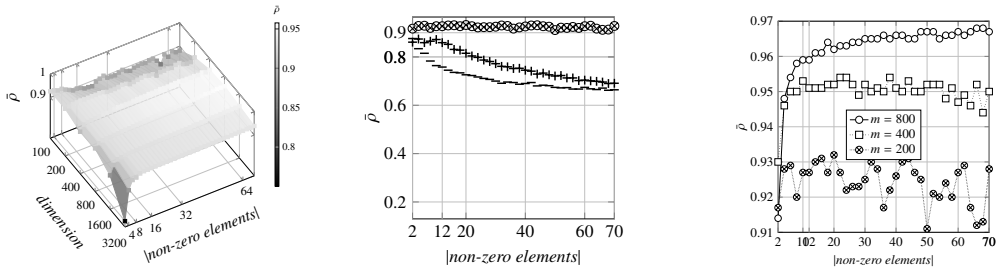
Figure 4.12: The observed results when the ℓ_1 distance in RMI-constructed VSMs is estimated using the geometric mean.

reliable for small values of m; particularly, when using the geometric mean, the worst observed correlations are higher than those observed when using the median estimator.

This experiment is also repeated over the document-by-word models that have been employed earlier in Section 4.2.1.1. Instead of the Euclidean distance, however, the constructed models are used to verify the ability of RMI-constructed VSMs to preserve ℓ_1 distances between vectors. Results are shown in Figure 4.13.

4.3.2 Random Manhattan Integer Indexing

The application of the RMI method is hindered by two obstacles: float arithmetic operations required for the construction and processing of the RMI-constructed VSMs and the calculation of the product of large numbers when ℓ_1 distances are estimated using the geometric mean.



(a) (b) (c)

Figure 4.13: The RMI's ability to preserve relative ℓ_1 distances in a document-by-word model: The performance is assessed using the observed $\bar{\rho}$ over a set of 10,000 documents chosen randomly from the *WaCkypedia_EN* in an experiment similar to Section 4.2.1.1. Figure 4.13a shows the overall observed result when the RMI' parameters are set differently. Figure 4.13b shows the same results only when the dimension of VSM is 200. In this figure, the minimum value of the $\bar{\rho}$-axis is set to the best observed correlation $\rho = 0.1375$ when distances are generated randomly (first baseline). The + and − marks show $\bar{\rho}$ when ℓ_1 distance is estimated in RI-constructed VSMs of dimensionality 1600 using the estimator in Equation 4.19 and the standard definition of the ℓ_1 distance, respectively. Figure 4.13c plots the same observed results only for RMI and when $m \in \{200, 400, 800\}$. These results are similar to the experiments with the word-by-document model. It can be verified that an increase in the dimension of VSM results in an increase in $\bar{\rho}$.

The proposed method for the generation of index vectors in RMI results in index vectors of non-zero elements that are real numbers. Consequently, index vectors and thus context vectors are arrays of floating point numbers. These vectors must be stored and accessed efficiently when the RMI technique is employed in an application. However, storing and processing floating numbers are resource intensive, and therefore not desirable in real-world applications—particularly when dealing with large corpora. Even if the requirement for the storage of index vectors is alleviated—for example, using a derandomisation technique for their generation—context vectors that are derived from these index vectors are still arrays of float numbers and their storage and process is of high space and time complexity.

To tackle this problem, I suggest substituting the value of non-zero elements of RMI's index vectors (given in Equation 4.14) from $\frac{1}{U}$ to integer values of $\lfloor \frac{1}{U} \rfloor$, where $\lfloor \frac{1}{U} \rfloor \neq 0$—that is:

$$r_i = \begin{cases} \lfloor \frac{1}{U_1} \rfloor & \text{with probability } \frac{s}{2} \\ 0 & \text{with probability } 1 - s \\ \lfloor \frac{1}{U_2} \rfloor & \text{with probability } \frac{s}{2} \end{cases}. \tag{4.20}$$

I argue that the resulting random projection matrix still has an asymptomatic Cauchy distribution. Therefore, the proposed methodology to estimate the ℓ_1 distance between vectors is still valid. The ℓ_1 distance between context vectors must be still estimated using either the median or the geometric mean.

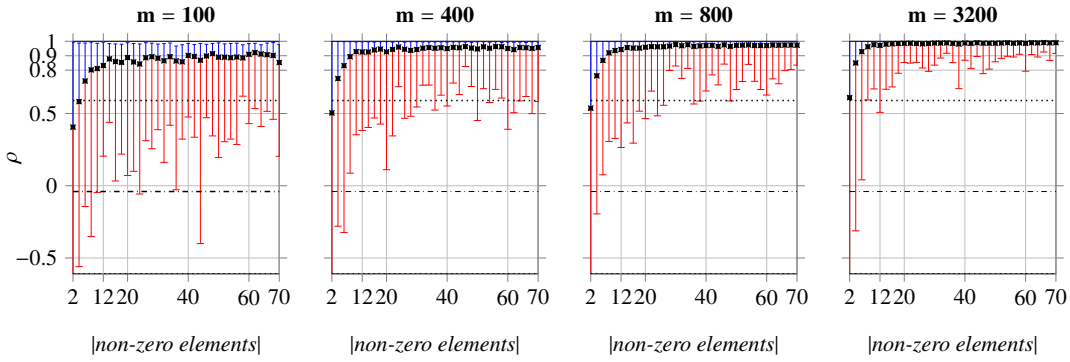


Figure 4.14: The observed results when using the RMII method for the construction and estimation of the ℓ_1 distances between vectors. The method is evaluated in the same setup as the RMI technique.

The use of the median estimator—for the reasons stated in Section 4.3.1.1—is not plausible. On the other hand, the computation of the geometric mean can be laborious as the overflow is highly likely to happen during its computation. Using the value of $\lfloor \frac{1}{U} \rfloor$ for non-zero elements of index vectors, it is evident that for any pair of context vectors $\vec{u} = (u_1, \cdots, u_m)$ and $\vec{v} = (v_1, \cdots, v_m)$, if $u_i \neq v_i$ then $|u_i - v_i| \geq 1$. Therefore, for $u_i \neq v_i$, $\ln |u_i - v_i| \geq 0$ and thus $\sum_{i=1}^{m} \ln(|u_i - v_i|) \geq 0$. In this case, the exponent in Equation 4.19 is a scale factor that can be discarded without a change in the relative distances between vectors.[1] Based on the intuition that the distance between a vector and itself is zero and the explanation given above, inspired by smoothing techniques and without being able to provide mathematical proofs, I suggest estimating the relative distances between vectors using

$$\hat{L}_1(\vec{u}, \vec{v}) = \sum_{\substack{i=1 \\ u_i \neq v_i}}^{m} \ln(|u_i - v_i|). \tag{4.21}$$

In order to distinguish the above changes in RMI, the resulting technique is called random Manhattan integer indexing (RMII). The experiment described in Section 4.3.1.2 is repeated using the RMII method. As shown in Figure 4.14, the obtained results are almost identical to the observed results when using the RMI technique. While RMI performs slightly better than RMII in lower dimensions—for example, $m = 400$—RMII shows more stable behaviour than RMI at higher dimensions—for example $m = 800$. However, in all these cases, RMII demands less memory and processing resources for its computations.

[1] Please note that according to the axioms in the distance definition, the distance between two numbers is always a non-negative value. When index vectors consist of non-zero elements of real numbers, the value of $|u_i - v_i|$ can be between 0 and 1, that is, $0 < |u_i - v_i| < 1$. Therefore, $\ln(|u_i - v_i|)$ can be a negative number and thus the exponent scale is required to make sure that the result is a non-negative number.

4.4 Comparing RMI and RI

RMI and RI utilise a similar two-step procedure consisting of the creation of index vectors and the construction of context vectors. In addition, both RMI and RI are incremental techniques that construct a VSM at reduced dimensionality directly, without requiring the VSM to be constructed at its original high dimension. Despite these similarities, RMI and RI are motivated by different applications and mathematical theorems. RMI is justified using asymptotic Cauchy random projections whereas RI is justified using asymptotic Gaussian random projections.

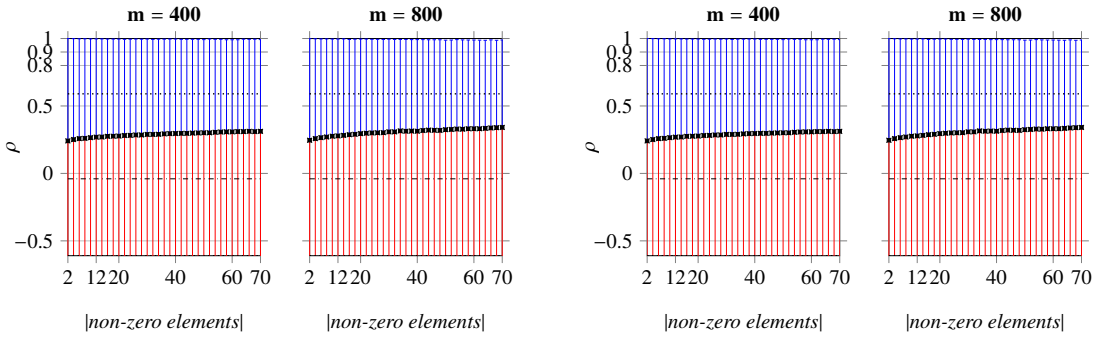
As described above, RMI approximates the ℓ_1 distance using a *non-linear estimator*, which has not yet been employed for the construction of VSMs and the calculation of ℓ_1 distances in distributional approaches to semantics. In contrast, RI approximates the ℓ_2 distance using a *linear estimator*. RI has initially been justified using the mathematical model of the sparse distributed memory (SDM). Later, as suggested in this chapter, the RI method was explained using the lemma proposed by Johnson and Lindenstrauss (1984)—which elucidates random projections in Euclidean spaces (see Section 4.2 for details). Although both the RMI and RI methods can be established as α-stable random projections—respectively for $\alpha = 1$ and $\alpha = 2$—the methods cannot be compared as they address different goals. If, for a given task, the ℓ_1 norm outperforms the ℓ_2 norm, then RMI is preferable to RI. Contrariwise, if the ℓ_2 norm outperforms the ℓ_1 norm, then RI is preferable to RMI. As implied in the reported evaluations and stated above, RI and RMI cannot be replaced with each other. As shown in the previous sections, using RI for dimensionality reduction causes a large distortion in the relative ℓ_1 distances between vectors. Reversely, RMI does not preserve the relative ℓ_2 distances between vectors.

To support the earlier claim that RI-constructed VSMs cannot be used for the ℓ_1 distance estimation, the RI method is evaluated in the experimental setup that has been used for the evaluation of RMI and RMII. In these experiments, however, RI is employed to construct vector spaces at reduced dimensionality and estimate the ℓ_1 distance using Equation 4.13 (the standard ℓ_1 distance definition) and Equation 4.15 (the median estimator) for $m \in 400, 800$. As shown in Figure 4.15, the experiments support this claim.

4.5 Summary

In this chapter, the applications of random projections for constructing vector spaces with reduced dimensionality are outlined. As discussed, these methods can be employed to enhance the performance in distributional semantic models.

This chapter has two contributions in particular. First, in Section 4.2, the random indexing method is explained mathematically; and its two-step procedure is delineated using sparse asymptotic Gaussian random projections. Consequently, criteria for setting



(a) The standard ℓ_1 distance definition

(b) The median estimator

Figure 4.15: Evaluation of RI for estimating ℓ_1 distances for $m = 400$ and $m = 800$ when the distances are calculated using (a) the standard definition of distance in ℓ_1-normed spaces and (b) the median estimator. The obtained results using RI do not show a correlation to the ℓ_1 distances in the original high-dimensional VSM.

the method's parameters are suggested. Second, in Section 4.3, a novel technique, named random Manhattan indexing (RMI), for the construction of ℓ_1-normed VSMs directly at reduced dimensionality is suggested. In addition, Section 4.3.2 introduces the random Manhattan integer indexing (RMII) technique—that is, a computationally enhanced version of the RMI technique. The ability of these methods to preserve ℓ_1 distances are demonstrated using empirical evaluations.

As discussed, the use of random projections in the incremental way suggested in this chapter has a number of benefits. First, it enhances the computational complexity of the construction of models by combining the process of collecting co-occurrences with the dimensionality reduction process. The result is a vector space model constructed directly with reduced dimensionality. Second, because of the reduced dimensionality of the vectors, the subsequent similarity computations are performed faster. Third, the proposed incremental method provides the capability of updating a model at any time during its use, which makes it suitable for frequently updated data, particularly, in the context of big-text data analytics.

As suggested in Section 4.4, vector spaces that are constructed using random projections, such as the RI and RMI techniques, are limited to the specific normed space that they are designed for. There are methods that claim they can overcome this restriction—for example, Li et al.'s (2006a) conditional random sampling. However, they have not yet been applied to the vector space models of semantics. The use of these methods is one way to extend the presented research in this chapter. In the proposed methods in this chapter, only one random projection is applied before estimating distances between vectors. However, it is possible to use a chain of projections—for example, as it is used in the implementations of neural network algorithms. Such combinations are also possible for RMI and RI.

Last but not least, the design principles employed in this chapter to reintroduce RI and propose RMI and RMII can be employed for normed spaces other than the ℓ_1 and the ℓ_2-normed. This is an exciting future research that has not yet been investigated for natural language processing applications. Random projections are a vibrant research topic in modern mathematics and statistics and the future advances in these fields will most definitely result in new efficient methods and techniques for big text data analytics.

This page is intentionally left blank.

Chapter 5

Identifying Co-Hyponym Terms: The Method and its Evaluation

In this chapter, the proposed method for identifying co-hyponym terms is explained and evaluated. The principles of automatic term recognition and distributional semantics are combined to implement a method that extracts terms from a category of similar concepts (i.e., co-hyponyms). After the extraction of candidate terms, stable random projections are employed to represent these candidate terms as low-dimensional vectors. These vectors are derived automatically from the co-occurrences of candidate terms and words that appear in their proximity (context-windows). In a memory-based k-nearest neighbours learning framework, and using a small set of manually annotated terms, co-hyponym terms are identifiedf by classifying these vectors.

Section 5.1 reintroduces the task and justifies the proposed method based on the principles of distributional semantics. This introduction is followed by delineating the method in Section 5.2. In Section 5.3, the evaluation framework and material are discussed. Results from a number of experiments in the defined evaluation framework are reported in Section 5.4 and discussed in Section 5.5. After suggesting an approach for improving the performance of the method for large recall values in Section 5.6, the chapter concludes with a summary in Section 5.7.[1]

[1]The proposed method in this chapter has been published and evaluated partly in Zadeh and Handschuh (2014b) and Zadeh and Handschuh (2014c).

5.1 Introduction

As is explained in Chapter 3, in automatic term recognition (ATR), given a special corpus, the goal is to automatically extract a specialised vocabulary—that is, in its simplest form, a terminological resource composed of a set of lexical units known as *terms*. Terms can be either *simple* or *complex*—that is, single-token or multi-token lexical units. A terminological resource is an indispensable component of a system employed to communicate a specialised knowledge. Hence, it signifies diverse concepts in the targeted domain knowledge. These concepts, and thus terms, are often organised according to a classification scheme, which is determined by a number of factors such as the intended application context. In an information system, this categorisation is often a major mechanism to reflect the structure of the (conceptualised) specialised knowledge—for example, as practised in *ontology engineering* (e.g., see L'Homme and Bernier-Colborne, 2012, for an overview) and as explained in Section 1.1.

For instance, the terms *lexicon*, *corpus*, *terminology*, *parsing* and *information extraction* are conceivable entries from a terminological resource in the domain of computational linguistics. In this list, *lexicon*, *parsing*, and *terminology* are simple terms, whereas *information extraction* is a complex term. According to a classification scheme (i.e., a conceptualisation of the domain knowledge), *lexicon* and *corpus* can be grouped under the concept category *language resource*. Similarly, *information extraction* and *parsing* can be classified under the category *technology and process* (Figure 5.1). It is worth mentioning that a term can appear in more than one category of concepts. In the given example, the term *terminology* appears in both categories, as a term that can signal both a language resource and a processing resource (see also Figure 3.2 in Chapter 3). Accordingly, as discussed in Section 1.1, terms under each category of concepts are in a co-hyponymy relationship since they share a similar hypernym. For instance, in the example given in Figure 5.1, *lexicon*, *terminology* and *corpus* are *co-hyponym terms*.

A number of research studies have attempted to extract and define a scheme for the categorisation of terms into co-hyponym groups, either implicitly using a clustering technique (e.g., as suggested in Dupuch et al., 2014; Cimiano et al., 2005), or explicitly by inducing inference rules—such as using an automatic or manual engineering of Hearst's (1992) lexico-syntactic patterns (e.g., as suggested in Maynard et al., 2009).[1] In a large number of applications, however, the classification scheme[2] is known (or, at least, a partial knowledge of it exists). In this case, finding co-hyponym terms that belong to a particular category of concepts is a typical task. In the context of ontology engineering, the former research is usually a sub-process of the *ontology learning* task, whereas the latter is often demanded for *ontology population* (see Buitelaar et al., 2005; Wong et al., 2012). The

[1] Note that the use of these patterns is not limited to taxonomy induction processes, as is shown in the next few pages.

[2] That is, the set of hypernyms in the conceptualisation of the domain knowledge under investigation.

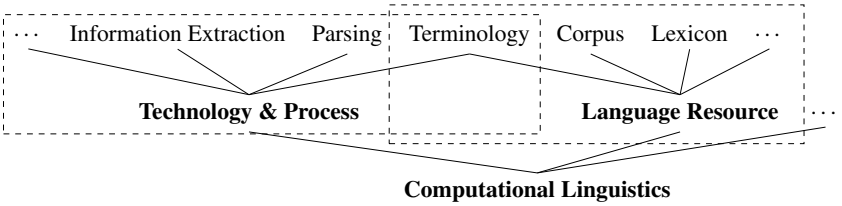


Figure 5.1: Taxonomy and co-hyponyms: This example shows a simple taxonomy in the domain of computational linguistics. Terms are classified into two categories: *language resource* and *technology and process*. Terms under each category (placed in boxes) form a group of *co-hyponyms*. A number of terms such as *terminology* in this example can be *polysemous*, hence classified under more than one category of concepts. An ideal ATR system extracts terms listed in the top row of this figure. As suggested, terms can be organised according to a taxonomy. One way to approach this task is to identify co-hyponyms.

focus in this Chapter is on the latter.

Entity extraction methods are commonly employed to distil co-hyponym terms, of which *bio-entity recognition tasks* are the most established examples (e.g., see Kim et al., 2004). As detailed in Section 1.2, these methods, however, are not suitable for a number of use cases due to their lack of flexibility and a mechanism for resembling the knowledge structure. Moreover, developing entity taggers is restricted by the availability of manually annotated corpora. In these corpora, individual mentions of terms and their concept category are required to be manually annotated. A few techniques exploit information redundancy in very large corpora to obviate this requirement.[1] However, in special (domain-specific) corpora, using information redundancy alone can be insufficient to automatically generate annotated data (e.g., as shown in Section 5.4.1.1). Last but not least, using an entity tagger for extracting co-hyponyms abandons an important characteristic of special corpora—that is, reduced lexical ambiguity.

As discussed in Chapter 1 and 3, in specialised languages, terms are often coined to facilitate communication by reducing lexical ambiguity. Therefore, synonymy and polysemy are less frequent in specialised languages than general language. The concept of *sense* can be defined differently depending on the context (e.g., see Cimiano et al., 2013, for an elaboration of a three faceted definition in the *ontology-lexicon* framework).[2] Thus, the meaning of *polysemy* can be interpreted differently. In the context of this discussion, I suggest that a domain ontology populated by instances extracted from a special corpus plays an analogical role to that of a lexical database supplemented by word senses in general language. Accordingly, I assume that the relationships between instances (i.e., terms in the special corpus) and concepts in the domain ontology is similar to the re-

[1] For instance, see Etzioni et al. (2005). These methods are often employed for the extraction of proper nouns in general language. Usually, the manual annotation is replaced by hand-crafted lexico-syntactic patterns, or a small number of *seed examples*.

[2] *Sense* has different senses and thus is polysemous, so to speak!

lationship between word forms and senses in a general language lexical database.[1] Let me explain the proposed argument by a comparison between WordNet and the *GENIA ontology* populated by annotations provided in the GENIA corpus.

WordNet is unarguably a general language vocabulary in which the proportion of polysemous words is approximately 17% (according to Miller, 1995). The GENIA corpus (which is a well-known corpus in the domain of molecular biology) provides manual annotations for 92,722 term mentions (Kim et al., 2003). These term annotations in the GENIA corpus are grounded on the GENIA ontology. The GENIA ontology consists of 45 classes that are organised in a hierarchical taxonomy of 6 levels. The annotated 92,722 term mentions form a vocabulary of 34,077 distinct entries (hereinafter *GENIA terminological resource*). Among them, individual mentions of 1,373 entries are annotated with at least two classes from the GENIA ontology. If these terms are considered polysemous, compared to WordNet, only a small fraction (i.e., $\frac{1372}{34077} = 4\%$) are polysemous.[2] Although a direct comparison of the two resources can be not accurate,[3] it is still a reliable evidence of the expected differences of the properties[4] of relationships between entries in a terminological resource and a general language lexical database. Disregarding the employed vocabulary for describing this phenomenon (i.e., whether or not to use the word *polysemy*), this thesis exploits the described phenomenon and suggests a method to identify co-hyponym terms.

Similar to ATR and in contrast to entity recognition tasks, the method proposed for identifying co-hyponym terms works at a corpus level and does not deal with individual occurrences of a term in text snippets. However, in contrast to ATR (which extracts terms from diverse categories of concepts in a domain knowledge) and similar to entity recognition, the objective is to extract a particular subset of terms that signify a similar hypernym.

The proposed method in the investigated use case has many practical applications: ranging from classic applications in information retrieval (e.g., see principles that are suggested by Rijsbergen, 1977, for *index term weighting*) to more recent so-called *ontology-based information systems* as (assistive) tools for maintaining and populating domain

[1] The conceptualisation behind a domain ontology (i.e., the number of classes and their relationship) plays a role in the proposed analogy and the subsequent proposed comparison in this section (i.e., the relationship between *granularity* vs. *condensation* of concepts in domain ontologies). For simplicity without the loss of generality, I discard this relationship.

[2] A same conclusion can be drawn by analysing the distribution of words senses in SemCor (Mihalcea, 1998)—that is, a corpus of general language text annotated with WordNet senses—and the GENIA corpus.

[3] For instance, as they are built from opposite viewpoints. In constructing WordNet, for a given word, an inventory of all the meanings of the word is made by searching the occurrences of the word in large text corpora. In the GENIA terminological resource, however, from a limited number of observations in the specialised corpus, all the concepts that terms represent are collected. In the proposed example, as put by Cimiano et al. (2013), a *reification* of the link between terms (lexical forms) and ontological references from the GENIA corpus are assumed to represent senses. According to this terminology, in this thesis, terms that are reified to the same ontological reference are considered co-hyponyms.

[4] Such as diversity and frequency.

ontologies. Apart from these two broad applications, there is a growing demand of information extraction tasks that, in fact, can be boiled down to the proposed co-hyponym identification task. For example, in the so-called expertise finding task (e.g., see Balog and de Rijke, 2008) a major process is the identification of *expertise topics* (e.g., Buitelaar and Eigner, 2008). In this scenario, the *expertise topics* are, in fact, a set of co-hyponym terms, and can thus be identified using the method proposed in this chapter. Classifying user-generated annotations in applications such as *tag-based information access* (e.g., Yi, 2010) is another example. A similar use case is presented by Chakraborty et al. (2014) for extracting information from unstructured text ads. Last but not least are applications such as *technology watch* that intend to provide *technological intelligence* by machine reading of large text corpora (e.g., as explained in QasemiZadeh, 2010).

As described in Section 1.2, the co-hyponym identification task can be formulated as a classification task (see Figure 1.1 in Chapter 1). Therefore, the proposed method is realised as an ad hoc term-weighting procedure on top of an ATR system's two-step procedure—candidate term extraction followed by term weighting and ranking. As described in Chapter 3, after the extraction of candidate terms, ATR combines the *unithood* and *termhood* scores to weight terms (Figure 5.2). Unithood characterises the strength of syntagmatic relationships between the tokens that compose complex terms. Termhood, however, characterises a paradigmatic relationship—that is, the association of candidate terms to the concepts in a specialised knowledge domain, which is verbalised by the special corpus under investigation. Termhood in ATR disregards terms' associations to different concept categories. In contrast, in the proposed task, a score that discerns these associations must be devised. This score, however, is similar to termhood in the sense that it characterises a *paradigmatic* relationship—that is, the co-hyponymy relationship between terms that are grouped under a category of concepts, such as the relationship between *lexicon*, *corpus*, and *terminology* exemplified in Figure 5.1.

A distributional approach is employed to design this score. By extending Harris's (1954) distributional hypothesis, one can claim that the context in which terms are used can be exploited to identify their concept category.[1] Hence, in this thesis, it is assumed that the association of a term to a concept category can be characterised using the syntagmatic relation of the term and its co-occurred words in windows of text extended in the vicinity of the term's mentions in the corpus (i.e., *context-windows* as shown in Figure 5.3).[2] Accordingly, I hypothesise that co-hyponym terms tend to have similar distributional properties in context-windows. In order to quantify these distributional similarities,

[1] With the assumption that multi-token complex terms have no compositional semantics.

[2] This claim is not new. The *syntagmatic consequences* of hyponymy relationships in particular—and, the syntagmatic consequences of paradigmatic relationships in general—have been widely exploited in research literature. The aforementioned Hearst's (1992) patterns is, perhaps, the most familiar example. Hearst exploits the syntagmatic consequences of hyponym relationships to suggest patterns such as $\cdots X$ *and other* $Y \cdots$ for the automatic acquisition of hyponymy relationships. In this thesis, this linguistic phenomena is articulated in the framework of distributional semantic models in order to characterise co-hyponymy relationships.

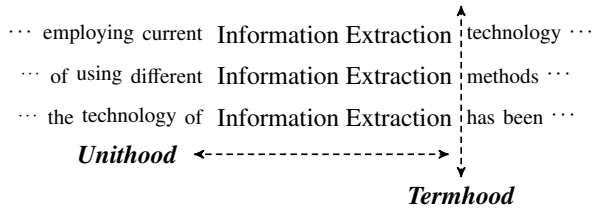


Figure 5.2: Unithood and termhood with respect to the terms usage in special corpus. In the given example, an ideal unithood measure identifies a strong association—that is, a syntagmatic relationship—between the two tokens *information* and *extraction*, and hence marks *information extraction* as a probable complex candidate term. Termhood, however, characterises the associations of specialised meanings to candidate terms: a paradigmatic relationship.

vector space models—which are described thoroughly in Chapter 2—are employed.

Words that appear in context-windows are represented by the elements of the standard basis of a vector space—that is, informally, dimensions of a vector space—and each candidate term is represented by a vector. In this vector space, the coordinates of vectors is determined by the co-occurrence frequency of words that appear in context-windows and candidate terms in a special corpus. Consequently, the values assigned to the coordinates of a vector represent the correlation of the candidate term that the vector represents and the words in context-windows. As a result, the vectors' proximity can be employed to compare the distributional similarities of candidate terms. As suggested by Sahlgren (2006), the result is a geometric metaphor of meaning: a semantic space, which, following previous research such as Schütze (1993), can be named a *term-space model*.

In this term-space model, a category of terms (i.e., co-hyponyms) is characterised using a set of *reference terms* (shown by R_s). R_s is a small number of terms that are manually annotated with their corresponding concept category. The distance between vectors that represent candidate terms and the vectors that represent R_s is assumed to determine the association of candidate terms to the concept categories represented by R_s. This association is computed using a k-nearest neighbours (k-nn) framework (Daelemans and van den Bosch, 2010). As is explained in Section 2.4 of Chapter 2, the memory-based k-nn learning technique provides a *similarity-based reasoning framework* that can be used to identify terms' categories without the need for formulating these associations using a meta-language, such as rules.

Previous research has confirmed the proposed term-space model's viability in capturing paradigmatic relationships between *words*, which can be taken as the evidence for the proposed method's practicability. However, as is described in Chapter 2, like other distributional approaches to semantics, finding a context-window's configurations that best characterises terms from similar concept categories is still a major research concern that must be investigated empirically. Besides the configuration of context-windows, the para-

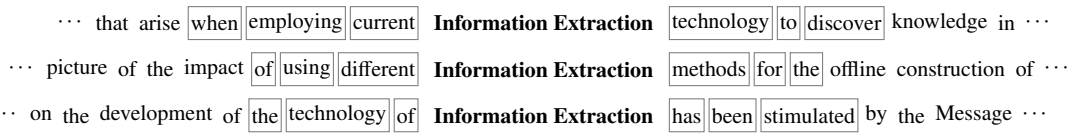


Figure 5.3: Illustration of a *context-window* of size 3 tokens that extend around a term: in the example above, this context-window is shown for the occurrences of the candidate term *information extraction* in three different sentences from a special corpus. For each occurrence of the candidate term *information extraction* in each line (i.e., a sentence), the context-window consists of words that are placed in rectangles. To construct a distributional model, the co-occurrences of *information extraction* and words within these context-windows are represented by a vector.

meters of the classification framework are additional elements that influence the method's performance. The employed metric for similarity measurement, and the neighbourhood size (k) are the parameters that can be set differently in the k-nn algorithm. Understandably, a change in these parameters alters the observed results. To grasp the method's behaviour, the effect of these parameters must be investigated empirically, too.

The remainder of this chapter is devoted to the delineation of the proposed method and the employed approach for its empirical investigation. Section 5.2 details the proposed method. The evaluation methodology and materials are described in Section 5.3. Subsequently, the observed results are reported in Section 5.4, which is followed by a summary in Section 5.7.

5.2 The Proposed Methodology

Figure 5.4 illustrates the method. It is assumed that an ATR system extracts a list of candidate terms and, perhaps, ranks them by its own weighting mechanism. The extracted list of candidate terms is then processed for constructing a vector space by scanning the corpus for the occurrences of the candidate terms. It is assumed that a small number of these candidate terms (e.g., 100) are annotated with their concept categories. Vectors that represent these annotated terms form a set of reference vectors R_s. In the constructed vector space, using a k-nn algorithm, R_s is employed to assign a concept category association weight c_w to the remaining candidate terms.

For a given candidate term represented by the vector $\vec{v}$, c_w is computed using

$$c_w(\vec{v}) = \sum_{i=1}^{k} s(\vec{v}, \vec{r_i})\delta(\vec{r_i}), \tag{5.1}$$

where $s(\vec{v}, \vec{r})$ denotes similarity between $\vec{v}$ and $\vec{r} \in R_s$, in which R_s is sorted by $s(\vec{v}, \vec{r})$ in descending order. If $\vec{r}$ represents a term from the targeted category of concepts, then $\delta(\vec{r}) = 1$, otherwise $\delta(\vec{r}) = 0$. The function s can be defined in a number of ways; three

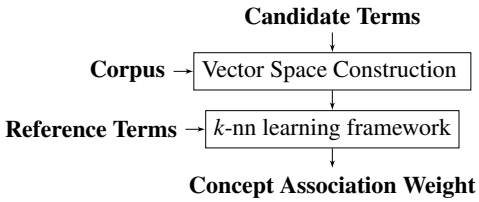


Figure 5.4: The proposed method for measuring the concept category associations.

widely used definitions are employed:[1]

- $s(\vec{v},\vec{r}) = \cos(\vec{v},\vec{r})$, that is, the cosine of the angles between $\vec{v}$ and $\vec{r}$;
- $s(\vec{v},\vec{r}) = \frac{1}{1+\ell_2}$, where ℓ_2 is the Euclidean distance between $\vec{v}$ and $\vec{r}$; and
- $s(\vec{v},\vec{r}) = \frac{1}{1+\ell_1}$, where ℓ_1 is the City block distance between $\vec{v}$ and $\vec{r}$.

As can be understood, the vector space construction is the major step in the proposed methodology, which is described in the following section.

5.2.1 Vector Space Construction Methodology

In distributional semantic models the curse of dimensionality is a common barrier, as is discussed in Chapter 2. In the proposed distributional method, due to the Zipfian distribution of terms and words in context-windows, the curse of dimensionality is an inevitable problem, too—that is, vectors that represent candidate terms are high-dimensional and sparse (i.e., most of the elements of vectors are zero). These properties of vectors hamper the subsequent classification process. To overcome this barrier, term-space models are constructed incrementally and at a reduced dimensionality using random projections techniques, which are proposed and justified in Chapter 4.

Each candidate term is assigned to an m-dimensional term vector $\vec{t}$. Term vectors are initially empty—that is, all the elements of $\vec{t}$ are set to zero. The corpus is then scanned for the occurrences of candidate terms and words that co-occur with them in context-windows. Each of these words is assigned exactly to one word vector $\vec{w}$. Similar to term vectors, word vectors are also m-dimensional. However, the elements w_j of each $\vec{w}$ are instantiated with random values with the following distributions:

$$w_j = \begin{cases} \lfloor \frac{-1}{U_1} \rfloor & \text{with probability } \frac{1}{2\alpha} \\ 0 & \text{with probability } 1 - \frac{1}{\alpha} \\ \lfloor \frac{1}{U_2} \rfloor & \text{with probability } \frac{1}{2\alpha} \end{cases}, \tag{5.2}$$

where α is a small value. As a result, most of the elements of w_j are set to zero and only a few have a non-zero value. Once a $\vec{w}$ is generated and assigned to a word, it is stored and

[1] See Section 2.3.4 of Chapter 2 for a long list of similarity measures.

kept for later usages.

If the similarity between $\vec{v}$ and $\vec{r}$ is measured using the cosine or the Euclidean distance (i.e., in an ℓ_2-normed space), then U_1 and U_2 are set to 1 and $\alpha = \mathrm{O}(\sqrt{|\vec{w}|})$, where $|\vec{w}|$ is the number of word vectors. In this case, $\vec{w}$ vectors resemble a random projection matrix that has a standard Gaussian distribution.[1] However, if the similarities are measured using the city block distance (i.e., in an ℓ_1-normed space), then U_1 and U_2 are two independent uniform random variables in $(0, 1)$ and $\alpha = \mathrm{O}(\sqrt{|\vec{w}|/100})$, where the constant factor 0.01 is an approximation of the sparsity of term-word co-occurrences in the corpus. In this case, $\vec{w}$ vectors resemble a random projection matrix with a standard Cauchy distribution.[2]

To capture the co-occurrence of a candidate term and a word in a context-window, the word vector $\vec{w}$ that represents the word is added to the term vector $\vec{v}$ that represents the candidate term—that is, $\vec{v} = \vec{t} + \vec{w}$. This procedure is repeated to capture all the co-occurrences of candidate terms and words that appear in context-windows in the input corpus. The result is a vector space that reflects the observed co-occurrences of terms and words, however, at the reduced dimension m.

Subsequent to the construction of a vector space using the method described above, the similarities between term vectors and reference term vectors in R_s must be computed. In the ℓ_2-normed constructed vector spaces, for the given vectors $\vec{v}$ and $\vec{u}$, the cosine between them is calculated using Equation 2.15 and their Euclidean distance using Equation 2.16. In the ℓ_1-normed spaces, the city block distance, however, is computed using the estimator proposed in Equation 4.21. Once computed, these distances and similarities between vectors are used to weight candidate terms according to Equation 5.1.

For instance, given the term *information extraction* in Figure 5.3, this term is first assigned to an empty m-dimensional term vector $\vec{t_{\text{ie}}}$. Assume that all the term's occurrences in the corpus are listed in this figure and the context-window is configured as shown (i.e., context-windows are stretched around the term for the size of three tokens). Then, each word placed in a rectangle is assigned exactly to one m-dimensional word vector $\vec{w}$. In this example, the result is 15 word vectors. The vector $\vec{t_{\text{ie}}}$ is then accumulated by these word vectors. Since the words *to* and *for* occur twice, their corresponding word vector is also accumulated twice. The generated word vectors are stored and used for constructing term vectors for candidate terms other than *information extraction*.

It becomes evident that the proposed method for constructing an ℓ_2-normed term-space model is equivalent to the random indexing technique. In the ℓ_1-normed spaces, however, the random Manhattan integer indexing (RMII) technique is employed. In Chapter 4, it is shown that the relative distances of vectors in these m-dimensional models are similar to the relative distances in the original high-dimensional vector spaces—that

[1] That is, a random projection matrix with asymptotic Gaussian distribution (see Chapter 4, Section 4.2 on Gaussian random projections).

[2] That is, a random projection matrix with asymptotic Cauchy distribution (see Chapter 4, Section 4.3 on random projections in ℓ_1-normed spaces).

Abstract	Sentence	Token	Type[1]
2,000	18,546	490,941	22,484 19,576

Table 5.1: A statistics summary of the GENIA corpus and its annotated terms.

is, when a term-space model is constructed using the classic *one-dimension-per-context-element* method. Chapter 4 also discusses criteria for setting the parameters of the vector space construction—that is, m and α. Simply put, it is shown that the value of m is determined independently of the original dimension of the vector space (i.e., the number of distinct words that appear in context-windows). It is, however, determined by the number of term vectors $\vec{t}$ in the model. It is also described that α is decided by the original dimension and the sparseness of the vector space in its original dimension. These criteria are employed for setting the method's parameters in the reported evaluations.

5.3 The Evaluation Framework

5.3.1 Corpus and Performance Measure

To evaluate the plausibility of the proposed method and to determine its performance, a set of experiments over the *GENIA corpus* are carried out and the obtained results are reported. The GENIA corpus is a collection of 2000 abstracts from the *domain of molecular biology* (Kim et al., 2003). The corpus comprises manual annotations of biological term mentions from several concept categories, which are organised in an ontology—also called the *GENIA ontology*. GENIA corpus is freely available and in the past decade has been used widely as a gold standard for benchmarking a variety of terminology mining methods. Table 5.1 gives a summary of the size of the corpus. Additional information about the GENIA corpus and its annotation process can be found in Kim et al. (2006).

In the GENIA ontology, terms are organised into 36 different categories such as *amino acids* (consisting of *proteins*, *peptides*, ...), *lipids*, *nucleic acids* (consisting of *DNA*, *RNA*, ...) and so on. To simplify the evaluation's reproducibility, a taxonomy of terms similar to the one suggested by Kim et al. (2004) in a shared-task for evaluating bio-entity taggers is employed. Manually annotated term mentions from the GENIA corpus[2] are thus collected to build a terminological resource, in which terms are organised according to the Kim et al.'s (2004) simplified taxonomy. To abridge the reports, unless otherwise stated, the focus is on the identification of terms belonging to the category of

[1]The first row shows the number of distinct part-of-speech tagged tokens (normalised to lowercase) while the second row shows the number of distinct tokens irrespective of their assigned part-of-speech tag and when they are normalised to lowercase (which are used in the reported experiments).

[2]Version 3.02, which can be downloaded from `http://www.nactem.ac.uk/GENIA/current/GENIA-corpus/Term/GENIAcorpus3.02.tgz`.

$T_{Mention}$	$P_{Mention}$	$T_{Distinct}$	$P_{Distinct}$	$T_{Polysemy}$	$P_{Polysemy}$
92,722	37,660	34,077	8,900	1,373	403[1]

Table 5.2: Statistics of the employed terminological resource: terms and *protein terms* are respectively abbreviated by T and P (note P ⊂ T). $T_{Polysemous}$ and $P_{Polysemous}$ show the number of distinct terms that are annotated with at least two different concept categories.

proteins—that is, the classification of *protein* and *non-protein* terms.

Table 5.2 shows the statistics for the extracted list of terms that is used as the gold standard. The reported statistics in Table 5.2 include mentions of both nested and non-nested terms. Amongst 97,876 '<cons>' mark-ups in the corpus that identify boundaries of terms, 5,154 mentions are not linked to the GENIA ontology and thus are not assigned to any concept category. From this list of 5,154 mentions with no concept category annotation, 1,440 distinct lexical units are not assigned to any concept category in the whole corpus. For instance, in

> ... are subject to tissue-specific and developmental stage-specific ...

the lexical unit *tissue-specific* is marked as a term but not assigned to any concept category. Similarly, in

> ... (SP and BP-14, 18, 19 kDs) isolated from splenic and brain cells...

splenic is marked as a term but not assigned to a category of concepts. These lexical units are removed from the list of compiled terms.

The collected mentions of terms are compiled independently of their concept category into a set of 34,077 distinct terms—that is, lexical units with identical surface structure are represented once in this set, even if they are annotated by two different concept categories. As reported in Table 5.2, only a small number of terms (i.e., 1,373) are polysemous and their mentions are annotated and classified in at least two concept categories. Amongst 8,900 terms that are classified as proteins, 403 terms are classified at least once in an additional concept category and as a result are considered polysemous (i.e., approximately 0.04% of all protein terms). For instance, in the following sentences

> ... using the murine B-cell lymphoma cell line A20, we show that ...

> ... correlate with expression of both BCL-2 and A20.

the mentions of the lexical unit *A20* are respectively annotated as a term of the concept categories *cell line* and *protein* (indicated by `G#cell_line` and `G#protein_molecule` in the GENIA corpus, respectively).

[1] In the GENIA corpus, protein terms themselves are classified into several categories such as *protein molecule*, *protein complex*, and so on. If this classification is considered, then the number of polysemous protein terms increases to 792.

	C	V	P
#Distinct Entry	58,558	19054	4,278
#Mentions	109,713	58,554	15,516
#Distinct$_{\text{Polysemous}}$	–	654	125
#Mentions$_{\text{Polysemous}}$	–	13,207	3,806

Table 5.3: Statistics of the extracted terminological resource using the YATEA system: candidate terms, valid terms and *protein terms* are respectively abbreviated by C, V, and P (note P ⊂ V ⊂ C). The statistics are also reported for Polysemous entries and their mentions in the corpus.

As stated earlier, the proposed method is built on top of an ATR system. Two methodologies are exploited for evaluations. In a set of experiments, in order to remove the effect of noise caused by the candidate term extraction process, the scope of ATR is limited only to the scoring and ranking process. Hence, it is assumed that the *noise-free* list of 34,077 terms in the GENIA corpus is known. Then, Frantzi et al.'s (2000b) c-*value* score is employed to rank these terms by the frequencies that are obtained from the GENIA corpus.[1] This set of ranked terms is denoted by $\{T\}^{\text{c-}value}_{\text{ideal}}$. A random baseline for choosing a term from the category of proteins in $\{T\}^{\text{c-}value}_{\text{ideal}}$ thus approaches to $\frac{8900}{34077} = 0.261$.

