National University of Ireland, Galway

Doctoral Thesis

A Transportable Distributional Semantics Architecture

Author:
Siamak Barzegar

Supervisors:
Dr. Andre Freitas
Dr. Brian Davis

Internal Examiner:
Dr. Colm O'Riordan

External Examiner:
Prof. Wlodek Zadrozny

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Knowledge Discover Unit
Insight Centre For Data Analytics

4th December, 2018

Declaration of Authorship

I, Siamak Barzegar, declare that this thesis titled, "A Transportable Distributional Semantics Architecture" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"Two there are who are never satisfied –
The lover of the world and the lover of knowledge."

— Rumi

NATIONAL UNIVERSITY OF IRELAND, GALWAY

Abstract

Engineering & Informatics
Insight Centre For Data Analytics

Doctor of Philosophy

A Transportable Distributional Semantics Architecture

by Siamak Barzegar

Distributional semantics is built upon the assumption that the context surrounding a given word in text provides important information about its meaning (Distributional hypothesis). A rephrasing of the distributional hypothesis states that words that occur in similar contexts tend to have a similar meaning. Distributional semantics focuses on the construction of a semantic representation of a word based on the statistical distribution of word co-occurrence in texts. Distributional Semantic Models (DSMs) represent co-occurrence patterns under a vector space representation. In recent years, word embedding/distributional semantic models have evolved to become a fundamental component in many natural language processing (NLP) architectures due to their ability of capturing and quantifying semantic associations at scale. Distributional Semantics have been applied for different tasks in NLP area such as finding similar or related phrase/words, the computation of semantic relatedness measures, semantic relation classification and so forth. Distributional semantic models are strongly dependent on the size and the quality of the reference corpora, which embeds the common-sense knowledge necessary to build comprehensive models. While high-quality texts containing large-scale common-sense information and domain-specific information are present in English, such as Wikipedia, other languages may lack sufficient textual support to build comprehensive distributional models. Distributional Semantic Models are also often limited to semantic similarity/relatedness between two entities/terms with no explicit relation type. Often, it is not possible to assign a direct semantic relation between entities. This thesis seeks to analyse transportability aspects (Language and Domain) as and explores both coarse & fine-grained semantics for direct and indirect relation classification using a unified architecture (INDRA) for developing language and domain independent DSM models with advanced (compositional) relation classification capabilities.

Acknowledgements

I would first like thank Prof. Dr. Siegfried Handschuh for given me the opportunity to continue my academic career at *Insight Centre for Data Analytics* by accepting me in his group at Insight (formerly DERI) as a PhD Student and also at Passau University as an Internship Student.

Most Importantly, I would like to thank my supervisors, Dr. Brian Davis and Dr. Andre Freitas for their patience, trust, supportiveness, and mentorship. I want in particular thank my friends and colleagues, Dr. Manel Zarrouk, Waqas Khawaja, Juliano Efson Sales and Macedo Sousa Maia at Knowledge Discovery Unit (KDU) and Digital Libraries and Web Information Systems Unit.

The period I spent in Insight/DSI and Passau University was a great experience, and I am happy to be a part of the history of them. I wish to publicly thank the members of my Graduate Research Committee during my PhD: Prof. Mathieu d'Aquin, Prof. Dietrich Rebholz-Schuhmann, Dr. Mihael Arcan, Dr. Souleiman Hasan and Dr. Paul Buitelaar, for their constructive feedback. and also I would like to sincerely thank the examination committee of my thesis: Prof. Wlodek Zadrozny, the external examiner, and Dr. Colm O'Riordan, the internal examiner.

I was also greatly supported by the Insight administrative and technical staff Claire Browne, Hilda Fitzpatrick, Christiane Leahy-Coen, Donal Carroll and Gerard Conneely.

During my study, I have made many friends which living without their friendships is unimaginable for me: Mahmmoud, Sahar, Amir, Shirin and Soheila.

I want to thank my mother and father, Fariba and Alireza for their encouragement to pursue my postgraduate study and their unlimited love, as well as my brother Babak and sister-in-law, Targol, and my cute nephew, Hanaa.

Finally I owe thanks to a very special person, my wife, Nahid for her continued and unfailing love, support and understanding during my pursuit of PhD degree that made the completion of thesis possible.

The research contributions reported by this thesis was supported by research funded in part from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645425 SSIX and Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

Contents

List of Figures

List of Tables

Chapter 1

Prologue

1.1 Introduction

The creation of real-world Artificial Intelligence (AI) applications is dependent on leveraging a large volume of commonsense knowledge. Simple semantic interpretation tasks such as understanding that if *A is married to B* then *A is the spouse of B* or that *car, vehicle, auto* have very similar meanings are examples of semantic approximation operations/inferences that are present in practically all applications of AI that interpret natural language. Many AI applications depend on being semantically flexible, i.e. coping with the large vocabulary variation that is permitted by natural language. Sentiment Analysis, Question Answering, Information Extraction, Semantic Search and Classification tasks are examples of tasks in which the ability to do semantic approximation is a central requirement. Natural Language Processing (*NLP*) is a sub-field of Artificial Intelligence concerned with bridging the communication divide between computers and humans. *NLP* enables computers to understand and process human language, to get closer to a human-level understanding of language. In order to understand human language it is necessary to interpret not only the linguistic entities (e.g., words and short phrases), but also knowing how they related together semantically. By analysing language for its meaning, *NLP* helps to resolving ambiguity in different languages and different domains and adds useful information for many different tasks such as translation, relation extraction, text entailment, sentiment analysis and question answering. These tasks require a certain amount of understanding of the "meaning" of the linguistic entities. Methods in computational semantics can be used that focus on how to automate the process of constructing and reasoning with meaning representations of linguistic entitles. Formal Semantics and Distributional Semantics are two significant semantic frameworks in *Computational Linguistics*. Formal Semantics study meaning by focusing mostly on grammatical words and the behaviour of the logical words (Aloni and Dekker, 2016). On the other hand, Distributional semantics are statistical and data-driven and focuses on aspects of meaning related to descriptive content (Boleda and Herbelot, 2016). Distributional Semantics Models and Word Vector models have emerged as successful approaches for supporting semantic approximations

due to their ability to build comprehensive semantic approximation models and to their simplicity of representation.

Distributional Semantics is built upon the assumption that the context surrounding a given word in a text provides important information about its meaning (Distributional hypothesis) (Harris, 1954). A rephrasing of the distributional hypothesis states that words that occur in similar contexts tend to have a similar meaning. Distributional semantics focuses on the construction of a semantic representation of a word, based on the statistical distribution of word co-occurrence in texts. Distributional Semantic Models (DSMs) represent co-occurrence patterns under a vector space representation (VSMs) and compute a semantic interpretation of a term as a set of weighted concepts.

Definition (Distributional hypothesis) (Harris, 1954): *"Words that occur in similar contexts tend to have similar meanings"*.

1.2 Distributional Semantic Models (DSMs): A Fundamental Part of the AI Infrastructure

In the last few years, word embedding/distributional semantic models have become a fundamental component for supporting automatic semantic interpretation in many natural language processing (*NLP*) architectures and AI infrastructure as pre-trained-vectors, due to their ability to capture and quantify semantic associations at scale. Distributional Semantics have been applied to different tasks in the `NLP` such as computation of semantic similarity and relatedness between words/phrases and Word Sense Disambiguation (Stevenson and Wilks, 2003) as well as relation classification among others. More recently, DSMs have been associated with many classification tasks in `NLP`, serving as a mechanism for coping with vocabulary variation (Freitas and Curry, 2014).

1.2.1 The Transportability Problem

Distributional semantic models are highly dependent on the size and the quality of the reference corpora, which embeds the commonsense knowledge necessary to build comprehensive models. While high-quality texts containing large-scale commonsense information and domain-specific information are present in English, such as *Wikipedia*, other languages may lack sufficient textual support to build comprehensive distributional models. Furthermore, the majority of discussions surrounding Distributional Semantic Models have concentrated on the construction and the evaluation of models based on open-domain and large-scale corpora and domain specific modes are under-resourced and under-researched. (Sahlgren,

2006; Speer, Chin, and Havasi, 2017; Kiela and Clark, 2014).

1.2.2 Coarse vs. Fine Grained Semantics

Distributional Semantic Models tend to compute only semantic similarity/relatedness between two entities/terms, which can be very informative **(course grained semantics)**. However, this does not provide any information about the **semantic type** of relation between two given entities **(fine grained semantics)**. On the other hand, structured knowledge based systems represent such relationships explicitly, but ignore the semantic similarity between entities and are bounded by the limitation of the human curation/annotation effort. The combination of distributional semantics and structured knowledge-based systems provide rich complementary semantic perspectives (More details are provided in Section 2.2).

1.2.3 Engineering Distributional Semantic Models

Building and consuming specific distributional semantic models require the setting of complex configurations, such as corpus-dependent parameters, distance measures as well as compositional models. Therefore, multiple models can be built, using different ways to compose vectors with different underlying approximation properties. Despite their increasing relevance as a component in `NLP` architectures, existing frameworks (*S-SPACE*: Jurgens and Stevens (2010a), *GENSIM*: Rehurek and Sojka (2011), *DEEPLEARNING4J*: Team (2016), *DISSECT*: Dinu and Baroni (2013)) provide limited options in their ability to systematically build, parameterize, compare and evaluate different models.

1.2.4 Heterogeneity of Existing Distributional Models

There is a large spectrum of distributional models. However, experimentation is limited to a constrained set, inhibiting the experimentation with different models. In this thesis, we assume that different models might fit better to different approximation tasks and requirements where there is not be a one-size-fits-all solution for DSMs. Hence the goal here is to reduce the barriers for experimenting with DSMs.

1.2.5 Performance

As mentioned earlier, Several complex configurations are needed to be set for building and consuming specific distributional semantic models. Also, there is a large spectrum of distributional models which makes the process of exploration

time-consuming and computationally costly. For querying operations of Distributional semantic models query execution time and index construction time play an important rule.

1.3 Core Requirements

1. **Language & Domain Transportability**:
   A system is transportable (As we explained at Section 1.2.1) if it can be easily adapted to new domains and languages (domain and language independence) (Grosz et al., 1987; Pearl and Bareinboim, 2014). In Machine Learning tasks, when there is insufficient data to train a model, transfer-learning technique can be used, which is a machine learning technique to adapt an existing model (trained model on a huge data) to new domains (Ruder, 2017; Lehmann and Voelker, 2014). Many Artificial Intelligence applications such as, Sentiment Analysis, Question Answering, Information Extraction, Semantic Search and Classification are increasingly using Distributional Semantics, which are often instantiated/generated across multiple languages and domains which may lack sufficient textual support to build comprehensive distributional models. Therefore it is necessary to have distributional semantics which is transportable across different languages and domains. In this thesis we analyse the role of machine translation approaches such as state-of-the-art machine translation (such as Google and Bing MT) and lightweight MT (the Combination of word translation table and English DSM (See Chapter 3, Section 3.2.2 for details)) to support the construction of better distributional vectors

2. **Coarse & Fine Grained Semantic**:
   Distributional semantic models should provide not only semantic relatedness/similarity **(Coarse-grained)** between entities, but also classify them and provide extra information**(Fine-grained),** such as semantic relations, on the top of semantic relatedness. Through Coarse-grained semantics on Distributional models we can cope with information incompleteness in large KBs and also eliminate non meaningful paths (See Chapter 3, Section 3.4.3.3 for more details). And also through fine-grained semantics on commonsense knowledge bases (KBs) we can enrich distributional semantic models (as we explained at Section Chapter 2, Section 2.1.3), which enable us to predict the existence of missing relationships and therefore completing the KBs. Therefore, unified distributional semantics and commonsense knowledge bases (KBs) provide complementary semantic perspectives (Bordes et al., 2013; Lin et al., 2015; Shi and Weninger, 2017; Freitas, Handschuh, and Curry, 2015).

3. **Expressive Set of Operations**:
   The architecture should have expressive querying operations (Vector Retrieval (Clark, 2015; Jurgens and Stevens, 2010a), Semantic Relatedness queries (Freitas et al., 2013b; Team, 2016), kNN[1] queries (Beyer et al., 1999; Rehurek, 2014)), and model construction operations (context windows, weighting schemes, dimensions).

4. **Reducing the Barriers for Experimenting with Different DSMs**:
   DSMs require the easy configuration of different parameters such as dimensionality, window size, context, weighting schemes (Kiela and Clark, 2014; Freitas, 2015a). These are due to the underlying parameter space, which is often too large to analyse and experiment exhaustively. Hence, there is a clear demand for an architecture which provides support for the generation of distributional semantic models directly from plain text files as well the configuration of the parameter space and the model generation itself, allowing users to experiment with different models and novel parameters.

5. **Single Point of Access/Unified API/Distributional Semantics as a Service (Modularization, Decoupling)**:
   With the growing and recurring use of word embeddings (Ruder, Vulić, and Søgaard, 2017), graph embeddings (Ristoski et al., 2018) and associated compositional functions (Kartsaklis, 2014), there is demand for an efficient and uniform API services for different DSMs across multiple languages and domains, enabling end-users and applications to consume and operate over multiple word embedding spaces. Natural language processing (NLP) tasks, specially Question Answering (QA) tasks usually contain of many individual functions that require to be used jointly to solve real-world issues. For example, a question answering task is: What types of relations can exist between two entities for a specific given domain and language?. In order to solve this task we need to use the different function/API, which requiring multi-lingual support, domain-specific support and composite semantic relation classification.

6. **Performance**: All operations over the transportable distributional semantic architecture should have fast response times for online requests ($10^2 - 10^3$ operations per second).

The set of requirements is used as qualitative dimensions to evaluate the effectiveness of a transportable distributional semantics architecture (Figure 1.1).

[1] k-Near Neighbours

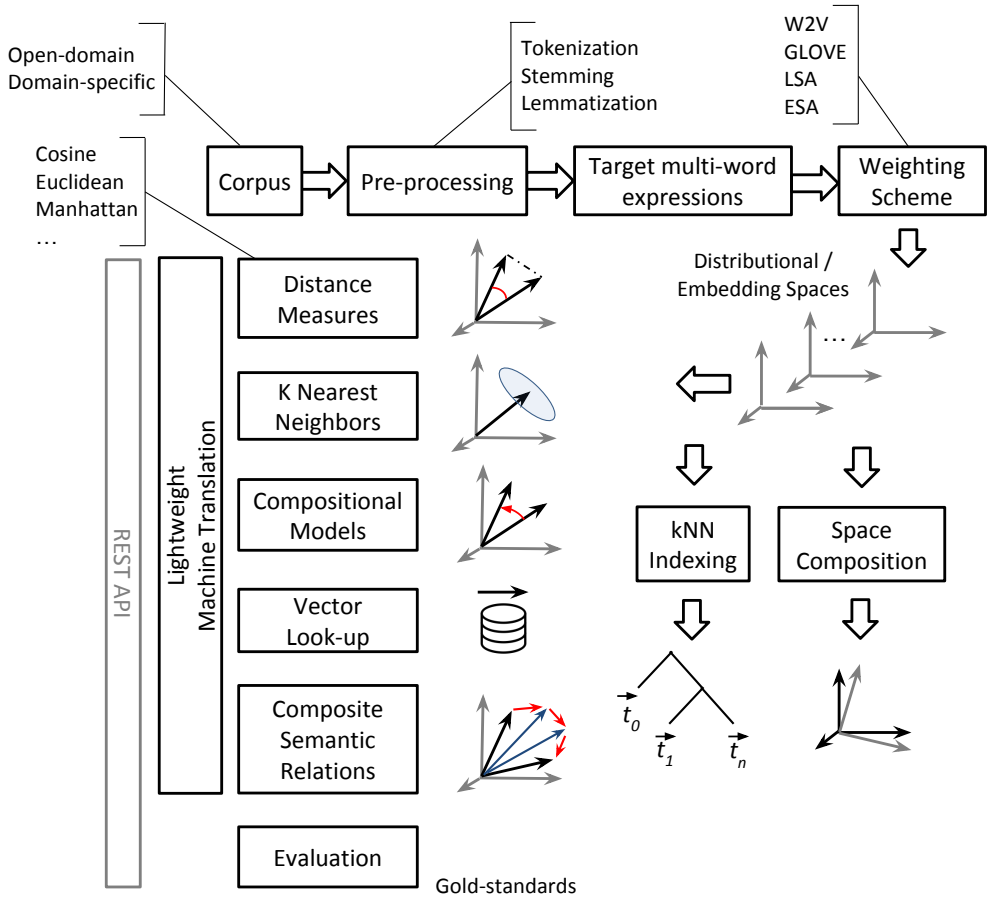

FIGURE 1.1: High-level Transportable Distributional Semantics Architecture

1.4 Research Questions

While distributional semantic models have been applied as a fundamental component in different tasks such as Sentiment Analysis, Question Answering, Information Extraction, Semantic Search and Classification tasks (Dai and Le, 2015; Gouws, Bengio, and Corrado, 2015; Tang, Qin, and Liu, 2015; Wang et al., 2016; Lai et al., 2015; Liu et al., 2015; Lin et al., 2015; Jurgens and Stevens, 2010a; Santos, Xiang, and Zhou, 2015b; Silva et al., 2018), there is no proper formalisation of the components which are underlying different distributional models. Moreover, DSMs have been mostly applied within the NLP scientific discourse in the context of open domain, English corpora. As a fundamental component of contemporary NLP practice, we argue that DSMs should be conceptualised as a principled architecture, reflecting the recurring operations for building and consuming these models and their associated products.