The second set of experiments embraces errors caused by the candidate term extraction process. In order to get a ranked list of terms, sentences of part-of-speech tagged, lemmatised words from the GENIA corpus are fed to the YATEA system: a state-of-the-art term extraction system (Aubin and Hamon, 2006).[2] Using part-of-speech tag sequence patterns for the extraction of candidate terms and its internal scoring mechanism, YATEA pulls out a sorted set of 59,988 candidate terms from the GENIA corpus.[3] The extracted terms are normalised by converting all their letters to lowercase; as a result, the size of the set is reduced to 58,558. This set of ranked terms is denoted by $\{T\}^{\text{YATEA}}_{\text{YATEA}}$. Amongst the set of 34,077 manually annotated terms derived as the gold standard form the GENIA corpus, 15,023 terms do not appear in $\{T\}^{\text{YATEA}}_{\text{YATEA}}$; 4,622 of these terms are from the concept category of proteins. As a result, $\{T\}^{\text{YATEA}}_{\text{YATEA}}$ contains only 4,278 terms that are once annotated as protein terms. Hence, a random baseline for choosing a term from the concept category of protein in $\{T\}^{\text{YATEA}}_{\text{YATEA}}$ approaches to $\frac{4278}{58558} = 0.073$. Table 5.3 provides a statistical summary of $\{T\}^{\text{YATEA}}_{\text{YATEA}}$. As can be inferred, errors caused by a candidate term extraction process can halve the recall in the extraction of a particular class of terms.

To measure the performance of the proposed method, two figures of merit are employed: *precision at n* ($P_{@n}$) and *non-interpolated precision at i* (NAP_i). $P_{@n}$ shows the

[1] The c-*value* score's definition is given by Equation 3.7, Chapter 3.

[2] Version 0.622, obtained from `http://search.cpan.org/~thhamon/Lingua-YaTeA/lib/Lingua/YaTeA.pm`.

[3] YATEA can be configured differently to boost its performance. For example, the part-of-speech sequence patterns for extracting candidate terms can be specified, or a set of verified terms may be provided to the system to enhance this process. However, to simplify reproducing the reported results, the system's default configuration is employed.

Denotation	Description
$\{T\}^{\text{c-}value}_{\text{ideal}}$	The set of terms extracted from the manual annotations in the GENIA corpus and sorted by the c-*value* score. This set does not contain invalid terms. The statistics for this set are reported in Table 5.2. Figure 5.5 shows the baseline performance computed in this set.
$\{T\}^{\text{YATEA}}_{\text{YATEA}}$	The set of candidate terms extracted and sorted by the YATEA system from the GENIA corpus. This set contains both valid and invalid terms. The statistics for this set are given in Table 5.3. Performance of protein term extraction in this set is reported in Figure 5.6.

Table 5.4: A summary of the resources that are employed in the experiments.

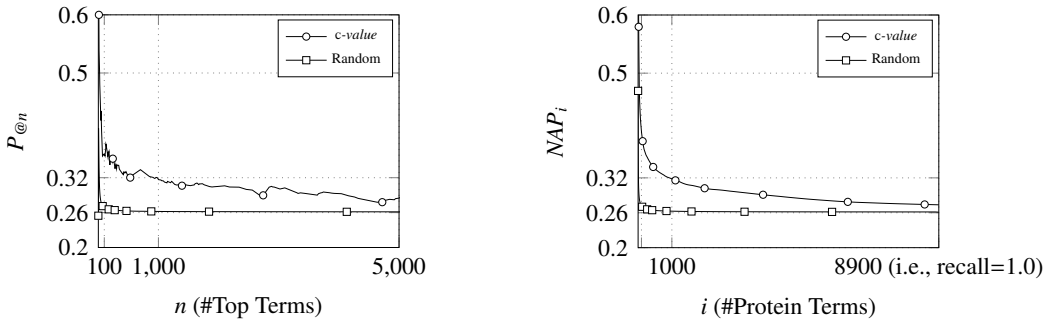


Figure 5.5: Baseline performance for protein term extraction in the $\{T\}^{\text{c-}value}_{\text{ideal}}$ ranked terms: (a) shows the proportion of protein terms in the top 5000 entries of the set of ranked candidate terms—that is, precision at n ($P_{@n}$) for $1 \leq n \leq 5000$; (b) shows the performance using non-interpolated precision at i (i.e., NAP_i) for $1 \leq i \leq 8900$; note that for $i = 8900$, recall is equal to 1.0. In both (a) and (b), a random baseline (computed by a simulation) is shown, too.

proportion of protein terms in the set of top n candidate terms that are sorted in descending order by their assigned weights (i.e., c_w given by Equation 5.1). NAP_i, however, reports the average of precision for finding the first i protein terms in a set of sorted terms (see Chapter 3, page 98). For the baseline, I report $P_{@n}$ and NAP_i that are observed in $\{T\}^{\text{c-}value}_{\text{ideal}}$ and $\{T\}^{\text{YATEA}}_{\text{YATEA}}$, which are plotted in Figures 5.5 and 5.6. Table 5.4 gives a summary of the datasets and the obtained baselines employed for the evaluation.

5.3.2 Parameters for the Configuration of the Context-Window

In the proposed methodology, once candidate terms are extracted, they are represented as vectors. The incremental method explained in Section 5.2.1 is employed to collect and represent the co-occurrences of candidate terms and words in context-windows. The co-occurrences of candidate terms and words, however, can be collected from context-

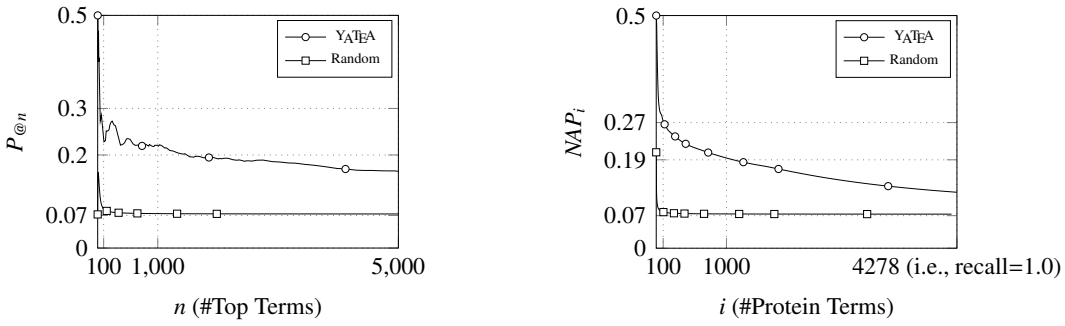


Figure 5.6: Baseline performance for protein term extraction in YATEA's extracted ranked terms (i.e., $\{T\}^{\text{YATEA}}_{\text{YATEA}}$): (a) the proportion of protein terms in the top 5000 entries of the set of ranked candidate terms—that is, $P_{@n}$ for $1 \leq n \leq 5000$; (b) non-interpolated precision (i.e., NAP_i) for $1 \leq i \leq 4278$; note that for $i = 4278$, recall is equal to 1.0. In both (a) and (b), random baselines are also shown.

windows that are configured differently.

5.3.2.1 Direction

In the proposed distributional method, context-windows can be configured differently regarding the position of the candidate terms in them and the direction in which they are stretched. Context-windows can be expanded (a) to the *left side* of a candidate term to collect the co-occurrences of the candidate term with preceding words in each sentence of the corpus, (b) to the *right* side to collect co-occurrences with the succeeding words or (c) *around* the candidate term, that is, in both left and right directions. For instance, in Figure 5.3, words that are placed in rectangles show context-windows that expand around candidate terms.

5.3.2.2 Size

The size of context-windows can also be modified—that is, the extent of the region of either side of a term for collecting and counting its co-occurrences with neighbouring words. For instance, Figure 5.3 illustrates context-windows of size $t = 3$ tokens. As stated in the literature (e.g., see Lenci, 2008; Baroni et al., 2014b), the optimum size of context-windows can only be established through experiments. However, research reports show that in contrast to wide context-windows (e.g., a paragraph or a document), narrow context-windows are more suitable to capture paradigmatic relations such as the intended concept category association in the proposed term classification task (e.g., Agirre et al., 2009; Zadeh and Handschuh, 2014c). In the performed experiments, therefore, the size of context-windows t is limited to $1 \leq t \leq 8$. The context-windows that expand around a candidate term are extended symmetrically in both directions.

5.3.2.3 Sequential order of words

Jones and Mewhort (2007) argue that the sequential order of words expresses information about lexical classes and grammatical behaviour, and therefore is important in the development of a comprehensive distributional semantic model. On the other hand, Landauer (2002) believes that 80% of the potential information in language is carried by the word choice regardless of the order in which they appear. He thus concludes that word order can be neglected in order to simplify the construction of vector spaces and their subsequent computations.[1] The influence of the inclusion of word order information on the performance of the method is investigated using a technique similar to the *permutation technique* proposed in Sahlgren et al. (2008); Recchia et al. (2015). However, instead of relying on intuition, I propose a mathematical justification based on the framework represented in Chapter 4.

5.3.2.3.1 Proposed method to capture the sequential order of words

One way to capture information about the sequential order of words in context-windows is to distinguish the appearances of words in different positions in these context-windows.[2]

This method could be best explained by giving the following example. In the first sentence of Figure 5.3, the word *technology* appears *after* the target term (i.e., *information extraction*) at the *position* $p = 1$ of the context-window. In the last sentence listed in Figure 5.3, the word *technology* also occurs, however, *before* the target term at the *position* $p = -2$. In this example, if the information about the sequential order of words is ignored, then the word *technology* is represented by only one standard basis $\vec{s}_t$ of the vector space—that is, one dimension of the model. The co-occurrence of the target term *information extraction* and the context word *technology* in these two sentences is then denoted by the coordinates $\vec{s}_t$ of the vector that represent the target term.

However, to capture information about the sequential order of words, the two appearances of the word *technology* must be distinguished and represented separately in the model. In doing so, additional dimensions must be appended to the model—one dimension per position per word. In the given example, this means that the occurrence of the word *technology* at the position $p = 1$ in context-windows must be presented by one standard basis $\vec{s}_a$ of the model, whereas the occurrence at the position $p = -2$ must be represented by another standard basis $\vec{s}_b$ of which $a \neq b$. In the same way, if the word *technology* appears at a location x other than $p = 1$ and -2 in context-windows, then it must be represented by an additional standard basis $\vec{s}_x$ of the vector space of which $x \neq a \neq b$. The co-occurrence of the target term *information extraction* and the context word

[1] Representing information about the order of words in context-windows usually entails appending additional dimensions to the underlying distributional model. Hence, computing similarities can demand more resources.

[2] Other methods are also conceivable, for instance, using n-grams, or even an additional vector space model that only captures the sequential order of words (e.g., as suggested by Jones and Mewhort, 2007).

technology at the two positions $p = 1$ and -2 is denoted by, respectively, the coordinates $\vec{s}_a$ and $\vec{s}_b$ of the vector that represents the target term *information extraction*.

If a vector space is constructed using the above-mentioned *one-dimension-per-context-element* methodology, then capturing information about the sequential order of words in context-windows drastically escalates the curse of dimensionality—as suggested by Landauer (2002), it is thus often discarded. However, as implied in Chapter 4, this problem can be easily obviated using random projections for the construction of a model. According to the principles discussed in Chapter 4 and based on the description given above, in order to capture the word order information in a vector space that is constructed using random projections, appearances of a word at different positions of context-windows are captured by assigning them to different word vectors.

Let us revisit the example given above and construct the model using random projections such as explained earlier in Section 5.2. If the word order information is ignored, then the word *technology* is assigned exactly to one word vector $\vec{w}_{\text{technology}}$, and both of its co-occurrences with the target term *information extraction* at $p = 1$ and -2 are captured by adding $\vec{w}_{\text{technology}}$ to the term vector $\vec{t}_{\text{information extraction}}$ that represents the target term at the reduced dimensionality—that is, $\vec{t}_{\text{information extraction}} = \vec{t}_{\text{information extraction}} + \vec{w}_{\text{technology}} + \vec{w}_{\text{technology}}$.

But, in order to model the sequential order of words, the appearances of the word *technology* at different positions in context-windows must be distinguished by assigning them to different word vectors. In the example above, the appearance of the word *technology* at $p = 1$ is captured by vector$\vec{w}^{p=1}_{\text{technology}}$ and its appearance at $p = -2$ is captured by $\vec{w}^{p=-2}_{\text{technology}}$ of which $\vec{w}^{p=-2}_{\text{technology}} \neq \vec{w}^{p=1}_{\text{technology}}$. The co-occurrences of the word *technology* and the term *information extraction* are then captured by accumulating these two different vectors to $\vec{t}_{\text{information extraction}}$—that is, $\vec{t}_{\text{information extraction}} = \vec{t}_{\text{information extraction}} + \vec{w}^{p=1}_{\text{technology}} + \vec{w}^{p=-2}_{\text{technology}}$. Both vectors $\vec{w}^{p=1}_{\text{technology}}$ and $\vec{w}^{p=-2}_{\text{technology}}$ are required to be stored for later usages—for example, to capture the co-occurrence of the word *technology* at $p = 1$ with another candidate term.

The method suggested above, however, is hampered by its required space for the storage and retrieval of word vectors. Instead of creating several word vectors for representing appearances of one word at different positions of context-windows and storing them separately for later usages, one can use the *permutation technique*.

The main idea is that shuffling randomly created word vectors creates new random vectors that can be used to represent context words at various positions in context-windows. For example, this shuffling can be defined using a permutation function. This permutation function is defined using the location of context words in context-windows. In my implementation, a circular shift function serves as the permutation function (as suggested in Sahlgren et al., 2008, too). If p is the number of tokens before or after a candidate term and a word in a context-window (i.e., the position of the word in context-windows), then the word vector $\vec{w}$ that represents the word is shifted p times circularly to

left or right prior to adding it to the candidate term's term vector. This circular shift of $\vec{w}$ results in a new random vector without the need to generate and store a new one. In this way, while the word order information is captured, the storage of additional word vectors is avoided. Hence, the method's space complexity enhances.[1]

5.3.3 Classification Parameters

In addition to various configurations of context-windows, the performance of the proposed term classification method is affected by the k-nn framework's parameters: (a) neighbourhood size selection (i.e., the value of k), (b) the size of the set of reference vectors (denoted by $|R_s|$)—that is, the number of training instances employed for the classification—and (c) the choice of similarity metric.

5.3.3.1 Neighbourhood size selection

The performance of k-nn is largely dependent on the value of k—that is, the neighbourhood size selection in the classification process. Using Bayesian mathematics, it is verified that if an infinite number of training samples are available (i.e., $|R_s| \rightarrow \infty$), then using a large value of k will result to the best-performing classification model (see Hastie et al., 2009, chap. 13). In the absence of a large R_s, in the employed memory-based learning framework, a small value for k can lead to *over-fitting* and sensitivity to noise, while a large neighbourhood estimation can reduce the discriminatory power of the classifier.

For a fixed R_s, if the underlying probability distribution of the term vectors in the vector space was known, the optimum k could be calculated. However, the underlying probability distribution is unknown and difficult to estimate. Therefore, the optimal value of k is usually obtained through experiments. To study the effect of neighbourhood size selection on the method's output, the performance is reported when k is set to different values. For instance, one can be interested in investigating whether the choice of k affects the choice for the best-performing context-windows configuration—that is to say, can one choose the most discriminative context-windows irrespective of the value of k?

5.3.3.2 Similarity metrics

Last but not least, the choice of the method for similarity measurement between vectors—that is, $s(v, r_i)$ in Equation 5.1, thus its underlying metric—is another important factor that influences the method's performance. For instance, in a classification task similar to the proposed method, Weeds et al. (2005) suggest that the city block distance outperforms other similarity metrics such as the cosine measure. Therefore, the performance of classifiers that exploit different similarity and distance measures are reported. As implied in

[1] From an alternative perspective beyond the scope of this thesis, the suggested method is known as a *derandomisation* technique.

Section 5.2, the method's performance is assessed when using the Euclidean distance, the city block distance and the cosine similarity.

5.3.4 Setting the Parameters of Random Projection

For the experiments that are carried over the $\{\mathrm{T}\}^{\text{c-}value}_{\text{ideal}}$ and $\{\mathrm{T}\}^{\mathrm{YATEA}}_{\mathrm{YATEA}}$, vector spaces are constructed at the reduced dimensionality of $m = 2000$. Considering the small number of candidate terms in these datasets—that is, 34,077 and 59,988, respectively for $\{\mathrm{T}\}^{\text{c-}value}_{\text{ideal}}$ and $\{\mathrm{T}\}^{\mathrm{YATEA}}_{\mathrm{YATEA}}$—and based on the justification provided in Chapter 4, it can be verified that the dimensionality $m = 2000$ is large enough to construct models that preserve the relative pairwise distances between vectors in the original high-dimensional spaces.

For the construction of the ℓ_2-normed vector spaces at the reduced dimensionality of $m = 2000$, word vectors with 8 non-zero elements are employed. This means that in the reported experiments, the value of α from Equation 5.2 is set to 250. For the ℓ_1-normed vector space construction, however, word vectors with 40 non-zero elements are employed–that is, $\alpha = 50$. Considering the proposed approach for collecting co-occurrence frequencies, the original dimensionality of a vector space constructed in the employed datasets (shown by n) is a product of the size of the vocabulary in the GENIA corpus (i.e., 19,576, as reported in Table 5.1). Moreover, a model constructed at the original dimension is extremely sparse: depending on the configuration of context-windows, the sparseness of vectors (shown by β) in the reported experiments is $\beta < 10^{-3}$. Therefore, the suggested values for α are conservative choices that meet the criteria for the number of non-zero elements—that is, $\mathrm{O}(n)$ and $\mathrm{O}(\beta n)$ for the ℓ_2 and ℓ_1-normed spaces, respectively.

Considering the number of candidate terms that are represented in the constructed m-dimensional models and the original dimensionality of them (i.e., n), setting $m > 2000$ or using more non-zero elements in word vectors would not affect the obtained performances.

5.3.5 Evaluation Methodology

To find the best performing models, an exhaustive search is performed over the Cartesian product of a set of values for the parameters of (a) context-widows configuration (Section 5.3.2), and (b) the k-nn classification framework (Section 5.3.3).

In the reported empirical results, the proposed methodology in Section 5.2.1 is performed to construct several vector spaces from each combination of values that must be set for the configurations of context-windows. These vector spaces are constructed for the list of candidate terms in $\{\mathrm{T}\}^{\text{c-}value}_{\text{ideal}}$ and $\{\mathrm{T}\}^{\mathrm{YATEA}}_{\mathrm{YATEA}}$ (see Table 5.4), in which the co-occurrence frequencies are collected from the GENIA corpus. Besides normalising text to lower-case letters and a *Penn Treebank* tokenisation, no other text preprocessing is performed. To summarise, context-windows are configured for:

- three directions: left, right, and around candidate terms;
- the size of context-windows is limited to t tokens, for $1 \leq t \leq 8$;
- inclusion and exclusion of information about sequential order of words.

Therefore, for each dataset 48 different vector spaces are constructed to encompass all the combinations of the values stated above.

The described term classification methodology in Section 5.2 is then employed to assign scores to the candidate terms in all the constructed vector spaces. The scoring procedure is also repeated for the combination of a set of values which can be set differently in the classification:

- three values for the neighbourhood size selection, that is $k = 1, 7, 25$;
- three similarity measures: cosine, the Euclidean, and the city block distance.

The top $n = 100$ entries from the list of ranked candidate terms in $\{\mathrm{T}\}^{c\text{-}value}_{\text{ideal}}$ and $\{\mathrm{T}\}^{\text{YATEA}}_{\text{YATEA}}$ are chosen to form the R_s. Hence, for each vector space, the scoring procedure is repeated 9 times in order to obtain 9 sets of ranked terms. The observed NAP_i in the obtained sets is then employed for their comparison and choosing the best combination of the evaluated parameters of the method.

5.4 Empirical Evaluations

5.4.1 Evaluation in $\{\mathrm{T}\}^{c\text{-}value}_{\text{ideal}}$: The Point of Departure

The first series of experiments are carried over $\{\mathrm{T}\}^{c\text{-}value}_{\text{ideal}}$ when the classification process is performed using a set of reference terms R_s of size 100 (i.e., $|R_s|$=100). In this experiment, R_s comprises of the top 100 entries from the ranked set of terms in $\{\mathrm{T}\}^{c\text{-}value}_{\text{ideal}}$ of which 36 entries are positive examples (i.e., protein terms). The experiments are duplicated for all of the context-window's configurations and the classification's parameters as explained in Section 5.3.5.

Tables 5.5, 5.6, and 5.7 report the results in detail when similarities are calculated using the cosine measure, the Euclidean distance, and the city block distance, respectively. The method's performance is denoted by the non-interpolated precision (i.e., NAP_i) for the identification of protein terms at $i = 200$ and $i = 8900$; note that $NAP_{i=200}$ and $NAP_{i=8900}$ denotes the method's performance when recall is 0.02 and 1.0, respectively. In these tables, the observed NAP_i over the list of sorted terms in $\{\mathrm{T}\}^{c\text{-}value}_{\text{ideal}}$ is reported as a baseline (see Figure 5.5). Figures 5.7 and 5.8 summarise and plot the reported numbers in these tables.

First and foremost, a glance at Figures 5.7a and 5.7b, and Figures 5.8a and 5.8b, indicates that choosing the best performing configuration for the method's parameters

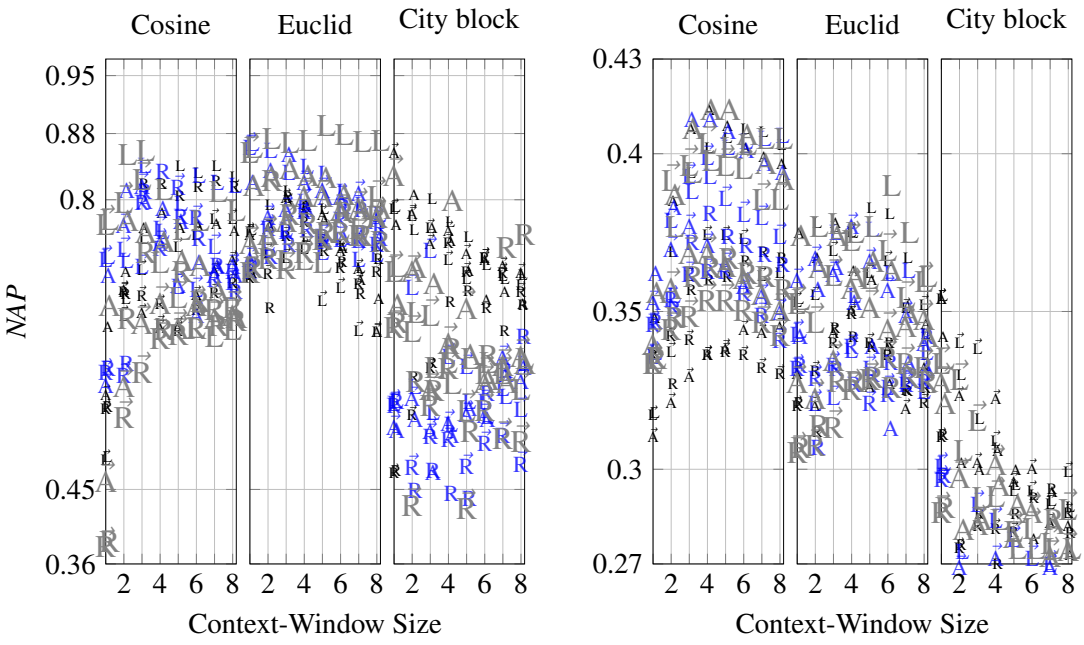
Context		$NAP_{i=200}$			$NAP_{i=8900}$		
dir	size	k			k		
		1	7	25	1	5	25
Around	1	0.647	0.709	0.673	0.348	0.362	0.356
	2	0.713	0.811	0.775	0.385	0.383	0.387
	3	0.766	0.798	0.761	0.408	0.411	0.408
	4	0.784	0.79	0.75	0.414	0.411	0.414
	5	0.766	0.774	0.727	0.409	0.407	0.414
	6	0.769	0.725	0.721	0.402	0.401	0.406
	7	0.771	0.72	0.691	0.399	0.397	0.397
	8	0.763	0.709	0.658	0.397	0.394	0.392
Left	1	0.489	0.732	0.776	0.318	0.337	0.335
	2	0.68	0.727	0.855	0.369	0.355	0.377
	3	0.753	0.842	0.86	0.394	0.388	0.394
	4	0.82	0.762	0.813	0.402	0.393	0.398
	5	0.841	0.796	0.764	0.405	0.402	0.402
	6	0.833	0.83	0.775	0.408	0.404	0.403
	7	0.841	0.821	0.81	0.407	0.405	0.405
	8	0.828	0.817	0.787	0.403	0.406	0.405
Right	1	0.55	0.596	0.387	0.337	0.347	0.335
	2	0.686	0.602	0.657	0.327	0.351	0.348
	3	0.819	0.81	0.736	0.357	0.372	0.361
	4	0.82	0.834	0.643	0.36	0.381	0.364
	5	0.805	0.78	0.714	0.361	0.374	0.367
	6	0.816	0.753	0.677	0.366	0.371	0.362
	7	0.823	0.721	0.648	0.369	0.368	0.361
	8	0.816	0.69	0.654	0.367	0.365	0.357
Baseline		**0.364**			**0.273**		

(a) Sequential Order of Words Discarded

$NAP_{i=200}$			$NAP_{i=8900}$		
k			k		
1	7	25	1	5	25
0.568	0.58	0.466	0.311	0.354	0.341
0.694	0.773	0.58	0.322	0.371	0.359
0.666	0.821	0.657	0.33	0.378	0.37
0.648	0.752	0.681	0.337	0.372	0.37
0.65	0.701	0.65	0.339	0.366	0.366
0.688	0.665	0.678	0.345	0.359	0.362
0.717	0.713	0.663	0.344	0.356	0.357
0.73	0.724	0.666	0.346	0.351	0.352
0.489	0.732	0.776	0.318	0.337	0.335
0.663	0.744	0.803	0.338	0.38	0.393
0.719	0.724	0.831	0.362	0.394	0.398
0.653	0.77	0.73	0.375	0.4	0.405
0.746	0.7	0.688	0.375	0.387	0.398
0.761	0.696	0.652	0.374	0.382	0.397
0.78	0.748	0.641	0.362	0.381	0.387
0.78	0.705	0.649	0.363	0.375	0.387
0.55	0.596	0.387	0.337	0.347	0.335
0.692	0.592	0.543	0.343	0.355	0.346
0.687	0.801	0.598	0.342	0.363	0.35
0.677	0.746	0.638	0.337	0.364	0.355
0.645	0.819	0.643	0.338	0.364	0.355
0.67	0.783	0.651	0.337	0.357	0.352
0.691	0.708	0.673	0.333	0.35	0.348
0.703	0.718	0.676	0.331	0.342	0.344
0.364			**0.273**		

(b) Sequential Order of Words Encoded

Table 5.5: The performances observed over the $\{T\}^{\text{c-}value}_{\text{ideal}}$ when $|R_s| = 100$ and similarities are computed using the *cosine* between vectors. The performance is shown with regards to the observed NAP_i, for $i = 200$ (i.e., recall = 0.02) and $i = 8900$ (i.e., recall = 1.0). The baseline shows the computed *NAP* when terms are sorted using the c-*value* (see Figure 5.5); (a) denotes the performance of models that ignore the sequential order of words in context-windows, whereas (b) shows the performance when this information is encoded.



(a) $\boldsymbol{NAP_{i=200}}$ (b) $\boldsymbol{NAP_{i=8900}}$

Figure 5.7: The results obtained over $\{\mathrm{T}\}^{c\text{-}value}_{\text{ideal}}$ when $|R_s| = 100$: The y-axis shows the observed performances (i.e., NAP_i) for the identification of protein terms when the method's parameters are set differently. Each box in the sub-figures denotes the performance for each of the employed similarity metric. In these boxes, the x-axis shows the size of context-windows. The letters A, L, and R denote the direction in which context-windows are stretched (i.e., respectively, Around, Left, or the Right side of the candidate terms). Models that encode word order information are denoted using the $\vec{\square}$ on top of the letters. The size of letters, however, shows the value of k. The smallest size denotes $k = 1$ (black colour), while the largest size denotes $k = 25$ (grey colour); the medium size represents $k = 7$ (blue colour). In these plots, the minimum value of y-axis shows the baseline.

is subject to the chosen i for computing and reporting NAP_i as the performance measure. Besides this observation, these figures suggest that obtaining the best performance is drastically dependant on the choice of similarity metrics (e.g., by comparing the average of the obtained performances over parameters of the method). If the performance is measured using NAP_i at a small i such as 200 in this experiment—that is, if the intention is the extraction of a small number of terms and hence precision is more important than recall—then the Euclidean distance seems to be a more desirable choice than the city block distance or the cosine measure. However, for NAP_i for a large i, for example $i = 8900$ (i.e., if a high recall is intended), then the cosine similarity seems to be a more robust choice than the other evaluated similarity metrics.

When the method's performance is evaluated using $NAP_{i=200}$, then the Euclidean

Context		$NAP_{i=200}$			$NAP_{i=8900}$		
dir	size	k			k		
		1	7	25	1	5	25
Around	1	0.712	0.821	0.817	0.375	0.342	0.374
	2	0.769	0.836	0.828	0.367	0.357	0.364
	3	0.799	0.855	0.838	0.378	0.364	0.374
	4	0.794	0.788	0.829	0.363	0.353	0.361
	5	0.776	0.803	0.827	0.36	0.339	0.366
	6	0.742	0.755	0.806	0.327	0.313	0.363
	7	0.704	0.756	0.788	0.352	0.334	0.349
	8	0.68	0.753	0.79	0.346	0.332	0.342
Left	1	0.764	0.868	0.863	0.354	0.344	0.354
	2	0.794	0.86	0.874	0.378	0.365	0.377
	3	0.811	0.833	0.869	0.377	0.375	0.378
	4	0.756	0.84	0.872	0.375	0.355	0.375
	5	0.789	0.832	0.89	0.383	0.376	0.377
	6	0.744	0.81	0.879	0.381	0.362	0.39
	7	0.718	0.794	0.871	0.367	0.363	0.375
	8	0.745	0.783	0.872	0.343	0.337	0.361
Right	1	0.709	0.718	0.718	0.321	0.333	0.307
	2	0.67	0.732	0.823	0.312	0.307	0.314
	3	0.804	0.746	0.782	0.345	0.332	0.328
	4	0.792	0.764	0.772	0.342	0.327	0.328
	5	0.738	0.75	0.761	0.339	0.33	0.329
	6	0.725	0.792	0.785	0.34	0.334	0.326
	7	0.687	0.788	0.757	0.325	0.329	0.329
	8	0.729	0.767	0.75	0.326	0.342	0.328
Baseline		0.364			0.273		

(a) Sequential Order of Words Discarded

$NAP_{i=200}$			$NAP_{i=8900}$		
k			k		
1	7	25	1	5	25
0.761	0.732	0.753	0.331	0.361	0.329
0.768	0.783	0.764	0.321	0.368	0.324
0.805	0.793	0.78	0.344	0.363	0.336
0.789	0.825	0.81	0.351	0.378	0.359
0.785	0.816	0.81	0.327	0.367	0.353
0.741	0.807	0.793	0.339	0.358	0.346
0.72	0.812	0.796	0.32	0.35	0.347
0.643	0.778	0.785	0.333	0.341	0.349
0.764	0.868	0.863	0.354	0.344	0.354
0.739	0.752	0.751	0.356	0.322	0.348
0.733	0.769	0.733	0.37	0.324	0.363
0.744	0.802	0.734	0.376	0.339	0.374
0.681	0.741	0.73	0.363	0.333	0.362
0.697	0.757	0.77	0.367	0.327	0.371
0.645	0.733	0.771	0.352	0.329	0.356
0.644	0.738	0.81	0.353	0.326	0.361
0.709	0.718	0.718	0.321	0.333	0.307
0.714	0.77	0.735	0.332	0.33	0.309
0.76	0.766	0.723	0.341	0.334	0.315
0.777	0.79	0.742	0.35	0.34	0.327
0.752	0.76	0.758	0.341	0.322	0.33
0.721	0.781	0.767	0.337	0.325	0.337
0.702	0.756	0.763	0.326	0.328	0.335
0.716	0.751	0.786	0.322	0.329	0.334
0.364			0.273		

(b) Sequential Order of Words Encoded

Table 5.6: The results observed in $\{\mathrm{T}\}^{\text{c-}value}_{\text{ideal}}$ when $|R_s| = 100$ and similarities are computed using the *Euclidean* distance. The presentation format is similar to Table 5.5.

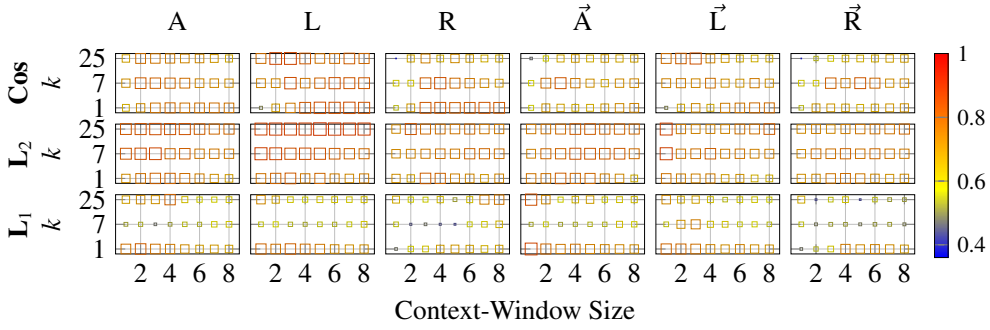
Context		$NAP_{i=200}$			$NAP_{i=8900}$		
dir	size	k			k		
		1	7	25	1	5	25
Around	1	0.77	0.521	0.674	0.342	0.259	0.321
	2	0.806	0.535	0.718	0.302	0.259	0.281
	3	0.766	0.471	0.699	0.3	0.253	0.284
	4	0.751	0.516	0.799	0.306	0.262	0.295
	5	0.714	0.548	0.613	0.3	0.262	0.289
	6	0.679	0.535	0.583	0.278	0.263	0.269
	7	0.689	0.589	0.58	0.293	0.264	0.283
	8	0.675	0.598	0.602	0.273	0.261	0.266
Left	1	0.792	0.558	0.72	0.355	0.302	0.335
	2	0.764	0.575	0.664	0.33	0.274	0.306
	3	0.801	0.543	0.597	0.316	0.264	0.291
	4	0.773	0.529	0.578	0.307	0.269	0.284
	5	0.755	0.556	0.608	0.295	0.266	0.275
	6	0.721	0.538	0.603	0.292	0.266	0.275
	7	0.705	0.563	0.594	0.29	0.265	0.275
	8	0.709	0.602	0.608	0.288	0.267	0.275
Right	1	0.473	0.556	0.652	0.311	0.298	0.288
	2	0.586	0.449	0.581	0.276	0.243	0.25
	3	0.603	0.473	0.57	0.286	0.251	0.257
	4	0.677	0.446	0.621	0.27	0.242	0.25
	5	0.69	0.439	0.52	0.286	0.264	0.249
	6	0.67	0.566	0.612	0.291	0.265	0.267
	7	0.642	0.594	0.743	0.294	0.257	0.284
	8	0.673	0.635	0.758	0.28	0.256	0.267
Baseline		0.364			0.273		

(a) Sequential Order of Words Discarded

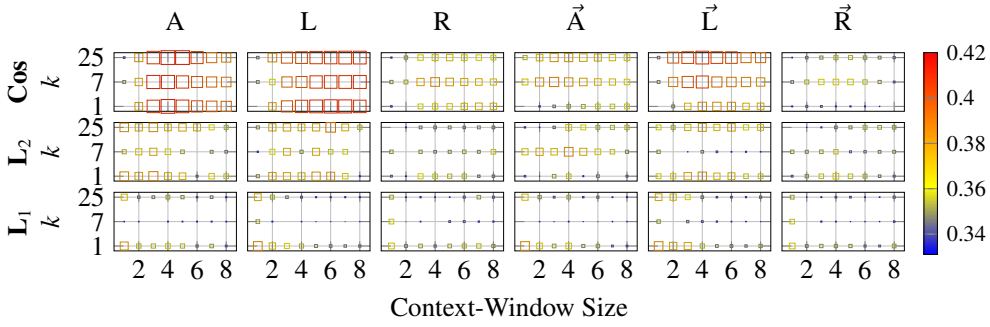
$NAP_{i=200}$			$NAP_{i=8900}$		
k			k		
1	7	25	1	5	25
0.86	0.529	0.836	0.355	0.265	0.329
0.786	0.564	0.719	0.324	0.27	0.299
0.75	0.525	0.577	0.303	0.267	0.284
0.766	0.534	0.633	0.323	0.273	0.303
0.738	0.545	0.675	0.297	0.267	0.28
0.73	0.544	0.585	0.301	0.266	0.288
0.723	0.614	0.606	0.284	0.27	0.274
0.715	0.585	0.617	0.283	0.269	0.276
0.792	0.558	0.72	0.355	0.302	0.335
0.784	0.699	0.647	0.341	0.266	0.33
0.769	0.743	0.666	0.339	0.29	0.317
0.73	0.615	0.609	0.31	0.285	0.293
0.708	0.603	0.552	0.294	0.279	0.282
0.733	0.561	0.596	0.294	0.273	0.283
0.711	0.576	0.596	0.29	0.271	0.278
0.7	0.551	0.577	0.3	0.268	0.289
0.473	0.556	0.652	0.311	0.298	0.288
0.543	0.481	0.437	0.276	0.243	0.247
0.6	0.519	0.558	0.283	0.257	0.254
0.751	0.515	0.602	0.282	0.252	0.263
0.716	0.482	0.433	0.281	0.246	0.263
0.733	0.51	0.551	0.268	0.247	0.251
0.715	0.52	0.518	0.265	0.247	0.251
0.692	0.485	0.514	0.287	0.261	0.269
0.364			0.273		

(b) Sequential Order of Words Encoded

Table 5.7: The performances observed in $\{\mathrm{T}\}^{\text{c-}value}_{\text{ideal}}$ when $|R_s| = 100$ and similarities are computed using the *city block* distance. The presentation format is similar to Table 5.5.



(a) $\boldsymbol{NAP_{i=200}}$

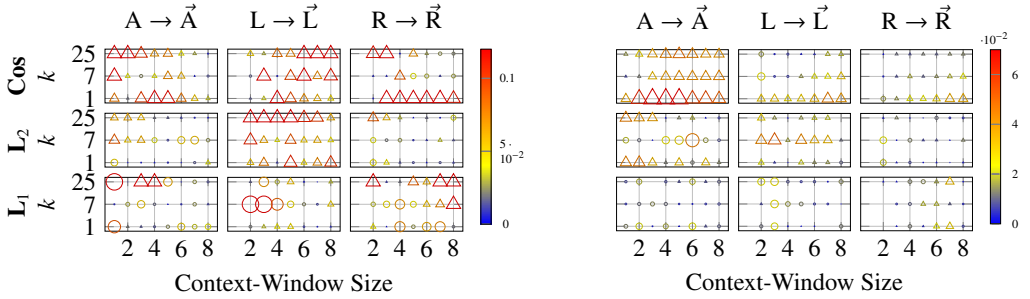


(b) $\boldsymbol{NAP_{i=8900}}$

Figure 5.8: The results obtained over $\{\mathrm{T}\}^{c\text{-}value}_{\text{ideal}}$ when $|R_s| = 100$: shown are the obtained results grouped by the type of similarity metrics and values of k (i.e., the rows), as well as the direction and the size of context-windows (i.e., the columns). The letters A, L, and R denote the direction in which context-windows are stretched (i.e., respectively, Around, Left, or the Right side of the candidate terms). Models that encode word order information are denoted using the $\vec{\square}$ on top of the letters. The size of squares in these plots denote the value for NAP_i at $i = 200$ (i.e., Figure 5.8a), and $i = 8900$ (i.e, Figure 5.8b).

distance seems to be the least sensitive similarity metric to the changes in the values of the remaining method's parameters—that is, it shows the least variance in the performance. Although this behaviour of the Euclidean distance changes when the performance is measured using $NAP_{i=8900}$. When it comes to choosing a value for the neighbourhood size in the classification process (i.e., k), the city block distance is the most sensitive measure. Except $k = 1$, the city block distance does not show an acceptable performance in these experiments.

In these experiments, context-windows that extend to the left side or around the terms usually outperform context-windows that only extend to the right of candidate terms. If the obtained performances are averaged independently of the value of k, the employed similarity metric, and the size of context-windows, then context-windows that extend around terms are preferable to those that only extend to the left side of candidate



(a) $NAP_{i=200}$ (b) $NAP_{i=8900}$

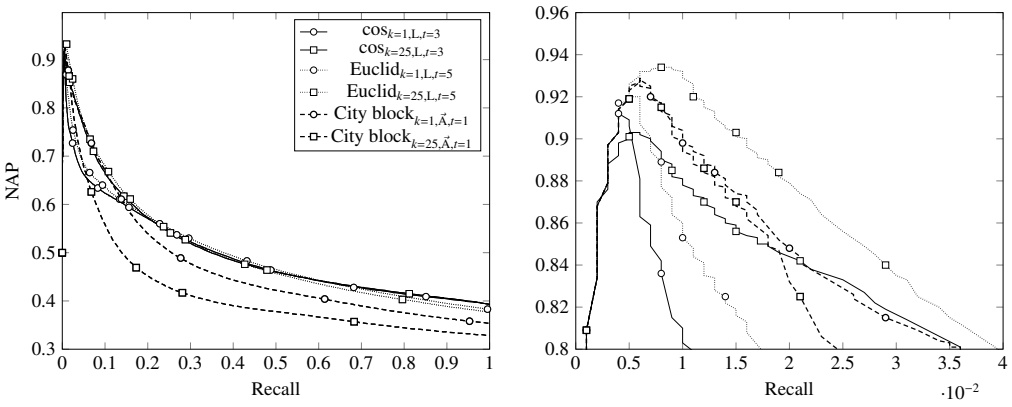
Figure 5.9: Observed differences in the performance of method when the sequential order of words are encoded in the context-windows. The letters A, L, and R denote the three directions of Around, Left, or Right side of the candidate terms, respectively. Accordingly, the notation X $\rightarrow \vec{\text{X}}$ shows the difference in the performance of the models constructed by collecting co-occurrences at the direction X before and after encoding information about the sequential order of words. An increase in the performance is marked by a circle, whereas a reduction is shown by a triangle. The size of shapes shows the intensity.

terms. As can be inferred from the reported results, encoding information about the sequential order of words in context-windows does not necessarily enhance the observed result. The effect of encoding this information in the observed performances is shown in detail in Figure 5.9. As shown in the figure, except when using the city block distance for measuring similarities, encoding word order information has a negative impact on the performance.