Taking all the requirements list above into account , the following dimensions of i) language and ii) domain transportability iii) the intersections between different DSMs iv) and relation classification models all emerge as gaps in delivering

a complete architecture (aiming for the completeness of the requirements and for delivering the architecture depicted in Figure 1.1). This thesis concentrates on these two dimensions, which are reflected and detailed in the following research questions:

For the language transportability aspect, we aim to answer the following research questions:

- **RQ 1.1** How do different distributional semantic models built from corpora in different languages and with different sizes perform in computing semantic relatedness similarity and relatedness tasks?
- **RQ 1.2** Does machine translation of non English input to English perform better than the word vectors in the original language (for which languages and for which distributional semantic models)?
- **RQ 1.3** Which DSMs and languages benefit more and less from the translation
- **RQ 1.4** What is the quality of state-of-the-art machine translation approaches for word pairs (for each language)?
- **RQ 1.5** Can a lightweight MT model over an English DSM provide higher quality word vectors compared to native word vectors?
- **RQ 1.6** How does a lightweight MT model compare with state-of-the-art MT models?
- **RQ 1.7** Are there DSMs which are more/less robust with respect to the quality of the MT?

For domain transportability aspect, we answer the following research questions:

- **RQ 2.1** Which types of Distributional Semantic Models are most cost-effective with respect to smaller, domain-specific corpora?
- **RQ 2.2** What type of discourse expressed in small-scale corpora leads to better domain-specific Distributional Semantic Models (dsDSMs)?
- **RQ 2.3** How to evaluate dsDSMs?

In additional, we answer the following research question in relation to coarse & fine grained semantics:

- **RQ 3.1:** How do we complement distributional semantic relatedness models with composite semantic relations?
- **RQ 3.2:** How to evaluate composite semantic relation classification?

Also we address the following research questions for having a single infrastructure:

- **RQ 4.1** What are the fundamental components of Distributional Semantic Models and how they can be unified into a single architecture?
- **RQ 4.2** How do they these components interact with each other?

1.5 Hypothesis

This thesis focuses on the corroboration of the following core research hypothesis:

- "*Distributional Semantics Models* can be realised in a transportable distributional semantics architecture."

The core research hypothesis can be detailed into the following research hypotheses:

- **Research Hypothesis I**: A lightweight machine translation provides higher quality multi-lingual Semantic Relatedness.
- **Research Hypothesis II**: High quality Distributional Semantic Models can be generated from domain-specific corpora
- **Research Hypothesis III**: The proposed DSM based classification of composite semantic relations can support a coarse & a fine grained semantic model.
- **Research Hypothesis IV**: The transportable distributional semantics architecture provide a single infrastructure that is language & domain transportable and also provide coarse and fine grained semantics, and allows users to have uniform access to all canonical (recurrently used) functionalities of a Distributional Semantics Architecture.

The hypotheses can be directly mapped to the core requirements (Section 1.3) of the proposed architecture (Table 1.1).

Table 1.1: Mapping of the core requirements to the hypothesis.

Requirements	**Hypothesis**
Language & Domain Transportability	Hyp. I & II
Coarse & Fine Grained Semantic	Hyp. III
Expressive Set of Operations	Hyp. IV
Reducing the barriers for Experimenting with Different DSMs	Hyp. IV
Single Point of Access/Unified API/Distributional Semantics as a Service	Hyp. IV
Performance	Hyp. IV

1.6 Research Methodology

The research methodology in this thesis aims at providing a rigorous method of validating the hypotheses defined in the previous section. This thesis follows the research methodology described below:

1. Comprehensive literature survey of the state-of-the-art in the problem space.
2. Comparative Analysis & evaluation of Distributional Semantic Models.
3. Developing a lightweight machine translation for multi-lingual aspect of DSMs.
4. Evaluation complex machine translations as baselines for multi-lingual aspect of DSMs.
    - Creation of multi-lingual test collections.
5. Creation of two domain-specific knowledge discovery test sets & evaluation them.
6. Provide a systematic methodology for evaluating Distributional Semantic models under relation classification.
    - Creation of a test collection for composite relation classification task.
    - Evaluation of the results (*measures: precision, recall, f1 score and accuracy*).
7. Post-processing the distributional semantic models on specific semantic relations and lexical categories.
8. Implementation of the transportable distributional semantics architecture.
9. Analysis of the results and conclusions.

1.7 Contributions

This work provides the following contributions:

1. The experimental set-up consists of:
    - The instantiation of **four** distributional semantic models: Explicit Semantic Analysis - *ESA* (Gabrilovich and Markovitch, 2007a), Latent Semantic Analysis - *LSA* (Landauer, Foltz, and Laham, 1998), Word2Vector - *W2V* (Mikolov et al., 2013a) and Global Vectors - *GloVe* (Pennington, Socher, and Manning, 2014a).

- Furthermore the above models are instantiated in **11 different languages** - *English, German, French, Italian, Spanish, Portuguese, Dutch, Russian, Swedish, Arabic and Farsi* - using *Wikipedia* (2014 and 2017) as a corpus.
- For the experiment the vector dimensions for LSA, W2V and Glove were finally set to 300 dimensions, while ESA was defined with 1500.

2. Each distributional model is evaluated for the task of computing semantic similarity and relatedness measures for each word pair using three human annotated gold standards: *Miller & Charles - MC* (Miller and Charles, 1991), *Rubenstein & Goodenough - RG* (Rubenstein and Goodenough, 1965), *WordSimilarity 353 - WS353* (Finkelstein et al., 2001) and *SIMLEX-999* (Hill, Reichart, and Korhonen, 2016). As these gold standards were originally in English, for every other language, the word pairs were translated and reviewed by professional translators, skilled in data localisation tasks.

3. Two complex and a lightweight machine translation approaches were evaluated: *Google* and *Bing* as complex Machine Translations and our *lightweight Machine Translation*.

4. Creation of two finance domain test sets.
   - Syntagmatic Word Pairs
   - Paradigmatic Word Pairs

1.8 Thesis Outline

The thesis is structured in the following chapters:

- *Chapter II - Background*: This chapter explains Distributional Semantic Models and Word Vector Models as successful approaches for supporting semantic approximations due to their ability to capture and quantify semantic associations in a scalable manner and their simplicity of representation. The chapter also discusses the semantic relation classification task, the importance of capturing the relations between two concepts for semantic interpretation tasks and the importance of deep learning in these tasks.

- *Chapter III - A Transportable Distributional Semantics Architecture*:

  This Chapter introduces the main principles and the motivation behind the development of a transportable distributional semantic architecture. It provides i) a solution to the problem of language and domain transportability in the context of distributional semantic models ii) addresses the connection between coarse and fine grained semantic models iii) integrate these contributions to existing recurring operations into the distributional semantics space into a unified distributional semantic architectures.

- *Chapter IV - Software & Services*: The implemented transportable distributional semantics architecture provides software infrastructure, which facilitates the experimentation and customisation of multilingual Distributional Semantic Models, allowing end-users and applications to consume and operate over multiple word embedding spaces as a service. In this chapter we describe our DSM API services.

- *Chapter V - Evaluation of Language and Domain Transportability*: Describes the experimental methodology for Language and Domain transportability components of the implemented distributional semantics architecture, collects the evaluation metrics and analyses the results of the experiments.

- *Chapter VI - Evaluation of the Composite Semantic Relation Classification*: Describes the experimental methodology for the composite semantic relation classification, collects the evaluation metrics and analyses the results of the experiments.

- *Chapter VII - Epilogue*: This chapter analyses the results on the evaluation of the research hypotheses, discusses the limitations of the transportable distributional semantics architecture and proposes a future work research agenda based on the limitations.

1.9 Associated Publications

Different aspects of this work were disseminated on the following publications:

- **Siamak Barzegar**, Brian Davis, Siegfried Handschuh and André Freitas, "Classification of Composite Semantic Relations by a Novel Distributional-Relational Model", Data & Knowledge Engineering Journal, 2018.

- **Siamak Barzegar,** Brian Davis, Siegfried Handschuh and André Freitas, "Multi-lingual Semantic Relatedness using lightweight machine translation", Semantic Computing (ICSC), 2018 IEEE 12th International Conference on, IEEE, 2018.

- **Siamak Barzegar,** Brian Davis, Siegfried Handschuh and André Freitas, "SemR-11: A Multi-Lingual Gold-Standard for Semantic Similarity and Relatedness for Eleven Languages", Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, May, 2018.

- Juliano Efson Sales, Leonardo Souza, **Siamak Barzegar**, Brian Davis, Siegfried Handschuh and André Freitas, "Indra: A Word Embedding and Semantic Relatedness Server", Proceedings of the Eleventh International

Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, May, 2018.

- **Siamak Barzegar,** André Freitas, Siegfried Handschuh and Brian Davis, "Composite Semantic Relation Classification", 22nd International Conference on Natural Language & Information Systems (NLDB), 2017.

- André Freitas, **Siamak Barzegar**, Juliano E. Sales, Siegfried Handschuh and Brian Davis, "Semantic Relatedness for All (Languages): A Comparative Analysis of Multilingual Semantic Relatedness using Machine Translation", 20th International Conference on Knowledge Engineering and Knowledge Management (EKAW), 2016.

- **Siamak Barzegar**, Juliano Efson Sales, André Freitas, Siegfried Handschuh, Brian Davis, "DINFRA: A One Stop Shop for Computing Multilingual Semantic Relatedness", 38th Annual ACM SIGIR Conference, Santiago, Chile, 2015.

Chapter 2

Background

2.1 Distributional Semantic Model

Distributional semantics is built upon the assumption that the context surrounding a given word in a text provides important information about its meaning (Distributional hypothesis). A rephrasing of the *distributional hypothesis* states that words that occur in similar contexts tend to have similar meaning (Harris, 1954). Distributional semantics focuses on the construction of a semantic representation of words based on the statistical distribution of word co-occurrence in texts. Distributional Semantic Models (DSMs) represent co-occurrence patterns under a vector space representation.

Definition (Distributional hypothesis) ((Harris, 1954)): *"Words that occur in similar contexts tend to have similar meanings".*

2.1.1 Introduction

Computers need to understand the meaning of human language (Natural Language Processing) to have a high level interaction. In order to understand human language, it is necessary to understand not only the words, but also the concepts and how they are linked together to create meaning. By analysing language for its meaning, NLP helps to develop different tasks such as Translation, Relationship Extraction (Katukuri, Raghavan, and Xie, 2013), Question Answering (Freitas, 2015a; Molino et al., 2012), Textual Entailment (Mehdad, Moschitti, and Zanzotto, 2010; Baroni et al., 2012) and Sentiment Analysis (Malandrakis et al., 2013). Distributional Semantic Models due to their ability to capturing meaning of linguistic entitles (e.g., words and short phrases) from a given corpus, become a fundamental component in many natural language processing architectures.

A list of several hypothesis that are embraced by the term distributional semantics mentions below.

- Difference of meaning correlates with difference of distribution (Harris, 1954).

- Words which are similar in meaning occur in similar contexts (Rubenstein and Goodenough, 1965).
- The semantic similarity of two words is a critical function of their interchangeability, without a loss of plausibility (Miller and Charles, 1991).
- Words with similar meanings will occur with similar neighbours if enough text material is available (Schütze and Pedersen, 1995).
- Word meanings as a function of keeping track of how words are used in context (Lund and Burgess, 1996).
- A representation that captures much of how words are used in natural context will capture much of what we mean by meaning (Landauer and Dumais, 1997).
- Words with similar distributional properties have similar semantic properties (Sahlgren, 2006).
- 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 (Lenci, 2008).

Although, the general idea behind all distributional hypotheses is clear enough, — differences in the meanings of linguistic entities correlate in their distributional properties — but such hypotheses do not clarify the variety of distributional information that should be taken into account.

In order to generate a semantic distributional model to capture the distributional information, two fundamental questions must be answered (Sahlgren, 2006).

1. What kind of distributional properties of entities should be taken into account?
2. How should different kinds of distributional patterns be interpreted?

Based on which distributional properties of entities are chosen and how they should be interpreted, different types of semantic similarities can be captured. Section 2.1.2 answers to these questions.

There are at least two different representation frameworks for interpreting distributional semantics: (i) the probabilistic and (ii) vector space frameworks. A probabilistic-based model of distributional semantics takes advantage of probability theory and Bayesian statistics. A probabilistic method[1] associates each word with probability distributions based on the linguistic context surrounding

[1] Which indicates semantic similarity in this framework

that word, as well as calculating conditional and join probabilities of contexts. One example of this method is Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan, 2003). Vector Space models construct n-dimensional space from the given text collection. In these models, each linguistic entity/term is represents by a weighted vectors where each dimension represents a linguistic context in which the entity/term occurs in the text collection (Corpus). Semantic Similarity between entities/terms is computed by calculating the distance between their vectors (Entities are close together in n-dimensional space are semantically similar). The well-known example in this category is Latent Semantic Analysis (LSA) (Landauer, Foltz, and Laham, 1998). In this thesis, we focus on the vector space frameworks for distributional models of semantics.

2.1.2 Distributional Semantic Models (DSMs)

Distributional Semantic Models (DSMs) are represented as a *distributional vector space*. A vector space is defined as a real vector space $VS^{\mathbb{R}}$ is a set that is closed under finite vector addition ($V \times V \rightarrow V$) and scalar multiplication ($R \times V \rightarrow V$). In this section, the core components of a distributional semantic model are described. The following definition summarizes the core elements of a distributional semantic model.

A Distributional Semantic Model (DSM) is a tuple (T, C, R, W, M, d, S), where:

- T are the target words, i.e. the words for which the DSM provides a contextual representation.
- C are the context patterns in which T co-occur.
- R is the relation between T and the context patterns C.
- W is the context-weighting scheme.
- M is the distributional matrix, $T \times C$.
- d is the dimensional reduction function, $d : M \rightarrow M'$.
- S is the distance measure, between the vectors in M'.

DSMs represent co-occurrence patterns under a vector space representation, where each dimension represents a linguistic *context* C in which the *target word* T occurs in the text collection.

Definition 2.1.1. (Target word): A *target word* is the word in the text collection/corpora for which the distributional vector representation is generated.

Definition 2.1.2. (Context Pattern): Two different approaches for modelling co-occurrence pattern (*context pattern*) exist: window-based and dependency-based. In window-based methods, contexts are words (based on the window-size)

surrounding a target term, which can be a few words, sentences, paragraphs or documents. In dependency-based methods, contexts are co-occurrence words that have a particular syntactic relation with a target word. (e.g. target word *Mike* is the subject of love, where love_subj is the context).

Definition 2.1.3. (Weighting Schemes): *Weighting Schemes* help to underline the contexts are much more informative about the meaning of the target word. Having higher co-occurrence frequencies does not lead to have a strong association with the target word.

For example, using a *weighting scheme dog* co-occurs with *bark* is more informative than its co-occurrence with *the*. Different types of *weighting schemes* can apply (Kiela and Clark, 2014). Table 2.1 considers some of them. At this table, F_{ij} denotes the target word frequency in a particular context, f_i is the total target word frequency, f_j is the total context frequency, N is the total of all frequencies, n_j is the number of non-zero contexts).

TABLE 2.1: Term weighting schemes

Scheme	Definition
None	$w_{ij} = f_{ij}$
TF-IDF	$w_{ij} = \log(f_{ij}) \times log(\frac{N}{n_j})$
TF-ICF	$w_{ij} = \log(f_{ij}) \times log(\frac{N}{f_j})$
Okapi BM25	$w_{ij} = \frac{f_{ij}}{0.5+1.5\times\frac{f_j}{\frac{f_j}{j}}+f_{ij}} log(\frac{N-n_j+0.5}{f_{ij}+0.5})$
ATC	$w_{ij} = \frac{(0.5+0.5\times\frac{f_{ij}}{\max_f})log(\frac{N}{n_j})}{\sqrt{\sum_N^{i=1}[(0.5+0.5\times\frac{f_{ij}}{\max_f})log(\frac{N}{n_j})]^2}}$
LTU	$w_{ij} = \frac{(log(f_{ij})+1.0)\log(\frac{N}{n_j}))}{0.8+0.2\times f_j\times\frac{j}{f_j}}$
MI	$w_{ij} = \log\frac{P(t_{ij}\|c_j)}{P(t_{ij})P(c_j)}$ Which $P(t_{ij} \mid c_j)$ is defines as $\frac{f_{ij}}{f_j}$ and $P(t_{ij})$ as $\frac{f_{ij}}{N}$
PosMI	$w_{ij} = \max(0,\text{MI})$
T-Test	$w_{ij} = \frac{P(t_{ij}\|c_j)-P(t_{ij})P(c_j)}{\sqrt{P(t_{ij})P(c_j)}}$ Which $P(t_{ij} \mid c_j)$ is defines as $\frac{f_{ij}}{f_j}$ and $P(t_{ij})$ as $\frac{f_{ij}}{N}$
χ^2	See (Curran, 2004), P. 83
Lin981	$w_{ij} = \frac{f_{ij}\times f}{f_i\times f_j}$
Lin98b	$w_{ij} = -1 \times \log\frac{n_j}{N}$
Grecf94	$w_{ij} = \frac{\log f_{ij}+1}{\log n_i+1}$

Definition 2.1.4. (Distributional Matrix): A *Distributional Matrix* determines numeric associations between *target word* and *context patterns* (Figure 2.2).

This association between *target word* and *context patterns* in its simplest form is a binary value, which, shows the absence or presence of the occurrences of a *target word* with a *context pattern*. However, it can be a weighted value, which usually correspond to the frequency of the observation of the co-occurrences of *target word* and *context patterns* and applying the weighting scheme on them.

TABLE 2.2: Distributional matrix built from the context vectors of the target words.

		Context				
	school	husband	born	footbal	play	...
Target Words	wife	0	4	1	1	3
	girl	7	0	2	3	6
	boy	7	0	2	3	6
	...			...		

Definition 2.1.5. (Dimensionality Reduction): A Distributional Matrix builds on different distinct context patterns, where the number of distinct context patterns determines the dimensionality of the vector space. The dimensionality of the vector space has a solid impact[2] on the performance of the distributional semantic model. To address this issue *Dimensionality Reduction* techniques are applied to reduce the number of context pattern (dimensionality) employed for the construction of a vector space.

Truncated Singular Value Decomposition (`SVD`) is the most familiar dimensionality reduction technique in the vector space models. Truncated `SVD` is a linear transformation method which is used to obtain the optimal rank d factorisation by taking advantage of the *Euclidean Norm* of context elements (Deerwester et al., 1990).

Definition 2.1.6. (Similarity Measurement): *Similarity Measurement* is for calculating the similarity/relatedness between two vectors,

The Distributional Semantics Model (Vector Space Model) contains a set of words represented by their weighted co-occurrence context patterns. According to the distributional hypothesis, words that contain similar contexts will tend to have similar meanings. Also, words with similar contexts will tend to have vectors which are geometrically closer, in contrast to words in dissimilar contexts. We consider the similarity metrics in Table 2.3 (Kiela and Clark, 2014).

[2] computational impact

TABLE 2.3: Similarity Measures between vectors v and u, where v_i is the ith component of v

Measure	Definition
Euclidean	$\frac{1}{1+\sqrt{\sum_{i=1}^{n}(v_i-u_i)^2}}$
Cityblock	$\frac{1}{1+\sqrt{\sum_{i=1}^{n}\lvert v_i-u_i\rvert}}$
Chebyshev	$\frac{1}{1+\max_i\lvert v_i-u_i\rvert}$
Cosine	$\frac{u\cdot v}{\lvert u\rvert\lvert y\rvert}$
Correlation	$\frac{(u-\mu_u)\cdot(v-\mu_y)}{\lvert u\rvert\lvert y\rvert}$
Dice	$\frac{2\sum_{i=0}^{n}\min(u_i,y_i)}{\sum_{i=0}^{n}u_i+y_i}$
Jaccard	$\frac{u\cdot y}{\sum_{i=0}^{n}u_i+y_i}$
Jaccard2	$\frac{\sum_{i=0}^{n}\min(u_i,y_i)}{\sum_{i=0}^{n}\max(u_i,y_i)}$
Lin	$\frac{\sum_{i=0}^{n}u_i+y_i}{\lvert u\rvert+\lvert y\rvert}$
Tanimoto	$\frac{u\cdot v}{\lvert u\rvert+\lvert y\rvert-u\cdot v}$
Jensen-Shannon Div	$1-\frac{\frac{1}{2}(D(u\|\frac{u+v}{2})+D(y\|\frac{u+v}{2}))}{\sqrt{2\lg 2}}$
α-skew	$1-\frac{D(u\|\alpha v+(1-\alpha)u)}{\sqrt{2\lg 2}}$

2.1.2.1 Syntagmatic & Paradigmatic Relations

Distributional Semantic Models can be categorised by the type of semantic relation (syntagmatic or paradigmatic relations) between their respective *target items* .

- *Syntagmatic Relation* Two words have a syntagmatic relation if they co-occur together (more frequently than expected by chance) but have different grammatical roles in the sentence. In the other words, these relations are indicated by possibilities of combination. For instance, the relation between *green* and *paint* in the sentence *she buys a green paint* is a example of a *syntagmatic* relation.

- *Paradigmatic Relation* Two words have paradigmatic relation if they can substitute one another in a sentence without affecting the grammar of the sentence. For example, for the given sentences a *she buys a green paint* and *he eats blue clay*, the pair of words *she* and *he*, *buys* and *eats*, *green* and

blue, as well as *paint* and *clay* have a paradigmatic relationship. Synonymy and antonymy are also examples of such paradigmatic relations.

Distributional Semantic Models that count the co-occurrence of words capture a syntagmatic relationship between them. In contrast, if Distributional Semantic Models count the frequency of shared neighbours between words capture a paradigmatic relationship. Therefore wider windows sizes — such as paragraphs or documents— tend to capture syntagmatic relations[3]; while narrow windows sizes will capture paradigmatic relations[4](Sahlgren, 2008; Lenci, 2008).

2.1.2.2 Preprocessing

The most important part of generating a high quality *DSMs*, is preprocessing the text collection/corpus. A corpus pre-processor is responsible for defining the tokenisation strategy and the tokens' subsequent transformations. It defines, for example, if *United States of America* corresponds to a unique token or to multiple. *Stem*, *lowercase*, *stopwords*, *accent*, *replacement number* and *token size* are other popular transformations, which explained in below.

- **Stopwords**: Because stopwords[5] are often too uninformative, ignoring them not only reduces the model size and computational effort, but also to makes for a more informative distributional vector space (Kiela and Clark, 2014). However, it has to be done carefully, e.g. removing negation decreases performance of sentiment analysis.

- **Stem**: Stemming is clearly a useful transformation of our word-similarity representation, but on the other hand, it tends to lose information. For example, after applying a stemmer, the term *University* is represented in the model as *univers*. Soricut and Och (2015) have shown that keeping morphological relationships help to have a better representation of similarities between words.

- **Lowercase**: applying lowercase improves performance by eliminating the difference between capitalised and non-capitalised item words. It also improves performance because it reduces dimensionality, and puts Uppercase and Lowercase tokens into the same class, and this amplifies the most frequent patterns.

- **Accent**: Eliminating accents[6] from words leads to corpus and query be normalised.

[3]Words with different meaning which frequently co-occur in the same context, such as *car* and *road*, *child* and *cradle*.

[4]Words that occur in very similar syntagmatic contexts, typically synonyms and antonyms.

[5]Such as: *a*, *the*, *and*, *as*, *below*, *that* and so forth.

[6]For example, convert *á* to *a*.

- **Number**: The replacement with a meta token for numbers could be an important saving in unique vocabulary count for large corpus training.
- **Token Size**: Setting a minimum and a maximum acceptable token size lead to eliminate uninformative item words such as characters, punctuation and so forth.

2.1.2.3 Different Distributional Semantics Models (DSMs)

This section describes the five DSMs:

Latent Semantic Analysis (LSA) (Landauer, Foltz, and Laham, 1998) is an algorithm that uses a collection of documents to construct a semantic space. The algorithm constructs a word-by-document matrix where each row corresponds to a unique word in the document corpus and each column corresponds to a document. The value at each position is how many times the row's word occurs in the column's document. Singular Value Decomposition is calculated for the word-document matrix to produce three matrices ($U\Sigma V$), U - the wordspace, Σ – the singular values, and V – the document space. The columns of U are then truncated to a small number of dimensions (typically 300), which produces the final semantic vectors. LSA models can handle Synonymy problems to some extent, but it can not capture capture polysemy[7].

Random Indexing (RI) (Sahlgren, 2005) is a word co-occurrence based approach to statistical semantics. RI uses statistical approximations of the full word co-occurrence data to achieve dimensionality reduction. This results in a much quicker running time and fewer required dimensions.
In most co-occurrence models, a word-by-word matrix is constructed, where the values denote how many times the column's word occurred in the context of the row's word. RI instead represents co-occurrence through index vectors. Each word is assigned a high-dimensional, random vector that is known as its index vector. These index vectors are very sparse - typically 7 ± 2 non zero bits for a vector of length 2048, which ensures that the chance of any two arbitrary index vectors having an overlapping meaning (i.e. a cosine similarity that is non-zero) is very low. Word semantics are calculated for each word by keeping a running sum of all of the index vectors for the words that co-occur.

Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch, 2007a) is a vectorial representation of text (individual words or entire documents) that uses a document corpus as a knowledge base. Specifically, in ESA, a word is represented as a column vector in the $tf - idf$ matrix of the text corpus and a

[7] i.e., multiple meanings of a word.

document (string of words) is represented as the centroid of the vectors representing its words.

Global Vectors (GloVe) (Pennington, Socher, and Manning, 2014a) is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space. It can be used to solve word analogy problems like man is to king as woman is to *'?'*.
Word2Vector (W2V) (Mikolov et al., 2013a) provides an efficient implementation of the continuous bag-of-words and skip-gram architectures for computing vector representations of words. These representations can be subsequently used in many natural language processing applications and for further research.

2.1.2.4 Evaluation

The ability to automatically determine and quantify the degree of semantic similarity and semantic relatedness between pairs of words or expressions is one of the archetypal tasks for assessing the ability of a system to perform semantic interpretation. The ability to quantify semantic relatedness can provide a lightweight semantic interpretation operation which can be applied in different areas of Artificial Intelligence, Natural Language Processing and Information Retrieval. Examples of applications include coping with lexical and semantic gaps in Question Answering Systems (Freitas, 2015b; Freitas and Curry, 2014), using the semantic relatedness score as a ranking function in Information Retrieval systems (Freitas, Curry, and O'Riain, 2012a) and serving as a semantic scoping mechanism in deductive/abductive methods, which provide a semantic justification under the structure of the relational graph (Freitas et al., 2014).

Due to its simplicity in comparison to other tasks such as Question Answering, Text Entailment and Machine Translation, semantic similarity and relatedness gold standards have been initially used to support the evaluation of the interaction between semantic distance measures and of linguistic and knowledge resources (Resnik, 1995; Lin, 1991; Wu and Palmer, 1994; Agirre et al., 2009). With the advent of large-scale corpora, distributional semantic models automatically built from textual corpora were created (Turney and Pantel, 2010a) using, in most cases, a vector space representation of meaning. Since distributional semantic models can induce models with a more comprehensive underlying vocabulary and also capture a broader set of semantic relations, new gold-standards emerged (Finkelstein et al., 2001), evolving from capturing semantic similarity to semantic relatedness behaviour. More recently, the creation of neural/predictive word embedding models such as *Glove* (Pennington, Socher, and Manning, 2014a) and *Word2Vector* (Mikolov et al., 2013a) pushed semantic similarity and relatedness

gold-standards to evolve in the direction of quantifying more fine-grained semantic relations (Hill, Reichart, and Korhonen, 2015).

Currently, most of the existing gold-standards for evaluating semantic similarity and relatedness have focused on the English language, with some initiatives providing initial gold-standards for few other languages[8] (Faruqui and Dyer, 2014a). Multi-lingual DSMs are under resourced and researched

The problem of measuring the semantic similarity and relatedness of two concepts can be stated as follows: given two concepts A and B, determine a numerical measure f(A, B) which expresses the semantic similarity or relatedness between concepts A and B. The notion of semantic similarity is associated with taxonomic (is-a) relations, while semantic relatedness represents more general relations. *Car* and *train* are examples of similar concepts (both share a common taxonomic ancestor, *vehicle*) while *car* and *wheel* are related concepts (a wheel is part of a car). As a consequence, semantic similarity is considered a particular case of semantic relatedness.

Alternatively semantic similarity can also be defined as two concepts sharing a high number of salient features (attributes): synonymy (car/automobile), hypernymy (car/vehicle), co-hyponymy (car/van/truck), while semantic relatedness can be defined as two words semantically associated without being necessarily similar: function (car/drive), meronymy (car/tyre), location (car/road), attribute (car/fast) (Freitas, 2015b).

The four main gold standards are used are:

- **Wordsimilarity 353:** WS-353 (Finkelstein et al., 2001) is the most popular evaluation gold standard for distributional semantic models. The dataset is focused on **semantic relatedness**. The dataset contains two subsets: *set 1* (153 word pairs, evaluated by 13 subjects), and *set 2* (200 word pairs evaluated by 16 subjects) each one containing pairs from different parts-of-speech, a proper noun and pairs involving subjective bias.
- **Rubenstein & Goodenough:** RG (Rubenstein and Goodenough, 1965) contains 65 pairs which are often used to evaluate Distributional Semantic Models. RG reflects **semantic similarity** of words rather than their relatedness. It was engineered using 15 annotators to rate the semantic similarity of each pair.

[8] For WS-353 dataset: Arabic, French, German, Spanish, Romanian, Italian, Russian. For RG dataset: German, Spanish, Farsi. For MC: Arabic, Romanian, Spanish. For SIMLEX-999 dataset: German, Italian, Russian.

- **Miller & Charles:** MC (Miller and Charles, 1991) is a subset of 30 noun pairs from the RG gold standard which are re-annotated following new similarity guidelines. Ten pairs were selected from the highest level (between 3 and 4 on a scale from 0 to 4), ten pairs from the intermediate level (between 1 and 3), and ten pairs from the lowest level (0 to 1) of **semantic similarity.**

- **SIMLEX-999:** SIMLEX-999[9] (Hill, Reichart, and Korhonen, 2016; Leviant and Reichart, 2015) is aimed to measure how well Distributional Semantic Models capture **semantic similarity,** rather than relatedness. SIMLEX-999 contains a range of 111 adjective, 666 noun and 222 verb pairs with an independent rating for each pair. It was built by using 500 annotators via Amazon Mechanical Turk.

2.1.3 Enriched Distributional Semantic Models

This part concentrates on assessing the impact of enriching distributional semantic models by structured Knowledge Bases (KBs), which are a kind of information network that represents knowledge in a triple set (h, l, t) that composes two entities $h, t \in E$ the set of entities and a relationship $l \in L$ the set of relationships.

2.1.3.1 Post-Processing Word Embeddings

Faruqui et al. (2014) proposed a graph-based learning technique to obtain higher quality word embeddings by using lexical relational resources such as *Wordnet* (Fellbaum, 2005), *Freebase* (Bollacker et al., 2008). This technique known as *retrofitting*, brings semantically similar words close together while keeping them (relatively) close to their initial distributional vectors (Table 2.4) (Mrkšić et al., 2016). It is a post-processing approach, whereby semantic constraints are injected into existing distributional vector spaces. The inputs of retrofitting technique are an existed vector space model ($m \times n$ dense matrix) and a list of semantic relationships from a lexical relational resources such as *WordNet*. Retrofitting method helps to minimise a sum of distance of a word from its neighbours in the lexical relation resource and its distance from its original vector in the exited vector space models.

[9] In this thesis we called it SIMLEX

TABLE 2.4: Nearest neighbours for target words (First Row) using *GloVe* vectors before and after injecting synonym relation.

	east	**expensive**	**British**
Before	west	pricey	American
	north	cheaper	Australian
	south	costly	Britain
	southest	overpriced	European
	northest	inexpensive	England
After	eastward	costly	Brits
	eastern	pricy	London
	easterly	overpriced	BBC
		pricey	UK
		afford	Britain

Speer, Chin, and Havasi (2017) introduced an ensemble method known as *ConceptNet-Numberbatch*, which combines data from pre-trained word embeddings and knowledge graphs, using a variation on retrofitting (Faruqui et al., 2014) to produce a high-quality word embeddings. They achieve this goal by applying the following method:

- Expanding the retrofitting algorithm (Faruqui et al., 2014) to benefit from structured links outside the original vocabulary.
- Using *ConceptNet* (Speer and Havasi, 2012) as a resource of structured connections between words.
- Merging two pre-trained DSMs (Word2Vec and Glove) using a local linear interpolation. This combination performs better than each of the models separately.
- Applying expanded retrofitting method on the combined vector space model by using *ConceptNet* as a lexical relational resource.

Speer, Chin, and Havasi (2017) called their word embedding *ConceptNet-Numberbatch*, showing that the combined embedding outperforms *W2V* on word-similarity evaluations.