Choosing the best size for context-windows is also largely dependant on the chosen similarity metric and the direction in which context-windows are extended. However, according to the obtained results, context-windows of size $3 \leq t \leq 6$ tokens are often amongst the top performers. As shown in Figure 5.7, the city block distance is again an exception in which a small size for context-windows—that is, $1 \leq t \leq 2$ tokens—outperforms larger sizes of context-windows. Although the cosine measure on average results in a higher performance (particularly at 100% recall), using a distance metric is preferable to the use of cosine similarity if a small recall is targeted (see Figure 5.10).

5.4.1.1 Using an entity tagger as an additional baseline

To have a better understanding of the reported performance measures, an additional baseline is introduced. The same set of annotated candidate terms (i.e., R_s) used in the experiment reported in Section 5.4.1 is employed to train a biomedical named entity recogniser. Namely, the *ABNER* system, an entity tagger based on conditional random fields, is employed. ABNER exploits a variety of orthographic and contextual features designed for the analysis of biology text (Settles, 2005). If ABNER is trained using all the manual



(a) (b)

Figure 5.10: The performance of similarity metrics across the complete range of recall values: (a) shows NAP_i for $0 < i \leq 8900$ (i.e., the complete range of values for recall) for the three employed similarity metrics for particular configurations of context-windows. For recall less than 2%, as shown with minute details in (b), the choice of a distance measure results in a better performance than the use of cosine.

annotation that are provided for the mentions of terms in the GENIA corpus, it achieves a recall of 77.8 and precision of 68.1 for extracting protein terms. Consequently, it is one of the top-performing bio-entity recognition systems for extracting protein terms (see Kim et al., 2004, for the performance comparison of ABNER and several other entity taggers in a shared task).

To compute the second baseline, ABNER is trained using the mentions of the protein term that appear in R_s. To ensure that the provided training dataset for the development of the tagger's model is noise-free, manual annotations in the GENIA corpus are used to mark each of the term mentions. The 36 protein terms in R_s are mentioned 1,321 times in the GENIA corpus. Therefore, it is worthwhile mentioning that although the entity tagger is developed by the same number of distinct terms as appeared in R_s, in practice, preparing the training dataset for the development of the entity tagger's model requires more manual effort than preparing R_s.

After training the ABNER tagger, the obtained model is reapplied to the same corpus (i.e., GENIA) in order to extract additional mentions of protein terms. The extracted terms using ABNER are collected in a set and the number of distinct valid and invalid protein terms are reported as the baseline. The resulting model can only identify an additional 16 protein terms out of the remaining 8,864 terms in the corpus. Simply put, and as suggested in the introduction of this chapter, the 1,321 mentions of the 36 protein terms in R_s are not sufficient for the training of the ABNER system and extracting additional terms from the concept category of proteins.

5.4.2 Evaluation of the Method in $\{T\}^{Y_A T_E A}_{Y_A T_E A}$: The Method's Performance in the Presence of Noise

In this section, the observed results are reported when the method is applied to $\{T\}^{Y_A T_E A}_{Y_A T_E A}$. As stated earlier, 67% of candidate terms in $\{T\}^{Y_A T_E A}_{Y_A T_E A}$ are *invalid* terms (see Table 5.3). Hence, the performed experiments let us study the effect of noise caused by the process of candidate term extraction in the method's performance. Therefore, the evaluation process described in the previous section is repeated over the ranked set of candidate terms in $\{T\}^{Y_A T_E A}_{Y_A T_E A}$ using $|R_s| = 100$ (similar to the previous experiment). As a result, R_s contains 22 protein terms (i.e., positive examples). Tables 5.8, 5.9, and 5.10 report the performance using NAP_i at $i = 98$ (i.e., recall =0.02) and $i = 4278$ (i.e., recall =1) when similarities are calculated using the cosine measure, the Euclidean distance, and the city block distance, respectively. Apart from the method's performance for the identification of protein terms, these tables report NAP_i when the goal is to extract valid terms.

In this set of experiments, the cosine similarity outperforms both the Euclidean and the city block distance with a large margin. Although the method shows an acceptable performance for a small recall such as 0.02% (i.e., $NAP_{i=98}$ as shown in Figure 5.11a), its performance drastically drops for a complete recall (i.e., $NAP_{i=4278}$). If the goal is to extract all the protein terms in the corpus, the method underperforms the ATR's ranking baseline and it shows a performance similar to the random baseline (Figure 5.11b). As reported in the tables, the obtained scores for sorting candidate terms using the proposed method also decreases the number of valid terms in the obtained ranked sets of terms.

The major sources of errors in this set of experiments are invalid candidate terms in which a valid protein term appears nested. For example, in the performed experiments, *I kappa B* is a protein term, which often appears at the top of the obtained ranked lists. In addition to *I kappa B*, $\{T\}^{Y_A T_E A}_{Y_A T_E A}$ contains candidate terms such as *control of I kappa B-alpha proteolysis*, *inhibitor I kappa B*, *endogenous I kappa B*, and so on, of which many are invalid terms.[1] Since these candidate terms share a similar context with valid protein terms, they also appear at the top of the obtained ranked sets. As a result, their presence in the list of candidate terms deteriorate the method's performance. As suggested previously, combining a unithood measure with the score generated by the classification method can help to alleviate these errors, particularly for large recall values.

Independently of the direction in which context-windows are extended, for the nearest neighbour (i.e., $k = 1$), a positive correlation between the size of context-window and the performance of the method (both for detecting protein terms and valid candidate terms) is observable. However, when $k = 25$ and the context-windows are larger than

[1] For instance, in $\{T\}^{Y_A T_E A}_{Y_A T_E A}$, *I kappa B* appears nested in 221 terms.

Context		$NAP_{i=98}$			$NAP_{i=4278}$		
dir	size	k			k		
		1	7	25	1	7	25
Around	1	0.55(0.77)	0.31(0.81)	0.22(0.69)	0.11(0.36)	0.13(0.42)	0.14(0.42)
	2	0.62(0.79)	0.49(0.8)	0.47(0.75)	0.13(0.36)	0.15(0.41)	0.16(0.42)
	3	0.68(0.85)	0.47(0.73)	0.66(0.77)	0.14(0.37)	0.15(0.39)	0.17(0.4)
	4	0.68(0.85)	0.39(0.69)	0.53(0.73)	0.14(0.37)	0.15(0.38)	0.16(0.39)
	5	0.69(0.86)	0.48(0.8)	0.54(0.8)	0.15(0.37)	0.15(0.38)	0.16(0.38)
	6	0.68(0.88)	0.37(0.71)	0.41(0.61)	0.14(0.37)	0.14(0.37)	0.15(0.37)
	7	0.71(0.87)	0.41(0.71)	0.39(0.59)	0.14(0.37)	0.14(0.37)	0.14(0.37)
	8	0.71(0.87)	0.37(0.7)	0.37(0.57)	0.14(0.37)	0.13(0.36)	0.14(0.37)
Left	1	0.36(0.5)	0.24(0.67)	0.41(0.8)	0.1(0.39)	0.12(0.43)	0.13(0.46)
	2	0.6(0.78)	0.39(0.73)	0.46(0.8)	0.13(0.4)	0.14(0.43)	0.16(0.45)
	3	0.69(0.85)	0.49(0.71)	0.48(0.77)	0.13(0.38)	0.15(0.42)	0.16(0.43)
	4	0.72(0.85)	0.49(0.71)	0.56(0.79)	0.13(0.37)	0.15(0.41)	0.17(0.43)
	5	0.74(0.86)	0.52(0.72)	0.53(0.8)	0.14(0.37)	0.15(0.4)	0.17(0.42)
	6	0.76(0.89)	0.47(0.63)	0.47(0.69)	0.14(0.37)	0.15(0.4)	0.16(0.41)
	7	0.78(0.91)	0.41(0.62)	0.44(0.71)	0.14(0.37)	0.14(0.39)	0.16(0.41)
	8	0.78(0.91)	0.41(0.63)	0.43(0.67)	0.15(0.38)	0.14(0.39)	0.15(0.4)
Right	1	0.36(0.41)	0.22(0.41)	0.16(0.37)	0.09(0.31)	0.1(0.33)	0.09(0.33)
	2	0.42(0.43)	0.28(0.41)	0.41(0.58)	0.09(0.32)	0.1(0.32)	0.1(0.33)
	3	0.45(0.46)	0.33(0.52)	0.38(0.59)	0.09(0.32)	0.1(0.33)	0.11(0.33)
	4	0.49(0.5)	0.37(0.58)	0.53(0.69)	0.1(0.33)	0.11(0.33)	0.12(0.34)
	5	0.55(0.6)	0.35(0.57)	0.43(0.71)	0.1(0.33)	0.11(0.33)	0.12(0.35)
	6	0.55(0.68)	0.28(0.54)	0.37(0.63)	0.1(0.34)	0.1(0.33)	0.11(0.34)
	7	0.57(0.69)	0.29(0.51)	0.34(0.58)	0.11(0.34)	0.11(0.33)	0.11(0.34)
	8	0.59(0.74)	0.32(0.56)	0.3(0.54)	0.11(0.34)	0.11(0.34)	0.11(0.34)
Baseline		**0.273(0.87)**			**0.12(0.5)**		

(a) Sequential Order of Words Discarded

	$NAP_{i=98}$			$NAP_{i=4278}$		
	k			k		
	1	7	25	1	7	25
Around	0.42(0.56)	0.25(0.52)	0.28(0.56)	0.09(0.34)	0.11(0.37)	0.11(0.36)
	0.47(0.6)	0.33(0.55)	0.29(0.46)	0.09(0.34)	0.11(0.37)	0.12(0.36)
	0.43(0.58)	0.29(0.49)	0.38(0.53)	0.09(0.33)	0.11(0.35)	0.12(0.34)
	0.45(0.6)	0.3(0.51)	0.33(0.5)	0.09(0.33)	0.1(0.34)	0.11(0.34)
	0.46(0.63)	0.24(0.47)	0.24(0.41)	0.09(0.33)	0.1(0.33)	0.11(0.33)
	0.48(0.64)	0.27(0.45)	0.32(0.45)	0.09(0.34)	0.1(0.34)	0.11(0.34)
	0.5(0.66)	0.2(0.38)	0.3(0.43)	0.09(0.33)	0.1(0.33)	0.1(0.33)
	0.53(0.65)	0.19(0.37)	0.29(0.44)	0.09(0.33)	0.09(0.33)	0.1(0.33)
Left	0.36(0.5)	0.24(0.67)	0.41(0.8)	0.1(0.39)	0.12(0.43)	0.13(0.46)
	0.44(0.5)	0.39(0.61)	0.52(0.65)	0.09(0.32)	0.11(0.37)	0.12(0.37)
	0.52(0.68)	0.38(0.47)	0.36(0.5)	0.09(0.34)	0.1(0.35)	0.11(0.35)
	0.6(0.72)	0.37(0.5)	0.34(0.59)	0.1(0.34)	0.11(0.36)	0.11(0.35)
	0.6(0.73)	0.33(0.51)	0.39(0.59)	0.11(0.34)	0.11(0.36)	0.11(0.35)
	0.63(0.77)	0.37(0.61)	0.47(0.77)	0.1(0.34)	0.12(0.36)	0.14(0.4)
	0.63(0.79)	0.33(0.56)	0.5(0.71)	0.1(0.34)	0.11(0.36)	0.13(0.38)
	0.62(0.79)	0.41(0.58)	0.49(0.74)	0.1(0.34)	0.11(0.35)	0.13(0.38)
Right	0.36(0.41)	0.22(0.41)	0.16(0.37)	0.09(0.31)	0.1(0.33)	0.09(0.33)
	0.5(0.6)	0.34(0.56)	0.21(0.42)	0.1(0.32)	0.1(0.33)	0.1(0.33)
	0.53(0.59)	0.33(0.51)	0.25(0.43)	0.09(0.32)	0.1(0.32)	0.1(0.33)
	0.53(0.65)	0.26(0.49)	0.25(0.43)	0.09(0.32)	0.1(0.32)	0.1(0.32)
	0.49(0.61)	0.28(0.49)	0.23(0.39)	0.09(0.32)	0.1(0.32)	0.1(0.32)
	0.5(0.62)	0.21(0.44)	0.24(0.42)	0.09(0.32)	0.1(0.32)	0.1(0.32)
	0.48(0.6)	0.2(0.42)	0.24(0.36)	0.09(0.32)	0.09(0.31)	0.1(0.32)
	0.54(0.66)	0.21(0.43)	0.24(0.37)	0.09(0.32)	0.09(0.31)	0.1(0.32)
	0.273(0.87)			**0.12(0.5)**		

(b) Sequential Order of Words Encoded

Table 5.8: The performances obtained over the $\{\mathrm{T}\}^{\mathrm{YATEA}}_{\mathrm{YATEA}}$ when $|R_s| = 100$ and similarities are computed using the *cosine* between vectors. The performance is shown with regards to the observed *NAP*, for $i = 98$ (i.e., recall = 0.02) and $i = 4278$ (i.e., recall = 1.0). Parenthesised numbers show *NAP* for valid terms. The baseline shows the computed *NAP* when terms are sorted using the YATEA's weighting mechanism (see Figure 5.6); (a) shows the performance of models that ignore the sequential order of words in context-windows, whereas (b) shows the performance when this information is encoded.

Context		$NAP_{i=98}$			$NAP_{i=4278}$		
dir	size	k			k		
		1	7	25	1	7	25
Around	1	0.29(0.18)	0.38(0.44)	0.59(0.95)	0.08(0.3)	0.08(0.31)	0.12(0.5)
	2	0.27(0.17)	0.29(0.51)	0.6(0.9)	0.07(0.3)	0.08(0.32)	0.12(0.5)
	3	0.29(0.18)	0.32(0.79)	0.61(0.84)	0.08(0.3)	0.07(0.38)	0.12(0.45)
	4	0.31(0.19)	0.37(0.87)	0.66(0.87)	0.08(0.3)	0.1(0.49)	0.13(0.44)
	5	0.3(0.18)	0.38(0.85)	0.65(0.88)	0.08(0.3)	0.1(0.47)	0.14(0.47)
	6	0.3(0.18)	0.42(0.86)	0.64(0.88)	0.08(0.3)	0.11(0.47)	0.12(0.39)
	7	0.31(0.19)	0.42(0.86)	0.6(0.89)	0.08(0.31)	0.11(0.48)	0.13(0.46)
	8	0.31(0.2)	0.46(0.86)	0.59(0.88)	0.08(0.31)	0.12(0.48)	0.12(0.4)
Left	1	0.36(0.25)	0.61(0.91)	0.34(0.32)	0.08(0.31)	0.09(0.33)	0.09(0.36)
	2	0.58(0.52)	0.35(0.33)	0.36(0.34)	0.08(0.31)	0.12(0.4)	0.1(0.38)
	3	0.45(0.39)	0.31(0.3)	0.47(0.68)	0.08(0.31)	0.11(0.4)	0.11(0.41)
	4	0.35(0.24)	0.3(0.32)	0.36(0.44)	0.08(0.31)	0.11(0.39)	0.1(0.4)
	5	0.3(0.2)	0.34(0.36)	0.28(0.3)	0.08(0.31)	0.11(0.4)	0.1(0.4)
	6	0.29(0.19)	0.43(0.83)	0.35(0.39)	0.08(0.31)	0.12(0.43)	0.09(0.38)
	7	0.28(0.18)	0.47(0.88)	0.33(0.34)	0.08(0.3)	0.11(0.39)	0.09(0.35)
	8	0.29(0.26)	0.3(0.37)	0.28(0.28)	0.07(0.3)	0.09(0.38)	0.09(0.36)
Right	1	0.41(0.26)	0.47(0.63)	0.61(0.76)	0.08(0.31)	0.08(0.31)	0.09(0.35)
	2	0.31(0.19)	0.42(0.84)	0.51(0.79)	0.08(0.31)	0.09(0.44)	0.1(0.43)
	3	0.3(0.18)	0.41(0.83)	0.61(0.82)	0.08(0.3)	0.1(0.46)	0.11(0.38)
	4	0.31(0.2)	0.45(0.83)	0.59(0.79)	0.08(0.31)	0.11(0.47)	0.11(0.36)
	5	0.31(0.19)	0.42(0.83)	0.57(0.77)	0.08(0.31)	0.11(0.48)	0.12(0.44)
	6	0.32(0.18)	0.46(0.82)	0.54(0.79)	0.08(0.3)	0.1(0.44)	0.11(0.44)
	7	0.3(0.17)	0.47(0.81)	0.53(0.8)	0.08(0.3)	0.1(0.44)	0.11(0.44)
	8	0.32(0.18)	0.47(0.8)	0.51(0.8)	0.08(0.3)	0.11(0.46)	0.12(0.45)
Baseline		**0.273(0.87)**			**0.12(0.5)**		

(a) Sequential Order of Words Discarded

	$NAP_{i=98}$			$NAP_{i=4278}$		
	k			k		
	1	7	25	1	7	25
Around	0.41(0.27)	0.51(0.72)	0.65(0.8)	0.08(0.31)	0.08(0.32)	0.11(0.38)
	0.41(0.25)	0.55(0.8)	0.63(0.75)	0.08(0.31)	0.08(0.32)	0.12(0.39)
	0.42(0.26)	0.56(0.8)	0.63(0.73)	0.08(0.31)	0.08(0.32)	0.11(0.4)
	0.48(0.33)	0.56(0.81)	0.61(0.8)	0.08(0.31)	0.09(0.33)	0.13(0.47)
	0.58(0.5)	0.58(0.79)	0.64(0.79)	0.08(0.31)	0.11(0.34)	0.14(0.48)
	0.62(0.65)	0.56(0.75)	0.57(0.82)	0.08(0.31)	0.1(0.37)	0.11(0.41)
	0.68(0.79)	0.54(0.76)	0.56(0.82)	0.09(0.32)	0.11(0.4)	0.11(0.41)
	0.65(0.76)	0.56(0.8)	0.59(0.83)	0.09(0.31)	0.1(0.39)	0.11(0.41)
Left	0.36(0.25)	0.61(0.91)	0.34(0.32)	0.08(0.31)	0.09(0.33)	0.09(0.36)
	0.27(0.16)	0.6(0.68)	0.28(0.25)	0.07(0.3)	0.08(0.31)	0.08(0.3)
	0.27(0.16)	0.4(0.69)	0.33(0.3)	0.07(0.3)	0.1(0.34)	0.09(0.31)
	0.27(0.16)	0.48(0.68)	0.44(0.61)	0.07(0.3)	0.12(0.38)	0.1(0.34)
	0.27(0.16)	0.28(0.25)	0.28(0.26)	0.07(0.3)	0.09(0.32)	0.09(0.32)
	0.36(0.22)	0.31(0.29)	0.31(0.27)	0.08(0.31)	0.1(0.35)	0.1(0.36)
	0.33(0.21)	0.32(0.3)	0.58(0.73)	0.08(0.31)	0.1(0.36)	0.12(0.38)
	0.34(0.21)	0.33(0.32)	0.52(0.63)	0.08(0.31)	0.1(0.37)	0.11(0.38)
Right	0.41(0.26)	0.47(0.63)	0.61(0.76)	0.08(0.31)	0.08(0.31)	0.09(0.35)
	0.41(0.25)	0.53(0.8)	0.62(0.77)	0.08(0.31)	0.08(0.32)	0.11(0.33)
	0.42(0.27)	0.54(0.81)	0.61(0.73)	0.08(0.31)	0.09(0.32)	0.1(0.33)
	0.43(0.28)	0.51(0.81)	0.64(0.76)	0.08(0.31)	0.1(0.35)	0.11(0.38)
	0.45(0.31)	0.52(0.81)	0.63(0.75)	0.08(0.31)	0.12(0.38)	0.12(0.44)
	0.64(0.66)	0.49(0.56)	0.42(0.51)	0.09(0.31)	0.1(0.35)	0.08(0.35)
	0.68(0.76)	0.53(0.78)	0.42(0.5)	0.09(0.31)	0.11(0.38)	0.1(0.38)
	0.68(0.77)	0.52(0.71)	0.57(0.78)	0.09(0.31)	0.1(0.37)	0.11(0.39)
	0.27(0.87)			**0.12(0.5)**		

(b) Sequential Order of Words Encoded

Table 5.9: The performances observed over the $\{\mathrm{T}\}^{\mathrm{YATEA}}_{\mathrm{YATEA}}$ when $|R_s| = 100$ and similarities are computed using the *Euclidean* distance.

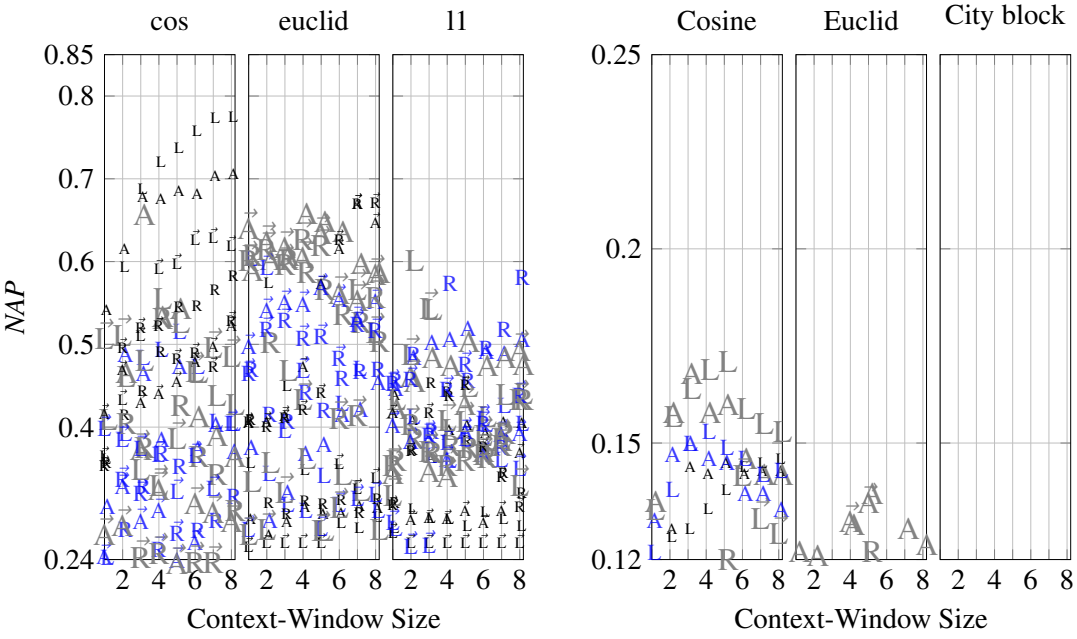
Context dir	size	$NAP_{i=98}$ k=1	$NAP_{i=98}$ k=7	$NAP_{i=98}$ k=25	$NAP_{i=4278}$ k=1	$NAP_{i=4278}$ k=7	$NAP_{i=4278}$ k=25
Around	1	0.44(0.33)	0.45(0.59)	0.43(0.62)	0.08(0.31)	0.07(0.34)	0.07(0.35)
	2	0.41(0.43)	0.49(0.7)	0.46(0.65)	0.08(0.31)	0.09(0.33)	0.09(0.33)
	3	0.37(0.24)	0.51(0.73)	0.49(0.73)	0.08(0.31)	0.09(0.32)	0.09(0.33)
	4	0.4(0.24)	0.51(0.77)	0.48(0.76)	0.08(0.31)	0.09(0.34)	0.1(0.34)
	5	0.41(0.24)	0.52(0.76)	0.51(0.79)	0.08(0.31)	0.1(0.35)	0.11(0.36)
	6	0.4(0.27)	0.49(0.77)	0.48(0.79)	0.08(0.31)	0.1(0.35)	0.1(0.36)
	7	0.41(0.27)	0.49(0.77)	0.49(0.79)	0.08(0.31)	0.1(0.36)	0.1(0.36)
	8	0.41(0.28)	0.51(0.76)	0.48(0.8)	0.08(0.31)	0.1(0.36)	0.11(0.37)
Left	1	0.28(0.23)	0.28(0.23)	0.44(0.55)	0.06(0.26)	0.06(0.26)	0.09(0.31)
	2	0.3(0.17)	0.42(0.61)	0.61(0.82)	0.08(0.3)	0.08(0.32)	0.09(0.33)
	3	0.29(0.16)	0.37(0.5)	0.54(0.78)	0.08(0.3)	0.08(0.31)	0.08(0.32)
	4	0.29(0.17)	0.36(0.54)	0.4(0.57)	0.08(0.3)	0.07(0.31)	0.09(0.35)
	5	0.28(0.16)	0.39(0.57)	0.45(0.67)	0.08(0.3)	0.08(0.31)	0.09(0.32)
	6	0.3(0.17)	0.41(0.59)	0.42(0.66)	0.08(0.3)	0.08(0.31)	0.08(0.32)
	7	0.29(0.17)	0.43(0.6)	0.44(0.71)	0.08(0.3)	0.08(0.31)	0.1(0.37)
	8	0.28(0.16)	0.41(0.62)	0.44(0.72)	0.08(0.3)	0.08(0.32)	0.1(0.37)
Right	1	0.31(0.19)	0.46(0.41)	0.36(0.6)	0.08(0.3)	0.08(0.31)	0.07(0.32)
	2	0.38(0.25)	0.49(0.69)	0.39(0.56)	0.08(0.31)	0.09(0.34)	0.08(0.33)
	3	0.46(0.41)	0.4(0.51)	0.37(0.5)	0.08(0.31)	0.08(0.32)	0.07(0.32)
	4	0.44(0.31)	0.58(0.65)	0.38(0.49)	0.08(0.31)	0.08(0.32)	0.08(0.31)
	5	0.39(0.23)	0.46(0.57)	0.38(0.51)	0.08(0.31)	0.09(0.34)	0.08(0.31)
	6	0.38(0.22)	0.5(0.63)	0.37(0.49)	0.08(0.31)	0.09(0.32)	0.08(0.31)
	7	0.35(0.19)	0.52(0.6)	0.4(0.53)	0.08(0.31)	0.08(0.31)	0.08(0.31)
	8	0.32(0.18)	0.58(0.74)	0.44(0.62)	0.08(0.3)	0.08(0.31)	0.09(0.32)
Baseline		**0.273(0.87)**			**0.12(0.5)**		

(a) Sequential Order of Words Discarded

$NAP_{i=98}$ k=1	$NAP_{i=98}$ k=7	$NAP_{i=98}$ k=25	$NAP_{i=4278}$ k=1	$NAP_{i=4278}$ k=7	$NAP_{i=4278}$ k=25
0.42(0.27)	0.41(0.61)	0.35(0.59)	0.08(0.31)	0.07(0.34)	0.08(0.36)
0.29(0.16)	0.39(0.53)	0.47(0.62)	0.08(0.3)	0.08(0.32)	0.08(0.33)
0.3(0.16)	0.39(0.48)	0.35(0.43)	0.08(0.3)	0.08(0.3)	0.07(0.3)
0.29(0.17)	0.37(0.49)	0.35(0.44)	0.08(0.3)	0.08(0.3)	0.07(0.3)
0.3(0.17)	0.4(0.51)	0.44(0.54)	0.08(0.3)	0.08(0.31)	0.08(0.3)
0.29(0.19)	0.37(0.43)	0.38(0.41)	0.08(0.31)	0.07(0.3)	0.07(0.3)
0.3(0.18)	0.4(0.45)	0.39(0.45)	0.08(0.3)	0.08(0.3)	0.07(0.3)
0.37(0.41)	0.4(0.47)	0.5(0.56)	0.08(0.31)	0.08(0.31)	0.08(0.31)
0.28(0.23)	0.28(0.23)	0.44(0.55)	0.06(0.26)	0.06(0.26)	0.09(0.31)
0.27(0.16)	0.26(0.17)	0.5(0.68)	0.07(0.3)	0.07(0.3)	0.1(0.33)
0.27(0.16)	0.26(0.16)	0.55(0.69)	0.07(0.3)	0.07(0.3)	0.08(0.31)
0.27(0.16)	0.39(0.58)	0.42(0.56)	0.07(0.3)	0.1(0.37)	0.08(0.3)
0.27(0.16)	0.39(0.51)	0.42(0.56)	0.07(0.3)	0.08(0.31)	0.08(0.31)
0.27(0.16)	0.42(0.49)	0.4(0.54)	0.07(0.3)	0.08(0.3)	0.08(0.3)
0.27(0.16)	0.37(0.49)	0.4(0.55)	0.07(0.3)	0.08(0.3)	0.08(0.31)
0.27(0.16)	0.36(0.51)	0.34(0.44)	0.07(0.3)	0.08(0.31)	0.08(0.33)
0.31(0.19)	0.46(0.41)	0.36(0.6)	0.08(0.3)	0.08(0.31)	0.07(0.32)
0.38(0.24)	0.46(0.68)	0.41(0.68)	0.08(0.31)	0.1(0.37)	0.08(0.33)
0.42(0.38)	0.4(0.52)	0.38(0.49)	0.08(0.31)	0.08(0.32)	0.07(0.32)
0.45(0.32)	0.45(0.59)	0.36(0.46)	0.08(0.31)	0.09(0.33)	0.07(0.31)
0.46(0.47)	0.48(0.49)	0.37(0.45)	0.08(0.31)	0.08(0.31)	0.07(0.31)
0.38(0.24)	0.41(0.5)	0.38(0.51)	0.08(0.31)	0.08(0.31)	0.07(0.31)
0.35(0.2)	0.38(0.48)	0.39(0.5)	0.08(0.31)	0.08(0.31)	0.07(0.31)
0.31(0.18)	0.45(0.57)	0.44(0.59)	0.08(0.3)	0.08(0.32)	0.08(0.32)
0.27(0.87)			**0.12(0.5)**		

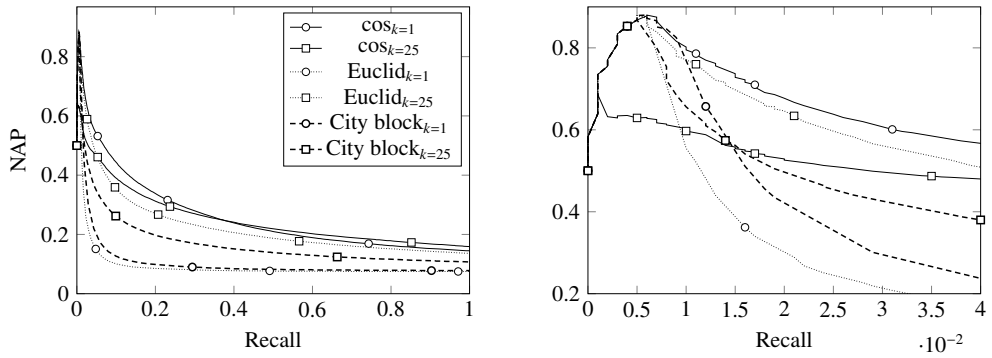
(b) Sequential Order of Words Encoded

Table 5.10: The performances observed over the $\{\mathrm{T}\}^{\mathrm{YATEA}}_{\mathrm{YATEA}}$ when $|R_s| = 100$ and similarities are computed using the *city block* distance.



(a) $NAP_{i=98}$ (b) $NAP_{i=4278}$

Figure 5.11: The performances observed in $\{\mathrm{T}\}^{\mathrm{YATEA}}_{\mathrm{YATEA}}$. The performance is reported using $NAP_i = 98$ (i.e., for 2% recall) and $NAP_i = 4278$ (i.e., 100% recall). Similar representation format as Figure 5.8 is used.



(a) (b)

Figure 5.12: The performances observed in $\{\mathrm{T}\}^{\mathrm{YATEA}}_{\mathrm{YATEA}}$ Using $|R_s| = 100$: NAP_i for $1 \leq i \leq 4278$ (i.e, various recall points) computed for context-windows of size 5 tokens that extend around candidate terms. Shown are results obtained for $k = 1, 25$ and the three similarity scores—that is, the cosine measure, the Euclidean and the city block distance; (a) shows NAP for $1 \leq i \leq 4278$, whereas (b) details the results for $1 \leq i \leq 172$ (i.e, recall less than 4%).

three tokens, then a low negative correlation between the size of context-windows and the performance of the method is observable. In these experiments, when the Euclidean or the city block distance are employed, then using $k = 25$ results in a more stable performance across different cut-off points for computing NAP_i than $k = 1$. As shown in Figure 5.12, if the goal is to extract only a small fraction of protein terms (e.g., 100), then using the city block or the Euclidean distance in the nearest neighbour framework gives the best performance. However, the performance of these combinations drops abruptly for larger recall values.

Similar to the previous experiment, encoding information about the sequential order of words in the context-windows does not improve the performances, particularly when using the cosine measure or the city block distance. Moreover, likewise experiments in $\{\mathrm{T}\}^{\text{c-}value}_{\text{ideal}}$, models constructed by collecting co-occurrences from context-windows that extend to left side of candidate terms, on average, show the best performances. However, using context-windows that extend around terms is a more cautious choice than choosing context-windows extending to the left or right side of candidate terms in the sense that they show less variance when parameters of the method, including the employed similarity metric, change.

Similar to the experiments over $\{\mathrm{T}\}^{\text{c-}value}_{\text{ideal}}$, the choice for the context-window's size remains dependant on the choices that are made for selecting the rest of the method's parameters. If the method's performance is averaged over k, then using context-windows of size 3 to 6 tokens is recommend. If $k = 1$, larger context-windows result in better performance; however, if k is large, then extending context-windows to more than 6 tokens reduces the performance. To investigate the similarity of the impact of the context-

Simialrity Metric	k	A	L	R	$\vec{A}$	$\vec{L}$	$\vec{R}$
Cosine	**1**	0.36	0.81	0.55	0.9	0.89	0.62
	7	0.83	0.4	0.81	0.48	0.05	0.07
	25	0.81	0.12	0.36	0.67	-0.36	0.33
Euclid	**1**	-0.4	0.9	-0.25	-0.6	-0.14	-0.07
	7	-0.88	0.19	0.41	0.36	0.62	0.0
	25	0.57	-0.32	-0.31	-0.4	0.6	-0.5
City block	**1**	0.33	0.01	0.4	-0.07	0.58	0.57
	7	0.17	0.6	0.46	0.11	-0.33	-0.55
	25	-0.5	0.4	0.43	0.05	0.78	-0.6

(a) Using *NAP* at 2% recall.

Simialrity Metric	k	A	L	R	$\vec{A}$	$\vec{L}$	$\vec{R}$
Cosine	**1**	0.6	0.83	0.97	0.21	0.28	0.14
	7	0.61	0.21	0.73	0.67	-0.17	0.33
	25	0.71	0.01	0.96	0.58	-0.71	0.69
Euclid	**1**	-0.85	0.29	-0.41	-0.24	-0.52	-0.34
	7	-0.86	0.53	0.19	-0.38	0.19	-0.76
	25	-0.29	0.44	0.76	0.6	0.24	-0.11
City block	**1**	0.18	-0.19	-0.52	-0.22	-0.58	-0.25
	7	0.65	-0.18	-0.08	0.5	-0.1	-0.16
	25	-0.43	-0.09	-0.14	0.08	0.75	-0.23

(b) Using *NAP* at 100% recall.

Table 5.11: Spearman's correlation coefficient (r_s) between the results obtained in $\{T\}^{c\text{-}value}_{ideal}$ and $\{T\}^{Y_A T_E A}_{Y_A T_E A}$ when the context-window's size used as the ranking variable and the remaining method's parameters are fixed. Tables (a) and (b) show the observed r_s when performances are computed using *NAP* at 2% and 100% recall, respectively.

window's size on the method's performance between experiments in $\{T\}^{Y_A T_E A}_{Y_A T_E A}$ and $\{T\}^{c\text{-}value}_{ideal}$, Table 5.11 reports the Spearman's coefficient correlation (r_s) when the size of context-windows is considered as the ranking variable.

In these tables, the results obtained in $\{T\}^{Y_A T_E A}_{Y_A T_E A}$ and $\{T\}^{c\text{-}value}_{ideal}$ are compared when the method's parameters, except the size of context-windows, are fixed. Each cell of Tables 5.11a and 5.11b shows the computed r_s between each column in Tables 5.8, 5.9, and 5.10 (from experiments in $\{T\}^{Y_A T_E A}_{Y_A T_E A}$) and the corresponding column in Tables 5.5, 5.6, and 5.7. Accordingly, if the choice for the best performing size of context-windows is similar in $\{T\}^{Y_A T_E A}_{Y_A T_E A}$ and $\{T\}^{c\text{-}value}_{ideal}$, then a high correlation (i.e., 1) is expected. As shown in Table 5.11, a high-correlation is observed only when using the cosine measure—that is, the same size of context-windows in both experiments results in a high performance.

5.4.3 Corpus Size: The Bigger the Better?

As described earlier, independent of the context-window's configuration for collecting co-occurrences, due to the Zipfian distribution of terms and words in context-windows, vectors that represent candidate terms are inevitably high-dimensional and *sparse* (i.e., most of the elements of vectors are zero). Whereas the high-dimensionality of vectors hinders the computation of similarities, their sparseness is likely to diminish the discriminatory power of the constructed distributional model. To overcome the high-dimensionality bar-

rier, random projections are employed in this research in order to reduce the dimension of vectors to a fixed certain size. Now that the vectors' dimension is set to a constant size, it is hypothesised that enlarging the size of the corpus reduces the number of zero elements in the vectors, and thus, the performance of the distributional model improves (e.g., see Bullinaria and Levy (2007), Pantel et al. (2009) as well as Gorman and Curran (2006b)).

In this section, the interplay between the size of the corpus and choosing the most discriminating configuration for context-windows in the proposed classification task is investigated. Two questions are investigated using empirical experiments, including (a) whether increasing the size of the corpus that is used for collecting co-occurrence frequencies enhances the performance of the classification task and (b) how doing so influences the choices that are made for configuring context-windows. The GENIA corpus is thus enlarged by fetching 223,316 abstracts from the PubMed repository,[1] of which each abstract contains at least three of the terms in the employed terminological resource.[2] Similar to the previous experiments, besides normalising text to lower-case letters and a Penn Treebank tokenisation, no other text preprocessing is performed. As a result, the enlarged corpus has more than 55 million tokens and a vocabulary of size 881,040.

This enlarged corpus contains 9,179,046 additional mentions of the terms ranked in the $\{T\}^{\textit{c-value}}_{\text{ideal}}$. As expected, the term frequencies in the enlarged corpus has a long tail distribution—that is, a small number of terms are frequent whereas the majority of terms are mentioned a few times (Figure 5.13). Using the vector space construction method explained in Section 5.2.1, the constructed vectors from $\{T\}^{\textit{c-value}}_{\text{ideal}}$ are augmented by the collected co-occurrences from the enlarged corpus. Hereafter, this set of vectors is denoted by $\{T\}^{\textit{c-value}}_{\text{Enlarged}}$. Nevertheless, the obtained vectors in $\{T\}^{\textit{c-value}}_{\text{Enlarged}}$ are less sparse than the previously built vectors in $\{T\}^{\textit{c-value}}_{\text{ideal}}$. For example, in $\{T\}^{\textit{c-value}}_{\text{ideal}}$, vectors that are constructed by collecting the co-occurrence frequencies from context-windows that extend by the size of one token around terms are approximately *five times* sparser than vectors in $\{T\}^{\textit{c-value}}_{\text{Enlarged}}$ that are constructed by collecting the co-occurrence frequencies from context-windows of the same configuration.[3]

An identical process employed for term classification in $\{T\}^{\textit{c-value}}_{\text{ideal}}$ (see, Section 5.4.1) is employed in $\{T\}^{\textit{c-value}}_{\text{Enlarged}}$. 48 different vector spaces are constructed, each reflects one of the possible combinations for context-window's configuration. The classification is then performed using three values of k (i.e., $k = 1, 7, 25$) and the same set of reference vectors (R_s) employed in the experiments reported in Section 5.4.1 (i.e., R_s comprises 100

[1] Accessible at `http://www.ncbi.nlm.nih.gov/pmc/tools/ftp/#Source_files`.

[2] The set of employed abstracts can be retrieved from `http://atmykitchen.info/phd/materials/genia/extended_abstracts.tar.gz`.

[3] Please note that in the proposed method, apart from the size of the corpus employed for collecting co-occurrences, the sparseness of vectors is also determined by the number of zero and non-zero elements in word vectors.

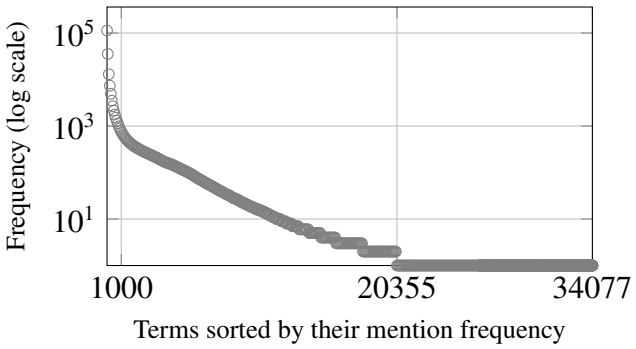


Figure 5.13: The frequency of terms in the enlarged corpus: more than 40% of the terms never appear in the enlarged corpus.

terms of which 36 are positive examples). Tables 5.12, 5.13, and 5.14 report the observed results when the cosine measure, Euclidean distance, and the city block distance are employed for computing similarities, respectively. Figure 5.14 plots numbers reported in these tables.

As shown in Figure 5.14b, likewise the previous experiments and independently of the size of the input corpus, if *NAP* is computed for a high recall such as 100%, then the cosine similarity outperforms both the Euclidean and City block distance. In $\{T\}^{\text{c-}value}_{\text{ideal}}$ and for $NAP_{i=8900}$, the best classification performance is observed using models that are built by collecting co-occurrence frequencies in context-windows of size 4 or 5 tokens that extend around terms. However, in $\{T\}^{\text{c-}value}_{\text{Enlarged}}$, models that are built using context-windows that expand to the left side of the candidate terms outperform models that are built by collecting co-occurrence frequencies in context-windows that expand around the terms. In addition, similar to the experiments in $\{T\}^{\text{c-}value}_{\text{ideal}}$, in $\{T\}^{\text{c-}value}_{\text{Enlarged}}$, a large value for k results in a more desirable performance than a small value such as $k = 1$, too.