Speer et al. consider the data in ConceptNet as a symmetric matrix of association between words to apply the expanded retrofitting method. Therefore,

they eliminate non-symmetric relations in ConceptNet and disregard the these relation types to generate new word embeddings. We argue that in order to achieve a high quality semantic relation classification, all relations must be taken into account. Hence a more comprehensive approach is needed which includes knowledge about how both asymmetric and symmetric allowing us to inject all semantic constraints into existing word embeddings.

2.1.3.2 Embedding Entities and Relations

Bordes et al. (2013) proposed an energy-based model for learning low-dimensional embeddings of entities which is materialised into the $TransE$ model. Relationships are represented as translations in the embedding space. In other words, the basic idea behind behind the model is, in a triple set (h, r, t) that composes two entities $h, t \in E$ the set of entities and a relationship $r \in L$ (the set of relationships), the embedding of the entity t should be close to the embedding of the head entity h plus some vector that depends on the relationship r.

$$h + r \simeq t$$

To learn such embeddings, they minimise a margin-based ranking criterion over the training set (Yang et al., 2014), where the scoring function of $TransE$ is

$$-(2g_r^a(y_h, y_t) - 2g_r^b(y_h, y_t) + \left\|V_r\right\|_2^2)$$

where:

$$g_r^a(y_h, y_t) = A_r^T \begin{pmatrix} y_h \\ y_t \end{pmatrix} \quad and \quad g_r^b(y_h, y_t) = y_h^T B_r y_t$$

and A_r^T, B_r are relation-specific parameters and equal to $(V_r^T \quad -V_r^T)$ and I, respectively.

The motivation behind translation-based parametrisation is that the structure of the hierarchical relationships are very common in Knowledge based systems and translations are the natural transformations for representing them. Their model relies on a reduced set of parameters as it learns only one low-dimensional vector for each entity and each relationship. The optimisation is carried out by stochastic gradient descent (using *minibatches*), and also the embedding vectors of the entities are normalised. $TransE$ has fewer parameters comparison with other approaches, leading to a simplification of the training process and preventing under-fitting.

2.2 Complementary Semantic Perspectives

Commonsense Knowledge Bases (KBs) are a kind of information network that represents knowledge in a triple set (h, l, t) that composes two entities $h, t \in E$ the set of entities and a relationship $l \in L$ the set of relationships. They have been applied to different tasks including semantic search (Ramkumar and Poorna, 2014), entity linking (Moussallem et al., 2017) and semantic relation classification (Girju, 2008) and they provide the fine grained semantics. However, most KBs are often noisy and incomplete. There are a few large commonsense KBs.

- **ConceptNet**: *ConceptNet* (Speer and Havasi, 2012) is a semantic network built from existing linguistic resources and crowd-sourced. It is built from nodes representing words or short phrases as observed in natural language and labelled abstract relationships between them.
- **WordNet**: WordNet (Miller, 1995) is a large lexical database including *Nouns*, *Verbs*, *Adjectives* and *Adverbs*, which grouped into different *synsets* (cognitive synonyms). The most frequently relation are *Synonymy*, *Hypernymy*[10], *Hyponymy*[11] and *Meronym*[12].
- **Microsoft Concept Graph**: *Microsoft Concept Graph* (Wang et al., 2015) is a taxonomy of English nouns[13] containing *IsA* relations
- **DBpedia**: *DBpedia* (Auer et al., 2007) is focused on named entities and basic factoid-style attributes.

Although Distributional Semantic Models provide only the coarse grained semantic, through Distributional Semantic Models, we can cope with information incompleteness in large KBs (Freitas et al., 2014) and eliminate non meaningful paths (See Chapter 3, Section 3.4.3.3 for more details). Also with post-processing techniques through commonsense knowledge bases (KBs) on word embeddings (As we explained at Section 2.1.3), we can enrich distributional semantic models, which are able to predict the existence of missing relationships and therefore completing the KBs. Therefore, unified distributional semantics and commonsense knowledge bases (KBs) provide complementary semantic perspectives (Shi and Weninger, 2017; Freitas, Handschuh, and Curry, 2015).

2.3 Neural Networks

As explained earlier, Natural Language Processing (NLP) aims to convert human language into a semantic presentation that is understandable for computers. However full natural understanding is still a distant goal. *Machine Learning/Deep*

[10] Y is a hypernym of X if every X is a (kind of) Y.
[11] Y is a hyponym of X if every Y is a (kind of) X.
[12] Y is a meronym of X if Y is a part of X.
[13] 5.4 million concepts

Learning algorithms already have made impressive advances in different fields such as computer vision. Deep Learning contains many layers to produce the output once a large data is fed, and nowadays has become popular because it can extract features and classification easily (in one shot) and we have access to very large data. Following this trend, much effort has been invested in applying *Machine Learning*/*Deep Learning* algorithms on *Natural Language Processing* tasks such as *part-of-speech (*POS*) tagging*, *machine translation* and *word embedding*. For several years, *Machine Learning* techniques such as *Support Vector Machine* (SVM) (Hearst et al., 1998), Decision Trees (Quinlan, 1986), Logistic Regression (Harrell, 2015) and shallow neural networks) have been applied on NLP tasks. In the last new years, *Deep Learning* techniques have been applied in NLP task increasingly, due to their abilities to learn the features. Early layers in Deep Learning model learn how to detect low level features and following layers combine features from earlier layers into a more complete representation (LeCun, Bengio, and Hinton, 2015). Deep Learning models can be categorised into **supervised** and **semi or unsupervised** approaches. In **Supervised Approaches** a predictive model is trained by a sufficient amount of labelled training data. Supervised learning models, such as *Recurrent neural networks* (RNNs) (Elman, 1990; Mikolov et al., 2010), *Long Term Short Memory* (LSTM) (Hochreiter and Schmidhuber, 1997) and *Convolutional deep neural networks (*CNNs*)* (Kim, 2014) can perform very well, but it is necessary to have an enough annotated data, which can be a critical bottleneck. Supervised Deep Learning can learn multi-level feature representation automatically. In contrast, traditional machine learning based NLP systems depend on hand-crafted features which are time-consuming and often incomplete. **Semi or Unsupervised Approaches** do not have the limitation of *supervised methods* for manually annotating of sufficient amounts of training data (LeCun, Bengio, and Hinton, 2015).

2.4 Semantic Relation Classifications

2.4.1 Introduction

Semantic relation classification is the task of classifying the underlying abstract semantic relations between target entities (Nominals) present in texts (Qin, Xu, and Guo, 2016). The goal of relation classification is defined as follows: given a sentence S with pairs of annotated target nominals e_1 and e_2, the relation classification system aims to classify the relations between e_1 and e_2 in given texts within the pre-defined relation set (Hendrickx et al., 2009). For instance, the relation between the nominal ***burst*** and ***pressure*** in the following example sentence is interpreted as ***Cause-Effect(***e_2, e_1***)***.

The $< e_1 >$ **burst** $< /\mathbf{e_1} >$ has been caused by water hammer $< e_2 >$ **pressure** $< /\mathbf{e_2} >$.

2.4.2 Approaches

Different approaches have been explored for relation classification, including **supervised** and **semi or unsupervised** relation discovery and classification (Zeng et al., 2014).

Supervised Approaches are machine learning methods in which a predictive model is trained by a sufficient amount of labelled training data to predict a true relation between two given entities on a test set. Supervised learning models can perform very well, given sufficient annotated data, which can be a critical bottleneck. Second, learning models should be designed very well to acquire a representatives set of features and the performance of supervised approaches strongly depend on the quality of their designed features. Approaches can be grouped into **two types:** *feature-based* and *kernel-based* (Zeng et al., 2014). In feature-based methods (Kambhatla, 2004), classifiers [14] combine a large number of features, such as Dependency Parse (De Marneffe, MacCartney, and Manning, 2006), WordNet (e.g. hyponyms, Common-parents, distance) (Miller et al., 1990), Part-Of-Speech (POS) (e.g. Noun, Verb, ...) and so forth. In kernel-based methods (Zelenko, Aone, and Richardella, 2003) relations have been identified through evaluating the similarity between two trees or sequences. Some example of supervised learning models are *Support Vector Machine* (VSM) *Long Short Term Memory* (LSTM) and Convolutional neural network (CNN) (See Chapter 3, Section 3.4.2 for more details).

Semi or Unsupervised Approaches do not have the limitation of *supervised methods* for manually annotating of sufficient amounts of training data which is time-consuming. Semi-supervised approaches solve this problem by pre-defining some initial seeds for any individual relation, and after that bootstrap from the seeds for acquisition of the relation. On the other hand, it is complicated to choose representative seeds for achieving high accuracy. The distributional hypothesis (Harris, 1954) indicates that words that have similar meanings, probably occur in the same context. Accordingly, we can assume that the pairs of words that occur in similar contexts tend to have similar relations.

[14] Such as *Support Vector Machine (SVM*, Max-Entropy model

2.4.3 Task & Dataset

SemEval 2010 Task 8 (Hendrickx et al., 2009) focuses on Multi-Way classification of semantic relations between pairs of nominals. For instance, *student* and *association* are in an ***Member-Collection*** relation in *"The student association is the voice of the undergraduate student population of the State University of New York at Buffalo"*. This task includes around 11,000 labelled sentence, which is enough data for deep learning methods to be learned.
The tasks provides for nine general relations plus *"OTHER"* is as follows:

- **Cause-Effect.** An event or object leads to an effect. *Example*: $< e_1 >$ *Smoking* $< /e_1 >$ causes $< e_2 >$ *cancer* $< /e_2 >$.
- **Instrument-Agency.** An agent uses an instrument. *Example*: $< e_1 >$ *Laser* $< e/_1 >$ $< e_2 >$ *printer* $< /e_2 >$
- **Product-Producer.** A producer causes a product to exist.*Example*: The $< e_1 >$ *farmer* $< /e_1 >$ grows $< e_2 >$ *apples* $< /e_2 >$
- **Content-Container.** An object is physically stored in a delineated area of space, the container. *Example*: $< e_1 >$ *Earth* $< /e_1 >$ is located in the $< e_2 >$ *MilkyWay* $< /e_2 >$.
- **Entity-Origin.** An entity is coming or is derived from an origin (e.g., position or material). *Example*: $< e_1 >$ *Letters* $< /e_1 >$ from $< e_2 >$ *foreigncountries* $< /e_2 >$.
- **Entity-Destination.** An entity is moving towards a destination. *Example*: The $< e_1 >$ *boy* $< /e_1 >$ went to $< e_2 >$ *bed* $< /e_2 >$.
- **Component-Whole.** An object is a component of a larger whole. *Example*: My $< e_1 >$ *apartment* $< /e_1 >$ has a large $< e_2 >$ *kitchen* $< /e_2 >$.
- **Member-Collection.** A member forms a nonfunctional part of a collection. *Example*: There are many $< e_1 >$ *trees* $< /e_1 >$ in the $< e_2 >$ *forest/* $< e_2 >$.
- **Communication-Topic.** An act of communication, whether written or spoken, is about a topic. *Example*: The $< e_1 >$ *lecture* $< /e_1 >$ was about $< e_2 >$ *semantics* $< /e_2 >$.

The dataset contains a set of 10,717 instances, where 8,000 instances are defined as the training set. Table 2.5 shows the distribution of categories for the dataset. The second column (Frequency) shows the absolute and relative frequencies of each relation.

TABLE 2.5: Annotation Statistics of relation types with absolute and relative frequency in the dataset.

Relation	**Frequency**
Cause-Effect	1331 (12.4%)
Component-Whole	1 253 (11.7%)
Entity-Destination	1137 (10.6%)
Entity-Origin	974 (9.1%)
Product-Producer	948 (8.8%)
Member-Collection	923 (8.6%)
Message-Topic	895 (8.4%)
Content-Container	732 (6.8%)
Instrument-Agency	660 (6.2%)
Other	1864 (17.4%)
Total	10717 (100%)

2.5 Summary

The discussion in this chapter started by giving an overview of the core technologies underlying this thesis. Section 2.1 provided a brief history of distributional semantic models, vector space models parameters and how distributional semantic models can be enriched by injected semantic constraints from knowledge base systems. We have also explained about the commonsense knowledge bases (KBs) and how distributional semantics provide complementary semantic perspectives in Section 2.2. We have discussed briefly the different *neural networks* in Section 2.3. Finally in Section 2.4 we described the semantic relation classification tasks, which is the basis for composite semantic relation classification which will be described in Chapter 3. The following research gaps in distributional semantics field can be summarised as : (i) a severe lack of efficient approaches for addressing multilingual distributional semantic models and a significant deficit availability of multilingual test collections for evaluation. (ii) need for more comprehensive analysis and evaluation on distributional semantic models with respect to smaller and domain-specific corpora . (iii) need for complementary semantic perspectives. (iv) Knowledge Graph completion which is one main part of having complementary semantic perspectives. The contribution of this thesis concentrates on the implementation of a transportable distributional semantic architecture which address these gaps in Chapter 3.

Chapter 3

A Transportable Distributional Semantics Architecture

3.1 Introduction

In Chapter 1 the motivation and the main principles behind the development of a transportable distributional semantic architecture were introduced. This chapter will describe the main contributions aimed by this work, namely:

1. Providing a solution to the problem of language transportability in the context of distributional semantic models. The key strategy used to address this problem is the development of a language l-to English lightweight machine translation layer, the output of which is sent to an English distributional model. This contribution is described in Section 3.2.

2. Providing a solution to the problem of domain transportability in the context of distributional semantic models. This is achieved by providing a systematic study on how different types of discourse and different types distributional model configurations perform in a domain-specific setting (financial domain). This is described in Section 3.3.

3. Addressing the connection between coarse-grained (distributional semantics) and fine-grained (relational) semantic models. Using the problem of Composite Semantic Relation Classification (CSRC) to link both types of models, where users can have access to both the distributional semantic scores between pairs of terms and also interpretation pathways (compositions of semantic relations) between the terms. This contribution is described in Section 3.4.

4. Integrate these contributions as existing recurring operations of the distributional semantics space into a unified distributional semantics architecture. This is addressed by the specification of a common architecture which encapsulates the previous contributions, integrating them with existing recurring operations. This contribution is described in Section 3.5.

3.2 Language Transportability

Distributional Semantic Models (DSM) have consolidated themselves as fundamental components for supporting automatic semantic interpretation in different application scenarios in natural language processing. From *question answering systems*, to *semantic search* and *text entailment*, distributional semantic models support a scalable approach for representing the meaning of words, which can automatically capture comprehensive associative commonsense information by analysing word-context patterns in large-scale corpora in an unsupervised or semi-supervised fashion (Freitas, 2015b; Turney and Pantel, 2010b; Sales et al., 2016).

However, such DSMs are strongly dependent on the size and the quality of the reference corpora, which embed the commonsense knowledge necessary to build comprehensive models. While high-quality texts containing large-scale commonsense and domain-specific information are present in English, other languages may lack sufficient textual support to build comprehensive distributional models.

To address this problem, this thesis investigates how different distributional semantic models built from corpora in different languages and with different sizes perform in computing semantic relatedness similarity and relatedness tasks. Additionally, we analyse the role of machine translation approaches to support the construction of better distributional vectors and for computing semantic similarity and relatedness measures for other languages. In other words, in the case that there is not enough information to create a DSM for a particular language, this work aims at evaluating whether the benefit of corpora volume for English outperforms the error introduced by machine translation. Also for more investigation this section proposes the combination of a lightweight machine translation (MT) model (See Section 3.2.2) and an English DSM as a mechanism to provide knowledge-rich word vectors for languages other than English. While the problem of delivering high-quality sentence MT requires large parallel corpora and resource-intensive ML models, we claim that the MT for accessing distributional word vectors can be achieved with a lightweight model. In the context of this work, a lightweight MT model is a model which accesses the unigram-level source-target probabilities which can be directly computed from the parallel corpora.

In the proposed model the word-pairs datasets are translated into English as a reference language and the distributional vectors are defined over the target end model (Figure 3.1). Despite the simplicity of the proposed method based on lightweight machine translation, there is a high relevance for the distributional semantics user/practitioner due to its simplicity of use and the significant improvement in the results (See Chapter 6, section 5.2).