Figure 5.15 plots the changes that are observed by enlarging the size of the input corpus when the performance is measured using $NAP_{i=8900}$. For instance, when using the cosine similarity and $k = 25$ in the model constructed using context-windows of size 6 tokens that neglect word order information and extend only to the *left* side of terms, the $NAP_{i=8900}$ in $\{T\}^{\text{c-}value}_{\text{Enlarged}}$ is 0.461 (see Table 5.12). However, the same classification parameters and configuration for context-windows in $\{T\}^{\text{c-}value}_{\text{ideal}}$ gives the performance of 0.405 (see Table 5.5). In Figure 5.15, this increase in the performance is marked by a wide circle at the corresponding position. Accordingly, the plotted results suggest that when the corpus size increases, the type of employed similarity measure plays a role in determining the changes in the performances. When similarities are calculated using the cosine measure, enlarging the size of the corpus enhances the performance. Similarly,

Context		$NAP_{i=200}$			$NAP_{i=8900}$		
dir	size	k			k		
		1	7	25	1	7	25
Around	1	0.817	0.724	0.714	0.371	0.388	0.389
	2	0.61	0.797	0.748	0.395	0.409	0.41
	3	0.641	0.69	0.646	0.41	0.416	0.414
	4	0.672	0.795	0.656	0.417	0.427	0.421
	5	0.686	0.794	0.659	0.416	0.428	0.422
	6	0.673	0.781	0.686	0.414	0.428	0.421
	7	0.688	0.797	0.723	0.416	0.43	0.42
	8	0.698	0.805	0.692	0.416	0.427	0.415
Left	1	0.643	0.716	0.772	0.333	0.372	0.356
	2	0.789	0.81	0.782	0.377	0.401	0.388
	3	0.834	0.798	0.808	0.398	0.421	0.417
	4	0.851	0.825	0.772	0.415	0.435	0.437
	5	0.838	0.823	0.781	0.42	0.445	0.451
	6	0.825	0.82	0.807	0.427	0.453	0.458
	7	0.813	0.823	0.833	0.426	0.456	0.461
	8	0.808	0.85	0.82	0.428	0.456	0.461
Right	1	0.765	0.655	0.609	0.349	0.383	0.361
	2	0.611	0.819	0.748	0.359	0.389	0.38
	3	0.664	0.763	0.641	0.374	0.39	0.382
	4	0.681	0.812	0.648	0.385	0.395	0.388
	5	0.692	0.837	0.707	0.389	0.397	0.389
	6	0.684	0.828	0.705	0.389	0.395	0.39
	7	0.708	0.834	0.727	0.39	0.392	0.389
	8	0.712	0.844	0.726	0.39	0.392	0.387
Baseline		0.364			0.273		

(a) Sequential Order of Words Discarded

$NAP_{i=200}$			$NAP_{i=8900}$		
k			k		
1	7	25	1	7	25
0.748	0.687	0.608	0.357	0.388	0.365
0.703	0.7	0.614	0.369	0.394	0.374
0.691	0.733	0.555	0.373	0.398	0.378
0.676	0.788	0.591	0.375	0.399	0.381
0.657	0.793	0.575	0.376	0.395	0.379
0.642	0.778	0.561	0.373	0.395	0.377
0.63	0.805	0.58	0.372	0.394	0.375
0.625	0.831	0.592	0.372	0.395	0.373
0.643	0.716	0.772	0.333	0.372	0.356
0.574	0.868	0.898	0.323	0.414	0.424
0.711	0.949	0.906	0.347	0.432	0.423
0.731	0.921	0.897	0.373	0.428	0.42
0.777	0.905	0.912	0.369	0.417	0.412
0.806	0.919	0.833	0.384	0.407	0.399
0.843	0.889	0.843	0.374	0.393	0.388
0.845	0.854	0.767	0.371	0.371	0.382
0.765	0.655	0.609	0.349	0.383	0.361
0.747	0.657	0.647	0.361	0.385	0.368
0.744	0.674	0.598	0.369	0.386	0.37
0.727	0.707	0.589	0.369	0.384	0.37
0.698	0.732	0.583	0.368	0.383	0.368
0.68	0.761	0.62	0.366	0.386	0.368
0.666	0.762	0.643	0.365	0.383	0.365
0.651	0.749	0.617	0.365	0.38	0.362
0.364			0.273		

(b) Sequential Order of Words Encoded

Table 5.12: The performances observed over the $\{\mathrm{T}\}^{\text{c-}value}_{\text{Enlarged}}$ when $|R_s| = 100$ and similarities are computed using the *cosine* between vectors. Similar to the experiments over $\{\mathrm{T}\}^{\text{c-}value}_{\text{ideal}}$, the performance is shown with regards to the observed NAP_i, for $i = 200$ (i.e., recall = 0.02) and $i = 8900$ (i.e., recall = 1.0).

Context		$NAP_{i=200}$			$NAP_{i=8900}$		
dir	size	k			k		
		1	7	25	1	5	25
Around	1	0.595	0.772	0.742	0.341	0.312	0.329
	2	0.639	0.771	0.852	0.349	0.331	0.315
	3	0.692	0.752	0.728	0.359	0.325	0.302
	4	0.627	0.773	0.71	0.302	0.322	0.298
	5	0.621	0.732	0.814	0.333	0.313	0.319
	6	0.623	0.783	0.767	0.345	0.32	0.328
	7	0.598	0.728	0.742	0.33	0.316	0.31
	8	0.59	0.736	0.76	0.322	0.316	0.306
Left	1	0.605	0.82	0.73	0.314	0.343	0.314
	2	0.593	0.777	0.706	0.322	0.338	0.319
	3	0.635	0.793	0.637	0.334	0.344	0.323
	4	0.66	0.81	0.632	0.339	0.34	0.315
	5	0.681	0.788	0.662	0.343	0.341	0.311
	6	0.655	0.831	0.667	0.338	0.334	0.31
	7	0.64	0.772	0.633	0.336	0.328	0.304
	8	0.629	0.772	0.648	0.335	0.322	0.303
Right	1	0.638	0.854	0.66	0.287	0.299	0.301
	2	0.552	0.85	0.599	0.282	0.305	0.282
	3	0.641	0.773	0.697	0.296	0.3	0.293
	4	0.556	0.713	0.57	0.279	0.299	0.283
	5	0.522	0.744	0.565	0.275	0.301	0.283
	6	0.552	0.729	0.773	0.28	0.309	0.298
	7	0.542	0.708	0.74	0.276	0.309	0.312
	8	0.544	0.727	0.633	0.276	0.313	0.293
Baseline		0.364			0.273		

(a) Sequential Order of Words Discarded

$NAP_{i=200}$			$NAP_{i=8900}$		
k			k		
1	7	25	1	5	25
0.682	0.886	0.668	0.292	0.304	0.302
0.694	0.886	0.686	0.296	0.313	0.311
0.698	0.897	0.696	0.297	0.317	0.319
0.683	0.893	0.698	0.296	0.322	0.323
0.672	0.898	0.71	0.291	0.325	0.322
0.657	0.906	0.709	0.292	0.331	0.322
0.618	0.894	0.71	0.286	0.332	0.322
0.609	0.867	0.709	0.282	0.33	0.317
0.605	0.82	0.73	0.314	0.343	0.314
0.646	0.749	0.77	0.335	0.301	0.323
0.663	0.844	0.748	0.327	0.324	0.333
0.694	0.837	0.748	0.344	0.331	0.331
0.743	0.833	0.757	0.351	0.329	0.336
0.716	0.801	0.831	0.347	0.328	0.332
0.742	0.776	0.833	0.349	0.322	0.332
0.693	0.842	0.832	0.337	0.328	0.323
0.638	0.854	0.66	0.287	0.299	0.301
0.634	0.88	0.68	0.285	0.302	0.304
0.623	0.879	0.679	0.286	0.309	0.309
0.635	0.862	0.686	0.285	0.31	0.316
0.638	0.847	0.698	0.282	0.311	0.315
0.635	0.855	0.702	0.284	0.318	0.317
0.603	0.877	0.707	0.281	0.32	0.318
0.582	0.84	0.696	0.279	0.323	0.318
0.364			0.273		

(b) Sequential Order of Words Encoded

Table 5.13: The results observed in $\{\mathrm{T}\}^{\text{c-}value}_{\text{Enlarged}}$ when $|R_s| = 100$ and similarities are computed using the *Euclidean* distance.

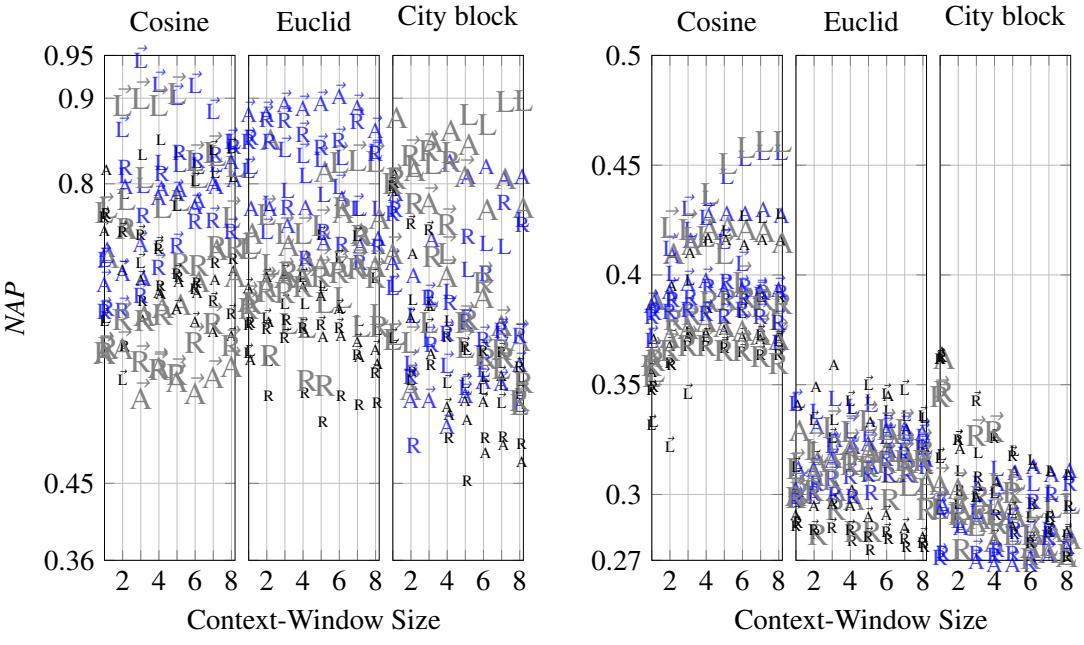
Context dir	size	$NAP_{i=200}$ k=1	k=7	k=25	$NAP_{i=8900}$ k=1	k=5	k=25
Around	1	0.812	0.789	0.878	0.365	0.295	0.362
	2	0.718	0.693	0.816	0.321	0.285	0.293
	3	0.601	0.735	0.85	0.304	0.29	0.289
	4	0.53	0.826	0.862	0.306	0.305	0.293
	5	0.524	0.809	0.846	0.311	0.31	0.29
	6	0.486	0.821	0.77	0.313	0.313	0.282
	7	0.495	0.807	0.806	0.311	0.309	0.283
	8	0.475	0.809	0.772	0.309	0.311	0.28
Left	1	0.623	0.683	0.624	0.318	0.296	0.296
	2	0.665	0.646	0.82	0.311	0.286	0.293
	3	0.658	0.667	0.824	0.319	0.305	0.296
	4	0.64	0.675	0.837	0.327	0.312	0.305
	5	0.606	0.7	0.881	0.32	0.305	0.307
	6	0.575	0.731	0.87	0.316	0.305	0.294
	7	0.565	0.727	0.896	0.311	0.301	0.294
	8	0.575	0.758	0.897	0.309	0.296	0.296
Right	1	0.801	0.77	0.806	0.363	0.273	0.347
	2	0.581	0.495	0.777	0.302	0.254	0.275
	3	0.589	0.613	0.715	0.308	0.267	0.299
	4	0.503	0.661	0.826	0.291	0.275	0.287
	5	0.453	0.739	0.642	0.282	0.284	0.267
	6	0.503	0.696	0.596	0.296	0.297	0.267
	7	0.505	0.778	0.597	0.29	0.3	0.266
	8	0.496	0.753	0.568	0.295	0.305	0.265
Baseline		**0.364**			**0.273**		

(a) Sequential Order of Words Discarded

$NAP_{i=200}$ k=1	k=7	k=25	$NAP_{i=8900}$ k=1	k=5	k=25
0.795	0.712	0.725	0.363	0.294	0.307
0.647	0.551	0.565	0.326	0.266	0.305
0.661	0.551	0.783	0.3	0.271	0.29
0.542	0.521	0.698	0.281	0.27	0.281
0.55	0.557	0.815	0.287	0.27	0.297
0.548	0.574	0.673	0.28	0.273	0.278
0.579	0.571	0.64	0.284	0.274	0.276
0.546	0.593	0.623	0.284	0.278	0.274
0.623	0.683	0.624	0.318	0.296	0.296
0.574	0.594	0.62	0.297	0.29	0.294
0.635	0.623	0.644	0.3	0.294	0.288
0.57	0.592	0.725	0.302	0.296	0.293
0.571	0.566	0.615	0.294	0.29	0.284
0.564	0.588	0.605	0.291	0.284	0.284
0.547	0.629	0.59	0.29	0.286	0.279
0.558	0.546	0.548	0.283	0.277	0.275
0.801	0.77	0.806	0.363	0.273	0.347
0.756	0.579	0.839	0.326	0.256	0.293
0.749	0.672	0.845	0.344	0.275	0.329
0.624	0.629	0.758	0.327	0.274	0.331
0.613	0.643	0.644	0.319	0.275	0.303
0.608	0.623	0.691	0.279	0.27	0.273
0.607	0.633	0.612	0.284	0.282	0.267
0.589	0.627	0.553	0.273	0.266	0.265
0.364			**0.273**		

(b) Sequential Order of Words Encoded

Table 5.14: The results observed in $\{T\}^{\text{c-}value}_{\text{Enlarged}}$ when $|R_s| = 100$ and similarities are computed using the *city block* distance.



(a) $NAP_{i=200}$ (b) $NAP_{i=8900}$

Figure 5.14: The NAP_i observed over $\{T\}^{\text{c-}value}_{\text{Enlarged}}$ for $i = 200$ (i.e., 2% recall) and $i = 8900$ (i.e., recall 100%) shown in (a) and (b), respectively.

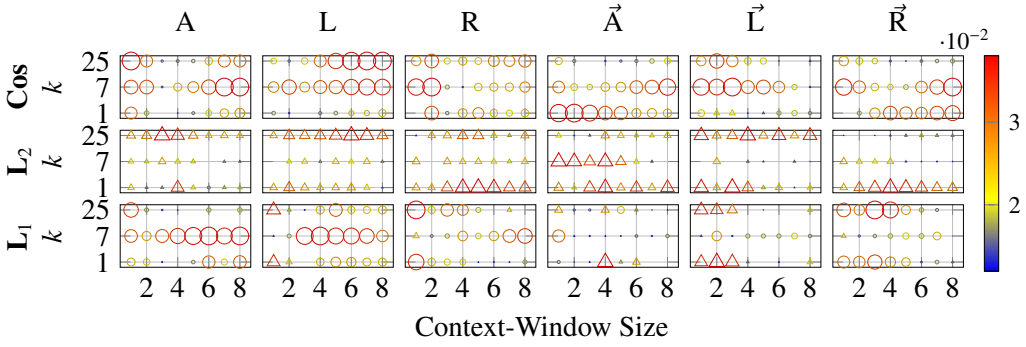


Figure 5.15: Changes in the performance of models caused by increasing the size of the input corpus. The method's performance is measured using *NAP* at 100% recall. The figure shows the absolute value of the difference between the performance obtained from a model constructed in $\{\mathrm{T}\}^{\text{c-}value}_{\text{ideal}}$ and the corresponding model in $\{\mathrm{T}\}^{\text{c-}value}_{\text{Enlarged}}$. Triangles denote negative change, while circles show positive change. The size/colour of shapes represents the amount of changes. The x-axis shows various configurations of context-windows (i.e., size, direction, and encoding word order information). The y-axis, however, represents classification parameters (i.e., the values of k and the employed measures for calculating similarities).

the city block distance shows a relatively better performance with larger input corpus. However, when similarities are measured using the Euclidean distance, an increase in the size of the corpus can drastically decline the performance.

Figures 5.14a and 5.16, similar to Figures 5.14b and 5.15, show the method's performance, however, when it is measured by $NAP_{i=200}$ (i.e., for the 2% recall). As shown, if the performance is assessed for a small recall, then all three measures equally perform well. In this case, increasing the size of the corpus enhances or diminishes the performance by approximately 20%. Again, the Euclidean distance is more susceptible to an increase in the corpus size. In contrary, the cosine measure consistently shows a better performance when the corpus size increases. Results suggest a similar conclusion for the city block distance. Although in this case, the enhancement is not as steady as the cosine measure and it depends on the value of k and the context-window's configuration, too. Particularly, for $k = 1$, the performance frequently drops when the size of the input corpus increases.

In the experiments over $\{\mathrm{T}\}^{\text{c-}value}_{\text{Enlarged}}$, with respect to the relationship between recall and performance, the behaviour of similarity measures is similar to the previous experiments. If the performance of the method is studied across recall values, city block distance outperforms the cosine measure then for a small recall. When using the city block distance, however, as shown in Figure 5.17, the performance drops abruptly as recall increases. Compared to Figure 5.10 and 5.12, for a number of context-window configurations, enlarging the corpus eminently enhances the performance of the cosine metric at small recall values and thus makes it a rival to the Euclidean and city block distance in

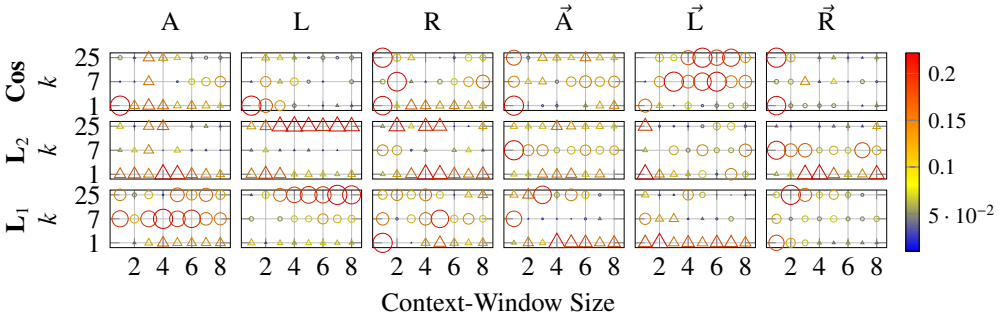


Figure 5.16: The changes in the performance of the method caused by increasing the size of the input corpus when the method's performance is measured using $NAP_{i=200}$ (i.e, 2% recall). The presentation format is similar to Figure 5.15: circles show positive effect whereas triangles show negative impact on the performance.

tasks that aim for extracting a small number of terms.

Similar to Table 5.11, Table 5.15 reports the Spearman's coefficient correlation (r_s) when the size of context-window is considered as the ranking variable (see page 168) and the results obtained in $\{T\}^{\text{c-}value}_{\text{Enlarged}}$ are compared with the results in $\{T\}^{\text{c-}value}_{\text{ideal}}$. A high positive correlation for the choice of the best performing sizes of context-windows is observable between these two experiments only when cosine is employed for computing similarities and the performance is assessed for a large recall value. Otherwise, as the Table 5.15 suggests, if the size of the corpus changes, unfortunately the choice for the size of context-window must be revised to achieve the best performance.

With respect to the effect of encoding information about the sequential order of words in the models, the observed results in $\{T\}^{\text{c-}value}_{\text{Enlarged}}$ (Figure 5.18) is also inconsistent with the observations that are made over $\{T\}^{\text{c-}value}_{\text{ideal}}$ (see Figure 5.9), specifically when the performances are assessed at a recall pint (i.e., $NAP_{i=200}$). In the experiments over $\{T\}^{\text{c-}value}_{\text{ideal}}$ for $NAP_{i=200}$, encoding information about the order of words in context-windows often worsened the performance of the Euclidean distance and the cosine similarity. However, this information improves the performance of the city block distance. In contrary, in the experiments over $\{T\}^{\text{c-}value}_{\text{Enlarged}}$, encoding information about the order of words improves the performance of the Euclidean distance, and exacerbates it for the city block distance. However, for a large recall value (i.e., $NAP_{i=8900}$), the observed results in both $\{T\}^{\text{c-}value}_{\text{ideal}}$ and $\{T\}^{\text{c-}value}_{\text{Enlarged}}$ are similar in the sense that models that encode this information are not among the top performers.

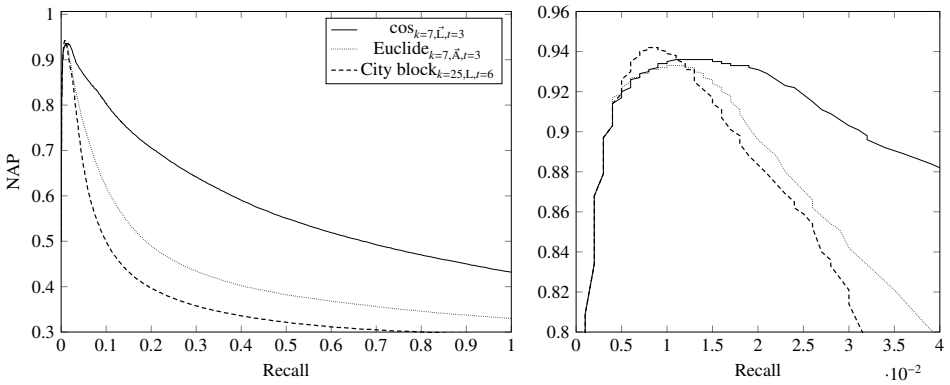


Figure 5.17: Performance of the similarity metrics over the range of recall values: shown the observed performances of the top performing models in the reported results. Similar to the previous experiments, when using the Euclidean or city block distance, the performance drops abruptly. However, if the aim is to extract only a small number of terms such as 100 (i.e., in this example, approximately a recall less than 1%), then the city block distance outperforms other similarity metrics.

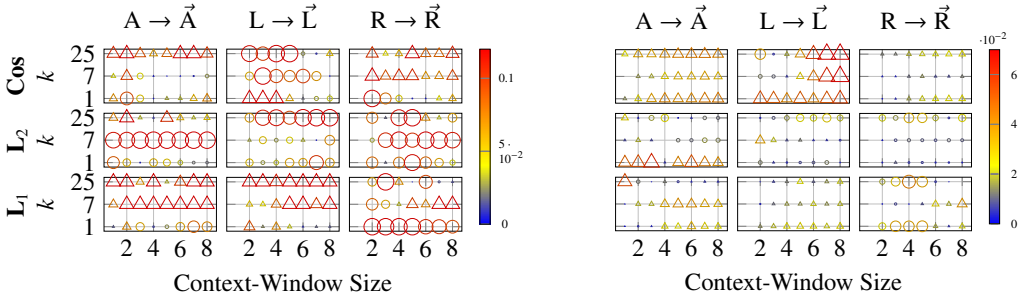
Simialrity Metric	k	A	L	R	$\vec{A}$	$\vec{L}$	$\vec{R}$
Cosine	**1**	-0.62	0.92	-0.32	-0.82	0.74	-0.53
	7	-0.11	0.51	0.39	0.14	-0.04	0.55
	25	-0.19	0.08	0.52	-0.65	0.38	-0.03
Euclid	**1**	0.82	-0.14	0.34	0.9	-0.77	0.36
	7	0.22	0.37	-0.88	0.5	0.24	0.26
	25	0.13	-0.21	-0.06	0.81	0.07	0.85
City block	**1**	0.82	0.87	-0.98	0.88	0.74	-0.95
	7	0.4	0.39	0.63	-0.11	0.05	0.82
	25	0.64	-0.85	-0.39	-0.01	0.25	0.28

(a) Using *NAP* at 2% recall.

Simialrity Metric	k	A	L	R	$\vec{A}$	$\vec{L}$	$\vec{R}$
Cosine	**1**	0.95	0.98	0.92	0.8	0.84	0.01
	7	0.83	0.98	0.85	0.64	0.8	0.66
	25	0.94	0.94	0.93	0.96	0.85	0.84
Euclid	**1**	0.2	0.39	0.18	0.41	0.21	0.3
	7	0.53	0.5	0.25	-0.45	0.78	-0.36
	25	0.33	0.19	-0.02	0.79	0.54	0.97
City block	**1**	0.79	0.17	0.72	0.83	0.78	0.68
	7	0.77	-0.38	0.15	-0.54	0.75	0.12
	25	0.9	-0.25	0.42	0.65	0.76	0.52

(b) Using *NAP* for %100 recall.

Table 5.15: Shown Spearman's correlation coefficient (r_s) between the results obtained in $\{\mathrm{T}\}^{\text{c-}value}_{\text{Enlarged}}$ and $\{\mathrm{T}\}^{\text{c-}value}_{\text{ideal}}$ when the context-window's size considered as the ranking variable and the remaining method's parameters are fixed. Table (a) and (b) shows the observed r_s when performances are computed using *NAP* at 2% and 100% recall, respectively.



(a) $NAP_{i=200}$ (b) $NAP_{i=8900}$

Figure 5.18: The effect of encoding information about the sequential order in the method's performance when dealing with the enlarged corpus. Results are shown in the same format as Figure 5.9. An enhancement in the performance is marked by a circle whereas a decrease is shown by a triangle; the size of the shapes shows the intensity.

5.4.3.1 The effect of enlarging the corpus in the presence of invalid terms

The aim is to investigate whether using a larger corpus can enhance the method's performance when the classification is carried out in the presence of invalid terms. Vectors constructed from the set of candidate terms in the $\{T\}^{Y_A T_E A}_{Y_A T_E A}$ are thus augmented by additional co-occurrence frequencies (hereafter, denote by $\{T\}^{Y_A T_E A}_{\text{Enlarged}}$). In the enlarged corpus $\{T\}^{Y_A T_E A}_{\text{Enlarged}}$, these candidate terms are mentioned more than two million times. Similar to the experiments over $\{T\}^{c\text{-}value}_{\text{Enlarged}}$, these mentions of terms are scanned in order to update the co-occurrence frequencies of term vectors. Afterwards, a classification process identical to the one applied to $\{T\}^{Y_A T_E A}_{Y_A T_E A}$ (see Section 5.4.2) is employed to classify the updated vectors obtained from $\{T\}^{Y_A T_E A}_{\text{Enlarged}}$.

The method's performance over $\{T\}^{Y_A T_E A}_{\text{Enlarged}}$ is reported in Tables 5.16, 5.17, and 5.18. These numbers are plotted in Figure 5.19. To study the effect of enlarging the corpus, these results are compared with their corresponding values obtained from the earlier experiment over $\{T\}^{Y_A T_E A}_{Y_A T_E A}$ (i.e., results reported in Section 5.4.2; see Figure 5.11). Here, enlarging the corpus size marginally enhances the best observed performance. Particularly, although enlarging the corpus enhances the discriminatory power of the models, it does not necessarily improve the method's ability to filter invalid terms (see *NAP* for valid terms reported in Tables 5.16, 5.17, and 5.18; that is, numbers placed in parentheses).

Figure 5.20 plots changes in the method's performance caused by enlarging the corpus at 100% recall (i.e., $NAP_{i=4278}$ in this experiment). Compared to experiments over $\{T\}^{c\text{-}value}_{\text{Enlarged}}$ (see Figure 5.15), increasing the size of corpus in $\{T\}^{Y_A T_E A}_{Y_A T_E A}$ enhances the performance disregarding the method's parameters. However, even with these enhancements, performances remain below the baseline for many combinations of parameters, particularly

Context		$NAP_{i=98}$			$NAP_{i=4278}$		
dir	size	k			k		
		1	7	25	1	7	25
Around	1	0.74(0.83)	0.35(0.72)	0.32(0.65)	0.13(0.36)	0.13(0.4)	0.15(0.4)
	2	0.54(0.69)	0.61(0.81)	0.58(0.77)	0.14(0.36)	0.16(0.4)	0.18(0.4)
	3	0.57(0.72)	0.55(0.67)	0.53(0.73)	0.15(0.36)	0.17(0.38)	0.17(0.38)
	4	0.62(0.78)	0.57(0.72)	0.49(0.71)	0.15(0.36)	0.17(0.37)	0.17(0.37)
	5	0.65(0.78)	0.6(0.68)	0.39(0.62)	0.15(0.35)	0.17(0.37)	0.16(0.37)
	6	0.64(0.77)	0.6(0.69)	0.43(0.58)	0.14(0.35)	0.17(0.36)	0.16(0.36)
	7	0.65(0.78)	0.64(0.74)	0.42(0.56)	0.14(0.35)	0.17(0.36)	0.16(0.36)
	8	0.65(0.78)	0.61(0.72)	0.38(0.54)	0.14(0.36)	0.16(0.36)	0.16(0.36)
Left	1	0.42(0.51)	0.41(0.71)	0.39(0.71)	0.12(0.39)	0.14(0.45)	0.15(0.49)
	2	0.68(0.8)	0.4(0.7)	0.44(0.76)	0.15(0.4)	0.16(0.43)	0.17(0.45)
	3	0.74(0.82)	0.59(0.74)	0.45(0.7)	0.15(0.37)	0.17(0.42)	0.18(0.44)
	4	0.79(0.86)	0.59(0.68)	0.49(0.65)	0.15(0.37)	0.17(0.4)	0.19(0.41)
	5	0.8(0.87)	0.63(0.78)	0.51(0.62)	0.16(0.37)	0.17(0.39)	0.19(0.4)
	6	0.78(0.85)	0.56(0.68)	0.43(0.55)	0.15(0.37)	0.17(0.39)	0.18(0.4)
	7	0.75(0.84)	0.54(0.64)	0.43(0.56)	0.15(0.36)	0.17(0.38)	0.18(0.39)
	8	0.74(0.83)	0.55(0.64)	0.41(0.51)	0.14(0.36)	0.17(0.37)	0.17(0.38)
Right	1	0.48(0.53)	0.24(0.43)	0.27(0.53)	0.1(0.32)	0.11(0.33)	0.11(0.35)
	2	0.45(0.52)	0.32(0.4)	0.32(0.53)	0.1(0.32)	0.11(0.33)	0.12(0.34)
	3	0.47(0.54)	0.35(0.45)	0.31(0.49)	0.1(0.32)	0.12(0.33)	0.12(0.34)
	4	0.53(0.6)	0.34(0.44)	0.34(0.54)	0.11(0.33)	0.12(0.33)	0.12(0.34)
	5	0.54(0.62)	0.48(0.56)	0.37(0.55)	0.11(0.33)	0.13(0.34)	0.13(0.35)
	6	0.56(0.64)	0.43(0.55)	0.47(0.6)	0.12(0.34)	0.13(0.34)	0.13(0.35)
	7	0.58(0.66)	0.38(0.53)	0.47(0.6)	0.12(0.34)	0.13(0.34)	0.13(0.35)
	8	0.59(0.67)	0.42(0.54)	0.38(0.53)	0.12(0.34)	0.13(0.34)	0.13(0.34)
Baseline		**0.273(0.87)**			**0.12(0.5)**		

(a) Sequential Order of Words Discarded

$NAP_{i=98}$			$NAP_{i=4278}$		
k			k		
1	7	25	1	7	25
0.49(0.55)	0.21(0.42)	0.27(0.58)	0.1(0.32)	0.11(0.35)	0.12(0.37)
0.49(0.64)	0.2(0.4)	0.3(0.6)	0.1(0.33)	0.11(0.34)	0.13(0.37)
0.49(0.62)	0.21(0.39)	0.3(0.6)	0.1(0.32)	0.11(0.34)	0.13(0.36)
0.48(0.6)	0.26(0.41)	0.32(0.61)	0.09(0.32)	0.11(0.34)	0.13(0.35)
0.47(0.58)	0.29(0.43)	0.34(0.64)	0.09(0.32)	0.11(0.33)	0.13(0.35)
0.46(0.57)	0.32(0.46)	0.33(0.63)	0.1(0.32)	0.11(0.33)	0.12(0.35)
0.46(0.57)	0.28(0.45)	0.32(0.61)	0.09(0.32)	0.11(0.33)	0.12(0.35)
0.46(0.57)	0.27(0.44)	0.36(0.63)	0.1(0.32)	0.11(0.33)	0.12(0.35)
0.42(0.51)	0.41(0.71)	0.39(0.71)	0.12(0.39)	0.14(0.45)	0.15(0.49)
0.66(0.77)	0.47(0.71)	0.48(0.76)	0.13(0.37)	0.14(0.42)	0.13(0.45)
0.66(0.78)	0.46(0.7)	0.47(0.76)	0.12(0.36)	0.14(0.4)	0.12(0.43)
0.67(0.78)	0.54(0.7)	0.45(0.74)	0.12(0.36)	0.15(0.4)	0.12(0.42)
0.67(0.78)	0.47(0.62)	0.42(0.68)	0.13(0.38)	0.15(0.4)	0.12(0.42)
0.68(0.78)	0.44(0.6)	0.42(0.65)	0.13(0.36)	0.13(0.38)	0.12(0.41)
0.68(0.78)	0.38(0.57)	0.4(0.63)	0.12(0.35)	0.12(0.37)	0.12(0.41)
0.63(0.76)	0.44(0.64)	0.36(0.61)	0.11(0.35)	0.12(0.37)	0.11(0.4)
0.48(0.53)	0.24(0.43)	0.27(0.53)	0.1(0.32)	0.11(0.33)	0.11(0.35)
0.56(0.71)	0.21(0.43)	0.3(0.57)	0.11(0.32)	0.11(0.33)	0.12(0.35)
0.54(0.69)	0.22(0.4)	0.32(0.62)	0.1(0.32)	0.11(0.33)	0.12(0.35)
0.53(0.65)	0.21(0.35)	0.33(0.59)	0.1(0.32)	0.11(0.33)	0.12(0.34)
0.51(0.62)	0.26(0.39)	0.3(0.62)	0.1(0.32)	0.11(0.33)	0.12(0.34)
0.5(0.61)	0.32(0.45)	0.32(0.64)	0.1(0.32)	0.11(0.33)	0.12(0.34)
0.49(0.6)	0.34(0.47)	0.33(0.63)	0.09(0.32)	0.11(0.33)	0.12(0.34)
0.48(0.6)	0.36(0.46)	0.31(0.65)	0.09(0.32)	0.11(0.32)	0.12(0.34)
0.273(0.87)			**0.12(0.5)**		

(b) Sequential Order of Words Encoded

Table 5.16: The performances observed over the $\{\mathrm{T}\}^{\mathrm{YATEA}}_{\mathrm{Enlarged}}$ when $|R_s| = 100$ and similarities are computed using the *cosine* between vectors. Similar to the previous experiments, the performance is reported with respect to the observed NAP_i, for $i = 98$ (i.e., recall = 0.02) and $i = 4278$ (i.e., recall 100%). Numbers placed in parentheses show NAP_i when the precision is computed with respect to the number of valid terms.

Context		$NAP_{i=98}$			$NAP_{i=4278}$		
dir	size	k			k		
		1	7	25	1	7	25
Around	1	0.53(0.74)	0.48(0.66)	0.48(0.58)	0.13(0.43)	0.12(0.48)	0.13(0.37)
	2	0.54(0.77)	0.47(0.66)	0.43(0.72)	0.12(0.46)	0.12(0.46)	0.1(0.45)
	3	0.49(0.73)	0.5(0.72)	0.52(0.73)	0.11(0.44)	0.11(0.44)	0.11(0.44)
	4	0.47(0.7)	0.56(0.75)	0.66(0.81)	0.13(0.36)	0.11(0.44)	0.12(0.37)
	5	0.43(0.67)	0.56(0.72)	0.66(0.75)	0.12(0.42)	0.11(0.43)	0.11(0.42)
	6	0.42(0.64)	0.57(0.74)	0.46(0.65)	0.11(0.41)	0.11(0.43)	0.1(0.41)
	7	0.44(0.64)	0.55(0.68)	0.42(0.63)	0.11(0.41)	0.11(0.43)	0.1(0.41)
	8	0.44(0.63)	0.53(0.71)	0.4(0.6)	0.11(0.41)	0.11(0.43)	0.1(0.41)
Left	1	0.64(0.79)	0.54(0.79)	0.61(0.82)	0.13(0.5)	0.13(0.49)	0.13(0.51)
	2	0.63(0.81)	0.58(0.8)	0.58(0.77)	0.13(0.48)	0.14(0.48)	0.13(0.49)
	3	0.62(0.76)	0.61(0.79)	0.57(0.78)	0.13(0.46)	0.13(0.46)	0.13(0.47)
	4	0.6(0.75)	0.54(0.75)	0.57(0.77)	0.12(0.46)	0.13(0.45)	0.12(0.46)
	5	0.6(0.75)	0.59(0.78)	0.53(0.75)	0.12(0.45)	0.12(0.45)	0.12(0.45)
	6	0.56(0.74)	0.59(0.78)	0.58(0.78)	0.12(0.41)	0.12(0.44)	0.12(0.42)
	7	0.55(0.74)	0.59(0.79)	0.6(0.75)	0.12(0.44)	0.12(0.44)	0.12(0.44)
	8	0.56(0.7)	0.54(0.73)	0.53(0.72)	0.11(0.44)	0.11(0.44)	0.11(0.44)
Right	1	0.37(0.67)	0.5(0.74)	0.57(0.74)	0.09(0.44)	0.1(0.43)	0.1(0.44)
	2	0.39(0.71)	0.51(0.69)	0.32(0.69)	0.1(0.44)	0.1(0.43)	0.09(0.44)
	3	0.36(0.69)	0.51(0.75)	0.32(0.68)	0.1(0.44)	0.1(0.43)	0.09(0.43)
	4	0.38(0.7)	0.55(0.74)	0.34(0.69)	0.1(0.43)	0.1(0.42)	0.09(0.43)
	5	0.38(0.68)	0.58(0.76)	0.36(0.69)	0.1(0.43)	0.11(0.42)	0.09(0.43)
	6	0.38(0.68)	0.57(0.7)	0.51(0.72)	0.11(0.42)	0.11(0.4)	0.1(0.42)
	7	0.38(0.67)	0.51(0.67)	0.53(0.69)	0.11(0.36)	0.12(0.35)	0.11(0.4)
	8	0.39(0.67)	0.49(0.67)	0.49(0.69)	0.1(0.36)	0.1(0.42)	0.09(0.35)
Baseline		**0.273(0.87)**			**0.12(0.5)**		

(a) Sequential Order of Words Discarded

$NAP_{i=98}$			$NAP_{i=4278}$		
k			k		
1	7	25	1	7	25
0.37(0.69)	0.5(0.76)	0.58(0.76)	0.1(0.45)	0.1(0.45)	0.1(0.45)
0.4(0.69)	0.5(0.77)	0.58(0.76)	0.1(0.44)	0.1(0.44)	0.1(0.45)
0.43(0.68)	0.53(0.78)	0.58(0.76)	0.1(0.44)	0.11(0.44)	0.11(0.44)
0.48(0.72)	0.53(0.76)	0.6(0.75)	0.11(0.44)	0.11(0.43)	0.11(0.44)
0.48(0.71)	0.52(0.76)	0.59(0.75)	0.11(0.43)	0.11(0.43)	0.12(0.44)
0.52(0.7)	0.59(0.76)	0.59(0.76)	0.11(0.34)	0.11(0.43)	0.11(0.34)
0.55(0.72)	0.63(0.75)	0.59(0.76)	0.12(0.37)	0.11(0.43)	0.13(0.36)
0.57(0.75)	0.67(0.74)	0.6(0.76)	0.12(0.39)	0.11(0.42)	0.13(0.38)
0.64(0.79)	0.54(0.79)	0.61(0.82)	0.13(0.5)	0.13(0.49)	0.13(0.51)
0.58(0.71)	0.67(0.78)	0.39(0.7)	0.11(0.37)	0.11(0.33)	0.1(0.35)
0.59(0.78)	0.59(0.79)	0.59(0.71)	0.12(0.41)	0.13(0.39)	0.11(0.38)
0.56(0.75)	0.64(0.75)	0.65(0.77)	0.13(0.43)	0.14(0.42)	0.13(0.41)
0.57(0.72)	0.69(0.76)	0.69(0.82)	0.13(0.42)	0.13(0.41)	0.13(0.42)
0.53(0.68)	0.66(0.73)	0.67(0.8)	0.13(0.4)	0.12(0.39)	0.12(0.42)
0.58(0.7)	0.51(0.58)	0.6(0.67)	0.12(0.42)	0.12(0.41)	0.12(0.41)
0.48(0.69)	0.51(0.58)	0.57(0.69)	0.11(0.42)	0.12(0.41)	0.11(0.41)
0.37(0.67)	0.5(0.74)	0.57(0.74)	0.09(0.44)	0.1(0.43)	0.1(0.44)
0.39(0.68)	0.53(0.77)	0.58(0.76)	0.1(0.44)	0.1(0.44)	0.1(0.45)
0.42(0.71)	0.54(0.77)	0.59(0.76)	0.1(0.44)	0.1(0.44)	0.11(0.44)
0.44(0.71)	0.54(0.77)	0.59(0.75)	0.11(0.44)	0.11(0.44)	0.11(0.44)
0.47(0.73)	0.52(0.76)	0.6(0.76)	0.11(0.44)	0.11(0.43)	0.11(0.44)
0.49(0.73)	0.53(0.77)	0.6(0.76)	0.11(0.43)	0.11(0.43)	0.11(0.44)
0.52(0.75)	0.56(0.77)	0.62(0.77)	0.11(0.43)	0.11(0.43)	0.11(0.44)
0.52(0.75)	0.56(0.76)	0.61(0.76)	0.11(0.42)	0.11(0.43)	0.12(0.42)
0.27(0.87)			**0.12(0.5)**		

(b) Sequential Order of Words Encoded

Table 5.17: The results observed in $\{\mathrm{T}\}^{\mathrm{YATEA}}_{\mathrm{Enlarged}}$ when $|R_s| = 100$ and similarities are computed using the *Euclidean* distance.

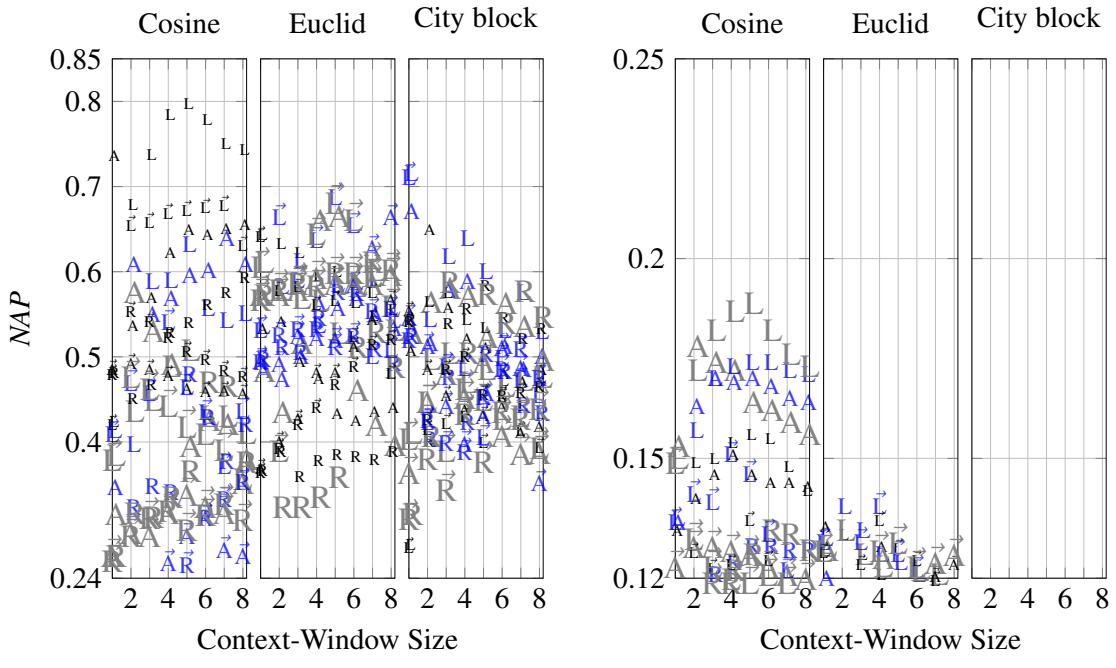
Context dir	size	$NAP_{i=98}$ k=1	$NAP_{i=98}$ k=7	$NAP_{i=98}$ k=25	$NAP_{i=4278}$ k=1	$NAP_{i=4278}$ k=7	$NAP_{i=4278}$ k=25
Around	1	0.51(0.69)	0.67(0.8)	0.53(0.74)	0.1(0.36)	0.1(0.34)	0.09(0.35)
	2	0.65(0.77)	0.52(0.67)	0.56(0.73)	0.1(0.36)	0.09(0.31)	0.1(0.35)
	3	0.56(0.69)	0.58(0.71)	0.58(0.72)	0.1(0.36)	0.1(0.32)	0.1(0.35)
	4	0.52(0.7)	0.59(0.7)	0.57(0.7)	0.1(0.37)	0.1(0.34)	0.1(0.36)
	5	0.51(0.67)	0.46(0.59)	0.52(0.68)	0.1(0.37)	0.1(0.35)	0.1(0.36)
	6	0.48(0.66)	0.51(0.6)	0.56(0.71)	0.1(0.35)	0.1(0.36)	0.1(0.33)
	7	0.47(0.64)	0.49(0.61)	0.54(0.71)	0.11(0.39)	0.1(0.35)	0.11(0.38)
	8	0.49(0.68)	0.5(0.61)	0.5(0.7)	0.11(0.38)	0.1(0.36)	0.11(0.36)
Left	1	0.28(0.21)	0.72(0.85)	0.41(0.72)	0.06(0.26)	0.09(0.32)	0.07(0.29)
	2	0.57(0.71)	0.55(0.74)	0.5(0.71)	0.1(0.36)	0.1(0.35)	0.09(0.32)
	3	0.58(0.73)	0.62(0.73)	0.5(0.7)	0.1(0.35)	0.11(0.33)	0.09(0.32)
	4	0.54(0.71)	0.64(0.74)	0.47(0.7)	0.1(0.35)	0.11(0.33)	0.09(0.33)
	5	0.53(0.7)	0.6(0.73)	0.49(0.68)	0.1(0.35)	0.11(0.34)	0.1(0.34)
	6	0.47(0.64)	0.53(0.7)	0.47(0.64)	0.1(0.34)	0.1(0.33)	0.09(0.34)
	7	0.44(0.63)	0.58(0.7)	0.47(0.64)	0.1(0.36)	0.1(0.33)	0.1(0.35)
	8	0.48(0.67)	0.53(0.64)	0.48(0.62)	0.1(0.36)	0.1(0.32)	0.1(0.35)
Right	1	0.54(0.7)	0.52(0.65)	0.32(0.57)	0.09(0.35)	0.09(0.35)	0.07(0.33)
	2	0.4(0.54)	0.4(0.51)	0.41(0.54)	0.08(0.33)	0.08(0.33)	0.08(0.31)
	3	0.54(0.64)	0.48(0.58)	0.59(0.65)	0.1(0.34)	0.09(0.33)	0.09(0.32)
	4	0.56(0.71)	0.42(0.6)	0.51(0.71)	0.1(0.33)	0.1(0.33)	0.09(0.31)
	5	0.58(0.73)	0.5(0.63)	0.57(0.65)	0.1(0.33)	0.1(0.33)	0.09(0.31)
	6	0.55(0.71)	0.48(0.63)	0.54(0.64)	0.1(0.33)	0.09(0.32)	0.09(0.31)
	7	0.52(0.7)	0.44(0.62)	0.58(0.69)	0.1(0.33)	0.09(0.33)	0.09(0.32)
	8	0.53(0.71)	0.43(0.61)	0.55(0.66)	0.1(0.34)	0.1(0.34)	0.09(0.32)
Baseline		**0.273(0.87)**			**0.12(0.5)**		

(a) Sequential Order of Words Discarded

$NAP_{i=98}$ k=1	$NAP_{i=98}$ k=7	$NAP_{i=98}$ k=25	$NAP_{i=4278}$ k=1	$NAP_{i=4278}$ k=7	$NAP_{i=4278}$ k=25
0.56(0.77)	0.55(0.7)	0.37(0.7)	0.1(0.37)	0.08(0.32)	0.08(0.36)
0.49(0.71)	0.52(0.67)	0.45(0.73)	0.09(0.34)	0.09(0.32)	0.09(0.33)
0.46(0.7)	0.45(0.54)	0.49(0.71)	0.09(0.33)	0.09(0.32)	0.09(0.31)
0.44(0.61)	0.4(0.55)	0.46(0.64)	0.09(0.32)	0.09(0.31)	0.09(0.31)
0.43(0.65)	0.43(0.62)	0.5(0.66)	0.09(0.31)	0.09(0.31)	0.09(0.3)
0.45(0.65)	0.51(0.66)	0.41(0.6)	0.08(0.3)	0.09(0.31)	0.08(0.3)
0.41(0.63)	0.49(0.66)	0.39(0.58)	0.08(0.31)	0.09(0.32)	0.08(0.31)
0.42(0.59)	0.36(0.53)	0.39(0.55)	0.08(0.31)	0.08(0.31)	0.08(0.3)
0.28(0.21)	0.72(0.85)	0.41(0.72)	0.06(0.26)	0.09(0.32)	0.07(0.29)
0.42(0.64)	0.43(0.6)	0.45(0.67)	0.09(0.35)	0.09(0.29)	0.09(0.35)
0.42(0.66)	0.49(0.73)	0.4(0.51)	0.09(0.31)	0.09(0.3)	0.08(0.3)
0.43(0.69)	0.49(0.67)	0.47(0.68)	0.1(0.33)	0.09(0.33)	0.09(0.31)
0.4(0.59)	0.41(0.51)	0.46(0.68)	0.09(0.31)	0.08(0.3)	0.09(0.3)
0.46(0.68)	0.5(0.61)	0.43(0.66)	0.09(0.31)	0.09(0.3)	0.09(0.31)
0.42(0.68)	0.48(0.65)	0.44(0.69)	0.09(0.32)	0.09(0.31)	0.09(0.32)
0.4(0.62)	0.46(0.64)	0.4(0.66)	0.09(0.32)	0.09(0.31)	0.09(0.32)
0.54(0.7)	0.52(0.65)	0.32(0.57)	0.09(0.35)	0.09(0.35)	0.07(0.33)
0.43(0.66)	0.43(0.62)	0.39(0.51)	0.09(0.37)	0.09(0.38)	0.08(0.3)
0.49(0.64)	0.4(0.62)	0.35(0.46)	0.08(0.3)	0.08(0.31)	0.08(0.3)
0.5(0.63)	0.39(0.61)	0.44(0.5)	0.09(0.32)	0.09(0.34)	0.08(0.3)
0.46(0.6)	0.46(0.61)	0.38(0.59)	0.09(0.35)	0.09(0.34)	0.09(0.32)
0.46(0.62)	0.49(0.64)	0.44(0.58)	0.08(0.31)	0.09(0.31)	0.08(0.31)
0.46(0.62)	0.51(0.63)	0.43(0.57)	0.08(0.31)	0.08(0.3)	0.08(0.3)
0.47(0.62)	0.48(0.6)	0.44(0.6)	0.09(0.31)	0.09(0.31)	0.08(0.31)
0.27(0.87)			**0.12(0.5)**		

(b) Sequential Order of Words Encoded

Table 5.18: The results obtained in $\{\mathrm{T}\}^{\mathrm{YATEA}}_{\mathrm{Enlarged}}$ when $|R_s| = 100$ and similarities are computed using the *city block* distance.



(a) $NAP_{i=98}$ (b) $NAP_{i=4278}$

Figure 5.19: The NAP_i observed in $\{\mathrm{T}\}^{\mathrm{YATEA}}_{\mathrm{Enlarged}}$ for $i = 98$ (i.e., 2% recall) and $i = 4278$ (i.e., recall 100%) are shown in (a) and (b), respectively.

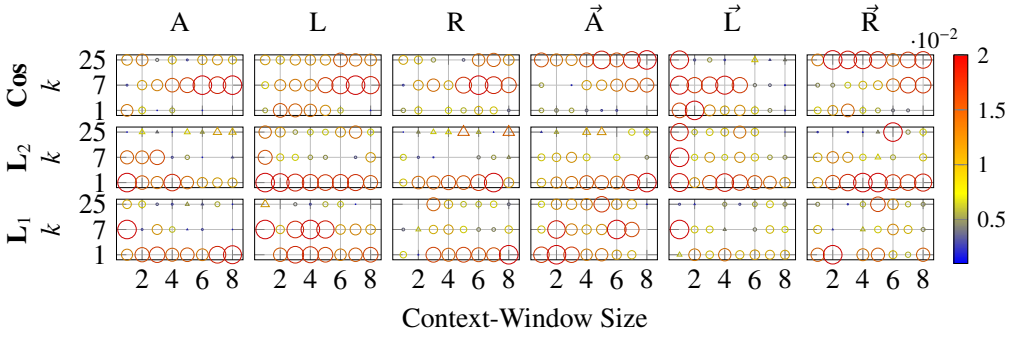


Figure 5.20: Changes in the performance of the method caused by increasing the size of the input corpus when the performance is measured using *NAP* for 100% recall in the experiments over $\{\mathrm{T}\}^{\mathrm{Y_ATE_A}}_{\mathrm{Enlarged}}$. Shown is the absolute value of the difference between the performance obtained from a model constructed in $\{\mathrm{T}\}^{\mathrm{Y_ATE_A}}_{\mathrm{Y_ATE_A}}$ and the corresponding model in $\{\mathrm{T}\}^{\mathrm{Y_ATE_A}}_{\mathrm{Enlarged}}$. Triangles denote negative change whereas circles show positive change. The size/colour of shapes represents the amount of changes. The *x*-axis shows various configurations of context-windows. The *y*-axis represents classification parameters.

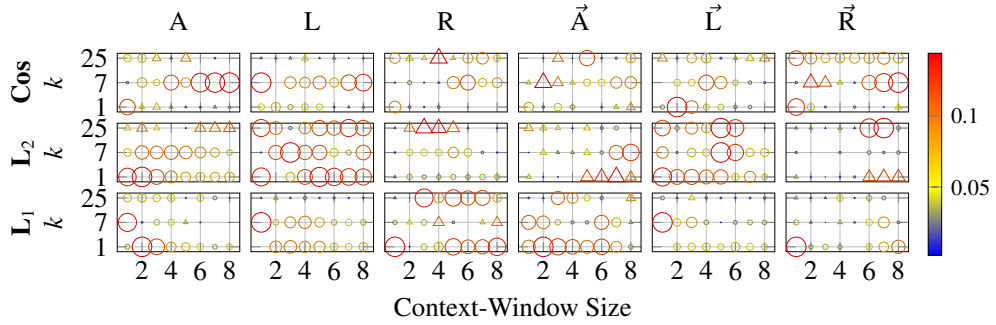


Figure 5.21: The changes in the performance caused by increasing the size of the input corpus when the method's performance is measured using $NAP_{i=200}$ (i.e, 2% recall) over $\{\mathrm{T}\}^{\mathrm{Y_ATE_A}}_{\mathrm{Enlarged}}$. The presentation format is similar to Figure 5.20: circles show positive effects whereas triangles show negative impacts.

when similarities are computed using the Euclidean and the city block distance. Similarly, Figure 5.21 shows how enlarging the corpus effects the observed performances at a small recall point such as $NAP_{i=98}$ (i.e., recall %2). As shown, compared to the experiments over $\{\mathrm{T}\}^{\textit{c-value}}_{\mathrm{Enlarged}}$ (see Figure 5.16), enlarging the corpus has a more steady positive effect on the discriminatory power of the constructed models when the classification task is accomplished for a set of candidate terms that contain invalid terms.

With an exception to the size of context-windows, parameters that gives in the best performance in $\{\mathrm{T}\}^{\mathrm{Y_ATE_A}}_{\mathrm{Y_ATE_A}}$ also results the best performance in $\{\mathrm{T}\}^{\mathrm{Y_ATE_A}}_{\mathrm{Enlarged}}$. In both $\{\mathrm{T}\}^{\mathrm{Y_ATE_A}}_{\mathrm{Y_ATE_A}}$ and $\{\mathrm{T}\}^{\mathrm{Y_ATE_A}}_{\mathrm{Enlarged}}$, the most discriminative models are built using context-windows that extend to the left side of candidate terms. In contrast to experiments over $\{\mathrm{T}\}^{\mathrm{Y_ATE_A}}_{\mathrm{Y_ATE_A}}$, in the experi-

ments over $\{T\}^{\text{YATEA}}_{\text{Enlarged}}$ extending context-windows more than 5 tokens often diminishes the performance. The cosine metric on average shows the best performance; in this case, a small k results in the best performance at small recall points whereas large k must be chosen at large recall points (e.g., $k = 1$ at 2% recall and $k = 25$ at 100% recall, respectively).

5.4.4 Evaluating Parameters Across Concept Categories

Terms are often classified in several categories of concepts; therefore, the identification of co-hyponym terms can go beyond one concept category. For instance, in the domain of *molecular biology* (and accordingly in the GENIA corpus), several categories of terms (e.g., *cell line*, *cell type*, etc.) other than *protein* are conceived. The question here is that whether the same configuration of the context-window and the classification's parameters can be used for identify terms from different concept categories. That is to say, if a model shows the best performance for identifying a category of terms such as *protein*, would it be also the top performer for extracting terms that belong to other categories such as , *cell line*, and *cell type*?

To answer the questions asked above, the reported evaluation in the Section 5.4.1 over $\{T\}^{\text{c-}value}_{\text{ideal}}$ are repeated; however, for identifying terms that are classified under the concept category of *cell type* and *cell line* (i.e., terms that are annotated as `G#cell_type` and `G#cell_line` in the GENIA corpus, respectively). Table 5.19 shows statistics for these two categories of term in the corpus. Similar to the description given in Section 5.3 for protein terms, terms that are annotated at least once as cell type or cell line are collected from the corpus. Those terms that are annotated in one additional category are marked as polysemous.

Accordingly, when extracting terms that belong to the concept category of cell type and cell line, the random baseline approaches to $\frac{2097}{34077} = 0.061$ and $\frac{2261}{34077} = 0.066$, respectively. Figure 5.22 also shows the c-*value* ranking baselines (i.e., the baselines computed using the set of ranked terms in $\{T\}^{\text{c-}value}_{\text{ideal}}$). As shown, at the small recall point of 2% (i.e., $NAP_{i=42}$ for *cell type* and $NAP_{i=45}$ for *cell line*), the c-*value* ranking baseline is 0.255 and 0.134 for the cell type and cell line categories, respectively. However, at 100% recall (i.e., *NAP* at $i = 2097$ and $i = 2261$ for terms in the category of cell type and cell line, respectively), the performance of the c-*value* baseline is similar to the random baseline. Since terms that belong to the category of cell line are less frequent than cell type, they are given lower ranks by the c-*value* measure. As a result, although the number of distinct terms annotated as cell line is larger than cell type, the computed c-*value* baseline for cell line is less than cell type.

A classification process identical to the one employed for identifying protein terms in Section 5.4.1 is carried out for extracting terms that belong to the category of cell type

Category	Frequency (mentions)	#Distinct Entry	#Polysemous Entry
Cell Type	8,257	2,097	178
Cell Line	5,944	2,261	154

Table 5.19: Shown are the statistics of the co-hyponym terms in the two categories of *cell type* and *virus* in the GENIA corpus. Polysemous entries are subset of distinct entries.

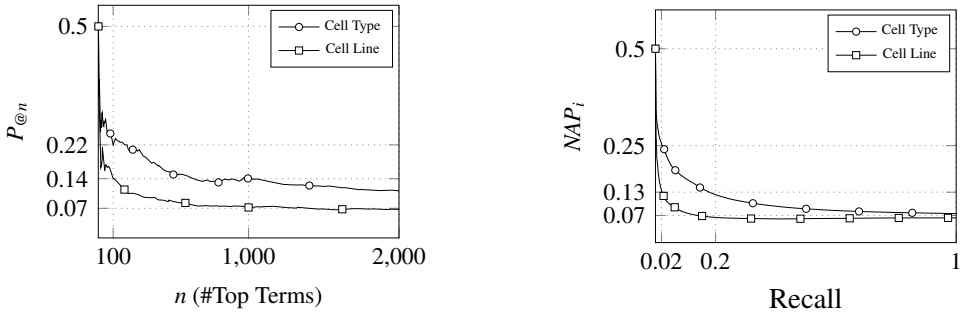
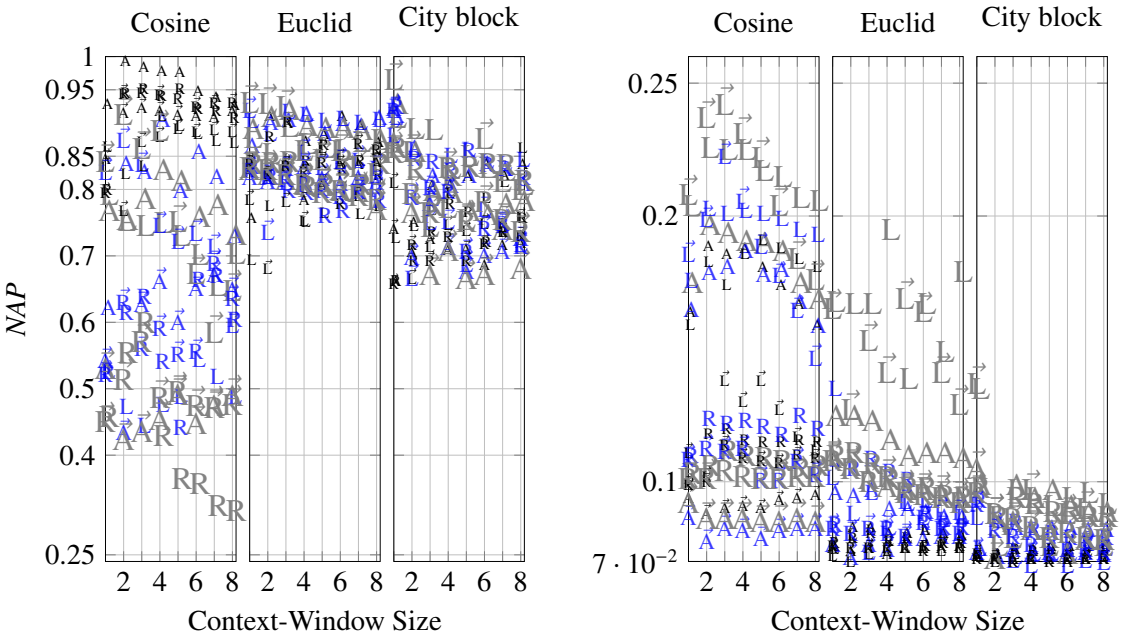


Figure 5.22: Baseline performances when extracting terms from the categories of *cell type* and *cell line* in $\{T\}^{\text{c-}value}_{\text{ideal}}$ ranked terms: (a) shows the proportion of terms in these categories in the top 2000 entries in $\{T\}^{\text{c-}value}_{\text{ideal}}$ (i.e., $P_{@n}$ for $1 \leq n \leq 2000$); (b) shows the performance computed by NAP_i for the complete range of recall values—that is, $i = 2,097$ and $i = 2261$ for terms in the category of *cell type* and *cell line*, respectively.

and cell line. As shown in Figure 5.22a, the employed R_s (i.e., the top 100 c-*value* ranked terms) contains 14 terms from the cell line category and 22 terms from the cell type category. The obtained results are plotted in Figures 5.23 and 5.24.

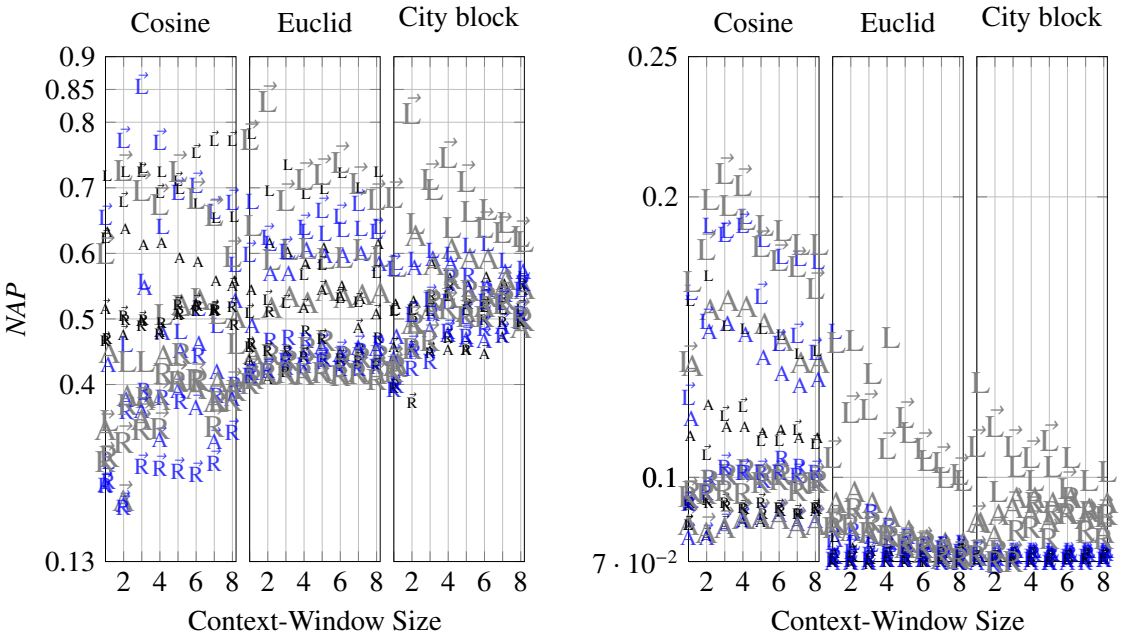
As an initial inspection of the results shows, answering the questions asked above is not straight forward, particularly, at small recall points. Assuming that the method's parameters are fixed, then the performance appears to be sensitive to the chosen targeted category of concepts. That is to say, to obtain the best performances for identifying each category of co-hyponyms terms, often context-windows must be reconfigured with respect to the evaluated parameters. For instance, at 2% recall, if similarities are computed using the cosine measure, then context-windows that extend *around* the terms shows the best performance for identifying *cell type* terms (Figure 5.23a). However, under the similar conditions, context-windows that extend to the *left* side of terms shows the best performance for identifying *cell line* terms (Figure 5.24a).

When it comes to the choice of choice of k in the classification process, a similar conclusion as to the parameters of context-windows can be drawn, too. For different categories of concepts, the best performances are obtained using different values of k (e.g., $k = 1$ in Figure 5.23a vs. $= 7$ in Figure 5.24a). However, concerning the choice of similarity metric, the observations are predominantly comparable across categories of concepts. Except for the small recall values, the cosine measure outperforms the other



(a) $\boldsymbol{NAP_{i=42}}$ (b) $\boldsymbol{NAP_{i=2097}}$

Figure 5.23: Performance over$\{\mathrm{T}\}^{\textit{c-value}}_{\text{ideal}}$ for extracting terms from the *cell type* category. The notation is similar to previous figures: letters show the direction in which context-windows are extended; their size/colour denote the value of k, and the $\vec{\square}$ implies encoding information about words order. The y-axis's minimum value shows the c-*value* baseline.



(a) $\boldsymbol{NAP_{i=45}}$ (b) $\boldsymbol{NAP_{i=2261}}$

Figure 5.24: Performance over $\{\mathrm{T}\}^{\textit{c-value}}_{\text{ideal}}$ for extracting terms from the category of *cell line*. The y-axis's minimum value shows the c-*value* baseline.

evaluated metrics (see discussions related to Figures 5.10, 5.12, and 5.17).

The experiments are also repeated over the enlarged corpus—that is, $\{T\}^{c\text{-}value}_{Enlarged}$. The obtained performances are abridged in Figure 5.25 and 5.26, which also corroborate the conclusions drawn above.

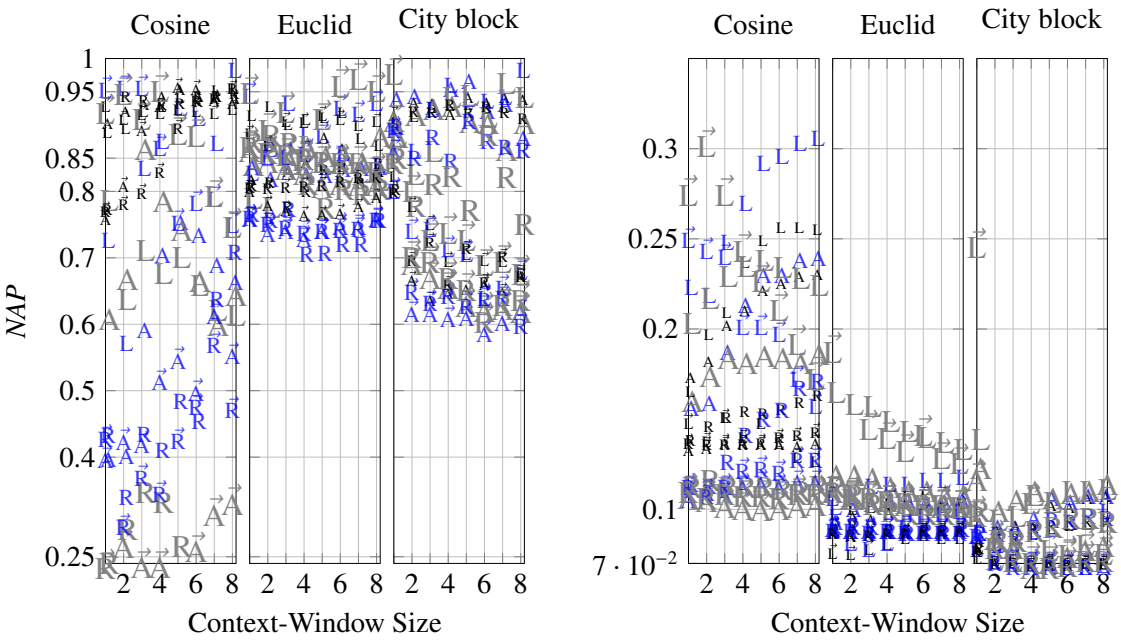
A comparison between the results that are plotted earlier in Figures 5.23 and 5.24, and the results reported in Figures 5.25 and 5.26 (i.e., comparing the method's performance in $\{T\}^{c\text{-}value}_{ideal}$ and $\{T\}^{c\text{-}value}_{Enlarged}$) leads to a discussion similar to the one proposed in Section 5.4.3: enlarging the corpus does not necessarily enhance the observed performances. Taking the results reported throughout this section into consideration, it becomes evident that the effect of enlarging the corpus not only depends on the configuration of context-windows and the chosen values for the classification's parameters (as suggested in Section 5.4.3), but also on the targeted category of concepts. For instance, when the method's performance is investigated at 100% recall and the cosine measure is employed to compute similarities, enlarging the corpus has a positive effect on the performance when extracting terms that belong to the category of cell type (compare Figures 5.23b and 5.25b). However, under the same conditions, the result is the opposite when extracting cell line terms—that is, a decrease in the performance is observed (compare the cosine section of Figures 5.24b and 5.26b).

5.4.5 Averaging Performances Across Concept Categories

The construction of a vector space model and configuring it for a particular category of concepts would result in the best possible performance, as shown in the previous section. However, this practice cannot be feasible for a few reasons. The construction of a model, even with a reduced dimensionality, demands computational resources that may not be available in order to construct a model for category of concepts. It is therefore likely that a single model is employed to identify a variety of co-hyponymy relationships in an application. In the context of this chapter, for example, a single model could be used to identify terms from the categories of protein, cell type, and cell line. One way to choose a configuration for this model is to use the average of performances across the categories.

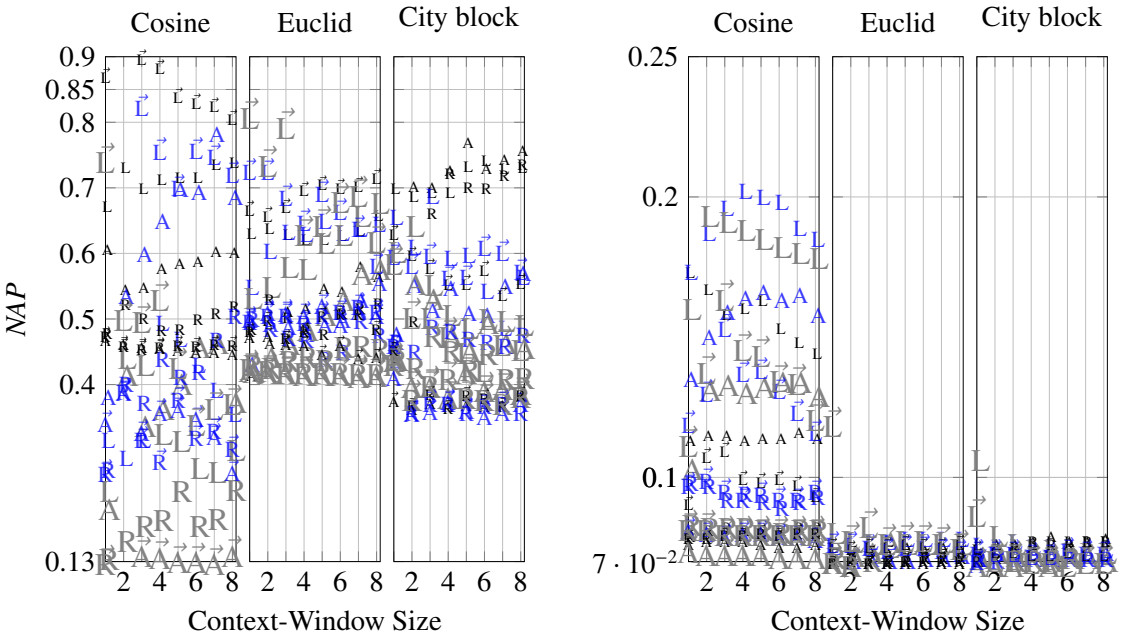
In this section, the average performance of the method across the concept categories of protein, cell type, and cell line are reported when the parameters of the context-window and the classification process are set differently. To do so, similar to the evaluation of an information retrieval system in a task that involves a set of queries (e.g., as suggested by Manning et al., 2009, chap. 8), the average of the arithmetic mean of recorded non-interpolated average precisions (i.e., NAP_i) is employed as a single-figure measure of the method's performance across categories of concepts. This arithmetic mean average of performances (MAP) is simply the sum of the observed NAP_i for the three aforementioned categories of terms divided by the number of categories (i.e., 3 in here).

For each of the evaluated datasets, the observed MAP is reported for the two re-



(a) $NAP_{i=42}$ (b) $NAP_{i=2097}$

Figure 5.25: The method's performance over $\{T\}^{c\text{-}value}_{Enlarged}$ for extracting terms in the category of *cell type*.



(a) $NAP_{i=45}$ (b) $NAP_{i=2261}$

Figure 5.26: The method's performance over $\{T\}^{c\text{-}value}_{Enlarged}$ for identifying terms in the category of *cell line*.

call points of 2% and 100%. Figures 5.27 and 5.28 show the results over the $\{T\}^{\textit{c-value}}_{\text{ideal}}$ and $\{T\}^{\textit{c-value}}_{\text{Enlarged}}$, respectively. Similarly, the computed *MAP* over $\{T\}^{\text{YATEA}}_{\text{YATEA}}$ and $\{T\}^{\text{YATEA}}_{\text{Enlarged}}$ are, respectively, plotted in Figures 5.29 and 5.30. In these figures, the baselines are the average of the performances obtained when using the c-*value* and YATEA based rankings for extracting the aforementioned categories of terms. For example, if the set of annotated terms in the GENIA corpus that are ranked by the c-*value* measure (i.e., $\{T\}^{\textit{c-value}}_{\text{ideal}}$), the computed NAP_i at 2% for the three categories of protein, cell line, and cell type are 0.37, 0.14, and 0.25, respectively. The mean average baseline is thus the sum of these numbers divided by three, which is ≈ 0.25 as reported in Figures 5.27a and 5.28a.

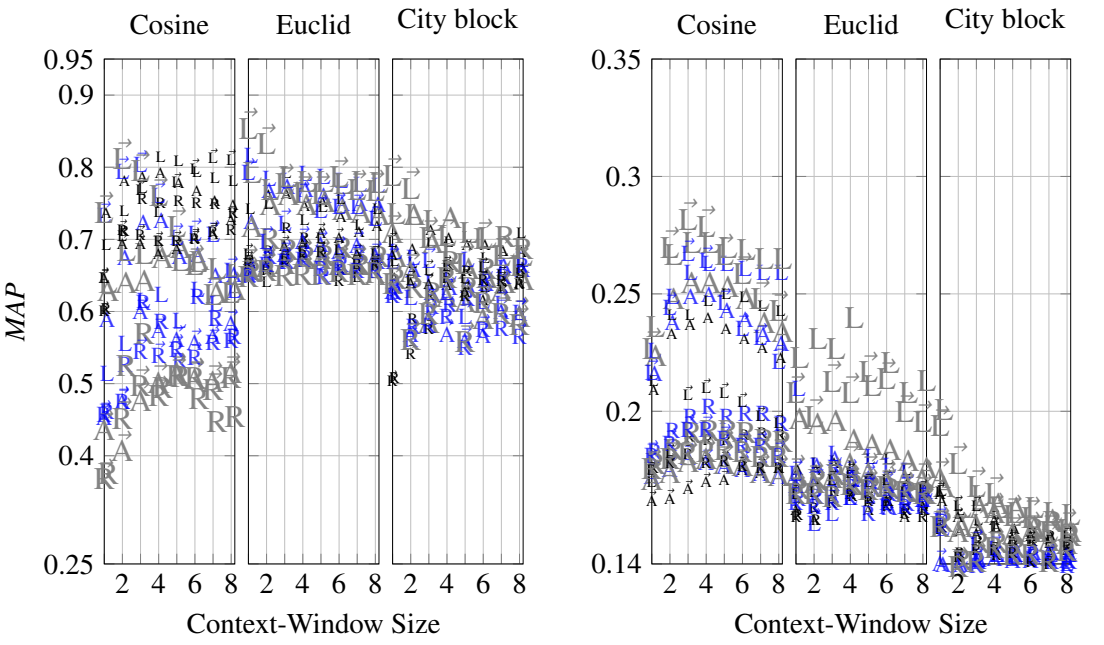
A series of discussions can follow the comparison of the plotted results in Figures 5.27, 5.28, 5.29, and 5.30, similar to the approach employed in the previous sections. Evidently, depending on factors such as the targeted recall point and the characteristics of the corpus, the method's parameters can be tuned differently to obtain the best-averaged performances.

5.5 Discussion

In Section 5.4, the use of the proposed distributional method for finding co-hyponym terms using a memory-based classification technique is investigated through a set of empirical experiments. Firstly, the results from these experiments allow one to accept the proposed hypothesis—that is, terms from a similar category of concepts appear in similar context, and that can be used for developing a distributional method for identifying co-hyponym terms. It is shown that with a small number of annotated reference terms (i.e., $|R_s| = 100$) and in the absence of sufficient training data for developing an entity tagger (i.e., as shown in Section 5.4.1.1), automatically constructed vector space models with reduced dimensionality can be used to address the proposed task with an acceptable performance (i.e., well above a general term recognition baseline, an entity tagger, and a random baseline). The result is satisfactory, particularly when the little amount of manual effort for developing a model is taken into consideration.

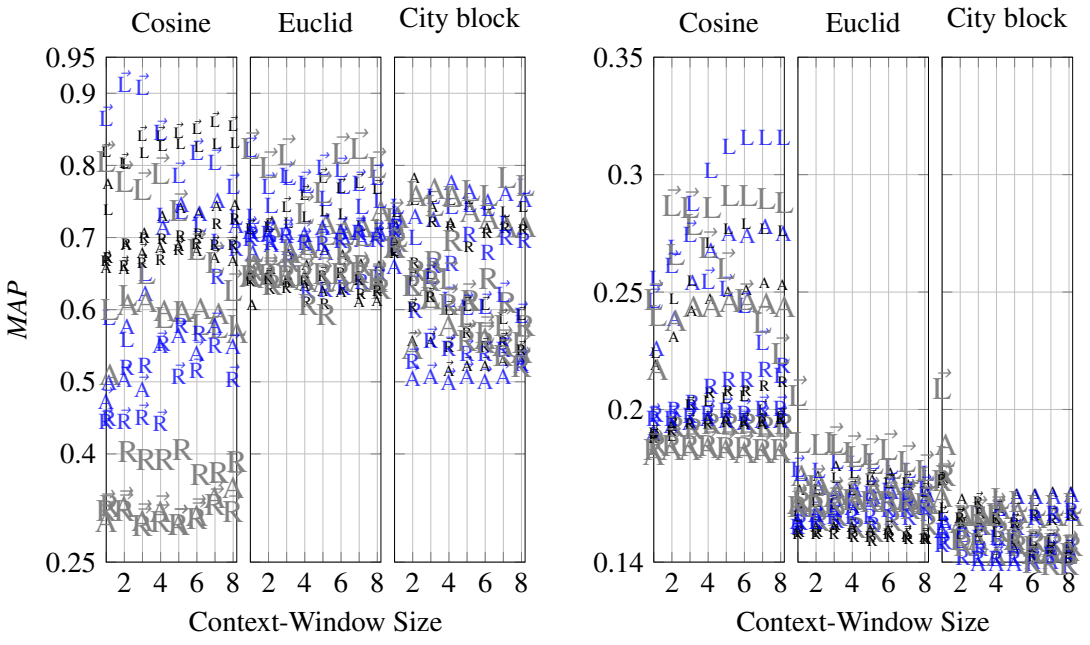
To address research questions proposed in Chapter 1 (Section 1.4), experiments are designed and carried out over the *Cartesian* product of a set of values for configuring the parameters of the context-widow (i.e., to address RQ 1.1, 1.2, and 1.3) and the classification framework (i.e., RQ 2.1 and 2.2). To cover the remaining research questions, these experiments are repeated over several sets of candidate terms, and in corpora of two different sizes (i.e., to investigate RQ 3), in order to extract terms from various categories of concepts (i.e., in pursuing RQ 4). The non-interpolated average precisions at two recall points (2% and 100%) are reported as the figure of merit.

To address research questions about the configuration of context-windows, sev-



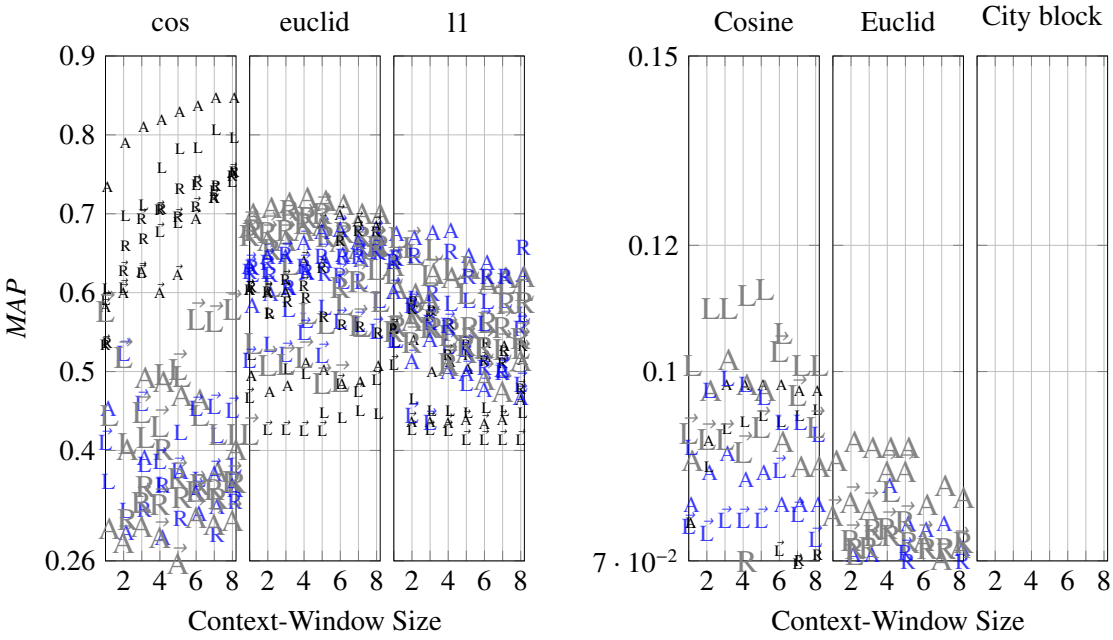
(a) $\boldsymbol{MAP_{2\%}}$ (b) $\boldsymbol{MAP_{100\%}}$

Figure 5.27: The mean average performance (i.e., *y*-axis) across concept categories observed in experiments over $\{\text{T}\}^{\textit{c-value}}_{\text{ideal}}$ using $|R_s| = 100$. The presentation format is similar to Figure 5.8: the letters show the direction in which the context-window is extended to collected co-occurrence frequency; their size (colour) denote the value of k; and the presence of $\vec{\square}$ on top of them indicates encoding information about the word order information.