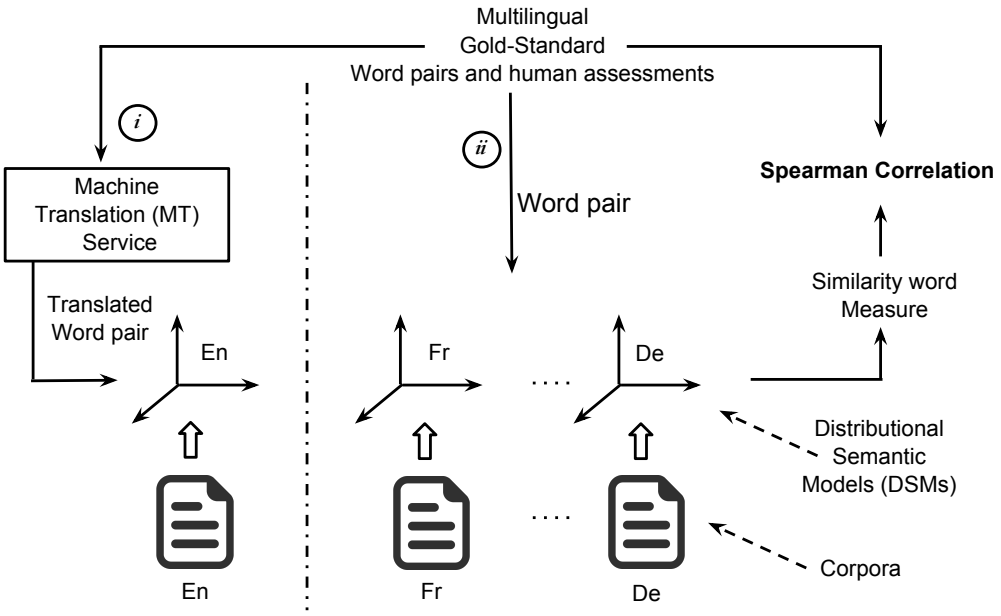

FIGURE 3.1: Depiction of the experimental setup of the experiment. i) Translate word-pairs into English by a Machine Translation and evaluate them over English DSM. ii) Evaluate words-pairs over language-specific models.

3.2.1 Related Work: Multi-lingual Distributional Semantic Models

The majority of related work has concentrated on leveraging joint multi-lingual information to improve the performance of semantic similarity/relatedness models. Faruqui and Dyer (2014b) use the distributional invariance across languages and propose a technique based on canonical correlation analysis (CCA) for merging multi-lingual evidence into vectors generated in a monolingual fashion. The authors evaluate the resulting word representations on semantic similarity/relatedness evaluation tasks, showing the improvement of the multi-lingual scenario over the monolingual one. Utt and Padó (2014) developed methods that take advantage of the availability of annotated corpora in English using a translation-based approach to transport the word-link-word co-occurrences to support the creation of syntax-based DSMs. Navigli and Ponzetto (2012) propose an approach to compute semantic relatedness exploiting the joint contribution of different languages mediated by lexical and semantic knowledge bases. The proposed model uses a graph-based approach of joint multi-lingual disambiguated senses which outperforms the monolingual scenario and achieves competitive results for both resource-rich and resource-poor languages. Zou et al. (2013) describe an unsupervised semantic embedding (bilingual embedding) for words across two languages that represent semantic information of monolingual words, but also semantic relationships across different languages. The motivation

for their work was on the difficulty of identifying semantic similarities across languages, especially when word co-occurrences are rare in the training parallel text. Al-Rfou, Perozzi, and Skiena (2013) produced multi-lingual word embeddings for about 100 languages using *Wikipedia* as the reference corpora.

3.2.2 Lightweight Machine Translation

The lightweight MT model is built by processing the set of source|target word alignments within the parallel corpora and by computing the $\omega(s|t)$ word translation table [1]. Given this alignment, it is quite straight-forward to estimate a maximum likelihood lexical translation table. Given a word pair w_1,w_2 in a language L other than English, the semantic similarity $sim(w_1, w_2)$ will be calculated by first collecting all English translations of w_1 and w_2 into the sets $\mathcal{T}_1$, $\mathcal{T}_2$. For a set which is defined by the cross product of $\mathcal{T}_1$, $\mathcal{T}_2$, the word vectors for each element τ_1^i, τ_2^j are produced ($\vec{\tau_1}^i$, $\vec{\tau_2}^j$). The final similarity score is given by getting the top-most similarity score $sim(\vec{\tau_1}^i, \vec{\tau_2}^j)$.

$$sim(w_1, w_2) = \arg\max_{\tau_1^i, \tau_2^j} sim(\vec{\tau_1}^i, \vec{\tau_2}^j)$$

Algorithm 1 describes the lightweight MT model.

Algorithm 1 The algorithm for computing the semantic similarity between two words with the translation.

WP : *word pair (w_1,w_2) in a language other than English*
$\tau_1 \leftarrow$ *Collects all English translations of* w_1 *from the Lexical translation table.*
$\tau_2 \leftarrow$ *Collects all English translations of* w_2 *from the Lexical translation table.*
CP : *Cross product of* τ_1 *and* τ_2
for all $pairs \in CP$ **do**:
    $Scores \leftarrow$ *Calculate* $sim(\vec{\tau_1}^i, \vec{\tau_2}^j)$.
end for
Return top-most similarity score in *Scores*

[1] IBM alignments models (Gal and Blunsom, 2013) or GIZA++ (Gal and Blunsom, 2013) can provide this information

Algorithm 2 The algorithm for looking up distributional vectors for a single word as a disambiguation mechanism.

SENT : *Sentence in a language other than English*
for all $W \in SENT$ **do**:
MW← *Meaningful words in SENT related to W* .
$\tau_w \leftarrow$ *Collecting all English translations of W from the Lexical translation table.*
for all $M \in MW$ **do**:
$\tau_m \leftarrow$ *Collecting all English translations of M from the Lexical translation table.*
end for
CP : *Cross product of* τ_w *and* τ_m
for all *pairs* $\in CP$ **do**:
Scores $\leftarrow$ *Calculate* $sim(\vec{\tau_w}^i, \vec{\tau_m}^j)$.
end for
$\vec{\tau_w}^i \leftarrow$ Based on the top-most similarity score in *Scores*
end for

In many cases, users of distributional semantic models need to use the word vectors directly instead of the similarity function (typically the case when using distributional word vectors as features for a machine learning model). An analogous procedure could be used as a disambiguation mechanism when looking up single word vectors. In this case, collocated words in the sentence can serve as a supporting mechanism for disambiguation. Algorithm 2 shows the variation of the model for looking up distributional vectors for a single word.

3.3 Domain Transportability

Distributional semantic models (DSMs) have emerged as simplified semantic representation models which are used in many different Natural Language Processing (NLP) tasks, such as Question Answering (Molino et al., 2012), Textual Entailment (Mehdad, Moschitti, and Zanzotto, 2010; Baroni et al., 2012) and Sentiment Analysis (Malandrakis et al., 2013). However, most of the discussions surrounding DSMs have concentrated on the construction and on the evaluation of models based on open-domain and large-scale corpora. This work aims to provide a systematic analysis of the construction, performance and usage of domain-specific DSMs.

Due to the reduced size and availability of domain-specific corpora, domain-specific DSMs (dsDSMs) suffer from scarcity of lexical associations, which are

much more frequent in very large generic corpora. This affects the parameterisation assumptions of dsDSMs: taking into account this scarcity, more systematic and rational decisions should be performed with regard to the selection of the parameters for a domain-specific setting. In addition to parameters such as weighting schemes, window size, distance measures and dimensionality reduction approaches, the quality and type of the discourse within the corpora (news, definitional and encyclopedic) may significantly impact the performance of the model.

This work investigates which distributional models and supporting corpora have better performance for domain-specific DSMs. Additionally, still within the scope of model rationalisation, we provide a model for the quantification of the amount of data necessary for building effective dsDSMs.

For that purpose, this section uses the financial domain of discourse for the analysis of dsDSMs. As existing test collections for evaluating dsDSMs are still limited, this work also introduces two new test collections, named *Syntagmatic Financial Word Pairs - 422 (SFWP-422)*[2] consists of 422 word pairs from the financial domain, which were manually annotated for semantic relatedness and *Paradigmatic FinNet Word Pairs - 3920 (PFWP-3920)*[3] consists of 3029 word pairs from the financial domain, which were annotated for semantic relatedness, to support the evaluation of the model.

3.3.1 Related Work: Domain-Specific Distributional Semantic Models

The contemporary understanding of distributional semantic models (DSMs) were introduced by (Landauer and Dumais, 1997; Schütze, 1998) and have since been applied to a vast range of semantic tasks in Natural Language Processing (see (Turney and Pantel, 2010a) for a survey). They have been deployed as lightweight semantic representations which are able to capture meaning at scale.

DSMs have been used to support semantic approximations in different tasks, improving the generalisation over classification tasks (as in Sentiment Analysis), where a word vector over a distributional vector space is used to represent a cluster of words within the vector space, allowing a concept-based generalisation, or to support semantic matching to address a vocabulary problem (Freitas et al., 2012; Furnas et al., 1987) as in Question Answering (Freitas, 2015a; Molino et al., 2012) and Textual Entailment (Mehdad, Moschitti, and Zanzotto, 2010; Baroni et al., 2012) tasks.

[2] Available at http://bit.ly/SFWP-422
[3] Available at http://bit.ly/PFWP-3920

Different applications induce variations in the way the parameterization of the DSM is defined, including the nature of the corpora, the size and type of context windows and their weighting schemes, the associated dimensional reduction strategy and the distance measures. While systematic evaluations of DSMs have been performed (Kiela and Clark, 2014), and different applications have done systematic comparisons between different models, there has not been a systematic evaluation of domain-specific DSMs with regards to the parameterizations required for smaller and more domain-specific corpora.

Domain-specific DSMs have been explored for the domains of health (Ghosh et al., 2016; Henriksson, 2015; Henriksson et al., 2015; Moen et al., 2015), recruitment (Shalaby et al., 2016), and Spoken Language Understanding (Anastasakos, Kim, and Deoras, 2014). In these domains, where annotated data are often scarce, large amounts of unstructured text are leveraged as lightweight semantic representations. Some applications require the injection of small amounts of structured knowledge.

Ghosh et al. (2016) build a word2vec model on the HealthMap corpus (Freifeld et al., 2008), a collection of Internet media reports on disease outbreaks. Structured information is added in the shape of disease vocabularies. They build not-curated taxonomy of health terms and their attributes in an unsupervised fashion which outperforms traditional distributional methods and only shows some taxonomic gaps due to the nature of reporting.

Henriksson (2015) create an ensemble of DSMs on clinical text for Named Entity Recognition (NER). Distributional representations for each category of Named Entities (NEs) are learned by calculating the centroid vector over all NEs of a given type. The results show that these distributed representations, when used as features in supervised classification, significantly enhance supervised NER. Henriksson et al. (2015) make use of DSMs in every step of their Information Extraction process for identifying adverse drug events in clinical text (NER, attribute labelling, Relation Extraction), using distributed representations as features in supervised classification. They report performance gains on all tasks when using DSMs.

Moen et al. (2015) focus on Finnish Electronic Health Records (EHRs), which are unstructured clinical notes about patients and their medical history. Using DSMs built on EHRs combined with semi-structured disease classification codes, they outperform the state-of-the-art in Information Retrieval. The role of the DSMs here is a lightweight representation which replaces a costly knowledge base or human judgement.

Shalaby et al. (2016) focus on entity classification in the recruitment domain. They combine a generic DSM learned from a Wikipedia-based corpus with a domain-specific DSM trained on 60 million job postings. The addition of this domain-specific component significantly outperforms the Wikipedia-based state-of-the-art.

Anastasakos, Kim, and Deoras (2014) apply DSMs to Spoken Language Understanding (SLU). They build DSMs on noisily labelled spoken queries from three domains (games, movies, music) and use them as additional features in a Conditional Random Field SLU system. The addition of domain-specific distributional representations improves results compared to using only generic data.

3.4 Composite Semantic Relation Classification

3.4.1 Semantic Relation Classification

Semantic relation classification is the task of classifying the underlying abstract semantic relations between target entities (terms) present in texts (Qin, Xu, and Guo, 2016). The goal of relation classification is defined as follows: given a sentence S with pairs of annotated target nominals e_1 and e_2, the relation classification system aims to classify the relations between e_1 and e_2 in given texts within the pre-defined relation set (Hendrickx et al., 2009). For instance, the relation between the nominal ***burst*** and ***pressure*** in the following example sentence is interpreted as ***Cause-Effect***(e_2, e_1).

> The $< e_1 >$ **burst** $< /\mathbf{e_1} >$ has been caused by water hammer $< e_2 >$ **pressure** $< /\mathbf{e_2} >$.

SemEval 2010 Task 8 (Hendrickx et al., 2009) focuses on Multi-Way classification of semantic relations between pairs of nominals. For instance, *student* and *association* are in an ***Member-Collection*** relation in *"The student association is the voice of the undergraduate student population of the State University of New York at Buffalo"*. They selected nine general relations plus *"OTHER"* is as follows:

- **Cause-Effect.** An event or object leads to an effect. *Example*: $< e_1 >$ *Smoking* $< /e_1 >$ causes $< e_2 >$ *cancer* $< /e_2 >$.
- **Instrument-Agency.** An agent uses an instrument. *Example*: $< e_1 >$ *Laser* $< e/_1 >$ $< e_2 >$ *printer* $< /e_2 >$
- **Product-Producer.** A producer causes a product to exist.*Example*: The $< e_1 >$ $farmer$ $< /e_1 >$ grows $< e_2 >$ *apples* $< /e_2 >$

- **Content-Container.** An object is physically stored in a delineated area of space, the container. *Example*: $< e_1 > Earth < /e_1 >$ is located in the $< e_2 > MilkyWay < /e_2 >$.
- **Entity-Origin.** An entity is coming or is derived from an origin (e.g., position or material). *Example*: $< e_1 > Letters < /e_1 >$ from $< e_2 > foreigncountries < /e_2 >$.
- **Entity-Destination.** An entity is moving towards a destination. *Example*: The $< e_1 > boy < /e_1 >$ went to $< e_2 > bed < /e_2 >$.
- **Component-Whole.** An object is a component of a larger whole. *Example*: My $< e_1 > apartment < /e_1 >$ has a large $< e_2 > kitchen < /e_2 >$.
- **Member-Collection.** A member forms a nonfunctional part of a collection. *Example*: There are many $< e_1 > trees < /e_1 >$ in the $< e_2 > forest/ < e_2 >$.
- **Communication-Topic.** An act of communication, whether written or spoken, is about a topic. *Example*: The $< e_1 > lecture < /e_1 >$ was about $< e_2 > semantics < /e_2 >$.

The final dataset contains a set of $10,717$ instances, where $8,000$ instances are defined as the training set. Table 3.1 shows the distribution of categories for the dataset. The second column (Frequency) shows the absolute and relative frequencies of each relation.

TABLE 3.1: Annotation Statistics of relation types with absolute and relative frequency in the dataset.

Relation	**Frequency**
Cause-Effect	1331 (12.4%)
Component-Whole	1 253 (11.7%)
Entity-Destination	1137 (10.6%)
Entity-Origin	974 (9.1%)
Product-Producer	948 (8.8%)
Member-Collection	923 (8.6%)
Message-Topic	895 (8.4%)
Content-Container	732 (6.8%)
Instrument-Agency	660 (6.2%)
Other	1864 (17.4%)
Total	10717 (100%)

3.4.2 Existing Approaches for Single Semantic Relation Classification

Different approaches have been explored for relation classification, including **unsupervised** and **supervised** relation discovery and classification. Existing literature have proposed various features to identify the relations between entities using different methods, which are described in the following paragraphs.

In the unsupervised methods, contextual features are used. The distributional hypothesis Harris (1954) indicates that words that have similar meanings, probably occur in the same context. Accordingly, it is assumed that the pairs of words that occur in similar contexts tend to have similar relations. Hasegawa, Sekine, and Grishman (2004) used the contexts of nominal words using a hierarchical clustering method which represents the relationship between the words by using the most frequent words in the contexts as contextual features. Further work by Chen et al. (2005) suggested an unsupervised algorithm based on model-order selection and discriminative label identification.

In the supervised methods, approaches can be grouped into two types: *feature-based* and *kernel-based* (see Zeng et al. (2014) for more details). The performance of these models strongly depend on the quality of the designed features. Recently, neural network-based approaches have achieved significant improvement over traditional methods based on human-designed features (Qin, Xu, and Guo, 2016). Existing neural networks for relation classification are usually based on shallow architectures (e.g., one-layer convolutional neural networks or recurrent networks). In exploring the potential representation space at different abstraction levels, they may fail to perform (Xu et al., 2016). The performance of supervised approaches strongly depends on the quality of the designed features (Zeng et al., 2014). Complementarily, some models are exploring automatic feature learning strategies. Xu et al. (2015b) apply gated recurrent networks, in particular, Long Short-Term Memories (LSTMs) to relation classification. Zeng et al. (2014) also use Convolutional Neural Networks (CNNs) for the same task. Additionally, Santos, Xiang, and Zhou (2015a) replace the common Softmax loss function with a ranking loss in their CNN model. Xu et al. (2015a) design a negative sampling method based on CNNs. From the viewpoint of model ensembles, Liu et al. (2015) combine CNNs and recursive networks along the Shortest Dependency Path (SDP), Nguyen and Grishman (2015) incorporate CNNs with Recurrent Neural Networks (RNNs).

Additionally, much effort has been invested in relational learning methods that can scale to large knowledge bases. The best performing neural-embedding models are NTN (Socher et al., 2013a) and TransE and TATEC models (Bordes et al.,

2013; Garcia-Duran et al., 2016), respectively.

In summary, supervised learning approaches can perform very well, depending on how shallow or deep the models describing the features of a given relation are.(Nastase et al., 2013).

3.4.3 From Single to Composite Semantic Relation Classification

3.4.3.1 Introduction

The goal of this work is to propose an approach for semantic relation classification using one or more relations between entities/term mentions. The reason for the emphasis on composite semantic relation classification (CSRC) is the fact that distributional semantic relatedness cannot always be associated with a single named relation, and the link between the quantified view of distributional semantic relatedness scores (called in the context of this thesis coarse-grained semantics) and the named semantic relations (fine-grained semantics) are achieved with CSRC.

In the example below, the relationship between *Child* and *Cradle* cannot be directly expressed by one of the nine abstract semantic relations from the set described in (Hendrickx et al., 2009).

The $< e_1 >$ *child* $< /e_1 >$ was carefully wrapped and bound into the $< e_2 >$ *cradle* $< /e_2 >$ by means of a cord.

Assume R_1 be a relation from X to Y, and R_2 be a relation from Y to Z. Then a relation written as $R_1 \circ R_2$ is called a composite relation of R_1 and R_2 where

$$R_1 \circ R_2 = \{(x, z) | x \in X \wedge z \in Z \wedge (\exists y)(y \in Y \wedge (x, y) \in R_1 \wedge (y, z) \in R_2))\}$$

We can also write the composition as

$$R_1 \circ R_2 = \{(x, z) | x \in X \wedge z \in Z \wedge (\exists y)(y \in Y \wedge xR_1y \wedge yR_2z)\}$$

Based on the definition of relation composition, we need to instantiate the relation set taking into account a specific *ontology*. While the Semeval dataset provides a possible ontology, a commonsense KB (in this case, *ConceptNet* $V5.4$), can provide an initial instantiated set of composite relationships, while keeping a similar set of relations to Semeval. For the previous example:

$< e_1 >$ *child* $< /e_1 >$ **CreatedBy** $\circ$ **Causes** $\circ$ **AtLocation** $< e_2 >$ *cradle* $< /e_2 >$

With the increase in the number of edges which can be included in the set of semantic relation compositions (the size of the semantic relationship path), there is a dramatic increase in the number of paths which connect the two entities. For example, for the words *child* and *cradle* there are 15 paths of size 2, "1,079" paths of size 3 and "95,380" paths of size 4. Additionally, as the path size grows, many non-relevant relationships (less meaningful or redundant relations) will be included.

The challenge in *composite semantic relation classification* is to provide a classification method that provides the most meaningful (see more details on Section 3.4.3.3) set of relations for the context at hand. This task can be challenging because, as previously mentioned, a simple KB lookup based approach would provide all semantic associations at hand. To achieve this goal we propose an approach which combines *sequence-based machine learning models*, *distributional semantic models* and *commonsense relational knowledge bases* to provide an accurate method for composite semantic relation classification. The proposed model (Fig 3.2) relies on the combination of the following approaches:

i Using existing structured commonsense KBs to define an initial set of semantic relation compositions.

ii Using a pre-filtering method based on the Distributional Navigational Algorithm (DNA) as proposed by (Freitas et al., 2014; Silva, Handschuh, and Freitas, 2018a).

iii Using a sequence-based Neural Network based model to quantify the sequence probabilities of the semantic relation compositions. We call this model Neural Entity/Relation Model (`NERM`); an analogy to a Language Model.

3.4.3.2 Commonsense KB Lookup

The first step consists in the use of a large commonsense knowledge base for providing a reference for a sequence of semantic relations. *ConceptNet* (Speer and Havasi, 2012) is a semantic network built from existing linguistic resources and crowd-sourced. It is built from nodes representing words or short phrases as observed in natural language and labelled abstract relationships between them. There are a few alternatives for large commonsense KBs, such as *WordNet*, *Microsoft Concept Graph* and *DBpedia*. In the list below, *ConceptNet* is contrasted to other KBs:

- **WordNet**: *ConceptNet* has more relation types than *WordNet* (Miller, 1995). Additionally, *ConceptNet*'s vocabulary[4] is much larger and contains

[4] 28 million statements

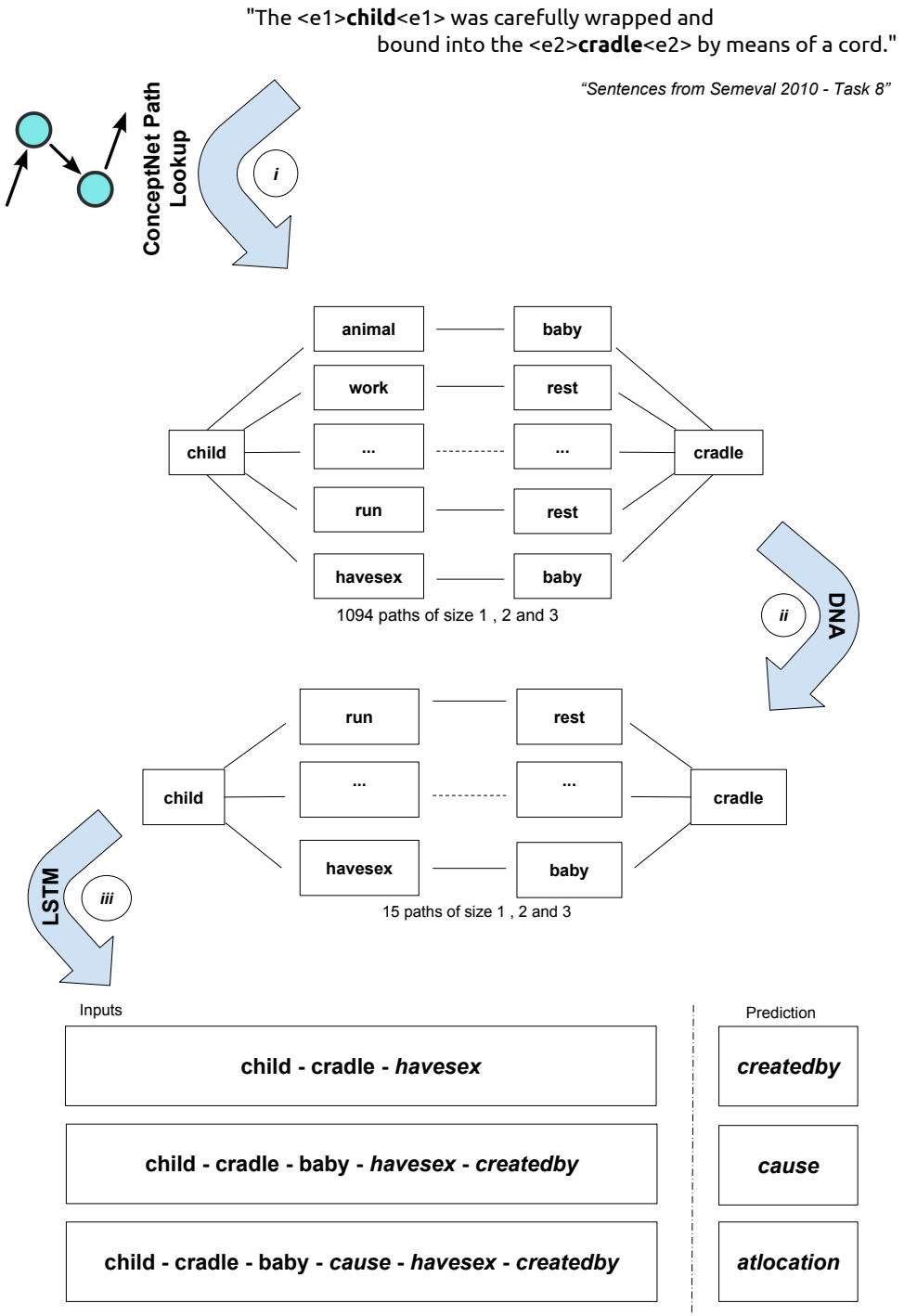

FIGURE 3.2: Depiction of the proposed model relies on the combination of our three approaches.

more links between the Concepts. *ConceptNet* does not assume that words fall into *synsets*. Furthermore, *synonymy* in ConceptNet is a relation like any other. *ConceptNet* reuses some relations from WordNet. Moreover their importance is weighted higher, given that the knowledge in *WordNet* is handcrafted, accurate and of high quality.

- **Microsoft Concept Graph**: *Microsoft Concept Graph* (Wang et al., 2015) is a taxonomy of English nouns[5] containing *IsA* relations
- **DBpedia**: *DBpedia* (Auer et al., 2007) is focused on named entities and basic factoid-style attributes. In contrast, *ConceptNet* focuses on noun/verb-level entities and their abstract relations.

ConceptNet is used as a large commonsense knowledge base for the proposed model. The intuition is that any type of relation classification task would need to be based on large-scale commonsense knowledge either in a distributional or structured/relational form. ConceptNet[6] has been built from several sources, such as:

- Information extracted from parsing Wiktionary (Zesch, Müller, and Gurevych, 2008).
- Open Multilingual WordNet (Bond and Foster, 2013).
- Open Mind Common Sense (Singh, 2002).
- A subset of DBpedia (Auer et al., 2007).

ConceptNet has a long-tail distribution of relations. However, the more frequent relations expressed at *ConceptNet* are[7]:

- **Symmetric relations:** *Antonym, DistinctFrom, EtymologicallyRelatedTo, LocatedNear, RelatedTo, SimilarTo, and Synonym.*
- **Asymmetric relations:** *AtLocation, CapableOf, Causes, CausesDesire, CreatedBy, DefinedAs, DerivedFrom, Desires, Entails, ExternalURL, FormOf, HasA, HasContext, HasFirstSubevent, HasLastSubevent, HasPrerequisite, HasProperty, InstanceOf, IsA, MadeOf, MannerOf, MotivatedByGoal, ObstructedBy, PartOf, ReceivesAction, SenseOf, SymbolOf, and UsedFor.*

For our target example, 1,094 paths were extracted from *ConceptNet*for two given entities (e.g. *child* and *cradle*) such that they contained **no corresponding semantic relation** from the *SemEval* 2010 Task 8 test collection (Figure 3.2(i)). Examples of paths are[8]:

- **child/CanBe/baby/AtLocation/cradle**
- child/IsA/animal/HasA/baby/AtLocation/cradle
- child/HasProperty/work/CausesDesire/rest/Synonym/cradle

[5] 5.4 million concepts
[6] Version 5.5
[7] https://github.com/commonsense/conceptnet5/wiki/Relations
[8] Paths in bold are considered semantically relevant.

- child/InstanceOf/person/Desires/baby/AtLocation/cradle
- child/DesireOf/run/CausesDesire/rest/Synonym/cradle
- **child/CreatedBy/havesex/Causes/baby/AtLocation/cradle**

Although ConceptNet can provide a large commonsense Knowledge Base, as we transcend immediate single relations, paths start to conceptually drift away from the *source* and *target* concepts. In order to filter these relations into a set of semantically relevant paths, we apply the Distributional Navigational Algorithm (DNA), described in the next section.

3.4.3.3 Distributional Navigational Algorithm (DNA)

The Distributional Navigational Algorithm (*DNA*) consists of an approach which uses distributional semantic models as a relevance-based heuristic for selecting relevant facts attached to a contextual query over a structured KB. DNA provides an abductive reasoning style mechanism which operates over Distributional-Relational Models (Freitas, 2015a; Freitas, Handschuh, and Curry, 2015; Freitas and Silva, 2014; Freitas, Curry, and Handschuh, 2014; Freitas et al., 2013a), i.e. models which enrich structured logical (triple-style) KBs with word-embedding style information.

The DNA approach focuses on addressing the following problems: (i) providing a semantic selection mechanism for facts which are relevant and meaningful in a particular reasoning & querying context and (ii) allowing coping with information incompleteness in large KBs. The DNA model starts from the *source* entity and navigates through the KB, computing the distributional semantic relatedness between the set of lexical elements associated with neighbouring nodes in the graph and the *target* entity. The semantic relatedness function is defined as:

$$sr(\overrightarrow{p_1}, \overrightarrow{p_2}) = cos(\theta) = \overrightarrow{p_1}.\overrightarrow{p_2}$$

where $sr : VS^{dist} \times VS^{dist} \rightarrow [0, 1]$.

An important point to emphasise is the fact that the distributional semantic relatedness function is defined *over an external/independent corpus* (in contrast to many existing approaches which define the embeddings based on the KB). The DNA method is not coupled to a specific distributional/word embedding model and can use different types of models.

A threshold $\eta \in [0, 1]$ can be used to establish the desired semantic relatedness between two vectors: $sr(\overrightarrow{p_1}, \overrightarrow{p_2}) > \eta$. The information provided by the semantic relatedness function sr is used to identify elements in the KB with a

similar meaning from the reference corpus perspective. The threshold is calculated following the semantic differential approach proposed in (Freitas et al., 2014). Multiword phrases are handled by calculating the centroid between the concept vectors defined by each word in the Distributional Navigation Algorithm (DNA) (Algorithm 3) (Freitas et al., 2014; Silva, Handschuh, and Freitas, 2018a).

Algorithm 3 Distributional Navigational Algorithm

η : *threshold*
$(source, target)$: *pair of terms such as that* $sr(\overrightarrow{source}, \overrightarrow{target}) > \eta$
l : *path length*
$RankedPaths$: *a set of ranked score paths* $< (t_0, ..., t_1), score >$ *such that* $t_0 = source$ *and* $t_1 = target$
- -
$t_0 \leftarrow source$
$Paths \leftarrow 0$
$ExplorePaths \leftarrow [(< t_0 >, sr(\overrightarrow{t_0}, \overrightarrow{target}))]$
while $ExplorePaths \neq 0$ **do**
  remove $(< t_0, ..., t_k >, sr(\overrightarrow{t_k}, \overrightarrow{target}))$ **from** $ExplorePaths$
  if $k < l$ - *1* **then**
    for all $(n \in neighbours(t_k) : sr(\overrightarrow{n}, \overrightarrow{target}) > \eta$ *and* $n \notin \{ t_0, ..., t_k \})$ **do**
      append $(< t_0, ..., t_k, n >, sr(\overrightarrow{n}, \overrightarrow{target}))$ **to** $ExplorePaths$
    end for
  else if $k = l$ - *1* **then**
    append $(< t_0, ..., t_k, target >, 1)$ **to** $Paths$
  end if
end while
$RankedPaths \leftarrow sort(Paths)$
return $RankedPaths$

In summary, given two semantically related terms *source* and *target* wrt a threshold η, the algorithm finds all paths from *source* to *target*, with length l, formed by concepts semantically related to *target* wrt η.

The *source* term is the first element in all paths. From the set of paths to be explored (*ExplorePaths*), the *DNA* selects a path and expands it with all neighbours of the last term in the selected path that are semantically related *wrt* the threshold η and but do not appear in that path. The stop condition is $sr(target, target) = 1$ or when the maximum path length is reached.

The paths $p =< t_0, t_1, \ldots, t_l >$ $(where \quad t_0 = source \quad and \quad t_l = target)$ found by *DNA* are ranked according to the following formula:

$$rank(p) = \sum_{i=0}^{l} sr(\overrightarrow{t_i}, \overrightarrow{target})$$

Algorithm 3 can be modified to use a heuristic that allows to expand only the paths for which the semantic relatedness between all the nodes in the path and the target term increases along the path. The differential in the semantic relatedness for two consecutive iterations is defined as $\Delta target(t_1, t_2) = sr(\overrightarrow{t_2}, \overrightarrow{target})$ - $sr(\overrightarrow{t_1}, \overrightarrow{target})$, for terms t_1, t_2 and $target$. This heuristic is implemented by including an extra test i.e., $\Delta target(t_k, n) > 0$.

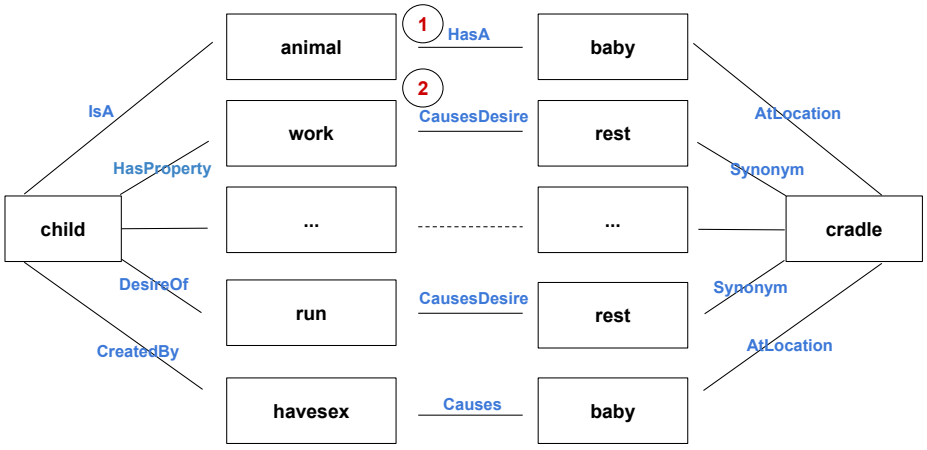

FIGURE 3.3: Selection of semantically relevant paths.