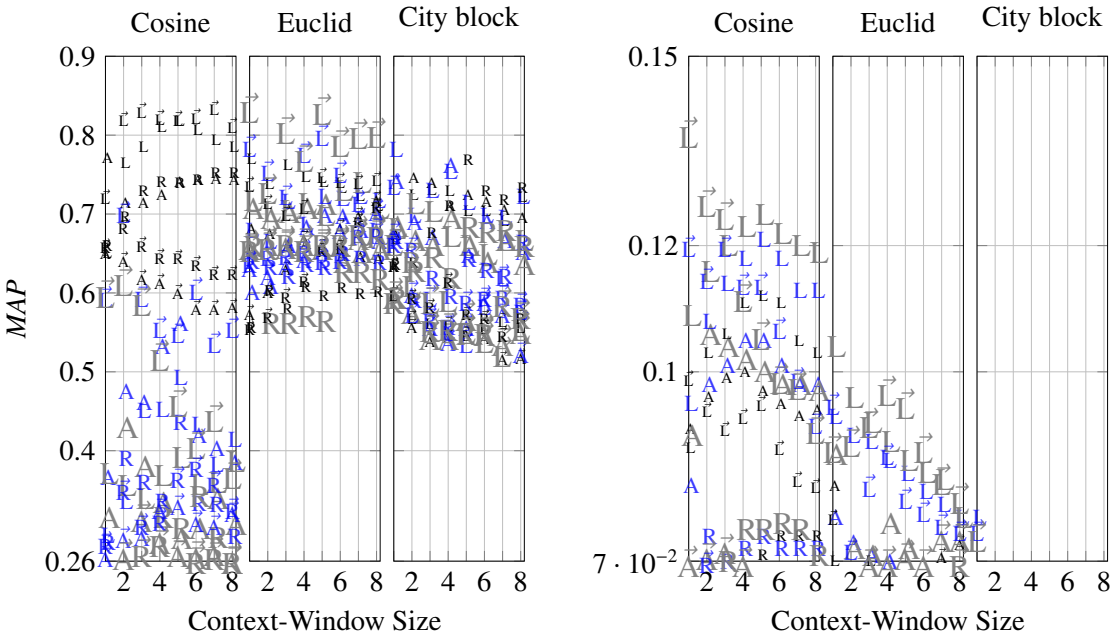
(a) $\boldsymbol{MAP_{2\%}}$ (b) $\boldsymbol{MAP_{100\%}}$

Figure 5.28: The mean average performance across concept categories observed in experiments over $\{\text{T}\}^{\textit{c-value}}_{\text{Enlarged}}$ using $|R_s| = 100$.



(a) $\boldsymbol{MAP_{2\%}}$ (b) $\boldsymbol{MAP_{100\%}}$

Figure 5.29: The mean average performance across concept categories observed in experiments over $\{\mathrm{T}\}^{\mathrm{YATEA}}_{\mathrm{YATEA}}$ using $|R_s| = 100$. At 100% recall, if the city block distance is employed to compute similarities, the method underperforms the computed baseline.



(a) $\boldsymbol{MAP_{2\%}}$ (b) $\boldsymbol{MAP_{100\%}}$

Figure 5.30: The mean average performance across concept categories observed in experiments over $\{\mathrm{T}\}^{\mathrm{YATEA}}_{\mathrm{Enlarged}}$ using $|R_s| = 100$. Similar to Figure 5.29, at 100% recall, the use of city block distance results in performances below the baseline.

eral models are constructed when the context-windows are extended to three different *directions*: only to the left, only to the right, and in both directions around the candidate terms (see RQ 1.1); with variable *sizes* of $1 \leq t \leq 8$ tokens (see RQ 1.2); and when the *sequential order* of words in the context-windows are encoded and neglected (see RQ 1.3). Hence, 48 different models are constructed for each set of candidate terms and each corpus employed in the experiments. To address questions about the parameters of the similarity-based reasoning framework, the weighting process is carried out using three *similarity metrics*: the city block distance, the cosine measure, and the Euclidean distance (see RQ 2.1). This is done for three different values of the *neighbourhood size* k (see RQ 2.2)—therefore, the categorisation process is repeated for $k \in 1, 7, 25$.

In Section 5.4.1, the experiments begin with the evaluation of the method for identifying terms from the category of *proteins* in the constructed terminological resource from the GENIA corpus and using this corpus for collecting the co-occurrence frequencies (i.e., $\{\mathrm{T}\}^{\textit{c-value}}_{\text{ideal}}$, which is free from invalid candidate terms). Accordingly, the method's performances are obtained when it is configured using the aforementioned values for its parameters. In Section 5.4.2, the experiments are repeated in the same corpus, however, using a set of candidate terms that are extracted using a state-of-the-art term extractor system (i.e., $\{\mathrm{T}\}^{\mathrm{YATEA}}_{\mathrm{YATEA}}$, which contains invalid candidate terms).[1] In experiments that are performed over both $\{\mathrm{T}\}^{\textit{c-value}}_{\text{ideal}}$ and $\{\mathrm{T}\}^{\mathrm{YATEA}}_{\mathrm{YATEA}}$, it is observed that choosing the best performing configuration is largely dependant on the recall value that is targeted.

While it is not possible to choose a one best value for the size of context-windows, it is verified that extending the context-windows to more than 5 tokens does not improve the computed performances, particularly for large recall values. With respect to the direction in which context-windows are extended to collect co-occurrences, the conclusion is similar: depending on the employed similarity metric and the targeted recall value, the best performing models are constructed when they are stretched in different directions. However, more than often, context-windows extended to the left of candidate terms outperform context-windows that are stretched in the other directions. However, in experiments over $\{\mathrm{T}\}^{\mathrm{YATEA}}_{\mathrm{YATEA}}$, specially when using the cosine measure, the context-windows that extend around the candidate terms can outperform those that extend only to the left. As discussed in Section 5.4.2, one explanation for this observation is that invalid terms in $\{\mathrm{T}\}^{\mathrm{YATEA}}_{\mathrm{YATEA}}$ often contain valid terms that appear nested at one side of invalid terms.

A similar conclusion can be drawn for deciding upon the inclusion of information about the order of words in context-windows. It is shown that word order information does not necessarily enhance the observed performances (see Figures 5.9b and 5.18). Distance metrics are understood to respond differently to the inclusion of word order information (i.e., if this information improves the performance when using one of them, it does not necessarily enhance the result when using the other one). Apart from the employed metric for similarity measurement and the configuration of context-windows, the results show

[1] See Table 5.4.

that the targeted recall value is an important factor when deciding on the inclusion of word order information. For small recall values, in many experiments, context-windows that encode information about the order of words are among the top performers.

The discussion about the parameters of the classification framework (see RQ 2.1 as well as RQ 2.2) is comparable to the discussion about the configuration of context-windows in the sense that the targeted recall value plays an important role in choosing the best performing configuration. In general, for small recall values—and, when the number of extracted terms is not much larger than $|R_s|$—the Euclidean and the city block distance metrics perform better than the cosine measure. However, the performance of similarity measures that are based on the distance metrics drops abruptly for large recall values. The cosine measure thus seems to be a preferable choice in the majority of applications. Particularly, the cosine measure seems to have a more stable behaviour in the sense that a higher correlation between the observed results is obtained when the set of candidate terms are altered (i.e., when the performances obtained in $\{\mathrm{T}\}^{\textit{c-value}}_{\text{ideal}}$ and $\{\mathrm{T}\}^{\text{YATEA}}_{\text{YATEA}}$ are compared). Concerning the neighbourhood selection value (i.e., k), in the majority of the experiments and on average, a large value (i.e., $k = 25$) shows a better performance than a small value such as $k = 1$, or 7. However, when using the cosine measure and for small recall values, a small value of k can result in a higher performance than a large k.

In order to investigate the effect of the corpus size in the method's performance (see RQ 3), the experiments are continued by fetching additional text and enlarging the GENIA corpus from half a million to 55 million tokens. In turn, as reported in Section 5.4.3, the interplay between the size of the corpus that is used for the construction of the models, the configuration of context-windows (i.e., the way co-occurrence frequencies are collected), and the metrics that are employed to measure similarity between vectors is investigated.

The experiments show that increasing the size of the input corpus for collecting co-occurrence frequencies can improve the performance of the method if a suitable configuration of context-windows and classification parameters (particularly, similarity metric) are employed. It is observed that the top performer parameters in the original corpus of a small size are not necessarily the top performers when the corpus size increases. In addition, it is noticed that choosing the best performing parameters largely depends on the criteria set for the performance assessment. For instance, the city block distance showed a poor performance when the method is assessed at 100% recall. However, at a small recall point, the city block showed a superior performance. These observations can perhaps justify a number of contradictory reports in literature on the effect of the corpus size in the performance of distributional models.

On average, compared to the Euclidean and the city block distance, cosine showed a better performance and a more positive and stable response to an increases in the size of the input corpus. This result can be expected intuitively, since cosine shows the degree of commonality between the elements of two vectors. One can suspect that frequency norm-

alisation and smoothing can enhance the results when using the Euclidean distance. However, an initial experiment to investigate this mater has resulted in even poorer results.[1] The entries of specialized vocabularies are rare and less frequent than general vocabularies. For example, a handful of terms in the GENIA corpus (e.g., the term *physiologic cell lineage*) are so rare that they have appeared only once in the enlarged corpus. Hence, enlarging the corpus will not change the collected co-occurrence frequencies for a relatively large number of terms (see Figure 5.13).

Lastly, to investigate the method's performance across categories of concepts (see RQ 4), the evaluation of the method is extended to a few categories of co-hyponym terms in the GENIA corpus. In Section 5.4.4, it is shown that despite similarities in the configurations of the method that give the best performances for identifying terms in each category (e.g., as shown in Figures 5.23, 5.24, 5.25, and 5.26, using context-windows that extend to the left of candidate terms and the cosine measure for computing similarities often results in the best observed performances), suggesting that it is not possible to recommend a one best configuration for context-windows and the classification parameters across all the categories.

This observation can be utilised when a clustering technique (e.g., as proposed in Dupuch et al., 2014) is employed for identifying co-hyponym terms. The aforementioned observation—that is, the performance of the method, particularly, with respect to the configuration of context-windows is different from one co-hyponym category to another—is often overlooked in these clustering tasks. That is to say, one single configuration of context-windows for collecting co-occurrence frequencies is employed to construct a single model and to perform the clustering process. Using several models in parallel that are constructed by collecting co-occurrences from context-windows of different configurations could, perhaps, enhance the performance of these techniques. If this is not feasible (e.g., due to the lack of computational resources), then a model can be chosen by averaging the performances across categories of concepts, such as the one proposed in Section 5.4.5.

5.6 Improving the Performance for Large Recall Values

Tuning the evaluated parameters of the proposed method enhances the observed performances, particularly for small recall values. However, with the settings employed for its evaluation in the previous sections (particularly, using $|R_s| = 100$), the method suffers from a low performance (precision) when a large recall value (e.g., 100%) is desirable. This problem can be solved by additional reference vectors (i.e., training samples) and enlarging the size of R_s, for example, as reported in our experiments in Zadeh and Handschuh (2014c).

[1] The results from these experiments are thus not reported in details.

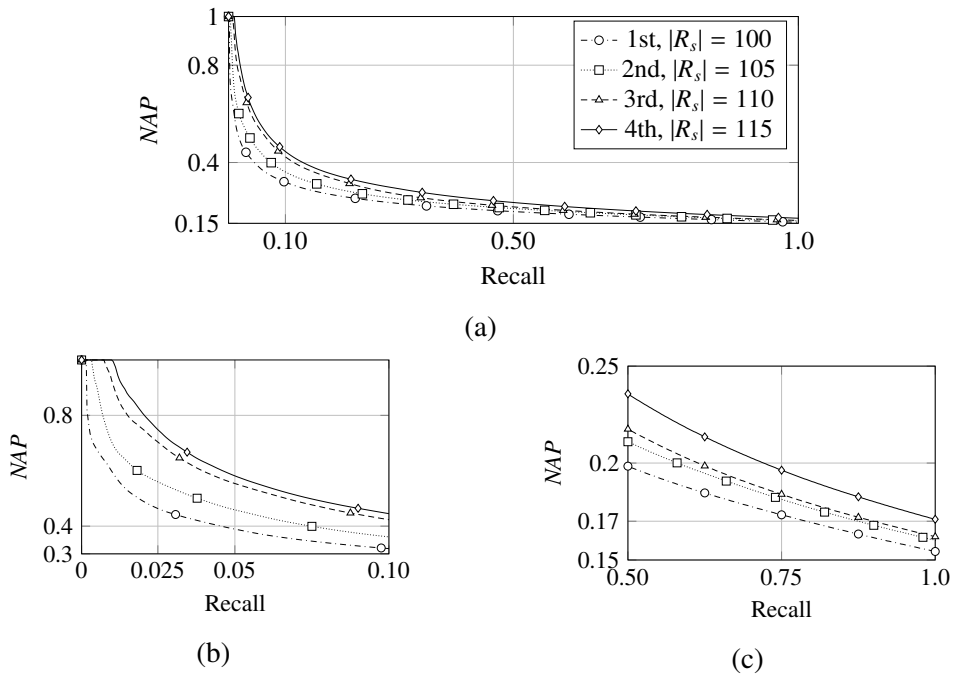


Figure 5.31: Bootstrap learning: the observed non-interpolated precision (*NAP*) for four iterations in the performed experiment: (a) plots the observed *NAP* (i.e., *y*-axis) over the complete range of recall values (i.e., *x*-axis); (b) and (c) provide minute details. In each iteration, *NAP* improves slightly.

In the suggested method, the use of the example-based learning technique allows the addition of training instances and enlarging the R_s during the life cycle of the proposed system. Hence, in some applications, R_s can be extended *manually*, for example, through iterative interactions between the user and the system. Whereas this can be a reasonable solution in a number of use cases, it still may not be favourable in some situations. In this case, an alternative solution is the use of the *bootstrap learning methodology*.

In the bootstrap learning technique, the available annotated data (i.e., R_s) is used to train a classifier, and to label some of the unlabelled data. The resulting labelled data is then employed to extend the available training dataset, to develop a new model, and to label additional unlabelled data. This process is often repeated several times until no improvements are observed. In the context of natural language processing, this methodology is often known as the Yarowsky algorithm (Yarowsky, 1995). Despite errors that are inevitable due to the automatic expansion of the training data, which may limit the performance of this methodology (e.g., as addressed in McIntosh and Curran, 2009),[1] the Yarowsky algorithm has been applied successfully to many information extraction problems.

In the proposed co-hyponym identification task, the Yarowsky algorithm can be

[1]Often known as the problem of *concept drifting*, or *semantic drifting*.

employed to resolve the problem of low performance at large recall points. Originally, Yarowsky proposed his unsupervised learning algorithm for word sense disambiguation based on the observation that words often express only one major sense in a given discourse or document. As stated earlier, in special corpora, the proportion of polysemous terms is very low (e.g., 4% in the GENIA corpus vs. 17% in WordNet). Evidently, for the proposed co-hyponym term extraction task, the prerequisite condition for a successful application of the Yarowsky algorithm is met.

Whereas the study of this algorithm is well beyond the scope of this thesis (e.g., see discussions in Abney, 2004, for an in-depth understanding of the important parameters in the Yarowsky algorithm), as a proof of concept, the observed results from a limited experiment, which is performed over the $\{\mathrm{T}\}^{\mathrm{YATEA}}_{\mathrm{YATEA}}$, are reported. In this experiment, for a particular configuration of context-windows (i.e., using context-windows of size three tokens that are extended to the left of candidate terms), and classification parameters (i.e., using the cosine similarity and the $k = 25$), the classification process is repeated for several iterations. In each iteration, after ranking the candidate terms by their assigned weight in that iteration, the top five candidate terms are added as positive examples to R_s. Figure 5.31 shows the observed results in the first four iterations. As shown, in each iteration, the performance of the method improves slightly.

5.7 Summary

In this chapter, the main method for identifying co-hyponym terms is proposed and evaluated. Terminological resources are often structured by organising terms into a number of categories in the domain of expertise that they represent. In Section 5.1, it is described that terms that are placed under each category of concepts are in a co-hyponymy relationship, which can be modelled—linguistically—as a kind of paradigmatic relationship. Consequently, it is explained that the principles of automatic term recognition (explained in Chapter 3) and distributional semantics (described in Chapter 2) can be combined to extract co-hyponym terms.

Section 5.2 details the method. After the extraction of candidate terms, they are represented in vector space models that are constructed automatically by collecting their co-occurrence frequencies with words appear in narrow context-windows in their vicinity. Exploiting this method, however, is hindered by the high dimensionality of vector spaces—that is, the curse of dimensionality problem (see also RQ 5). To tackle this problem, based on the principles introduced in Chapter 4, random projections are employed for the incremental construction of vectors spaces with a reduced dimensionality. In turn, in these vector space models, the task of identifying co-hyponym terms is accomplished by using an example-based k-nearest neighbours learning framework and a small number of annotated terms as reference vectors R_s, of which $|R_s| = 100$.

As discussed in Section 5.3.5, a number of factors play roles in the performance of

the proposed method: (a) the configuration of context-windows for the collection of co-occurrence frequencies; and, (b) setting the parameters of the learning framework—that is, the neighbourhood size (k) selection and the employed metric for similarity measurements. These parameters are evaluated systematically in Section 5.4. Apart from the influence of these parameters on the performance of the method for identifying co-hyponym terms, the method's performance is studied with respect to (a) the presence of noise (i.e., invalid terms) in the list of candidate terms (see Section 5.4.2), and (b) the size of input corpus for collecting co-occurrence frequencies (i.e., enlarging the corpus as described in Section 5.4.3). In Section 5.4.4, the reported experiments are followed by investigating the method's performance across concept categories. In Section 5.5, the results observed in these experiments are discussed and linked to the research questions proposed in Chapter 1.

Lastly, to improve the performance of the method when the extraction of co-hyponym terms at large recall values is desirable, Section 5.6 suggests the use of a bootstrap learning technique. It is proposed that an unsupervised learning method such as the Yarowsky algorithm can be employed to enlarge R_s iteratively, and thus enhance the observed performances across the complete range of recall values. To support the claim, the results observed in a limited number of experiments are reported.

This page is intentionally left blank.

Part IV

Epilogue

Chapter 6

Conclusion and Future Work

This thesis began with an emphasis on the importance of the extraction of co-hyponym terms—that is, terms that characterise a particular category of concepts in a knowledge domain—for facilitating the process of knowledge acquisition from text. It is explained how the principles of distributional semantics and automatic term extraction can be combined to bridge the semantic gap, to decipher the meaning of terms, and to address the task of extracting co-hyponym terms. Using random projections, vector space models with reduced dimensionality are constructed to represent the distributional properties of terms. In turn, an example-based learning framework is utilised to implement a similarity-based reasoning mechanism in order to identify co-hyponym terms. This thesis details the design and evaluation of the proposed methodology.

This chapter is divided into three sections. Section 6.1 restates the research contributions. Section 6.2 discusses a number of open research questions, as well as a few topics for future research. Finally, Section 6.3 concludes this thesis by a short summary.

6.1 Research Contributions

6.1.1 The Proposed Method for Identifying Co-Hyponym Terms

The main contribution of this thesis is a new perspective of and a novel approach to the identification of co-hyponym terms: a problem that has been so far overlooked in the acquisition of knowledge from text. The proposed distributional representation of terms in a vector space model and the use of similarity-based reasoning for deciphering their meaning (see Section 5.1 and 5.2, Chapter 5) alleviate a number of concerns that arise with respect to the flexibility and the user-friendliness of previously proposed techniques.

Categorisation in general, and, in particular term categorisation, is a major mechanism for organising knowledge and improving the performance of information systems. Based on the specification of a proposed abstraction from a knowledge domain, concepts (and thus terms) are organised in a taxonomy consisting of several co-hyponym groups—each group containing a number of terms that share a *type-of* relationship with a common general concept (i.e., a hypernym). The fluid nature of knowledge is inevitably reflected by the changes in the way that the domain knowledge is abstracted, and, in turn, how these co-hyponym terms are defined in a knowledge structure.

To maintain, and to embody this dynamic structure using tightly supervised techniques—such as those employed in the development of entity extraction systems—is labour-intensive, and thus, expensive. In contrast to these methods, the approach proposed in this thesis is flexible and easy to maintain. In the proposed approach, the mechanism employed for the representation of terms' semantics (i.e., the vector space representation) is independent of the devised categorisation of terms. An update in the structure of knowledge is applied by providing new examples of a newly emerged taxon. Neither text annotation nor a training process is required for the development of a model (meta-language) that captures the structure of terms. In addition, the suggested incremental technique for the construction of a vector space model allows the model to be updated at any time during its use—for example, new terms can be added and removed, and the vectors that represent them can be updated independently of each other. Similarly, examples that are employed by the similarity-based reasoning framework can be updated (new examples are added or removed, and the existing ones modified) at any time during the model's life cycle.

From the perspective of a user of such a system—perhaps, an expert in the knowledge domain, who may have minimal or no training in natural language processing—the process of adapting an existing extraction system to a new task, domain, or even a new class of terms is cumbersome. If a rule-based methodology is employed, new rules must be devised; if a supervised learning technique is employed, a new model must be developed. In contrast, the proposed methodology is user-friendly and intuitive in the sense that the user requires to provide only a few samples of what is, and perhaps what is not,

desirable in a new category of terms. Coupled with the system's flexibility in the manipulation of vectors, feedback from the user can be easily incorporated into the system during its life cycle.

Lastly, the proposed technique is scalable both vertically and horizontally. The fixed dimension of vectors, which can be set and known prior to the extraction task, allows one to implement the method for an effective exploitation of the computational resources available to the system in a single node. This is particularly advantageous if GPU-accelerated[1] computing techniques are employed for similarity measurements. Needless to say, the combination of random projections and example-based similarity reasoning exploited in the proposed method suits parallel, distributed computing (e.g., using the *MapReduce* programming paradigm) extremely well.

Additional novel characteristics of the method proposed for identifying co-hyponym terms are listed in Section 1.2.

6.1.2 A Systematic Evaluation of the Proposed Method

A systematic evaluation of the method proposed in this thesis is carried out (see Section 5.4 of Chapter 5). In the experiments performed (see Sections 5.3), the viability of the proposed distributional hypothesis for identifying co-hyponym terms is verified.[2]

Several parameters that play a role in the performance of the proposed method and the reciprocal relationship between these parameters are investigated (see Discussion in Section 5.5). The discussion of the experiments in Chapter 5 focusses on finding the best configuration of the parameters for the context-window (i.e., the way co-occurrence frequencies are collected) and the parameters for the example-based learning method (i.e., similarity-based reasoning). The interdependence among these parameters, including the figure of merit employed for assessing the performance of the method (i.e., precision at small recall vs. large recall values), is also an important consideration. This is confirmed by the method for reporting observations.

In the experiments described here, context-windows are configured differently with respect to their size (i.e., the extent to which they stretch in the vicinity of candidate terms), direction (i.e., left, right, or around the candidate terms), and encoding information about the order of the words they contain. The example-based learning framework is evaluated with respect to the measure employed for computing similarities (i.e., the cosine measure, the Euclidean distance, and the city block distance) as well as the role of neighbourhood size selection (i.e., the number of examples taken into consideration during the weighting procedure). Moreover, the performance of the method is examined under different conditions, namely in the absence and the presence of noise, for corpora of different sizes, for several categories of concepts, and at both small and large recall

[1] That is, graphical processing units that are customised for linear algebra calculations.

[2] See also RQ 1 to 4.

values. Despite a number of similarities in the results, no single best configuration could be recommended for all the tasks (see Section 5.5). However, the following settings for the parameters can be recommended:

- With respect to the size of the context-window (shown by t):
  - often $2 \leq t \leq 4$ is sufficient. However, if the corpus is small or the targeted co-hyponym terms are infrequent, then a large t such as $4 \leq t \leq 6$ could be a better choice. This is particularly so if the set of candidate terms contains many invalid terms and small recall values are intended. At the same time, choosing a large value of t can introduce noise and thus decrease the performance, particularly if a distance metric is employed for computing similarities.
- With respect to the direction in which the context-window is extended to collect co-occurrences:
  - if the size of the corpus and the intended recall value are small, and the set of candidate terms contains many invalid terms, then context-windows that extend around candidate terms are a better choice. Otherwise, context-windows that expand to the left of candidate terms to collect co-occurrences with their preceding words are recommended.
- With respect to information about the sequential order of words in context-windows:
  - encoding this information does not necessarily enhance performance. If the corpus is large (or, the targeted co-hyponym terms are frequent), and the context-window is extended to the left of candidate terms, encoding the word order information can enhance the result by as much as 10%. It is observed that encoding this information often improves the performance of the best performing models whereas it diminishes the performance of other models.
- With respect to the selection of a similarity measure:
  - a distance measure is, perhaps, a better choice if a small recall value is intended or the intended recall is small in relation to the number of reference terms. However, the cosine measure is an obligate choice if a large recall value is intended, or the intended recall is very large in relation to the number of reference terms. Similarly, if the set of candidate terms contains a large number of invalid terms that share a common context with valid terms,[1] then cosine is a better choice than a distance metric.
- With respect to the neighbourhood size (k) selection in the k-nearest neighbours framework:

[1] For instance, as suggested in Chapter 5, when valid terms appear nested in invalid terms (e.g., the appearance of the valid term *computational linguistics* in invalid candidate terms such as *in computational linguistic studies*, *interesting computational linguistics*, and so on).

– if a small recall value is intended and cosine is employed for similarity measurements, then the nearest neighbour can outperform other choices of k. Otherwise, a large value of k is recommended. Particularly, if a distance metric is chosen, a large k is a more reliable choice than a small one. In addition, when the corpus becomes larger, a smaller value of k can be employed.

6.1.3 The Method for Incremental Construction of Vector Spaces

In this thesis, novel techniques for the incremental construction of vector spaces, particularly ℓ_1-normed spaces, are introduced (see Chapter 4). The proposed methods are employed to obviate the *curse of dimensionality* problem. The mathematical theorems behind the previously employed technique, known as the random indexing (RI) method, is explained and ameliorated by a guideline for setting its parameters. It is shown that RI is an incremental method for the construction of ℓ_2-normed spaces (i.e, Euclidean spaces), which is based on the principle of sparse random projections.[1]

The aforementioned principles are employed to introduce the random Manhattan indexing technique (RMI) and a variation of it named random Manhattan integer indexing (RMII). Both RMI and RMII implement random projections in ℓ_1-normed spaces using projections of randomly created matrices with an asymptotic *Cauchy* distribution. However, by a slight alteration in the distribution of the random projection matrices and a new distance estimator, the RMII method avoids floating point arithmetics during the construction of a vector space. This thesis employs proposed incremental vector space construction techniques for identifying co-hyponym terms. These, however, can be also used in many text analyses algorithms that employ vector space mathematics in general, and in big text data analytics in particular.

Finally, by the help of the principles that are introduced to justify RI, RMI, and RMII, a mathematical justification of a method known as the permutation technique is provided (see Section 5.3.2.3.1, Chapter 5). The permutation technique is employed to capture and to encode into a vector space model the sequential order of words in a text. In this thesis, the previous intuitive justification of the permutation method is complemented using the newly provided mathematical discussion in Chapter 4.

6.2 Open Questions and Future Work

From a very broad perspective, getting machines to understand natural languages, as they are used by people to communicate with and to understand each other, has been, and perhaps, will be one of the biggest research challenges for curious minds. What is obvious is that language as *an instrument of communication*[2] is a non-random complex system.

[1] See also RQ 5.

[2] Note that this is not necessarily an exclusive function of language, but one of many.

However, distinguishing useful patterns in this system, and translating them into machine-accessible semantics has remained an open research question.[1] In addition to this kind of question, the research presented in this thesis can be extended in several ways, as described below.

6.2.1 Semantic Compositionality

In the context of distributional semantics, the *compositionality* of semantics and meaning is currently receiving much attention (see Mitchell and Lapata, 2010, for an overview). Apart from numerous research efforts (e.g., see Baroni et al., 2014a; Coecke et al., 2011), many debates are also going on with respect to the limits and the theoretical foundations of the *compositionality* of semantics (e.g., see Goldberg, 2015). In compositional distributional semantics, research is focused on inferring the meanings of a linguistic entity from its smaller parts—such as words from morphemes, and phrases and sentences from words—using an algebraic structure (e.g., the vector space model studied and employed in this thesis).

For instance, in a number of approaches, given a vector space model of word co-occurrences and a finite number of mathematical operations such as adding, subtracting, and so on, the goal is to answer whether it is possible to infer the meaning of a multiword expression from the vectors of the words that construct the expression (e.g., see Kiela and Clark, 2013). Evidently, this research overlaps with the study of the meanings of terms, particularly *complex*[2] terms. The semantic compositionality of terms is not dealt with in the research presented here: *Are terms, particularly complex terms, irreducible linguistic units such as idioms?* Or, *do they show a degree of compositionality?*

A systematic study of the aforementioned question is one way to extend the proposed research in this thesis (e.g., by limiting the scope of the research proposed in Baldwin et al., 2003, to terminology). As discussed in Chapter 5, complex terms are very rare in special corpora; as a result, the collected co-occurrences in special corpora show a very long-tail statistical distribution.[3] If terms have compositional semantics, then the proposed techniques in compositional distributional semantics can be also used to address problems arising from a lack of data for collecting evidence that is required for establishing the meaning of terms.

6.2.2 Term Space Models for Relations Other Than Co-Hyponymy

Term space models implemented in this thesis are employed and evaluated for identifying co-hyponymy relationships between terms. However, these models can be used

[1] For example, will it be possible to find a comprehensive representation of text data other than the text data itself that meets all the requirements for a system with natural language understanding capability?

[2] That is, multi-token.

[3] Longer than the distribution of the co-occurrences of words in general language corpora.

to recognise relationships between terms other than co-hyponymy—for example, synonymy,[1] associative and relatedness relationships, etc. This is similar to the applications of these methods in general language lexicography, which has been recently encouraged for terminology, too (e.g., see Faber and L'Homme, 2014).[2] If co-hyponymy relationships between terms are employed to suggest an organisation of a specialised vocabulary, then identifying *compatible* and *incompatible* co-hyponyms seems an interesting future research.[3]

As briefly suggested in Chapter 1, the problem of *is-a* overload can be expected in this context. Investigating methods address this problem is also an interesting future research. The advantages of similarity-based reasoning offered by the term space methodology can be used as a complementary mechanism, not only to extract useful information from text but also to facilitate logical inference mechanisms. There is an exciting potential for integrating existing (*semi-*)manually-built formal knowledge resources (e.g., the open schemas and data contributed by the semantic Web research community) and distributional semantic models to build a comprehensive system of reasoning (e.g., see Angeli and Manning, 2014). To make this potential a reality, terminology—as a research discipline—could be the point of convergence for the systematic integration of these research efforts. That is to say, the suggested perspective in terminology[4] can provide a coherent theoretical basis for rational integration of empiricist corpus-based distributional methods and rationalist formal knowledge representation frameworks.

6.2.3 Extending the Scope of Evaluation

Extending evaluated context parameters and enhancing performance

This thesis evaluated the performance of the proposed method using the so-called *flat* distributional models—that is, no linguistic information, such as part-of-speech categories, lemmatisation, or syntactic relationships are employed during the construction of the models. Whereas constructing a flat model demands low computational power and scales out easily, the use of linguistic information could enhance the performance.[5] The evaluation presented here can thus be extended by taking into account the linguistic properties of context elements (i.e., the co-occurred words with candidate terms).

Moreover, context-windows are configured only for a few parameters. This can be easily extended. In the evaluations performed in this thesis, those context-windows that extend around terms are assumed to be *symmetrical* (e.g., 5 tokens to the left and 5 tokens to the right side of terms: that is, 5+5). However, these context-windows can be extended *asymmetrically* (e.g., 5 tokens to the left and 1 tokens to the right side of

[1] That is, to address the term variation problem.
[2] See also Chapter 3.
[3] For example, to find *disjoint* classes in a domain ontology.
[4] Which goes beyond the interpretation of *terms as labels for concepts*; see Chapter 3.
[5] See also related discussion in Chapter 2.

terms: that is, 5+1). The influence of extending context-windows asymmetrically in the performance of the method can be thus studied in the future. Exclusion of words in context-windows is also a possibility that can be investigated, too. For example, context-windows do not require to be extended in the immediate vicinity of terms, but with an offset of a few tokens (e.g., as suggested by Brown et al., 1992). This method for defining context-windows can perhaps reduce noise resulted from errors in identifying candidate terms.

Likewise, the evaluation can be extended by using various weighting mechanisms other than the raw frequencies of words, and similarity measures other than cosine and the ℓ_2 and the ℓ_1 distances. In this study, the evaluation is limited to the use of a fixed set of reference vectors. Investigating methods for choosing the best representative reference vectors would be another way to extend the reported evaluation. Although this question has been investigated from the data analytics perspective (e.g., see Garcia et al., 2012), it is interesting to explore linguistic characteristics of such instances.

As suggested in Chapter 5, bootstrap learning is a plausible solution for improving the method's performance when large recall values are intended; this is one of the limits of the method. In this case, a number of new parameters are introduced. For example, the way the set of reference vectors is extended and the way concept drifting is controlled. This must be investigated together with other parameters of the method.

Evaluation across sublanguages and domains

The evaluation presented in this thesis is limited to the scientific sublanguage from of the molecular biology domain (i.e., the GENIA corpus). Although our initial observations in a sublanguage other than molecular biology (see Zadeh and Handschuh, 2014b,c) is similar to the reported results here, further empirical investigations can be helpful to have a better understanding of the method's behaviour across sublanguages and to further demonstrate its applicability across domains.

Qualitative study of the method's output

The presented quantitative evaluation can be complemented by a qualitative evaluation.[1] The method's parameters for instance can be investigated with respect to the various characteristics of terms they extract. For example, Weeds et al. (2004) study the frequency characteristics of extracted words using different similarity measures. A similar approach can be adopted for studying the method's parameters and the effect of these parameters on various aspects of the properties of the extracted terms (e.g., the frequency of terms, their generality-specificity, etc.).

Not presented in the reported evaluations is the identification of co-hyponyms in *nested* and *hierarchical* structures. For instance, the category of protein terms in the

[1] See the discussion on the evaluation of term extraction methods in Section 3.7 of Chapter 3.

GENIA corpus is made of several sub-categories. The fine-grained identification of these categories of concepts and their evaluation can be beneficial for a number of tasks. The extracted set of co-hyponym terms using the proposed distributional model often consists of entries that are synonyms, metonyms, and hypernyms (as suggested in the previous sentence). Identifying these entries can enhance the quality of the generated set of co-hyponym terms (e.g., similar to what is addressed by Weeds et al., 2014a, for words in a general vocabulary).

Modelling additional elements of the communicative context

Last but not least, extending the evaluation parameters to additional elements of the communicative context is another interesting research quest. For example, extending a distributional model to learn from user interactions and integrating a model of behaviour in the underlying distributional model[1] is an interesting research with many practical applications. Whereas current research is focused mostly on the learning algorithms, the distributional semantic framework allows for flexible expression of this type of information in the knowledge base itself, instead of the learning (training) mechanism.

Diachronic investigation

In the presented study, the evaluation is limited to the extraction of co-hyponym terms at a synchronic level. However, a diachronic analysis of term categories (as well as their meanings), which has a number of important applications, such as *trend analysis*, remains an open research area. The investigation of diachronic aspects of terminology in particular, and in general adding a temporal dimension to distributional semantic models, is certainly an exciting untouched research challenge. The lack of systematic studies in such an important area is, perhaps, due to the lack of suitable language resources.

As reported in Zadeh and Handschuh (2014a), we are developing a language resource, named *ACL RD-TEC*, that can be used for investigating diachronic aspects of a terminology. The ACL RD-TEC dataset consists of manually annotated terms from scientific publications that are drawn from the ACL anthology reference corpus (ACL ARC). The ACL ARC is a fixed set of 10,921 scientific publications in the domain of computational linguistics from 1965 to 2006 (Bird et al., 2008). Term annotations in ACL RD-TEC can thus be mapped to this time-line in order to provide a benchmark for diachronic study of terms and their meanings.

Investigating interaction with the domain conceptualisation

The conceptualisation of a domain defines the co-hyponym groups, which the method proposed in this thesis identifies. This conceptualisation is dynamic and varies even from

[1] Other than, or in addition to, the manipulation of the set of reference terms (as is implied in Section 5.6 of chap. 5), such as the proposed solution for automatic spell checking in QasemiZadeh et al. (2006).

one person to another, as discussed in Chapter 1. The conceived granularity of concepts is especially important in the performance of the method (not only from the statistical point of view, but also from the linguistic and knowledge engineering perspectives). The presented evaluation does not answer questions that arise with respect to this factor. The design of an evaluation framework that can assess this interaction is thus necessary (e.g., as suggested by Rindflesch and Fiszman, 2003).

6.2.4 Further Generalisation of Random Projections

Random projections are modern mathematical tools, which are still relatively unexplored, both theoretically and empirically. This thesis proposed a new incremental technique for constructing vector space models using random projections. The discussion about these projections is limited to α-normed spaces, where $\alpha = 1$, or 2. However, as suggested in Chapter 4, the proposed methodology can be extended to α-normed spaces other than $\alpha = 1$, or 2. The application of these random projections in distributional semantics for the construction of vector space models remains an untouched research avenue. Whether these techniques are suitable for various text analytic applications, however, is an open research question that must be addressed in future research and through experiments.

In this thesis, a single random projection is employed for the construction of vector spaces. However, it is possible to combine random projections in different normed spaces and in different ways. For example, instead of using a single random projection from an n-dimensional to an m-dimensional space of which $n \ll m$, one can apply two different random projections; a projection from the n-dimensional space to an m_1-dimensional space, and then from the m_1-dimensional space to the m-dimensional space of which $n \ll m_1 \ll m$.[1] Using this multi-stage projection allows the approximation of similarities to be carried out in different normed space, if desirable. In addition, a trade-off between the dimension of the projected spaces and the expected errors in the approximated similarities can be considered,[2] allowing for a more efficient computation of similarities and perhaps enhancing the time complexity of a similarity-based reasoning process over big text data—a similar rational as is employed in *locality-sensitive hashing* techniques and *space partitioning* (e.g., see Datar et al., 2004; Dhesi and Kar, 2010).

6.3 Summary

To summarise, this thesis aimed at designing a framework for characterising the conceptual organisation of terms in a specialised vocabulary induced from a domain-specific corpus. To meet this goal, the construction of distributional semantic models with fixed

[1] Note that the trending multi-layer neural networks (i.e., the so-called *deep learning* techniques) are also based on the same mathematical principle.

[2] Since $m_1 \ll m$, it is expected that the approximated distances in the m_1-dimensional space are more accurate than the m-dimensional space.

reduced dimensionality using random projection techniques is studied. With the help of a similarity-based reasoning mechanism, the application of these models to characterise co-hyponymy relationships between terms is investigated.

This page is intentionally left blank.