In Freitas et al. (2014), DSMs are used as a complementary semantic layer to the relational model, which supports coping with semantic approximation and incompleteness. For large-scale and open domain commonsense reasoning scenarios, model completeness, and full materialisation cannot be assumed. A commonsense KB would contain vast amounts of facts, and a complete inference over the entire KB would not scale to its size. Although several meaningful paths may exist between two entities, there are a large number of paths which are not meaningful in a specific context. For instance, the reasoning path which goes through path (1) at Figure 3.3 is not related or relevant to the classification goal of the entity pairs (the relation between *Child* of human and *Cradle*) and should be eliminated by the application of the Distributional Navigation Algorithm (DNA) (Freitas et al., 2014; Silva, Handschuh, and Freitas, 2018a), which computes the distributional semantic relatedness between the entities and the intermediate entities in the knowledge base path as a measure of semantic coherence. In this case, the algorithm navigates from first entity (e_1) in the direction of target entity (e_2) in the Knowledge Base using distributional semantic relatedness between the target node e_2 and the intermediate nodes e_n as a heuristic method.

3.4.3.4 Neural Entity/Relation Model (NERM)

The Distributional Navigational Algorithm provides a pre-filtering of the relations maximising the semantic relatedness coherence. This can be complemented

by a predictive model which takes into account the likelihood of a sequence of relations, i.e. the likelihood of a composition sequence (Algorithm 4). The goal is to systematically compute the sequence of probabilities of entity-relation compositions, in a similar fashion to a language model. As such, the model will capture the notion of a sequence compatibility between entities and relations.

Various machine learning models are used in the context of NLP in order to induce language models and KB-based models for having a relation classification, *Recursive Neural Networks* (Socher et al., 2013b), *Recurrent Neural Networks* (Elman, 1990; Mikolov et al., 2010), *Long Short Term Memory Networks* (Hochreiter and Schmidhuber, 1997), *Neural Tensor Networks*(Socher et al., 2013a) and *Convolutional Neural Networks* (Kim, 2014). In this work, due to the nature of the prediction task, which is similar to the language model (identifying semantically coherent sequences of relations), we will target recurrent sequence classification types of models (see Figure 3.5).

Algorithm 4 Composite Semantic Relation Classification

I : *sentences of semeval 2010-Task 8 dataset*
O : *predefined entity pairs* (e_1, e_2)
W : *words in* I
R : *related relations of* w
for all $s \in I$ **do**:
  $S \leftarrow$ *If entities of* s *are connected in an* $OTHER$ *relation*
end for
for all $s \in S$ **do**:
  $ep \leftarrow$ *predefined entity pairs of* s
  $p \leftarrow$ *find all path of* ep *in ConceptNet (with maximum paths of size 3)*
  for all $i \in p$ **do**:
    $sq_i \leftarrow$ *avg similarity score between each word pairs (Barzegar et al., 2015)*
  end for
  $msq \leftarrow$ *find max* sq
  for all $i \in p$ **do**:
    filter i *If* $sq_i < msq$ - $\frac{msq}{2}$
  end for
  $dw \leftarrow$ *convert* s *into a suitable format for deep learning*
end for
$model \leftarrow$ *learning LSTM with* dw *dataset*

Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) is a type of recurrent neural network (*RNN*) architecture whose advantages over other RNN models have been proved in different tasks in NLP (Barzegar et al., 2017; Sutskever, Vinyals, and Le, 2014; Cortis et al., 2017). The advantage of *LSTM* against *RNN* is that allows the network to capture information from inputs for a long time using a special hidden unit (z_t). A LSTM unit at a timestep t is

described by the following equations:

$$i_t = \delta(W_{x,i}x_t + W_{z,i}z_{i-1})\ (input\ gate)$$

$$f_t = \delta(W_{x,f}x_t + W_{z,f}z_{f-1})\ (forget\ gate)$$

$$o_t = \delta(W_{x,o}x_t + W_{z,o}z_{o-1})\ (output\ gate)$$

$$g_t = tanh(W_{x,g}x_t + W_{z,g}z_{g-1})\ (input\ modulation\ gate)$$

$$m_t = f_t \circ m_{t-1} + i_t \circ g_t\ (memory\ cell)$$

$$z_t = o_t \circ tanh(m_t)(hidden\ state)$$

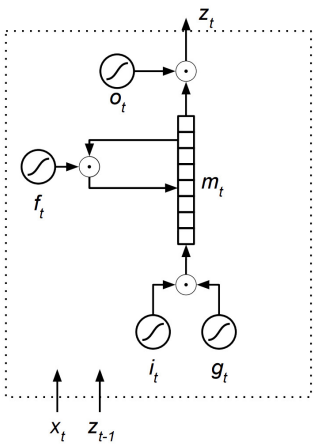

FIGURE 3.4: Long Short-Term Memory unit at timestep t. (Small circles with dots are elementwise vector multiplications).

The memory vector m_t is a function of two parts: (1) its previous value m_{t-1} modulated by the forget gate f_t (2) the information of the current input x_t and previous hidden state (z_t) which modulated by the input modulation gate (g_t). Long Short-Term Memory unit at timestep t has four nonlinearity nodes (i_t, g_t, f_t, and o_t) all have, as inputs, x_t and z_{t-1}. (Pichotta and Mooney, 2016; Kong et al., 2018). *LSTM* can memorise long sequences. The model's input is a sequence of entities and their relations with a specific order. For example, an input for our LSTM model is $child - cradle - baby - cause - havesex - creadtedby$ which *model* should predict *atlocation* label. The model is depicted graphically in Figure 3.4.

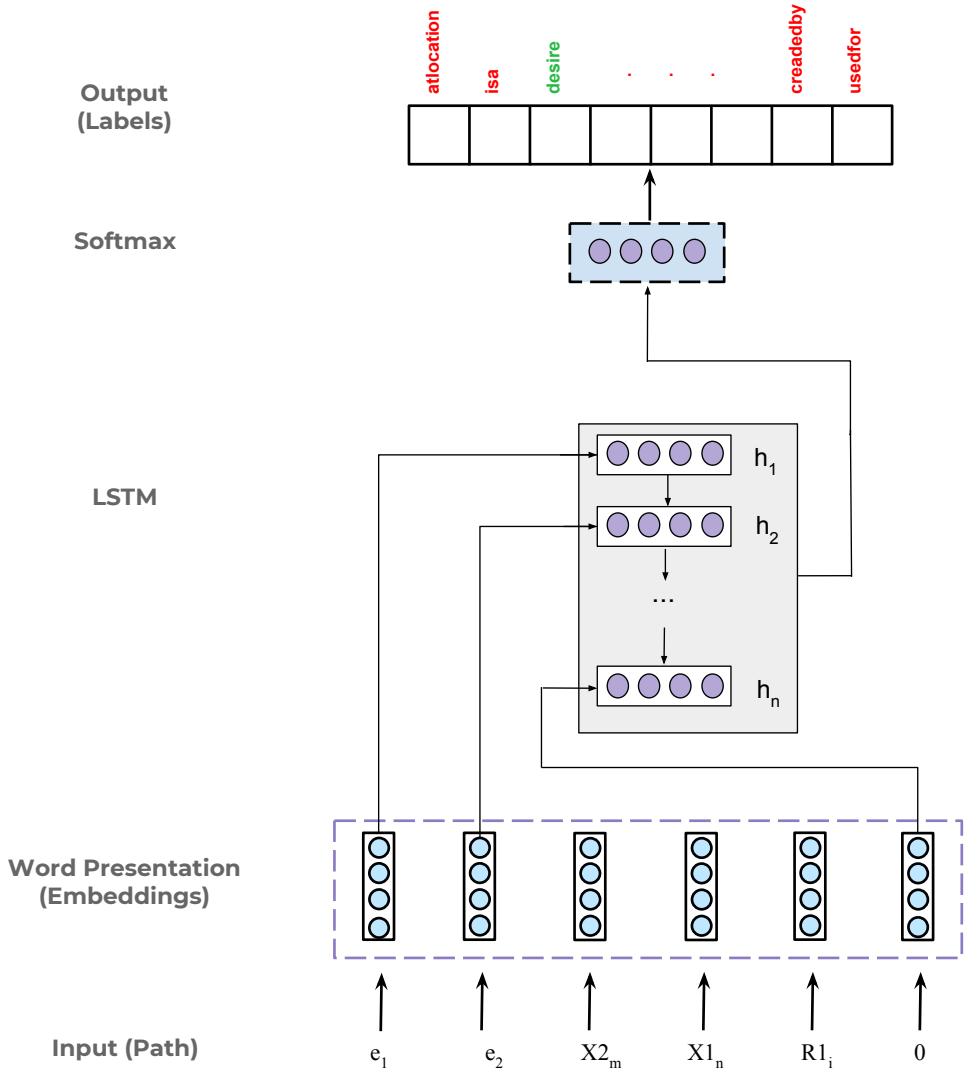

FIGURE 3.5: An overall picture of the Architecture of the Neural Entity/Relation Model that describes a predictive model which takes into account the likelihood of a sequence of relations

3.5 Indra: An Unified Architecture

Word embedding is a popular semantic model which represents words and sentences in computational linguistics systems and machine learning models. In recent years a large set of algorithms for both generating and consuming word embedding models (*WEMs*) have been proposed, which includes corpus preprocessing strategies, WEM algorithms or weighting schemes, vector compositions and distance measures (Turney and Pantel, 2010a; Lapesa and Evert, 2014; Mitchell and Lapata, 2010). Determining the optimal set of strategies for a given problem demands the support of a tool that facilitates the exploration of the

configuration space of parameters. Furthermore, given the applicability and maturity achieved by these systems and models, they have been promoted from academic prototypes to industry-level applications (Loebbecke and Picot, 2015; Hengstler, Enkel, and Duelli, 2016; Moro, Cortez, and Rita, 2015). In this new production scenario, a candidate tool should be able to scale to large number of requests and to the construction of models from large corpora, making use of parallel execution and traceability. From the functional point of view, *Integrated corpus pre-processing*, *Generation of predictive-based and count-based models* and *Unified access as a service* are key features.

In recent years *Distributional Semantic/Word Embedding* Models evolved to become a fundamental component in many natural language processing (*NLP*) architectures due to their ability of capturing and quantifying semantic associations at scale. Word embedding models can be used to satisfy recurrent tasks in NLP such as lexical and semantic generalisation in machine learning tasks, finding similar or related words and computing semantic relatedness of terms. However, building and consuming specific word embedding models require the setting of a large set of configurations, such as corpus-dependant parameters, distance measures as well as compositional models. Despite their increasing relevance as a component in NLP architectures, existing frameworks provide limited options in their ability to systematically build, parametrize, compare and evaluate different models. To answer this demand, this thesis describes INDRA as an unified architecture, a multi-lingual word embedding/distributional semantics framework which supports the creation, use and evaluation of word embedding models. INDRA provides a software infrastructure to facilitate the experimentation and customisation of multilingual *WEMs*, allowing end-users and applications to consume and operate over multiple word embedding spaces as a service or library.

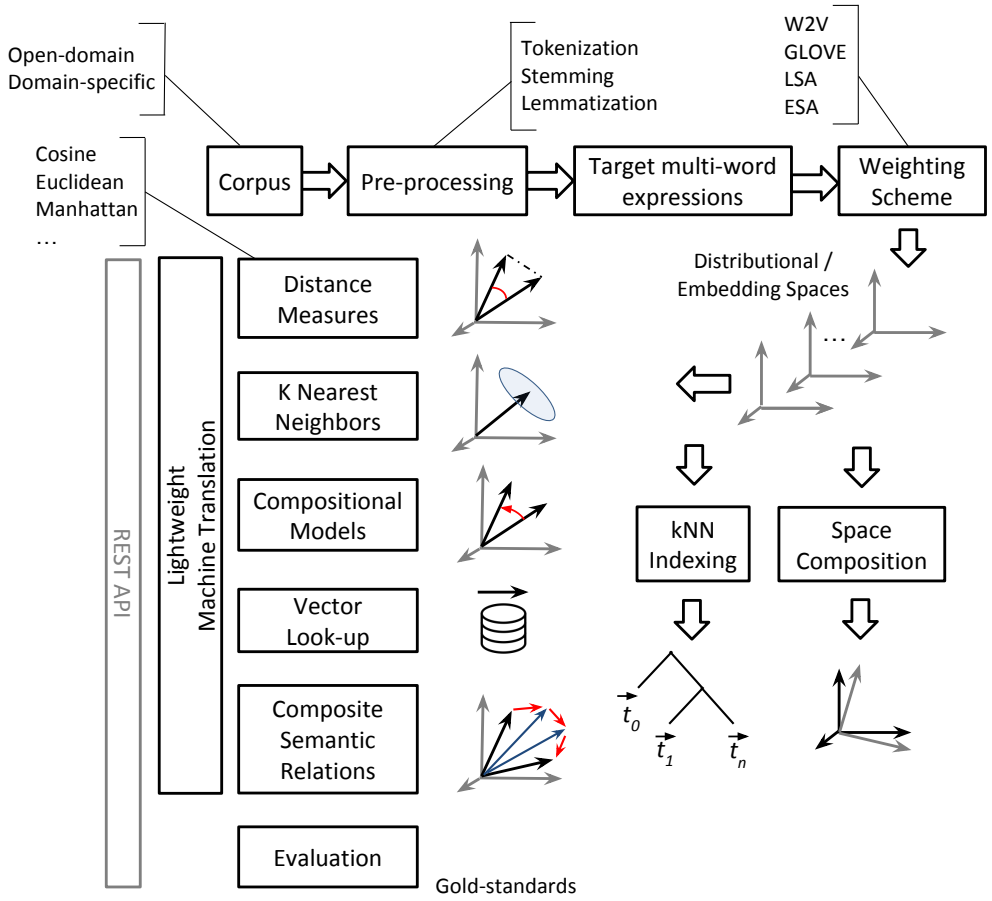

FIGURE 3.6: High-level architecture of the Indra framework including implemented features.

INDRA is designed to encapsulate the set of recurring semantic tasks associated with word embeddings (both their generation and their consumption) through a well defined API. This API is materialised as a software which can be consumed as a stand-alone library and also a REST service.

Figure 3.6 depicts the main components of its architecture. INDRAINDEXER supports the generation of WEMs directly from text files (Wikipedia-dump or plain-text formats), passing through the corpus pre-processing and multiword expression identification, to the model generation itself. INDRA dynamically builds the pipeline based on the metadata information produced during the model generation. This strategy guarantees that the same set of pre-processing operations are consistently applied to the input query. Additionally, the translation-based word embedding (Freitas et al., 2016; Barzegar et al., 2018a) can be conveniently activated in the pipeline as described in Chapter 4.

Different languages (Section 3.2), domains (Section 3.3) and application scenarios (including the link to semantic relation classification - See Section 3.4) require

different parameterizations of the underlying embedding models. Together with the availability of pre-generated models, INDRA's system architecture favors the exploration of a large grid of parameters. INDRA currently shares more than 65 pre-computed models which vary in languages, model algorithms and corpora (general-purpose and domain-specific). The list of available models is in the Github project's Wiki.

INDRA is available from two repositories (`github.com/Lambda-3/Indra` and `github.com/Lambda-3/IndraIndexer`) both licensed as open-source software. Additionally, INDRA also provides a Python client (`pyindra`) available via `pip` and from `github.com/Lambda-3/pyindra`.

3.5.1 Related Work: Distributional Semantics and Word Embeddings Libraries

S-SPACE is a library to support the construction of count-based distributional methods unifying different approaches in a common JAVA API (Jurgens and Stevens, 2010b). DEEPLEARNING4J[9], on the other hand, is a library which concentrates predictive-based models. DEEPLEARNING4J is also written in JAVA and its API contains methods to access word vectors and to find nearest neighbours (kNN). GENSIM is one of the most popular word-embedding tool-kits, mainly credited to its efficient implementation of nearest neighbours function (Řehůřek and Sojka, 2010). GENSIM is written in PYTHON and apart from its *kNN* function, it supports the generation of predictive-based models and methods to access word vectors. Following a different motivation, DISSECT (DIStributional SEmantics Composition Tookit) focuses on vector compositions (Dinu, Pham, and Baroni, 2013). DISSECT is a PYTHON library containing methods to generate vector representation of sentences from the vector of its constituting words. DISSECT partially supports the generation of count-based models and brings an integrated baseline framework for evaluation purposes. JOBIMTEXT is a semantic similarity tool that implements its own algorithm named JoBim (Biemann et al., 2013). The tool supports the construction of the JoBim model and also calculates semantic relatedness of pairs of terms, finds nearest neighbours and offers a native web server. EasyESA (Carvalho et al., 2014) and DInfra (Barzegar et al., 2015) are also two initiatives to deliver distributional semantics capabilities under a more specific set of distributional semantic models.