Reference List

Abney, S. (1992). Parsing by chunks. In *Principle-Based Parsing*, Studies in Linguistics and Philosophy, pages 257–278. Kluwer Academic Publishers. 78

Abney, S. (2004). Understanding the Yarowsky algorithm. *Computational Linguistics*, 30(3):365–395. 192

Achlioptas, D. (2001). Database-friendly random projections. In *Proceedings of the Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 274–281, Santa Barbara, CA, USA. ACM. 110, 111, 118

Acquaviva, P. (2014). The roots of nominality, the nominality of roots. In Alexiadou, A., Borer, H., and Schafer, F., editors, *The Syntax of Roots and the Roots of Syntax*, volume 51 of *Oxford Studies in Theoretical Linguistics*, pages 33–57. Oxford University Press. 6

Afzal, H., Stevens, R., and Nenadic, G. (2008). Towards semantic annotation of bioinformatics services: Building a controlled vocabulary. In Salakoski, T., Schuhmann, D. R., and Pyysalo, S., editors, *Proceedings of the Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008)*, pages 5–12, Turku, Finland. Turku Centre for Computer Science (TUCS). 9, 95

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., and Soroa, A. (2009). A study on similarity and relatedness using distributional and WordNet-based approaches. In *NAACL HLT 2009: Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, Boulder, Colorado. Association for Computational Linguistics. 152

Agres, K., McGregor, S., Purver, M., and Wiggins, G. (2015). Conceptualizing creativity: From distributional semantics to conceptual spaces. In Toivonen, H., Colton, S., Cook, M., and Ventura, D., editors, *Proceedings of the Sixth International Conference on Computational Creativity (ICCC 2015)*, pages 118–125, Utah, USA. The Association for Computational Creativity, Brigham Young University. 7

Ahmad, K., Gillam, L., and Tostevin, L. (1999). University of Surrey participation in TREC 8: Weirdness indexing for logical document extrapolation and retrieval (WILDER). In Voorhees, E. M. and Harman, D. K., editors, *NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC-8)*, pages 717–724. Department of Commerce, National Institute of Standards and Technology. 88

Ananiadou, S. (1994). A methodology for automatic term recognition. In *COLING 94: The 15th Conference on Computational Linguistics: Proceedings*, volume 2, pages 1034–1038, Kyoto, Japan. Association for Computational Linguistics. 69, 74, 76, 89

Anderson, A. J., Bruni, E., Bordignon, U., Poesio, M., and Baroni, M. (2013). Of words, eyes and brains: Correlating image-basedased distributional semantic models with neural representations of concepts. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1960–1970, Seattle, USA. Association for Computational Linguistics. 40

Anderson, A. J., Bruni, E., Uijlings, J., Bordignon, U., Baroni, M., and Poesio, M. (2012). Representational similarity between brain activity elicited by concrete nouns and image based semantic models. In *Proceedings of the Workshop on Vision and Language (VL'12)*, University of Sheffield, UK. 39

Andersson, L., Lupu, M., Palotti, J. R. M., Piroi, F., Hanbury, A., and Rauber, A. (2014). Insight to hyponymy lexical relation extraction in the patent genre versus other text genres. In Jung, H., Mandl, T., Womser-Hacker, C., and Xu, S., editors, *Proceedings of the First International Workshop on Patent Mining and its Applications (IPaMin 2014)*, volume 1292, Hildesheim, Germany. CEUR Workshop Proceedings. 92

Angeli, G. and Manning, C. D. (2014). NaturalLI: Natural logic inference for common sense reasoning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 534–545, Doha, Qatar. Association for Computational Linguistics. 39, 203

Anick, P., Verhagen, M., and Pustejovsky, J. (2014). Extracting aspects and polarity from patents. In Meyers, A., He, Y., and Grishman, R., editors, *Proceedings of the COLING Workshop on Synchronic and Diachronic Approaches to Analyzing Technical Language (SADAATL 2014)*. Association for Computational Linguistics and Dublin City University. 80, 92

Arriaga, R. and Vempala, S. (2006). An algorithmic theory of learning: Robust concepts and random projection. *Machine Learning*, 63(2):161–182. 118

Aubin, S. and Hamon, T. (2006). Improving term extraction with terminological resources. In Salakoski, T., Ginter, F., Pyysalo, S., and Pahikkala, T., editors, *Advances in Natural Language Processing*, volume 4139 of *Lecture Notes in Computer Science*, pages 380–387. Springer Berlin Heidelberg. 77, 83, 150

Bada, M., Eckert, M., Evans, D., Garcia, K., Shipley, K., Sitnikov, D., Baumgartner, W., Cohen, K., Verspoor, K., Blake, J., and Hunter, L. (2012). Concept annotation in the CRAFT corpus. *BMC Bioinformatics*, 13(1):161. 97

Baker, L. D. and McCallum, A. K. (1998). Distributional clustering of words for text classification. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 96–103, Melbourne, Australia. ACM. 47, 48

Baldwin, T., Bannard, C., Tanaka, T., and Widdows, D. (2003). An empirical model of multiword expression decomposability. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 89–96, Sapporo, Japan. Association for Computational Linguistics. 202

Baldwin, T. and Kim, S. N. (2010). Multiword expressions. In Indurkhya, N. and Damerau, F. J., editors, *Handbook of Natural Language Processing*, pages 267–292. CRC Press, Taylor and Francis Group, second edition. ISBN 978-1420085921. 75

Balog, K. and de Rijke, M. (2008). Associating people and documents. In Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., and White, R. W., editors, *Advances in Information Retrieval*, volume 4956 of *Lecture Notes in Computer Science*, pages 296–308. Springer Berlin Heidelberg. 143

Baroni, M. (2013). Dr. Strangestats or: How I learned to stop worrying and love distributional semantics. *Computational Models of Language Meaning in Context (Dagstuhl Seminar 13462)*, 3(11):85–86. 28

Baroni, M., Bernardi, R., and Zamparelli, R. (2014a). Frege in space: A program for compositional distributional semantics. *Linguistic Issues in Language Technologies*, 9(6):5–110. 202

Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. (2009). The WaCky Wide Web: A collection of very large linguistically processed Web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226. 114, 129

Baroni, M., Dinu, G., and Kruszewski, G. (2014b). Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, US. Association for Computational Linguistics. 152

Baroni, M. and Evert, S. (2009). Statistical methods for corpus exploitation. In Lüdeling, A. and Kytö, M., editors, *Corpus Linguistics: An International Handbook*, volume 2 of *Handbooks of Linguistics and Communication Science*, pages 777–802. Walter de Gruyter. 23

Baroni, M. and Lenci, A. (2010). Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721. 15

Baroni, M., Lenci, A., and Onnis, L. (2007). ISA meets Lara: An incremental word space model for cognitively plausible simulations of semantic learning. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 49–56, Prague, Czech Republic. Association for Computational Linguistics. 44, 119, 122

Baroni, M., Murphy, B., Barbu, E., and Poesio, M. (2010). Strudel: A corpus-based semantic model based on properties and types. *Cognitive Science*, 34(2):222–254. 36, 37, 38, 40

Barrett, R., Berry, M., Chan, T., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., and Van der Vorst, H. (1993). *Templates for the solution of linear systems: Building blocks for iterative methods*. SIAM. 42

Barrón-Cedeño, A., Sierra, G., and Ananiadou, P. D. S. (2009). An improved automatic term recognition method for Spanish. In Gelbukh, A., editor, *Computational Linguistics and Intelligent Text Processing*, volume 5449 of *Lecture Notes in Computer Science*, pages 125–136. Springer Berlin Heidelberg. 80

Basili, R., Moschitti, A., Pazienza, M. T., and Zanzotto, F. M. (2001). A contrastive approach to term extraction. In *TIA 2001: Terminologie et Intelligence Artificielle*, pages 119–128, Nancy, France. INIST-CNRS. 83

Baxendale, P. B. (1958). Machine-made index for technical literature – an experiment. *IBM Journal of Research and Development*, 2(4):354–361. 84

Bernier-Colborne, G. and Drouin, P. (2014). Creating a test corpus for term extractors through term annotation. *Terminology*, 20(1):50–73. 97

Bertels, A. and Speelman, D. (2014). Clustering for semantic purposess: Exploration of semantic similarity in a technical corpus. *Terminology: International Journal of Theoretical and Applied Issues in Specialized Communication*, 20(2):279–303. 92

Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. (1999). When is "nearest neighbor"meaningful? In Beeri, C. and Buneman, P., editors, *Database Theory – ICDT'99*, volume 1540 of *Lecture Notes in Computer Science*, pages 217–235. Springer Berlin Heidelberg. 45

Bingham, E. and Mannila, H. (2001). Random projection in dimensionality reduction: Applications to image and text data. In *KDD '01: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 245–250, New York, NY, USA. ACM. 118

Bins, J. and Draper, B. (2001). Feature selection from huge feature sets. In *Proceedings of Eighth IEEE International Conference on Computer Vision*, volume 2, pages 159–165, British Columbia, Canada. 45

Bird, S., Dale, R., Dorr, B., Gibson, B., Joseph, M., Kan, M.-Y., Lee, D., Powley, B., Radev, D., and Tan, Y. F. (2008). The ACL anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., and Tapias, D., editors, *Proceedings of the Sixth International Conference on Language Resources and Evaluation*, pages 1755–1759, Marrakech, Morocco. European Language Resources Association. 205

Blackburn, P. and Bos, J. (2005). *Representation and inference for natural language: A first course in computational semantics*. CLSI Studies in Computaional Linguistics. CSLI. 22

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022. 29, 62

Bonin, F., Dell'Orletta, F., Montemagni, S., and Venturi, G. (2010a). A contrastive approach to multi-word extraction from domain-specific corpora. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation*, pages 3222–3229, Valletta, Malta. European Language Resources Association. 80

Bonin, F., Dell'Orletta, F., Venturi, G., and Montemagni, S. (2010b). Contrastive filtering of domain-specific multi-word terms from different types of corpora. In *Proceedings of the 2010 Workshop on Multiword Expressions: from Theory to Applications (MWE 2010)*, pages 77–80, Beijing, China. Coling 2010 Organizing Committee. 83

Bourigault, D. (1992). Surface grammatical analysis for the extraction of terminological noun phrases. In *COLING 1992 Volume 3: The 15th International Conference on Computational Linguistics*, pages 977–981, Nantes, France. International Committee on Computational Linguistics. 78

Brand, M. (2006). Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra and its Applications*, 415(1):20–30. Special Issue on Large Scale Linear and Nonlinear Eigenvalue Problems. 108

Brewster, C., Jupp, S., Luciano, J. S., Shotton, D., Stevens, R. D., and Zhang, Z. (2009). Issues in learning an ontology from text. *BMC Bioinformatics*, 10(Suppl 5):S1. 4

Brewster, C. A. (2008). *Mind the Gap: Bridging from Text to Ontological Knowledge*. PhD thesis, University of Sheffield. 4

Brinkman, B. and Charikar, M. (2005). On the impossibility of dimension reduction in L1. *Journal of the ACM*, 52(5):766–788. 122, 124

Brown, P. F., de Souza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479. 7, 204

Bruni, E., Tran, N. K., and Baroni, M. (2014). Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49(1):1–47. 39

Bruni, E., Uijlings, J., Baroni, M., and Sebe, N. (2012). Distributional semantics with eyes: Using image analysis to improve computational representations of word meaning. In *MM'12: Proceedings of the 20th ACM International Conference on Multimedia*, pages 1219–1228, Nara, Japan. ACM. 28, 39

Brunzel, M. (2008). The XTREEM methods for ontology learning from Web documents. In Buitelaar, P. and Cimiano, P., editors, *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, volume 167 of *Frontiers in Artificial Intelligence and Applications*, pages 3–26. IOS Press, Amsterdam, The Netherlands. 82

Buitelaar, P., Cimiano, P., Haase, P., and Sintek, M. (2009). Towards linguistically grounded ontologies. In Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvonen, E., Mizoguchi, R., Oren, E., Sabou, M., and Simperl, E., editors, *The Semantic Web: Research and Applications*, volume 5554 of *Lecture Notes in Computer Science*, pages 111–125. Springer Berlin Heidelberg. 90

Buitelaar, P., Cimiano, P., and Magnini, B. (2005). Ontology learning from text: An overview. In Buitelaar, P., Cimiano, P., and Magnini, B., editors, *Ontology Learning from Text: Methods, Evaluation and Applications*, volume 123 of *Frontiers in Artificial Intelligence and Applications*, pages 3–12. IOS Press. 140

Buitelaar, P. and Eigner, T. (2008). Topic extraction from scientific literature for competency management. In Mochol, M., Zhdanova, A. V., Nixon, L., Breslin, J., and Polleres, A., editors, *Personal Identification and Collaborations: Knowledge Mediation and Extraction (PICKME 2008)*, volume 403, pages 55–66, Karlsruhe, Germany. CEUR Workshop Proceedings. 143

Bullinaria, J. A. and Levy, J. P. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510–526. xxv, 43, 56, 57, 58, 129, 169

Bullinaria, J. A. and Levy, J. P. (2012). Extracting semantic representations from word co-occurrence statistics: Stop-lists, stemming, and SVD. *Behavior Research Methods*, 44(3):890–907. 38, 47

Burgess, C. (2001). Representing and resolving semantic ambiguity: A contribution from high-dimensional memory modeling. In Gorfein, D. S., editor, *On the Consequences of Meaning Selection: Perspectives on Resolving Lexical Ambiguity*, Decade of Behavior, pages 233–260. American Psychological Association. 44

Bustos, B. and Navarro, G. (2004). Probabilistic proximity searching algorithms based on compact partitions. *Journal of Discrete Algorithms*, 2(1):115 – 134. The 9th International Symposium on String Processing and Information Retrieval. 65

Cabré, M. T. (1999). *Terminology: Theory, Methods and Applications*. John Benjamins. 70

Cabré, M. T. (2003). Theories of terminology their description, prescription and explanation. *Terminology*, 9(2):163–199. 68, 69, 71

Cabré, M. T. (2010). Terminology and translation. In dins Gambier, Y. and Van Doorslaer, L., editors, *Handbook of translation studies*, volume 1, pages 356–365. John Benjamins Publishing Company. 68

Cabré, M. T., Condamines, A., and Ibekwe-SanJuan, F. (2007). Introduction: Application-driven terminology engineering. In *Application-Driven Terminology Engineering*, volume vii, pages 1–19. John Benjamins. 72

Caid, W. R. and Oing, P. (1997). System and method of context vector generation and retrieval. 118

Campo, Á. (2013). *The reception of Eugen Wüster's work and the development of terminology*. PhD thesis, Université de Montréal. 69

Carlson, G. N. (1980). *Reference to kinds in English*. Outstanding Dissertations in Linguistics. Garland Publishing, rev. version of author's thesis, university of massachusetts, amherst, 1977 edition. 6

Cha, S.-H. (2007). Comprehensive survey on distance/similarity measures between probability density functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, 1(4):300–307. 55, 57

Chakraborty, S., Subramanian, L., and Nyarko, Y. (2014). Extraction of (key,value) pairs from unstructured ads. In *AAAI Fall Symposium Serie*, pages 10–17, Arlington, Virginia. AAAI Press. 92, 143

Chan, Y. S. and Ng, H. T. (2007). Domain adaptation with active learning for word sense disambiguation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 49–56, Prague, Czech Republic. Association for Computational Linguistics. 5

Chandler, D. (2007). *Semiotics: The Basics*. Routledge. 23

Charles, W. G. (2000). Contextual correlates of meaning. *Applied Psycholinguistics*, 21(4):505–524. 24

Chen, S. F. and Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13(4):359–393. 55

Chen, X., Hu, X., Zhou, Z., An, Y., He, T., and Park, E. (2012). Modeling semantic relations between visual attributes and object categories via Dirichlet forest prior. In *CIKM'12: Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 1263–1272, Hawaii, USA. ACM. 39

Church, K. W. and Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29. 85

Cimiano, P., Hotho, A., and Staab, S. (2005). Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligence Research*, 24:305–339. 140

Cimiano, P., McCrae, J., Buitelaar, P., and Montiel-Ponsoda, E. (2013). On the role of senses in the ontology-lexicon. In Oltramari, A., Vossen, P., Qin, L., and Hovy, E., editors, *New Trends of Research in Ontologies and Lexical Resources*, Theory and Applications of Natural Language Processing, pages 43–62. Springer Berlin Heidelberg. 6, 141, 142

Cimiano, P., Schultz, A., Sizov, S., Sorg, P., and Staab, S. (2009). Explicit versus latent concept models for cross-language information retrieval. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence*, pages 1513–1518, California, USA. IJCAI Organization. 49

Clark, S. (2015). Vector space models of lexical meaning. In Lappin, S. and Fox, C., editors, *Handbook of Contemporary Semantics*. Wiley-Blackwell, 2nd edition. 44

Coecke, B., Sadrzadeh, M., and Clark, S. (2011). Mathematical foundations for a compositional distributional model of meaning. In van Benthem, J. and Moortgat, M., editors, *Linguistic Analysis*, volume 36(1–4) of *A Festschrift for Joachim Lambek*, pages 345–384. 202

Cohen, T., Schvaneveldt, R., and Widdows, D. (2010). Reflective random indexing and indirect inference: A scalable method for discovery of implicit connections. *Journal of Biomedical Informatics*, 43(2):240–256. 119

Collins, M. and Duffy, N. (2002). Convolution kernels for natural language. In Dietterich, T., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, pages 625–632. MIT Press. 38

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27. 63

Cowie, J. and Wilks, Y. (2000). Information extraction. In Dale, R., Moisl, H., and Somers, H., editors, *Handbook of Natural Language Processing*. New York: Marcel Dekker. 99

Cruse, D. A. (1986). *Lexical Semantics*. Cambridge University Press. 24

Cunningham, P. and Delany, S. J. (2007). k-nearest neighbour classifiers. Technical Report UCD-CSI-2007-4, UCD School of Computer Science and Informatics. 64

Curran, J. R. (2004). *From Distributional to Semantic Similarity*. PhD thesis, University of Edinburgh. 37, 43, 58

Daelemans, W. (1999). Memory-based language processing: Introduction to the special issue. *Journal of Experimental and Theoretical Artificial Intelligence*, 11(3):287–292. 65

Daelemans, W. and van den Bosch, A. (2005). *Memory-Based Language Processing*. Studies in Natural Language Processing. Cambridge University Press. 65

Daelemans, W. and van den Bosch, A. (2010). Memory-based learning. In Clark, A., Fox, C., and Lappin, S., editors, *The Handbook of Computational Linguistics and Natural Language Processing*, pages 154–179. Wiley-Blackwell. 10, 65, 144

Daelemans, W., Zavrel, J., van der Sloot, K., and van den Bosc, A. (2009). TiMBL: Tilburg memory-based learner version 6.3 reference guide. Technical Report ILK Technical Report –ILK 10-01, ILK Research Group. 64

Daille, B. (1995). Combined approach for terminology extraction: Lexical statistics and linguistic filtering. In Wilson, A. and McEnery, T., editors, *UCREL Technical Papers*. Lancaster University. 78, 80, 85, 86

Dasgupta, S. and Gupta, A. (2003). An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures and Algorithms*, 22(1):60–65. 110

Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V. S. (2004). Locality-sensitive hashing scheme based on p-stable distributions. In *SCG '04: Proceedings of the Twentieth Annual Symposium on Computational Geometry*, pages 253–262, Brooklyn, NY, USA. ACM. 206

De Vine, L. (2013). Some extensions to representation and encoding of structure in models of distributional semantics. Master's thesis, Queensland University of Technology. 54

De Vries, C. M. (2014). *Document clustering algorithms, representations and evaluation for information retrieval*. PhD thesis, Queensland University of Technology. 54

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407. 27, 35, 37, 40, 48

Deza, E. and Deza, M. (2006). *Dictionary of distances*. Elsevier. 55

Deza, M. M. and Deza, E. (2014). *Encyclopedia of Distances*. Springer-Verlag Berlin Heidelberg, 3 edition. 55

Dhesi, A. and Kar, P. (2010). Random projection trees revisited. In Lafferty, J. D., Williams, C. K. I., J., S.-T., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*. Curran Associates, Inc. 206

Dias, G. and Kaalep, H.-J. (2003). Automatic extraction of multiword units for Estonian: Phrasal verbs. In Metslang, H. and Rannut, M., editors, *Languages in Development*, volume 41 of *Linguistics Edition*, pages 81–90. LINCOM. 85

Dinu, A., Dinu, L., and Sorodoc, I. (2014). Aggregation methods for efficient collocation detection. In Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 4041–4045, Reykjavik, Iceland. European Language Resources Association. 87

Dorji, T. C., sayed Atlam, E., Yata, S., Fuketa, M., Morita, K., and ichi Aoe, J. (2011). Extraction, selection and ranking of field association (FA) terms from domain-specific corpora for building a comprehensive FA terms dictionary. *Knowledge and Information Systems*, 27(1):141–161. 79

Dredze, M., McNamee, P., Rao, D., Gerber, A., and Finin, T. (2010). Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 277–285, Beijing, China. Coling 2010 Organizing Committee. 93

Drouin, P. (2003). Term extraction using non-technical corpora as a point of leverage. *Terminology*, 9(1):99–115. 82

Drouin, P. (2004). Detection of domain specific terminology using corpora comparison. In Lino, M. T., Xavier, M. F., Ferreira, F., Costa, R., Silva, R., Pereira, C., Carvalho, F., Lopes, M., Catarino, M., and Barros, S., editors, *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, pages 79–82, Lisbon, Portugal. European Language Resources Association. 82, 88

Dunbar, R. (1996). *The Trouble with Science*. Harvard University Press. 23

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74. 86

Dupuch, M., Dupuch, L., Hamon, T., and Grabar, N. (2014). Exploitation of semantic methods to cluster pharmacovigilance terms. *Journal of Biomedical Semantics*, 5(18). 73, 92, 140, 190

Dupuch, M., Hamo, T., Dupuch, L., and Grabar, N. (2012). Semantic distance and terminology structuring methods for the detection of semantically close terms. In *Proceedings of the 2012 Workshop on Biomedical Natural Language Processing (BioNLP 2012)*, pages 20–28, Montreal, Canda. Association for Computational Linguistics. 92

Eck, N. J. V., Waltman, L., Noyons, E. C., and Buter, R. K. (2010). Automatic term identification for bibliometric mapping. *Scientometrics*, 82:581–596. 79, 80

Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218. 49

Eddington, D. (2008). Linguistics and the scientific method. *The International Journal of The Linguistic Association Of The Southwest (LASSO)*, 27(2):1–16. 27

Ellis, J., Li, X., Griffitt, K., Strassel, S. M., and Wright, J. (2012). Linguistic resources for 2012 knowledge base population evaluations. In *Proceedings of the Fifth Text Analysis Conference (TAC 2012)*, Maryland, USA. National Institute of Standards and Technology. 92

Erk, K. (2012). Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653. 28

Erk, K. and Padó, S. (2008). A structured vector space model for word meaning in context. In *EMNLP 2008: 2008 Conference on Empirical Methods in Natural Language Processing: Proceedings of the Conference*, pages 897–906, Honolulu, Hawaii. Association for Computational Linguistics. 38

Esuli, A. and Sebastiani, F. (2010). Evaluating information extraction. In Agosti, M., Ferro, N., Peters, C., Rijke, M., and Smeaton, A., editors, *Multilingual and Multimodal Information Access Evaluation*, volume 6360 of *Lecture Notes in Computer Science*, pages 100–111. Springer Berlin Heidelberg. 99

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2005). Unsupervised named-entity extraction from the Web: An experimental study. *Artificial Intelligence*, 165(1):91–134. 141

Evans, D. A. and Zhai, C. (1996). Noun-phrase analysis in unrestricted text for information retrieval. In *34th annual meeting on Association for Computational Linguistics: Proceedings of the Conference*, pages 17–24, California, USA. Association for Computational Linguistics. 81

Evert, S. (2004). *The Statistics of Word Cooccurrences Word Pairs and Collocations*. PhD thesis, Institut für maschinelle Sprachverarbeitung, Universität Stuttgart. 71, 84

Evert, S. (2009). Corpora and collocations. In Lüdeling, A. and Kytö, M., editors, *Corpus Linguistics: An International Handbook*, volume 2 of *Handbooks of Linguistics and Communication Science*. Mouton de Gruyter. 87

Evgeniou, T., Pontil, M., and Poggio, T. (1999). A unified framework for regularization networks and support vector machines. Technical report, MIT, AI Lab and CBCL, Cambridge, MA, USA. Retrieved from `ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-1654.pdf`. 62

Faber, P. and L'Homme, M.-C. (2014). Lexical semantic approaches to terminology: An introduction. *Terminology*, 20(2):143–150. 14, 69, 70, 90, 203

Faber, P. and Rodríguez, C. I. L. (2012). Terminology and specialized language. In Faber, P., editor, *A Cognitive Linguistics View of Terminology and Specialized Language*, volume 20 of *Applications of Cognitive Linguistics*, pages 9–33. Walter de Gruyter. 68

Fahmi, I. (2009). *Automatic term and relation extraction for medical question answering system*. PhD thesis, University Library Groningen. 98

Fan, T.-K. and Chang, C.-H. (2008). Exploring evolutionary technical trends from academic research papers. In Kise, K. and Sako, H., editors, *DAS 2008: Proceedings of the 8th IAPR International Workshop on Document Analysis Systems*, pages 574–581, Nara, Japan. IEEE Computer Society. 81

Feigenbaum, E. A. (1980). Knowledge engineering: The applied side of artificial intelligence. Technical Report STAN-CS-80-812 (HPP-80-21), Computer Science Department, Stanford University. 4

Feiyu, X., Kurz, D., Piskorski, J., and Schmeier, S. (2002). A domain adaptive approach to automatic acquisition of domain relevant terms and their relations with bootstrapping. In *Proceedings of the 3rd International Conference on Language Resources an Evaluation*, number 224–230, Las Palmas, Canary Islands, Spain. European Language Resources Association. 85

Felber, H. (1982). Computerized terminology in Termnet: The role of terminological data banks. In *Term banks for tomorrow's world: Translating and the Computer 4*, pages 8–20, London, UK. Aslib. Conference Proceedings. 69

Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database*. MIT Press. 5

Ferraresi, A., Zanchetta, E., Baroni, M., and Bernardini, S. (2008). Introducing and evaluating ukWaC, a very large web-derived corpus of English. In Evert, S., Kilgarriff, A., and Sharoff, S., editors, *Proceedings of the 4th Web as Corpus Workshop (WAC-4): Can we beat Google?*, pages 47–54. 129

Firth, J. (1957). *Papers in Linguistics 1934–51*. Oxford University Press. 24

Fisher, R. A. (1936). Has mendel's work been rediscovered? *Annals of Science*, 1:115–137. Obtained from `https://drmc.library.adelaide.edu.au/dspace/bitstream/2440/15123/1/144.pdf`. 27

Fiszman, M., Rosemblat, G., Ahlers, C. B., and Rindflesch, T. C. (2007). Identifying risk factors for metabolic syndrome in biomedical text. In *AMIA Annual Symposium Proceedings*, volume 2007, page 249. American Medical Informatics Association. 82

Fodor, J. and Lepore, E. (2012). What sort of science is semantics? In Peter, G. and KrauSSe, R.-M., editors, *Selbstbeobachtung der modernen Gesellschaft und die neuen Grenzen des Sozialen*, pages 217–226. Springer Fachmedien Wiesbaden. 71

Foo, J. and Merkel, M. (2010). Using machine learning to perform automatic term recognition. In *Proceedings of the LREC 2010 Workshop on Methods for Automatic Acquisition of Language Resources and their Evaluation Methods*, pages 49–54. 95

Frantzi, K. T. (1997). Incorporating context information for the extraction of terms. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, EACL '97, pages 501–503, Stroudsburg, PA, USA. Association for Computational Linguistics. 78, 80

Frantzi, K. T. and Ananiadou, S. (1996). Extracting nested collocations. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, pages 41–46, Copenhagen, Denmark. Association for Computational Linguistics. 85, 88

Frantzi, K. T., Ananiadou, S., and Mima, H. (2000a). Automatic recognition of multi-word terms: The C-value/NC-value method. *International Journal on Digital Libraries*, 3(2):115–130. 80

Frantzi, K. T., Ananiadou, S., and Tsujii, J. (2000b). The C-value/NC-value method of automatic recognition for multi-word terms. In *Research and Advanced Technology for Digital Libraries*, volume 3 of *Lecture Notes in Computer Science*, pages 585–604. Springer Berlin Heidelberg. 88, 150

Freixa, J. (2006). Causes of denominative variation in terminology. *Terminology*, 12(1):51–77. 6

Fukuda, K., Tsunoda, T., Tamura, A., and Takagi, T. (1998). Toward information extraction: Identifying protein names from biological papers. In *Pacific Symposium on Biocomputing*, volume 3, pages 707–718. 92

Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI-07)*, pages 1606–1611, Hyderabad, India. AAAI Press. 39, 40

Gale, W. A., Church, K. W., and Yarowsky, D. (1992). One sense per discourse. In *Proceedings of Speech and Natural Language Workshop (HLT'92)*, pages 233–237, Harriman, New York. Morgan Kaufmann Publishers. 5

Gallant, S. I. (1991). A practical approach for representing context and for performing word sense disambiguation using neural networks. *Neural Computation*, 3(3):293–309. 42, 118

Gallant, S. I. (1994). Methods for generating or revising context vectors for a plurality of word stems. US Patent 5,325,298. 42

Gallant, S. I. (2000). Context vectors: A step toward a grand unified representation. In Wermter, S. and Sun, R., editors, *Hybrid Neural Systems*, volume 1778 of *Lecture Notes in Computer Science*, pages 204–210. Springer Berlin Heidelberg. 42

Garcia, S., Derrac, J., Cano, J. R., and Herrera, F. (2012). Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):417–435. 204

Gardner, M., Talukdar, P., Krishnamurthy, J., and Mitchell, T. (2014). Incorporating vector space similarity in random walk inference over knowledge bases. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 397–406, Doha, Qatar. Association for Computational Linguistics. 39

Geva, S. and De Vries, C. M. (2011). TOPSIG: Topology preserving document signatures. In Berendt, B., de Vries, A., Fan, W., Macdonald, C., Ounis, I., and Ruthven, I., editors, *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 333–338, Glasgow, Scotland, UK. ACM. 43, 119

Gionis, A., Indyk, P., Motwani, R., et al. (1999). Similarity search in high dimensions via hashing. In *VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases*, pages 518–529. 54

Gliozzo, A. and Strapparava, C. (2009). Semantic domains. In *Semantic Domains in Computational Linguistics*, pages 13–32. Springer Berlin Heidelberg. 13

Goldberg, A. E. (2015). Compositionality. In Riemer, N., editor, *The Routledge Handbook of Semantics*. Route ledge. 202

Gooch, P. and Roudsari, A. V. (2011). Automated recognition and post-coordination of complex clinical terms. In Borycki, E. M., Bartle-Clar, J. A., Househ, M. S., Kuziemsky, C. E., and Schraa, E. G., editors, *International Perspectives in Health Informatics*, Studies in Health Technology and Informatics, pages 8–12. IOS Press. 82

Gorman, J. and Curran, J. R. (2006a). Random indexing using statistical weight functions. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 457–464, Sydney, Australia. Association for Computational Linguistics. 44

Gorman, J. and Curran, J. R. (2006b). Scaling distributional similarity to large corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 361–368, Sydney. Association for Computational Linguistics. 169

Grefenstette, G. (1994). *Explorations in Automatic Thesaurus Discovery*, volume 278 of *The Springer International Series in Engineering and Computer Science*. Springer US, Norwell, MA, USA, 1 edition. 38

Grishman, R. and Kittredge, R., editors (2014). *Analyzing language in restricted domains: Sublanguage description and processing*. Psychology Press, New York, NY, US. First published 1986 by Lawrence Erlbaum Associates. 13

Guarino, N. (1998). Some ontological principles for designing upper level lexical resources. In Calzolari, N., Choukri, K., Hoege, H., Maegaard, B., Mariani, J., Municio, A. M., and Zampolli, A., editors, *First International Conference on Language Resources and Evaluation*, Granada, Spain. European Language Resources Association. 7

Gufler, B., Augsten, N., Reiser, A., and Kemper, A. (2012). Load balancing in MapReduce based on scalable cardinality estimates. In *ICDEW'12: Proceedings of the 2012 IEEE 28th International Conference on Data Engineering Workshops*, pages 522–533, Virginia, USA. IEEE Computer Society. 121

Hall, M. A. (1999). *Correlation-based Feature Selection for Machine Learning*. PhD thesis, The University of Waikato. 47

Halliday, M. A. K. and Hasan, R. (2013). *Cohesion in English*. English Language Series. Routledge. First published by Longman Group in 1976. 5

Hamon, T., Engström, C., and Silvestrov, S. (2014). Term ranking adaptation to the domain: Genetic algorithm-based optimisation of the C-value. In Przepiórkowski, A. and Ogrodniczuk, M., editors, *Advances in Natural Language Processing*, volume 8686 of *Lecture Notes in Computer Science*, pages 71–83. Springer International Publishing. 94

Hanks, P. (1996). Contextual dependency and lexical sets. *International Journal of Corpus Linguistics*, 1(1):75–98. 24

Harris, Z. (1968). *Mathematical structures of language*. Number 21 in Interscience tracts in pure and applied mathematics. John Wiley and Sons. 12, 13

Harris, Z. (1998). *Language and information*. Columbia University Press. 13

Harris, Z. S. (1954). Distributional structure. *Word, The Journal of the International Linguistic Association*, 10:146–162. 12, 21, 22, 23, 24, 143

Harris, Z. S. (2002). The structure of science information. *Journal of Biomedical Informatics*, 35(4):215–221. Sublanguage - Zellig Harris Memorial. 14

Hartmann, S., Szarvas, G., and Gurevych, I. (2011). Mining multiword terms from Wikipedia. In Pazienza, M. T. and Stellato, A., editors, *Semi-Automatic Ontology Development: Processes and Resources*, pages 226–258. IGI Global, Hershey, PA, USA. 82

Hartung, M. and Frank, A. (2010). A structured vector space model for hidden attribute meaning in adjective-noun phrases. In *Coling 2010: 23rd International Conference on Computational Linguistics: Proceedings of the Conference*, pages 430–438, Beijing, China. Association for Computational Linguistics/Tsinghua University Press. 38

Hartung, M. and Frank, A. (2011). Exploring supervised LDA models for assigning attributes to adjective-noun phrases. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 540–551, Edinburgh, Scotland, UK. Association for Computational Linguistics. 40

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer Science+Business Media, 2nd edition. 155

Hazen, R., Esbroeck, A., Mongkolwat, P., and Channin, D. (2011). Automatic extraction of concepts to extend RadLex. *Journal of Digital Imaging*, 24:165–169. 82

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In Boitet, C., editor, *Proceedings of the fifteenth International Conference on Computational Linguistics: COLING-92*, volume II, pages 539–545, Nantes, France. GETA (IMAG) and Association Champollion / Association for Computational Linguistics. 140, 143

Hecht-Nielsen, R. (1994). Context vectors: General purpose approximate meaning representations self-organized from raw data. In *Computational Intelligence: Imitating Life*, pages 43–56. IEEE Press. Papers presented at the 1994 World Congress on Computational Intelligence (WCCI) held in summer in Orlando, Florida. 118

Heid, U. and Gojun, A. (2012). Term candidate extraction for terminography and CAT: An overview of TTC. In Fjeld, R. V. and Torjusen, J. M., editors, *Proceedings of the 15th Euralex International Congress*, pages 585–594, University of Oslo, Norway. 70

Herbelo, A. (2015). Mr Darcy and Mr Toad, gentlemen: distributional names and their kinds. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 151–161, London, UK. Association for Computational Linguistics. 7

Hippisley, A., Cheng, D., and Ahmad, K. (2005). The head-modifier principle and multilingual term extraction. *Natural Language Engineering*, 11(2):129–157. 81

Hirst, D. (2007). For the love of god [platinum, diamonds and human teeth sculpture]. Photographed by Prudence Cuming Associates ® Damien Hirst and Science Ltd. Retrieved from `http://www.damienhirst.com/for-the-love-of-god`. xii

Hoang, H. H., Kim, S. N., and Kan, M.-Y. (2009). A re-examination of lexical association measures. In *MWE 2009: Proceedings of the 2009 Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, pages 31–39, Suntec, Singapore. The Association for Computational Linguistics and The Asian Federation of Natural Language Processing. 84

Hobbs, J. R. and Riloff, E. (2010). Information extraction. In Indurkhya, N. and Damerau, F. J., editors, *Handbook of Natural Language Processing*. CRC Press, Taylor and Francis Group, Boca Raton, FL. ISBN 978-1420085921. 13

Honkela, T. (1997). *Self-organizing maps in natural language processing*. PhD thesis, Helsinki University of Technology. 52

Houngbo, H. and Mercer, E. R. (2012). Method mention extraction from scientific research papers. In Kay, M. and Boite, C., editors, *Proceedings of COLING 2012: Technical Papers*, pages 1211–1222, Mumbai, India. The COLING 2012 Organizing Committee. 95

Hsu, L.-F. (2010). Mining on terms extraction from Web news. In Pan, J.-S., Chen, S.-M., and Nguyen, N., editors, *Computational Collective Intelligence. Technologies and Applications*, volume 6421 of *Lecture Notes in Computer Science*, pages 188–194. Springer Berlin Heidelberg, Kaohsiung, Taiwan. 80

Huang, W. and Yin, H. (2012). On nonlinear dimensionality reduction for face recognition. *Image and Vision Computing*, 30(4–5):355–366. 52

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 216–223, Sapporo, Japan. Association for Computational Linguistics. 77, 80, 81, 83

Indyk, P. (2000). Stable distributions, pseudorandom generators, embeddings and data stream computation. In *Proceedings: 41st Annual Symposium on Foundations of Computer Science*, pages 189–197, Redondo Beach, California. IEEE Computer Society. 125, 126, 127

Indyk, P. (2006). Stable distributions, pseudorandom generators, embeddings, and data stream computation. *Journal of the ACM*, 53(3):307–323. 122, 126

International Organization for Standardization (2000). ISO 1087-1:2000(en) terminology — vocabulary — part 1: Theory and application.

Irsoy, O. and Cardie, C. (2014). Deep recursive neural networks for compositionality in language. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2096–2104. Curran Associates, Inc. 45

Ittoo, A. and Bouma, G. (2013). Term extraction from sparse, ungrammatical domain-specific documents. *Expert Systems with Applications*, 40(7):2530–2540. 88

Ittoo, A., Maruster, L., Wortmann, H., and Bouma, G. (2010). Textractor: A framework for extracting relevant domain concepts from irregular corporate textual datasets. In Abramowicz, W. and Tolksdorf, R., editors, *Business Information Systems*, volume 47 of *Lecture Notes in Business Information Processing*, pages 71–82. Springer. 79

Jackson, J. E. (2004). *A User's Guide to Principal Components*, chapter Scaling of Data, pages 63–79. John Wiley and Sons, Inc. 50

Jacquemin, C. and Tzoukermann, E. (1999). NLP for term variant extraction: Synergy between morphology, lexicon, and syntax. In Strzalkowski, T., editor, *Natural Language Information Retrieval*, volume 7 of *Text, Speech and Language Technology*, pages 25–74. Springer Netherlands. 81

Jain, A., Duin, R. P. W., and Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37. 62

Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P., and Suchomel, V. (2014). Finding terms in corpora for many languages with the Sketch Engine. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 53–56, Gothenburg, Sweden. Association for Computational Linguistics. 81

Johnson, W. and Lindenstrauss, J. (1984). Extensions of Lipschitz mappings into a Hilbert space. In Beals, R., Beck, A., Bellow, A., and Hajian, A., editors, *Conference on Modern Analysis and Probability (1982: Yale University)*, volume 26 of *Contemporary Mathematics*, pages 189–206. American Mathematical Society. 110, 135

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21. 35, 85

Jones, K. S. (1986). *Synonymy And Semantic Classification*, volume 1 of *Edinburgh Information Technology Series*. Edinburgh University Press. The book comprises Jones's Ph.D. thesis, which is approved in 1964 at the University of Cambridge. 6

Jones, K. S. and Galliers, J. R. (1995). *Evaluating Natural Language Processing Systems: An Analysis and Review*, volume 1083 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag Berlin Heidelberg, Secaucus, NJ, USA, 1 edition. 96

Jones, K. S. and Kay, M. (1973). *Linguistics and information science*. Academic Press. 12

Jones, M. N. and Mewhort, D. J. K. (2007). Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, 114(1):1–37. 37, 153

Jones, W. P. and Furnas, G. W. (1987). Pictures of relevance: A geometric analysis of similarity measures. *Journal of the Association for Information Science and Technology*, 38(6):420–442. 58, 59

Jonnalagadda, S., Cohen, T., Wu, S., and Gonzalez, G. (2012). Enhancing clinical concept extraction with distributional semantics. *Journal of Biomedical Informatics*, 45(1):129–140. 38

Jonnalagadda, S., Leaman, R., Cohen, T., and Gonzalez, G. (2010). A distributional semantics approach to simultaneous recognition of multiple classes of named entities. In Gelbukh, A., editor, *Computational Linguistics and Intelligent Text Processing*, volume 6008 of *Lecture Notes in Computer Science*, pages 224–235. Springer Berlin Heidelberg. 40

Judea, A., Schütze, H., and Bruegmann, S. (2014). Unsupervised training set generation for automatic acquisition of technical terminology in patents. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 290–300, Dublin, Ireland. Dublin City University and Association for Computational Linguistics. 95

Jurgens, D., Mohammad, S., Turney, P., and Holyoak, K. (2012). SemEval-2012 Task 2: Measuring degrees of relational similarity. In *First Joint Conference on Lexical and Computational Semantic (*SEM)*, pages 356–364, Montréal, Canada. Association for Computational Linguistics. 36

Jurgens, D. and Stevens, K. (2009). Event detection in blogs using temporal random indexing. In Constantin Orasan, L. H. and Forascu, C., editors, *Proceedings of the Workshop on Events in Emerging Text Types*, pages 9–16, Borovets, Bulgaria. Association for Computational Linguistics. 118, 121

Jurgens, D. and Stevens, K. (2010). HERMIT: Flexible clustering for the SemEval-2 WSI task. In *SemEval 2010: 5th International Workshop on Semantic Evaluation: Proceedings of the Workshop*, pages 359–362, Uppsala, Sweden. Association for Computational Linguistics. 38, 118

Justeson, J. S. and Katz, S. M. (1995). Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1):9–27. 77, 78

Kageura, K. (1999). On the study of dynamics of terminology: A proposal of a theoretical framework. *Research Bulletin of the NACSIS*, 11:1–10. 68

Kageura, K. and Umino, B. (1996). Methods of automatic term recognition: A review. *Terminology*, 3(2):259–289. 71, 84, 85

Kamp, H. (2002). A theory of truth and semantic representation. In Portner, P. H. and Partee, B. H., editors, *Formal Semantics: The Essential Readings*. Wiley-Blackwell. 22

Kanejiya, D., Kumar, A., and Prasad, S. (2003). Automatic evaluation of students' answers using syntactically enhanced LSA. In Burstein, J. and Leacock, C., editors, *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing*, pages 53–60, Edmonton, Canada. Association for Computational Linguistics. 40

Kanerva, P. (1993). Sparse Distributed Memory and related models. In Hassoun, M. H., editor, *Associative neural memories: theory and implementation*, chapter 3, pages 50–76. Oxford University Press, New York, NY, USA. 109

Kanerva, P., Kristoferson, J., and Holst, A. (2000). Random indexing of text samples for latent semantic analysis. In Gleitman, L. R. and Josh, A. K., editors, *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, page 1036, Mahwah, New Jersey. Erlbaum. 42, 51, 109, 112, 117, 118

Karlgren, J. (1993). Sublanguages and registers: A note on terminology. *Interacting with Computers*, 5(3):348–350. 14

Karlgren, J., Sahlgren, M., Olsson, F., Espinoza, F., and Hamfors, O. (2012). Profiling reputation of corporate entities in semantic space: Notebook for RepLab at CLEF 2012. In *CLEF (Online Working Notes/Labs/Workshop)*. 121

Kaski, S. (1998). Dimensionality reduction by random mapping: Fast similarity computation for clustering. In *The 1998 IEEE International Joint Conference on Neural Networks Proceedings: IEEE Worl Congress on Computational Intelligence*, volume 1, pages 413–418, Alaska, USA. 115, 118

Ke, Q. and Kanade, T. (2003). Robust subspace computation using ℓ_1 norm. Technical Report CMU-CS-03-172, Carnegie Mellon University. 124

Khoo, C. S. and Na, J.-C. (2006). Semantic relations in information science. *Annual Review of Information Science and Technology*, 40(1):157–228. 90

Kiela, D. and Clark, S. (2013). Detecting compositionality of multi-word expressions using nearest neighbours in vector space models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1427–1432, Seattle, Washington, USA. Association for Computational Linguistics. 202

Kilgarriff, A. (1996). Which words are particularly characteristic of a text? A survey of statistical approaches. Technical Report ITRI-96-08, University of Brighton, Brighton, UK. 86

Kilgarriff, A. (1997). "I don't believe in word senses". *Computers and the Humanities*, 31(2):91–113. 91

Kilgarriff, A. (2001). Comparing corpora. *International journal of Corpus Linguistics*, 6(1):97–133. 88

Kilgarriff, A. (2005). Language is never, ever, ever, random. *Corpus Linguistics and Linguistic Theory*, 1(2):1613–7027. 87

Kilgarriff, A. (2006). Word senses. In Agirre, E. and Edmonds, P., editors, *Word Sense Disambiguation: Algorithms and Applications*, volume 33 of *Text, Speech and Language Technology*, pages 29–46. Springer Netherlands. 24

Kim, J.-D., Ohta, T., Tateisi, Y., and Tsujii, J. (2003). GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(Suppl 1):i180–i182. 94, 97, 142, 148

Kim, J.-D., Ohta, T., Tateisi, Y., and Tsujii, J. (2006). GENIA corpus manual. Technical Report TR-NLP-UT-2006-1, Tsujii Laboratory. 148

Kim, J.-D., Ohta, T., Tsuruoka, Y., Tateisi, Y., and Collier, N. (2004). Introduction to the bio-entity recognition task at JNLPBA. In Collier, N., Ruch, P., and Nazarenko, A., editors, *JNLPBA: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications*, pages 70–75, Geneva, Switzerland. Association for Computational Linguistics. 8, 92, 94, 141, 148, 163

Kit, Chunyu, and Liu, X. (2008). Measuring mono-word termhood by rank difference via corpus comparison. *Terminology*, 14(2):204–229. 100

Kittredge, R. and Lehrberger, J. (1982). Variation and homogeneity of sublanguages. In *Sublanguage: Studies of Language in Restricted Semantic Domains*. Walter de Gruyter. 13

Knoth, P., Schmidt, M., Smrz, P., and Zdrahal, Z. (2009). Towards a framework for comparing automatic term recognition methods. In *Conference Znalosti 2009*, Brno, Czech Republic. 88

Koan (2015). In *Oxford Dictionary of English*. Oxford University Press. Retrieved June 20, 2015, from from `http://www.oxforddictionaries.com/definition/english/koan`. xi

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480. 52

Korkontzelos, I., Klapaftis, I. P., and Manandhar, S. (2008). Reviewing and evaluating automatic term recognition techniques. In Nordström, B. and Ranta, A., editors, *Advances in Natural Language Processing*, volume 5221 of *Lecture Notes in Computer Science*, pages 248–259. Springer Berlin Heidelberg. 86

Kovačević, A., Konjović, Z., Milosavljević, B., and Nenadic, G. (2012). Mining methodologies from NLP publications: A case study in automatic terminology recognition. *Computer Speech and Language*, 26(2):105–126. 9, 95

Krauthammer, M. and Nenadic, G. (2004). Term identification in the biomedical literature. *Journal of Biomedical Informatics*, 37(6):512–526. 73

Kulis, B. (2013). Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364. 56

Kulkarni, S. and Harman, G. (2011). *An elementary introduction to statistical learning theory*, volume 853 of *Wiley Series in Probability and Statistics*. John Wiley and Sons. 63

Kwak, N. (2008). Principal component analysis based on L1-norm maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(9):1672–1680. 124

Lamp, J. W. and Milton, S. K. (2012). The social life of categories: An empirical study of term categorization. *Applied Ontology*, 7(4):449–470. 8

Landauer, T. K. (2002). On the computational basis of learning and cognition: Arguments from LSA. *The Psychology of Learning and Motivation*, 41:43–84. 153, 154

Landauer, T. K. and Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2):211–240. 24, 29

Lapesa, G. and Evert, S. (2013). Evaluating neighbor rank and distance measures as predictors of semantic priming. In *Proceedings of the Fourth Annual Workshop on Cognitive Modeling and Computational Linguistics*, pages 66–74, Sofia, Bulgaria. Association for Computational Linguistics. 122

Laurence, S. and Margolis, E. (1999). Concepts and cognitive science. In Laurence, S. and Margolis, E., editors, *Concepts: Core Readings*, pages 3–81. MIT Press Cambridge, MA. 71

Lauriston, A. (1995). Criteria for measuring term recognition. In *Proceedings of the Seventh Conference on European Chapter of the Association for Computational Linguistics*, pages 17–22, Dublin, Ireland. Association for Computational Linguistics. 100

Lavelli, A., Califf, M., Ciravegna, F., Freitag, D., Giuliano, C., Kushmerick, N., Romano, L., and Ireson, N. (2008). Evaluation of machine learning-based information extraction algorithms: Criticisms and recommendations. *Language Resources and Evaluation*, 42(4):361–393. 99

Lee, L. (1999). Measures of distributional similarity. In *37th Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, pages 25–32, Maryland, USA. Association for Computational Linguistics. 59, 124

Lee, L. (2001). On the effectiveness of the skew divergence for statistical language analysis. In *Artificial Intelligence and Statistics*, pages 65–72. 58

Lehnert, W., Cardie, C., Fisher, D., McCarthy, J., Riloff, E., and Soderland, S. (1994). Evaluating an information extraction system. *Journal of Integrated Computer-Aided Engineering*, 1(6). 97, 99

Lehrer, A. (1978). Structures of the lexicon and transfer of meaning. *Lingua*, 45(2):95–123. 7

Lenci, A. (2008). Distributional semantics in linguistic and cognitive research. *From Context to Meaning: Distributional Models of the Lexicon in Linguistics and Cognitive Science, Special Issue of the Italian Journal of Linguistics*, 20(1):1–31. 23, 24, 25, 152

Leopold, E. (2005). On semantic spaces. *LDV-Forum (Special Issue on Text Mining)*, 20(1):63–86. 49

L'Homme, M.-C. (2014). Terminologies and taxonomies. In Taylor, J. R., editor, *The Oxford Handbook of the Word*. Oxford University Press. 68, 73, 90

L'Homme, M.-C. and Bernier-Colborne, G. (2012). Terms as labels for concepts, terms as lexical units: A comparative analysis in ontologies and specialized dictionaries. *Applied Ontology*, 7(4):387–400. 4, 90, 140

Li, P. (2007). Very sparse stable random projections for dimension reduction in l_α ($0 \leq \alpha \leq 2$) norm. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 440–449, San Jose, US. ACM. 126, 128

Li, P. (2008). Estimators and tail bounds for dimension reduction in ℓ_α ($0 \leq \alpha \leq 2$) using stable random projections. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 10–19, CA, USA. Association for Computing Machinary and Society for Industrial and Applied Mathematics. 127

Li, P., Church, K. W., and Hastie, T. J. (2006a). Conditional random sampling: A sketch-based sampling technique for sparse data. In Schölkopf, B., Platt, J., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19*, pages 873–880. MIT Press, Cambridge, MA. 136

Li, P., Hastie, T. J., and Church, K. W. (2006b). Very sparse random projections. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 287–296, New York, NY, USA. ACM. 111, 113

Li, P., Hastie, T. J., and Church, K. W. (2007). Nonlinear estimators and tail bounds for dimension reduction in L_1 using Cauchy random projections. *Journal of Machine Learning Research*, 8:2497–2532. 127, 128

Li, P., Samorodnitsk, G., and Hopcroft, J. (2013). Sign Cauchy projections and chi-square kernel. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 2571–2579. Curran Associates, Inc. 122

Lin, D. (1997). Using syntactic dependency as local context to resolve word sense ambiguity. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics: Proceedings of the Confernce*, pages 64–71, Madrid, Spain. Association for Computational Linguistics. 24

Lin, D. (1998a). Automatic retrieval and clustering of similar words. In *COLING-ACL'98: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics: Proceedings of the Conference*, volume 1, pages 768–774, Montreal, Quebec, Canada. Association for Computational Linguistics. 40

Lin, D. (1998b). An information-theoretic definition of similarity. In Shavlik, J. W., editor, *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304, Wisconsin, USA. Morgan Kaufmann Publishers Inc. 59

Lin, D. and Pantel, P. (2001). DIRT – Discovery of inference rules from text. In *KDD-2001: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 323–328, San Francisco, CA, USA. ACM. 24, 40

Linial, N., London, E., and Rabinovich, Y. (1995). The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245. 118

Liu, X. and Kit, C. (2008). An improved corpus comparison approach to domain specific term recognition. In *Proceedings of the 22nd Pacific Asia Conference on Language, Information and Computation: PACLIC 22*, pages 253–261, Cebu City, Philippines. De La Salle University. 88

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. (2002). Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444. 38

Lops, P., de Gemmis, M., Semeraro, G., Musto, C., and Narducci, F. (2013). Content-based and collaborative techniques for tag recommendation: An empirical evaluation. *Journal of Intelligent Information Systems*, 40(1):41–61. 40

Lund, K. and Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, and Computers*, 28(2):203–208. 24, 27, 35, 37, 40, 44

Lupu, M. (2014). On the usability of random indexing in patent retrieval. In Hernandez, N., Jäschke, R., and Croitoru, M., editors, *Graph-Based Representation and Reasoning*, volume 8577 of *Lecture Notes in Computer Science*, pages 202–216. Springer International Publishing. 112

Manning, C. D., Raghavan, P., and Schütze, H. (2009). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, draft april 1, 2009 edition. 43, 183

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press. 22, 86

Margolis, E. and Laurence, S. (2014). Concepts. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Center for the Study of Language and Information (CSLI), spring 2014 edition. 4

Markie, P. (2015). Rationalism vs. Empiricism. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Stanford University, spring 2015 edition. 12

Martin, C. D. and Porter, M. A. (2012). The extraordinary SVD. *The American Mathematical Monthly*, 119(10):838–851. 49

Martin, D. I. and Berry, M. W. (2011). Mathematical foundations behind latent semantic analysis. In Landauer, T. K., McNamara, D. S., Dennis, S., and Kintsch, W., editors, *Handbook of Latent Semantic Analysis*, chapter Mathematical foundations behind latent semantic analysis, pages 35–55. Routledge. 49, 50

Martinez, D. and Agirre, E. (2000). One sense per collocation and genre/topic variations. In *2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP)*, pages 207–215, Hong Kong, China. Association for Computational Linguistics. 5

Matoušek, J. (2008). On variants of the Johnson-Lindenstrauss lemma. *Random Structures and Algorithms*, 33(2):142–156. 111

Mayfield, J., McNamee, P., Harmon, C., Finin, T., and Lawrie, D. (2014). KELVIN: Extracting knowledge from large text collections. In *Natural Language Access to Big Data: Papers from the 2014 AAAI Fall Symposium*, pages 34–41, Arlington, Virginia. AAAI Press. 92

Maynard, D. (2005). Benchmarking ontology-based annotation tools for the Semantic Web. In Cox, S., editor, *Proceedings of the UK e-Science All Hands Conference*, Nottingham, UK. Engineering and Physical Sciences Research Council. 100

Maynard, D. and Ananiadou, S. (2000). Identifying terms by their family and friends. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 530–536, Saarbrucken, Germany. Association for Computational Linguistics. 89

Maynard, D. and Ananiadou, S. (2001). Term extraction using a similarity-based approach. In Bourigault, D., Jacquemin, C., and L'Homme, M.-C., editors, *Recent Advances in Computational Terminology*, volume xviii, pages 261–278. John Benjamins. 76, 95

Maynard, D., Funk, A., and Peters, W. (2009). Using lexico-syntactic ontology design patterns for ontology creation and population. In Blomqvist, E., Sandkuhl, K., Scharffe, F., and Svatek, V., editors, *Workshop on Ontology Patterns: WOP 2009: Papers and Patterns from the ISWC workshop*, volume 516, pages 39–52, Washington D.C., USA. CEUR Workshop Proceedings. 140

Maynard, D., Li, Y., and Peters, W. (2008). NLP techniques for term extraction and ontology population. In Buitelaar, P. and Cimiano, P., editors, *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, volume 167 of *Frontiers in Artificial Intelligence and Applications*, pages 3–26. IOS Press, Amsterdam, The Netherlands. 91, 92, 100

Maynard, D. G. (2000). *Term recognition using combined knowledge sources*. PhD thesis, Manchester Metropolitan University. 78, 85

McIntosh, T. and Curran, J. R. (2009). Reducing semantic drift with bagging and distributional similarity. In *ACL-IJCNLP 2009: Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and 4th International Joint Conference on Natural Language Processing of the AFNLP: Proceedings of the Conference*, volume 1, pages 396–404, Suntec, Singapore. Association for Computational Linguistics and the Asian Federation of Natural Language Processing. 191

McNally, L. (2015). Kinds, descriptions of kinds, concepts, and distributions. Technical report, Universitat Pompeu Fabra. First Presented at Workshop Bridging Formal and Conceptual Semantics (BRIDGE-14). 7

Mehdad, Y., Moschitti, A., and Zanzotto, F. M. (2010). Syntactic/semantic structures for textual entailment recognition. In *NAACL HLT 2010: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Proceedings of the Main Conference*, pages 1020–1028, Los Angeles, California. Association for Computational Linguistics. 39

Meyers, A., Glass, Z., Grieve-Smith, A., He, Y., Liao, S., and Grishman, R. (2014). Jargon-term extraction by chunking. In Meyers, A., He, Y., and Grishman, R., editors, *Proceedings of the COLING Workshop on Synchronic and Diachronic Approaches to Analyzing Technical Language (SADAATL 2014)*, pages 11–20. Association for Computational Linguistics and Dublin City University. 78

Mihalcea, R. (1998). Semcor. Available for download from `http://web.eecs.umich.edu/~mihalcea/downloads/semcor/semcor3.0.tar.gz`. 142

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781. 45

Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41. 90, 142

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244. 6

Miller, G. A. and Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28. 24

Minsky, M. (1974). A framework for representing knowledge. Artificial Intelligence Lab Publications AIM-306, Massachusetts Institute of Technology, Cambridge, MA, USA. 5

Mitchell, J. and Lapata, M. (2010). Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429. 202

Mitchell, T. M., Shinkareva, S. V., Carlson, A., Chang, K.-M., Malave, V. L., Mason, R. A., and Just, M. A. (2008). Predicting human brain activity associated with the meanings of nouns. *Science*, 320(5880):1191–1195. 39

Mohit, B. (2014). Named entity recognition. In Zitouni, I., editor, *Natural Language Processing of Semitic Languages*, Theory and Applications of Natural Language Processing, pages 221–245. Springer. 94

Morris, J. and Hirst, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48. 24

Moskovich, W. (1976). Perspective paper: Quantitative linguistics. In *Natural Language in Information Science*, pages 57–74. Skriptor. 12

Mostow, J., Chang, K.-M., and Nelson, J. (2011). Toward exploiting EEG input in a reading tutor. In Biswas, G., Bull, S., Kay, J., and Mitrovic, A., editors, *Artificial Intelligence in Education*, volume 6738 of *Lecture Notes in Computer Science*, pages 230–237, Berlin, Heidelberg. Springer Berlin Heidelberg. 39

Murphy, B., Talukdar, P., and Mitchell, T. (2012). Selecting corpus-semantic models for neurolinguistic decoding. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics: Volume 1: Proceedings of the main conference and the shared task*, pages 114–123, Montréal, Canada. Association for Computational Linguistics. 38

Murphy, G. L. (2002). *The Big Book of Concepts*. The MIT Press. 5

Musto, C., Narducci, F., Lops, P., Semeraro, G., Gemmis, M., Barbieri, M., Korst, J., Pronk, V., and Clout, R. (2012). Enhanced semantic TV-show representation for personalized electronic program guides. In Masthoff, J., Mobasher, B., Desmarais, M., and Nkambou, R., editors, *User Modeling, Adaptation, and Personalization*, volume 7379 of *Lecture Notes in Computer Science*, pages 188–199. Springer Berlin Heidelberg. 118

Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Special issue of Linguisticae Investigationes*, 30(1):3–26. 94

Nakagawa, H. (2001a). Automatic term recognition based on statistics of compound nouns. *Terminology*, 6(2):195–210. 74, 81, 85

Nakagawa, H. (2001b). Disambiguation of single noun translations extracted from bilingual comparable corpora. *Terminology*, 7(1):63–83. 80

Nallapati, R. (2004). Discriminative models for information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 64–71, Sheffield, UK. ACM. 29

Navigli, R. and Ponzetto, S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250. 28, 95

Nevin, B., editor (2002). *The Legacy of Zellig Harris: Language and Information Into the 21st Century*, volume 1: Philosophy of science, syntax and semantics of *Amsterdam Studies in the Theory and History of Linguistic Sc*. John Benjamins Publishing Company. 12

Nevin, B. E. (1984). [review of the book *A Grammar of English on Mathematical Principles*, by Zellig Harris]. *Computational Linguistics*, 10:203–211. Formerly the American Journal of Computational Linguistics. 13

Nigel, C. N., Collier, N., and Tsujii, J. (1999). Automatic term identification and classification in biology texts. In *Proceedings of the 5th Natural Language Pacific Rim Symposium (NLPRS'99)*, pages 369–374, Beijing, China. 8, 9, 73, 92, 95

Nimb, S., Pedersen, B. S., Braasch, A., Sorensen, N. H., and Troelsgard, T. (2013). Enriching a wordnet from a thesaurus. In Borin, L., Fjeld, R. V., Forsberg, M., Nimb, S., Nugues, P., and Pedersen, B. S., editors, *Proceedings of the Workshop on Lexical Semantic Resources for NLP at NODALIDA 2013*, volume 19 of *NEALT Proceedings Series*, Oslo, Norway. Linkoping University Electronic Press. 7

Padó, S. and Lapata, M. (2007). Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199. 38, 40

Palmer, D. D. (2010). Text pre-processing. In Indurkhya, N. and Damerau, F. J., editors, *Handbook of Natural Language Processing*, Chapman and Hall/CRC Machine Learning & Pattern Recognition. CRC Press, Taylor and Francis Group, Boca Raton, FL, 2nd edition. ISBN 978-1420085921. 41

Pantel, P. (2005). Inducing ontological co-occurrence vectors. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 125–132, Ann Arbor, Michigan. Association for Computational Linguistics. 24

Pantel, P., Crestan, E., Borkovsky, A., Popescu, A.-M., and Vyas, V. (2009). Web-scale distributional similarity and entity set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 938–947, Singapore. Association for Computational Linguistics. 169

Paradis, C. (2012). Lexical semantics. In Chapelle, C., editor, *The Encyclopedia of Applied Linguistics*. Blackwell Publishing Ltd. 25

Park, Y., Byrd, R. J., and Boguraev, B. K. (2002). Automatic glossary extraction: Beyond terminology identification. In Tseng, S.-C., editor, *COLING 2002: Proceedings of the 19th International Conference on Computational Linguistics*, Taipei, Taiwan. Association for Computational Linguistics and Chinese Language Processing. 78

Partee, B. H. (2011). Formal semantics: Origins, issues, early impact. In Glanzberg, M., Partee, B. H., and Skilters, J., editors, *Baltic International Yearbook of Cognition, Logic and Communication*, volume 6 of *Formal Semantics and Pragmatics: Discourse, Context, and Models*, pages 1–52. New Prairie Press, Manhattan, KS. 22

Pavoine, S., Vallet, J., Dufour, A.-B., Gachet, S., and Daniel, H. (2009). On the challenge of treating various types of variables: Application for improving the measurement of functional diversity. *Oikos*, 118(3):391–402. 55

Pearson, J. (1998). *Terms in Context*, volume 1 of *Studies in Corpus Linguistics*. John Benjamins, Amsterdam, The Netherlands. 14, 88

Pecina, P. (2008). A machine learning approach to multiword expression extraction. In *Proceedings of the LREC Workshop: Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 54–57, Marrakech, Morocco. European Language Resources Association. 93

Pecina, P. (2010). Lexical association measures and collocation extraction. *Language Resources and Evaluation*, 44(1-2):137–158. 84

Périnet, A. and Hamon, T. (2014a). Distributional context generalisation and normalisation as a mean to reduce data sparsity: Evaluation of medical corpora. In Przepiorkowski, A. and Ogrodniczuk, M., editors, *Advances in Natural Language Processing*, volume 8686 of *Lecture Notes in Computer Science*, pages 128–135. Springer International Publishing. 47

Périnet, A. and Hamon, T. (2014b). *Proceedings of the 4th International Workshop on Computational Terminology (Computerm)*, chapter Generalising and Normalising Distributional Contexts to Reduce Data Sparsity: Application to Medical Corpora, pages 1–10. Association for Computational Linguistics and Dublin City University. 44, 47

Pinnis, M., Ljubešić, N., Ştefănescu, D., Skadiņa, I., Tadić, M., and Gornostay, T. (2012). Term extraction, tagging, and mapping tools for under-resourced languages. In de CeaMari Carmen Suarez-Figueroa Raul Garcia-Castro Elena Montiel-Ponsoda, G. A., editor, *Proceedings of the 10th Conference on Terminology and Knowledge Engineering: New frontiers in the constructive symbiosis of terminology and knowledge engineering*, pages 193–208, Spain, Madrid. 77

Plank, B. and Moschitti, A. (2013). Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *51st Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, pages 1498–1507, Sofia, Bulgaria. Association for Computational Linguistics. 39

Polajnar, T. and Clark, S. (2014). Improving distributional semantic vectors through context selection and normalisation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 230–238, Gothenburg, Sweden. Association for Computational Linguistics. 112

Pollack, J. B. (1990). Recursive distributed representations. *Artificial Intelligence*, 46(1–2):77–105. 118

Pustejovsky, J., Anick, P., and Bergler, S. (1993). Lexical semantic techniques for corpus analysis. *Computational Linguistics*, 19(2):331–358. 90

QasemiZadeh, B. (2010). Towards technology structure mining from text by linguistics analysis. Tech Report DERI TR 010-02-15, Digital Enterprise Research Institute, Lower Dangan, Galway. 143

QasemiZadeh, B. (2015). Random indexing revisited. In Biemann, C., Handschuh, S., Freitas, A., Meziane, F., and Metais, E., editors, *Natural Language Processing and Information Systems*, volume 9103 of *Lecture Notes in Computer Science*, pages 437–442. Springer International Publishing. 105

QasemiZadeh, B., Buitelaar, P., Chen, T. Q., and Bordea, G. (2012). Semi-supervised technical term tagging with minimal user feedback. In Calzolari, N., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 617–621, Istanbul, Turkey. European Language Resources Association (ELRA). 95

QasemiZadeh, B. and Handschuh, S. (2015). Random indexing explained with high probability. In Kral, P. and Matousek, V., editors, *Text, Speech and Dialogue (TSD)*, volume 9302 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 480–489, Pilsen, Czech. Springer International Publishing Switzerland. 105

QasemiZadeh, B., Ilkhani, A., and Ganjeii, A. (2006). Adaptive language independent spell checking using intelligent traverse on a tree. In *2006 IEEE Conference on Cybernetics and Intelligent Systems*, pages 1–6, Bangkok, Thailand. IEEE. 205

Rao, D., McNamee, P., and Dredze, M. (2013). Entity linking: Finding extracted entities in a knowledge base. In Poibeau, T., Saggion, H., Piskorski, J., and Yangarber, R., editors, *Multi-source, Multilingual Information Extraction and Summarization*, Theory and Applications of Natural Language Processing, pages 93–115. Springer Berlin Heidelberg. 73

Rapp, R. (2003). Word sense discovery based on sense descriptor dissimilarity. In *MT Summit IX: Proceedings of the Ninth Machine Translation Summit*, pages 23–27, New Orleans, USA. 27

Rayson, P., Berridge, D., and Francis, B. (2004). Extending the Cochran rule for the comparison of word frequencies between corpora. In G., P., C., F., and A., D., editors, *Proceedings of the 7th International Conference on Statistical Analysis of Textual Data (JADT 2004)*, volume II, pages 926–936, Louvain-la-Neuve, Belgium. Presses universitaires de Louvain. 86

Rayson, P. and Garside, R. (2000). Comparing corpora using frequency profiling. In *WCC '00: Proceedings of the Workshop on Comparing Corpora*, volume 9, pages 1–6, Hong Kong. Association for Computational Linguistics. 88

Recchia, G., Sahlgren, M., Kanerva, P., and Jones, M. N. (2015). Encoding sequential information in semantic space models: Comparing holographic reduced representation and random permutation. *Computational Intelligence and Neuroscience*, 2015:1–18. 153

Resnik, P. (1993). *Selection and information: a class-based approach to lexical relationships*. PhD thesis, University of Pennsylvania. 6, 7

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, volume 1, pages 448–453, Montreal, Quebec, Canada. Morgan Kaufmann Publishers Inc. 28

Rijsbergen, C. J. V. (1977). A theoretical basis for the use of co-cccurrence data in information retrieval. *Journal of Documentation*, 33(2):106–119. 142

Rindflesch, T. C. and Fiszman, M. (2003). The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text. *Journal of Biomedical Informatics*, 36(6):462–477. Unified Medical Language System. 206

Rindflesch, T. C., Rajan, J. V., and Hunter, L. (2000). Extracting molecular binding relationships from biomedical text. In *6th Conference on Applied Natural Language Processing: Proceedings of the Conference (ANLP-2000)*, pages 188–195, Seattle, Washington. Association for Computational Linguistics. 82

Riordan, B. and Jones, M. N. (2011). Redundancy in perceptual and linguistic experience: Comparing feature-based and distributional models of semantic representation. *Topics in Cognitive Science*, 3(2):303–345. 30

Ritter, H. and Kohonen, T. (1989). Self-organizing semantic maps. *Biological Cybernetics*, 61(4):241–254. 118

Roark, B. and Charniak, E. (1998). Noun-phrase co-occurence statistics for semi-automatic semantic lexicon construction. In *COLING-ACL'98: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics: Proceedings of the Conference*, volume 2, pages 1110–1116, Montréal, Quebec, Canada. Morgan Kaufmann Publishers. 98

Roddenberry, G. (1965-Now). Star trek. American science fiction entertainment franchise. 4

Roller, S. and Schulte im Walde, S. (2013). A multimodal LDA model integrating textual, cognitive and visual modalities. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1146–1157, Seattle, Washington, USA. Association for Computational Linguistics. 39

Rubenstein, H. and Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633. 24

Russell, B. (2014). *The Problems of Philosophy*. Bibliotech Press. Originally published by Oxford University Press in 1912. 26

Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, USA, 2nd edition. 50

Sager, J. C. (1990). *A practical course in terminology processing*. John Benjamins Publishing. 68, 71

Sager, N. (1975). Sublanguage grammars in science information processing. *Journal of the American Society for Information Science*, 26:10–16. 13

Sahlgren, M. (2005). An introduction to random indexing. Technical report, Swedish ICT (SICS). Retrived from `https://www.sics.se/~mange/papers/RI_intro.pdf`. 46, 51, 109, 112, 117

Sahlgren, M. (2006). *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. PhD thesis, Stockholm University. 10, 23, 24, 29, 36, 37, 46, 49, 51, 117, 144

Sahlgren, M. (2008). The distributional hypothesis. *Italian Journal of Linguistics*, 20(1):33–54. 15, 25

Sahlgren, M., Holst, A., and Kanerva, P. (2008). Permutations as a means to encode order in word space. In Sloutsky, V., Love, B., and Mcrae, K., editors, *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, pages 1300–1305. Cognitive Science Society, Austin, TX. 37, 153, 154

Sahlgren, M. and Karlgren, J. (2005). Automatic bilingual lexicon acquisition using random indexing of parallel corpora. *Natural Language Engineering*, 11(3):327–341. 117

Sahlgren, M. and Karlgren, J. (2009). Terminology mining in social media. In *CIKM'09: Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 405–414, Hong Kong, China. ACM. 121

Sahlgren, M., Karlgren, J., Coster, R., and Jorvinen, T. (2003). SICS at CLEF 2002: Automatic query expansion using random indexing. In Peters, C., Braschler, M., and Gonzalo, J., editors, *Advances in Cross-Language Information Retrieval*, volume 2785 of *Lecture Notes in Computer Science*, pages 311–320. Springer Berlin Heidelberg. 44

Salton, G. (1992). The state of retrieval system evaluation. *Information processing & management*, 28(4):441–449. 85

Salton, G., Wong, A., and Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620. xxi, 34, 35, 40, 107

Schone, P. and Jurafsky, D. (2001). Is knowledge-free induction of multiword unit dictionary headwords a solved problem? In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, Pittsburgh, PA USA. Association for Computational Linguistics. 98

Schultz, M. and Joachims, T. (2004). Learning a distance metric from relative comparisons. In Thrun, S., Saul, L., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA. 56

Schütze, H. (1993). Word space. In Hanson, S., Cowan, J., and Giles, C., editors, *Advances in Neural Information Processing Systems 5 (NIPS 1992)*, pages 895–902, San Francisco, CA, USA. Morgan-Kaufmann. 10, 144

Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123. 42

Schütze, H. and Pedersen, J. (1995). Information retrieval based on word senses. In *Proceedings: Fourth Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Nevada, USA. 24

Séaghdha, D. O. and Korhonen, A. (2011). Probabilistic models of similarity in syntactic context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1047–1057, Edinburgh, Scotland, UK. Association for Computational Linguistics. 38, 40

Seretan, V., Nerima, L., and Wehrli, E. (2004). A tool for multi-word collocation extraction and visualization in multilingual corpora. In Williams, G. and Vessier, S., editors, *Proceedings of the Eleventh EURALEX International Congress, EURALEX 2004*, volume II, pages 755–766, Lorient, France. Universite de Bretagne. 82

Settles, B. (2005). ABNER: An open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics*, 21(14):3191–3192. 162

Shepard, R. N. (1987). Toward a universal law of generalization for psychological science. *Science*, 237(4820):1317–1323. 64

Shimizu, N., Hagiwara, M., Ogawa, Y., Toyama, K., and Nakagawa, H. (2008). Metric learning for synonym acquisition. In *Coling 2008: 22nd International Conference on Computational Linguistics: Proceedings of the Conference*, pages 793–800, Manchester, UK. Association for Computational Linguistics. 58

Sichel, H. S. (1975). On a distribution law for word frequencies. *Journal of the American Statistical Association*, 70(351):542–547. 50

Silberer, C., Ferrari, V., and Lapata, M. (2013). Models of semantic representation with visual attributes. In *51st Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, pages 572–582, Sofia, Bulgaria. Association for Computational Linguistics. 39

Sinclair, J. (1996). Preliminary recommendations on corpus typology. Technical Report EAG–TCWG–CTYP/P, Expert Advisory Group on Language Engineering Standards (EAGLES). 14, 68

Sinclair, J., Jones, S., and Daley, R. (1970/2004). *English Collocation Studies: The OSTI Report*. London: Continuum (Originally mimeo 1970). 24

Singhal, A., Buckley, C., and Mitra, M. (1996). Pivoted document length normalization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29, New York, NY, USA. ACM. 44

Snaider, J. (2012). *Integer Sparse Distributed Memory and Modular Composite Representation*. PhD thesis, Computer Science. 117

Spasić, I. and Ananiadou, S. (2004). Using automatically learnt verb selectional preferences for classification of biomedical terms. *Journal of Biomedical Informatics*, 37(6):483–497. 94, 95

Spasic, I. and Ananiadou, S. (2005). A flexible measure of contextual similarity for biomedical terms. In Altman, R. B., Dunker, A. K., Hunter, L., Jung, T. A., and Klein, T. E., editors, *Pacific Symposium on Biocomputing 2005*, pages 197–208, Hawaii, USA. World Scientific. 100

Stanford, K. (2013). Underdetermination of scientific theory. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Winter 2013 edition. 27

Stein, B. (2007). Principles of hash-based text retrieval. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 527–534, Amsterdam, The Netherlands. ACM. 113

Stubbs, M. (2009). Memorial article: John Sinclair (1933–2007): The search for units of meaning: Sinclair on empirical semantics. *Applied Linguistics*, 30(1):115–137. 24

Tanev, H. and Magnini, B. (2008). Weakly supervised approaches for ontology population. In Buitelaar, P. and Cimiano, P., editors, *Ontology Learning and Population: Bridging the Gap Between Text and Knowledge*, volume 167 of *Frontiers in Artificial Intelligence and Applications*, pages 129–143. IOS Press, Amsterdam, The Netherlands. 92

Taylor, A., Marcus, M., and Santorini, B. (2003). The Penn treebank: An overview. In Abeillé, A., editor, *Treebanks*, volume 20 of *Text, Speech and Language Technology*, pages 5–22. Springer Netherlands. 78

Temnikova, I., Jr., W. A. B., Hailu, N. D., Nikolova, I., Mcenery, T., Kilgarriff, A., Angelova, G., and Cohen, K. B. (2014). Sublanguage corpus analysis toolkit: A tool for assessing the representativeness and sublanguage characteristics of corpora. In Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 1714–1718, Reykjavik, Iceland. European Language Resources Association. 14

Term (2014). In *Oxford Dictionary of English*. Oxford University Press. Retrieved June 20, 2015, from from `http://www.oxforddictionaries.com/definition/english/term`. 68

Thater, S., Fürstenau, H., and Pinkal, M. (2010). Contextualizing semantic representations using syntactically enriched vector models. In *ACL 2010: 48th Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, pages 948–957, Uppsala, Sweden. Association for Computational Linguistics. 38

The EAGLES Evaluation Working Group (1996). Evaluation of natural language processing systems. FINAL REPORT EAGLES DOCUMENT EAG-EWG-PR.2, Expert Advisory Group on Language Engineering. 96

Tognini-Bonelli, E. (2001). *Corpus Linguistics at Work*, volume 6 of *Studies in Computational Linguistics*. John Benjamins. 12

Toral, A. and Munoz, R. (2006). A proposal to automatically build and maintain gazetteers for named entity recognition by using Wikipedia. In *Proceedings of the ACL 2006 Workshop on New Text Wikis and Blogs and Other Dynamic Text Sources*, Trento, Italy. Association for Computational Linguistics. 82

Trask, L. R. (2013). *A Dictionary of Grammatical Terms in Linguistics*. Routledge. First published 1992. 5

Trier, J. (1934). Das sprachliche feld: Eine auseinandersetzung. *Neue Fachbuecher fuer Wissenschaft und Jugendbildung*, 10:428–449. 13

Tsvetkov, Y. and Wintner, S. (2014). Identification of multiword expressions by combining multiple linguistic information sources. *Computational Linguistics*, 40(2):449–468. 87

Turney, P. D. and Littman, M. L. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4):315–346. 45

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188. 9, 23, 36, 41, 42, 43, 46, 49

Turtle, H. R. and Croft, W. B. (1992). A comparison of text retrieval models. *Computer Journal*, 35(3):279–290. 29

Van der Maaten, L. J. P., Postma, E. O., and Van den Herik, H. J. (2009). Dimensionality reduction: A comparative review. Technical Report TiCC TR 2009-005, Tilburg University. 52

Vivaldi, J., Màrquez, L., and Rodríguez, H. (2001). Improving term extraction by system combination using boosting. In De Raedt, L. and Flach, P., editors, *Machine Learning: ECML 2001*, volume 2167 of *Lecture Notes in Computer Science*, pages 515–526, London, UK. Springer Berlin Heidelberg. 93

Vivaldi, J. and Rodríguez, H. (2007). Evaluation of terms and term extraction systems: A practical approach. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 13:225–248. 100

Weeds, J., Clarke, D., Reffin, J., Weir, D., and Keller, B. (2014a). Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2249–2259, Dublin, Ireland. Dublin City University and Association for Computational Linguistics. 205

Weeds, J., Dowdall, J., Schneider, G., Keller, B., and Weir, D. (2005). Using distributional similarity to organise BioMedical terminology. *Terminology*, 11(1):3–4. 91, 124, 155

Weeds, J., Weir, D., and McCarthy, D. (2004). Characterising measures of lexical distributional similarity. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, Geneva, Switzerland. Association for Computational Linguistics. 58, 204

Weeds, J., Weir, D., and Reffin, J. (2014b). Distributional composition using higher-order dependency vectors. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 11–20, Gothenburg, Sweden. Association for Computational Linguistics. 38

Wermter, J. and Hahn, U. (2005). Finding new terminology in very large corpora. In *Proceedings of the 3rd International Conference on Knowledge Capture*, Alberta, Canada. ACM Press. 85

Wheeler, E. S. (1983). [review of the book *A Grammar of English on Mathematical Principles*, by Zellig Harris]. *Computers and the Humanities*, 17(2):88–92. 13

Widdows, D. (2003). Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *HLT-NAACL 2006: Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics: Proceedings of the Main Conference*, pages 197–204, New York, USA. Association for Computational Linguistics. 38, 40

Widdows, D. (2004). *Geometry and Meaning*. Number 172 in CSLI Lecture Notes. CSLI Publications, Stanford, CA. 10, 29

Wilks, Y. A. and Brewster, C. A. (2009). *Natural Language Processing as a Foundation of the Semantic Web*, volume 1 of *Foundation and Trends® in Web Science*. now Publishing Inc. 5

Wilks, Y. A. and Tait, J. I. (2005a). A retrospective view of synonymy and semantic classification. In Tait, J. I., editor, *Charting a New Course: Natural Language Processing and Information Retrieval: Essays in Honour of Karen Sparck Jones*, volume 16 of *The Kluwer International Series on Information Retrieval*, pages 1–11. Springer Netherlands. 5, 6

Wilks, Y. A. and Tait, J. I. (2005b). A retrospective view of synonymy and semantic classification. In Tait, J. I., editor, *Charting a New Course: Natural Language Processing and Information Retrieval*, volume 16 of *The Kluwer International Series on Information Retrieval*, pages 1–11. Springer Netherlands, 1 edition. 35

William J. Gilbert, W. K. N. (2004). *Modern Algebra with Applications*. John Wiley & Sons, Inc., second edition. 33

Wilson, A. and McEnery, T. (1996). *Corpus Linguistics*. Edinburgh Textbooks in Empirical Linguistics. Edinburgh University Press, 2nd edition. 12

Wong, W. (2009). Determination of unithood and termhood for term recognition. In Song, M. and Wu, Y.-F. B., editors, *Handbook of Research on Text and Web Mining Technologies*, chapter 30, pages 500–529. IGI Global. 81

Wong, W., Liu, W., and Bennamoun, M. (2012). Ontology learning from text: A look back and into the future. *ACM Computing Surveys*, 44(4):20:1–20:36. 95, 140

Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *32nd Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, pages 133–138, New Mexico, USA. Association for Computational Linguistics. 89

Wüster, E. (1974). Die allgemeine Terminologielehre–ein Grenzgebiet zwischen Sprachwissenschaft, Logik, Ontologie, Informatik und den Sachwissenschaften. *Linguistics*, 12(119):61–106. 69

Xing, E. P., Ng, A. Y., Jordan, M. I., and Russell, S. (2002). Distance metric learning, with application to clustering with side-information. In Becker, S., Thrun, S., and Obermayer, K., editors, *Advances in Neural Information Processing Systems 15*, pages 505–512. MIT Press. 56

Yamamoto, K. and Asakura, T. (2010). Even unassociated features can improve lexical distributional similarity. In *Proceedings of the Second Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010)*, pages 32–39, Beijing, China. Coling 2010 Organizing Committee. 45

Yamamoto, K., Kudo, T., Konagaya, A., and Matsumoto, Y. (2003). Protein name tagging for biomedical annotation in text. In Ananiadou, S. and Tsujii, J., editors, *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pages 65–72, Sapporo, Japan. Association for Computational Linguistics. 94

Yang, C. (2013). Who's afraid of George Kingsley Zipf? or: Do children and chimps have language? *Significance: The Magazine of the Royal Statistical Society and the American Statistical Soceity*, 10(6):29–34. 46

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1-2):69–90. 65

Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In Fisher, D. H., editor, *Proceedings of the Fourteenth International Conference on Machine Learning (ICML '97)*, pages 412–420, Tennessee, USA. Morgan Kaufmann. 47

Yangarber, R., Grishman, R., Tapanainen, P., and Huttunen, S. (2000). Automatic acquisition of domain knowledge for information extraction. In *The 18th Conference on Computational Linguistics: Proceedings of the Conference*, volume 2, pages 940–946, Saarbrucken, Germany. Association for Computational Linguistics. 100

Yangarber, R., Lin, W., and Grishman, R. (2002). Unsupervised learning of generalized names. In *COLING 2002: The 19th International Conference on Computational Linguistics*, Taipei, Taiwan. Association for Computational Linguistics. 73, 94

Yannakoudakis, H. and Briscoe, T. (2012). Modeling coherence in ESOL learner texts. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 33–43, Montreal, Canada. Association for Computational Linguistics. 118

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, pages 189–196, Cambridge, Massachusetts, USA. Association for Computational Linguistics. 191, 192, 193

Yi, K. (2010). A semantic similarity approach to predicting library of congress subject headings for social tags. *Journal of the American Society for Information Science and Technology*, 61(8):1658–1672. 143

Zadeh, B. Q. and Handschuh, S. (2014a). The ACL RD-TEC: A dataset for benchmarking terminology extraction and classification in computational linguistics. In Drouin, P., Grabar, N., Hamon, T., and Kageura, K., editors, *COLING 2014: Computerm 2014: 4th International Workshop on Computational Terminology: Proceedings of the Workshop*, pages 52–63, Dublin, Ireland. Association for Computational Linguistics and Dublin City University. 77, 81, 97, 205

Zadeh, B. Q. and Handschuh, S. (2014b). Evaluation of technology term recognition with random indexing. In Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, Reykjavik, Iceland. European Language Resources Association. ACL Anthology Identifier: L14-1703. 139, 204

Zadeh, B. Q. and Handschuh, S. (2014c). Investigating context parameters in technology term recognition. In Meyers, A., He, Y., and Grishman, R., editors, *Proceedings of the COLING Workshop on Synchronic and Diachronic Approaches to Analyzing Technical Language (SADAATL 2014)*, pages 1–10. Association for Computational Linguistics and Dublin City University. 139, 152, 190, 204

Zadeh, B. Q. and Handschuh, S. (2014d). Random Manhattan indexing. In *25th International Workshop on Database and Expert Systems Applications (DEXA)*, pages 203–208, Munich, Germany. IEEE. 105

Zadeh, B. Q. and Handschuh, S. (2014e). Random Manhattan integer indexing: Incremental L1 normed vector space construction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1713–1723, Doha, Qatar. Association for Computational Linguistics. 105

Zadeh, L. A. (2010). Computing with words and perceptions—A paradigm shift. In Arabnia, H. R., Chiu, S. C., Gravvanis, G. A., Ito, M., Joe, K., Nishikawa, H., and Solo, A. M. G., editors, *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2010)*, pages 3–5, Nevada, USA. CSREA Press. 28

Zeng, D., Liu, K., Lai, S., Zhou, G., and Zhao, J. (2014). Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344. Dublin City University and Association for Computational Linguistics. 41, 45

Zervanou, K. (2010). UvT: The UvT term extraction system in the keyphrase extraction task. In *SemEval 2010: 5th International Workshop on Semantic Evaluation: Proceedings of the Workshop*, pages 194–197, Uppsala, Sweden. Association for Computational Linguistics. 80

Zhang, Z., Iria, J., Brewster, C., and Ciravegna, F. (2008). A comparative evaluation of term recognition algorithms. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., and Tapias, D., editors, *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 2108–2113, Marrakech, Morocco. European Language Resources Association. 87, 98

Zhitomirsky-Geffet, M. and Dagan, I. (2009). Bootstrapping distributional feature vector quality. *Computational Linguistics*, 35(3):435–461. 45

Zweigenbaum, P. and Grabar, N. (1999). Automatic acquisition of morphological knowledge for medical language processing. In Horn, W., Shahar, Y., Lindberg, G., Andreassen, S., and Wyatt, J., editors, *Artificial Intelligence in Medicine*, volume 1620 of *Lecture Notes in Computer Science*, pages 416–420. Springer Berlin Heidelberg. 74

This page is intentionally left blank.