Table 3.2 summarises a comparative analysis of the main frameworks and their features. Apart from INDRA, none of the listed frameworks gives support to corpora pre-processing (which will be detailed in Section 3.5.2.1). Other limitations addressed by INDRA are *(i)* the generation of both count-based and

[9] https://deeplearning4j.org/

predictive-based models, *(ii)* the support for vector composition and *(iii)* the support for translation-based models. Finally those libraries offer a limited set of pre-computed models, which makes the process of exploration time-consuming and computationally costly. INDRA shares more than 65 pre-computed models in 14 languages. INDRA aims at covering these gaps by providing an end-to-end infrastructure to build, consume and evaluate multi-lingual word embedding models.

TABLE 3.2: List of functionalities and framework coverage. In the line *Support to model generation*, P stands for *predictive-based models* and C for *count-based models*. *DISSECT partially supports the generation of count-based models.

Features	**Indra**	**GenSim**	**DeepLearning4J**	**S-Space**	**JoBim**	**DISSECT**
			Word Embeddings			
Simple vectors	✓	✓	✓	✓		✓
Composed vectors	✓					✓
Translation-based vectors	✓					
Word embeddings as a service	✓					
			Semantic Relatedness			
Word pair relatedness	✓	✓	✓	✓	✓	✓
Top-k nearest neighbours	✓	✓	✓	✓	✓	
Score-based nearest neighbours	✓				✓	
Translation-based relatedness	✓					
Composed-vector relatedness	✓					✓
Multiple score functions	✓					✓
Relatedness as a service	✓				✓	
			Model Generation and Other Functions			
Integrated corpus pre-processing	✓					
Support to model generation	P/C	P	P	C	JoBim	C*
Support to sparse vector models	✓			✓	✓	✓
Built-in word disambiguation					✓	
English pre-computed models	✓	✓			✓	
Multi-lingual pre-computed models	✓					
Multi-model querying	✓					
Semantic Relation Classification	✓					

3.5.2 Distributional Semantic Models

3.5.2.1 Text Pre-processing

One important step in the construction of word embeddings models is pre-processing the texts. Defining the tokenization strategy, which depends on the language, whether or not words must be lower-cased or stemmed is part of the pre-processing step. Furthermore, the pre-process strategy should be stated consistently during both construction and consumption phases as exemplified further.
The corpus pre-processor is responsible for defining the tokenization strategy and the tokens' subsequent transformations. It defines, for example, if `United States of America` corresponds to a unique or to multiple tokens. *Stem* and *lowercase* are two other popular transformations also supported by the pre-processor. INDRA uses the Lucene's `StandardTokenizer`, which implements the Unicode Text Segmentation algorithm based on the *word breaking* rules defined in the Unicode Standard Annex #29 (Davis and Iancu, 2017). Additionally, INDRA allows users to specify a customised list of multi-word expressions which will be considered a unique token, independently of the tokenizer rules. This mechanism allows, for example, modelling a unique vector for named entities such as *Nelson Mandela* and *Republic of Austria*.

As in the context of WEM numbers are usually disregarded tokens, the pre-processing step allows replacing them by a default placeholder (`<NUMBER>`). INDRA pre-processor also allows specifying stopwords, whose occurrences are removed from the text. Table 3.3 shows the full list of operations supported by the pre-processor.

The pre-processor is defined as a package that is attached to both INDRAINDEXER and INDRA in order to guarantee that the consuming functions apply the same set of operations in retrieval time.

TABLE 3.3: Parameters supported by the INDRA's pre-processing package.

Parameter	Description/Options
input format	Wikipedia-dump format or plain texts from one or multiple files.
language	14 supported languages.
set of stopwords	a set of tokens to be removed.
set of multi-word expressions	set of sequences of tokens that should be considered a unique token.
apply lowercase	lowercase the tokens.
apply stemmer	applies the Poter Stemmer in the tokens.
remove accents	remove the accents of words.
replace numbers	replaces numbers for the place holder $< NUMBER >$.
min	set a minimum acceptable token size.
max	set a maximum acceptable token size.

3.5.2.2 Model Generation

IndraIndexer is the module responsible for the generation of word embedding models. It defines a unified interface to generate predictive-based models (e.g. Skip-gram (Mikolov et al., 2013b) and Global Vectors (Pennington, Socher, and Manning, 2014b)) and count-based models (e.g. Latent Semantic Analysis (Dumais et al., 1988) and Explicit Semantic Analysis (Gabrilovich and Markovitch, 2007b)) whose implementation comes from the libraries DeepLearning4J[10] and S-Space (Jurgens and Stevens, 2010b) respectively. In addition to creating a unified interface for WEM algorithms, IndraIndexer integrates the corpus pre-processor package.

IndraIndexer receives as input the pre-processed corpus and outputs the vectors in binary files in a format compatible with *GenSim* (Řehůřek and Sojka, 2010). In addition to the vector file, IndraIndexer also generates a metadata file containing all the parameters from both the pre-processing and generation steps. Figure 3.7 shows an example of a metadata file.

```
{
   "windowSize" : 5,
   "minWordFrequency" : 5,
   "corpusMetadata" : {
      "corpusName" : "wiki-2014",
      "stopWords" : ["been", "don't", ...],
      "replaceNumbers" : false,
      "applyStemmer" : 3,
      "removeAccents" : true,
      "maxTokenLength" : 100,
      "minTokenLength" : 3,
      "description" : null,
      "language" : "en",
      "encoding" : "UTF-8",
      "applyLowercase" : true
   },
   "vocabSize" : 1181258,
   "sparse"  :false,
   "model" : "w2v",
   "dimensions" : 300
}
```

Figure 3.7: Example of metadata file generated by IndraIndexer which describes how the user configured both the pre-processor and the WEM generator.

During the consumption phase, Indra applies the same set of options to guarantee consistence. For instance, let's assume a given model was generated by applying the stemmer and lowercase to the tokens. It means that the term `University`

[10] https://deeplearning4j.org/

is represented in the model as `univers`. When it is required to retrieve the vector representation of `University`, INDRA guarantees this consistence by executing the pre-processing steps in the query at runtime. This method simplifies the execution of experiments that consumes models using different set of pre-processing transformations.

3.5.3 Semantic Relatedness

Natural language understanding systems use semantic relatedness in fine-grained tasks such as *word disambiguation* (Freitas et al., 2013c) or more coarse-grained such as *paraphrase detection* (Sales et al., 2016; Silva, Handschuh, and Freitas, 2018b), *semantic parsing* (Sales, Freitas, and Handschuh, 2018) and *question answering* (Freitas, 2015b). INDRA implements two semantic relatedness methods. The first is the pair-wise semantic relatedness in which the user provides pairs of terms to calculate their semantic relatedness. The other option is integrated to the nearest neighbours function which returns the relatedness of the k closest terms. Additionally Indra can support the application of various distance or correlation measures (Lapesa and Evert, 2014). Currently INDRA supports more than ten different distance and correlation functions, including *Cosine*, *Alphaskew*, *Chebyshev*, *CityBlock*, *Pearson*, *Dice*, *Euclidean*, *Jaccard*, *Jaccard2*, *JensenShannon* and *Spearman Correlation*.

3.5.4 Nearest Neighbours

Given a term and an integer k, the Nearest Neighbours function lists the set of its k closest terms. This method is applied, for instance, in *topic modeling* (Řehůřek and Sojka, 2010) and *vocabulary expansion* (Atzori, Balloccu, and Bellanti, 2018). INDRA implements this function using the SPOTIFY ANNOY library[11], since a preliminary study suggests ANNOY's performance is an order of magnitude better performing than GENSIM's (Řehůřek, 2014). In addition to the identification of term's neighbours, the function also accepts a vector as input. Another related function present in INDRA is the selection based on thresholds, in which INDRA gets a query term and a set of target terms as inputs, and returns those target terms whose relatedness score is greater (or lower) than a given threshold. The threshold can be determined both statically or dynamically (Freitas, Curry, and O'Riain, 2012b).

3.5.5 Vector Compositions

As a primary use, INDRA acts as a central repository of WEMs, serving vectors for terms in different languages and models. The set of pre-processed models allows the user to experiment different WEMs configurations in a one-stop-shop fashion. INDRA can act as a central server in an enterprise context, or as a local

[11] https://github.com/spotify/annoy

library in more constrained environments. In simple terms, *vector composition* aims at generating a vector representation of phrases and sentences from the combination of individual vectors of its compound terms. The popular methods are *Sum (Mitchell and Lapata, 2008), Normalised Sum, Average, Tensor product (Clark and Pulman, 2007), Deep learning models (such as recursive neural networks Socher et al. (2012)* (Kartsaklis, 2014). For example, the vector representation of `modern Democratic Party` is generated by the composition of the corresponding vectors of the three compounding terms `modern`, `Democratic` and `Party`. Currently, INDRA implements three composition methods (*Sum*, *Normalised Sum* and *Average*) and supports the extension of user-defined functions as described in Section 3.5.8. Vector composition is automatically associated to the semantic relatedness function or the retrieval of vectors. Whenever a expression comprehending more than one token is submitted, INDRA composes their corresponding vectors before executing the required function.

3.5.6 Support for Translation-based Models

Some languages do not have large text corpora publicly available. As word embedding models are sensitive to the corpus size, (Freitas et al., 2016; Barzegar et al., 2018b) propose the use of *translation-based models*. In simple words, the translation-based strategy translates the original query terms to a second language for which a high quality WEM is available. On the other hands, Lample et al., 2018 assumed that they have access to high quality word embeddings for each language. and they trained an MT model without access to any translation resources at training time. INDRA gives native support to this operation as described in Section 3.2 and Chapter 4, Section 4.2.5.

3.5.7 Single/Composite Semantic Relation Classification

INDRA supports single and composite Semantic relation classification which is the task of classifying the underlying abstract one or more semantic relations between target entities (terms). The link between the quantified view of distributional semantic relatedness scores (called in the context of this thesis coarse-grained semantics) and the named semantic relations (fine-grained semantics) are achieved with Composite Semantic Relation Classification(CSCW) function. This function is combined of *sequence-based machine learning models*, *distributional semantic models* and *commonsense relational knowledge bases*.

3.5.8 Extensibility

INDRA implements a plugin-based extensible mechanism built on the top of the JAVA SERVICE API which allows including new *compositional methods*, *score functions* and *threshold functions* without recompiling Indra's code. To do so, it

is required to pack the new functions' implementations in a JAR file and place it in the INDRA's *classpath*[12].

3.6 Summary

All contributions which are aimed and described by this thesis, have been implemented and explained at Chapter 4.

[12]For more information about The Java Archive (JAR) and CLASSPATH, please refer to official Java documentation.

Chapter 4

Software and Services

This chapter describes the details of INDRA as a transportable distributional semantic architecture.

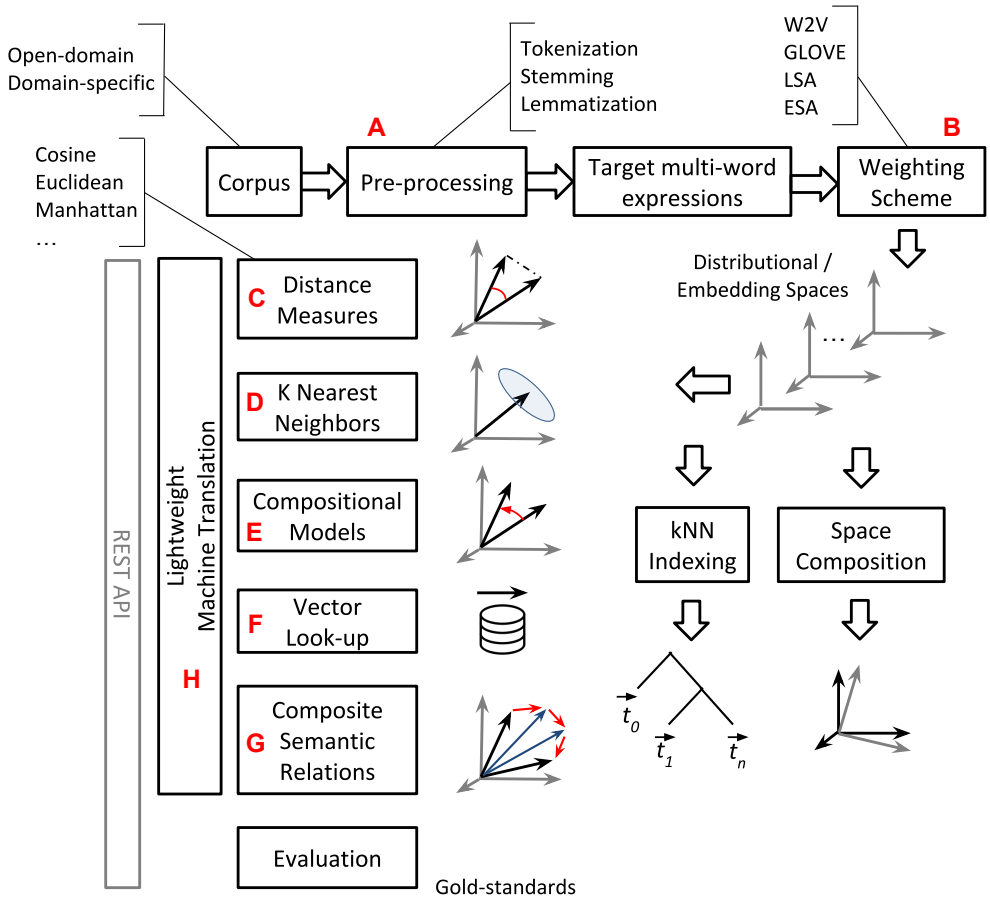

FIGURE 4.1: High-level Transportable Distributional Semantics Architecture

The INDRA PROJECT is divided into two major modules: INDRAINDEXER and INDRA. INDRAINDEXER is responsible for the pre-processing the corpora and generation of the models, whereas INDRA implements the consumption components such as *Vector Look-Up*, *Vector Compositions*,*Nearest Neighbours*, *Semantic Relatedness* and so forth. Figure 4.1 depicts the main components of Indra which are described in the below.

4.1 IndraIndexer

4.1.1 Pre-processing

Pre-Processing (Figure 4.1(A))is one most important component of the transportable distributional semantic architecture. Defining the tokenization strategy, which depends on the language, whether or not words must be lower-cased or stemmed as part of the pre-processing step. Furthermore, the pre-processing strategy should be stated consistently during both construction and consumption phases as exemplified below. Table 4.1 shows the parameters are supported by pre-processing component.

Table 4.1: Parameters supported by the Indra's pre-processing package.

Parameter	Description/Options
input format	Wikipedia-dump format or plain texts from one or multiple files.
language	14 supported languages.
set of stopwords	a set of tokens to be removed.
set of multi-word expressions	set of sequences of tokens that should be considered a unique token.
apply lowercase	lowercase the tokens.
apply stemmer	applies the Poter Stemmer in the tokens.
remove accents	remove the accents of words.
replace numbers	replaces numbers for the place holder $< NUMBER >$.
min	set a minimum acceptable token size.
max	set a maximum acceptable token size.

For using the command-line tool to pre-process a given text file, the following command (Figure 4.2) has been provided.

4.1.2 Model Generation

IndraIndexer (Figure 4.1(B)) is the component responsible for the generation of word embedding models. At Figure 4.2, the command-line tool to generate a model from a *pre-processed* corpora has been provided. There are several parameters that is necessary to set in the *Indexing* component such as, `Weighting Scheme` which the user can select between *W2V*, *GLOVE*, *LSA* and *ESA*, `Dimension`[1], `Windows Size` and min-word-frequency (low frequency cut-off) (See Chapter 2, Section 2.1 for more details of these parameters). The generation time mostly is dependent to the size of both available corpora and sufficient computer memory.

The final generated model stores the set of applied transformations as a meta-data information. When using models, Indra applies the same set of options to

[1] Vector Size

```
$ java -cp indra-preprocessing.jar IndraPreProcessorCommandLine

Usage: IndraPreProcessor [options] [command] [command options]
 Options:
   --help
     You know this..
 Commands:
   pp      Pre process text corpora.
     Usage: pp [options]
       Options:
       * -f, --files
           Input text corpus files or directories.
       * -o, --output
           The output directory.
       * -ft, --file-type
           File type (wiki or text)
           Possible Values: [TEXT, WIKI]
       * -ct, --contentType
           Content type (line or file)
           Possible Values: [LINE, FILE]
       * -n, --name
           Corpus name.
       * -l, --lang
           Corpus language.
         -r, --regex
           Regex to filter files into the directories.
         -d, --desc
           Corpus description.
         -e, --encoding
           File text encoding.
         --stemmer
           Number of times the stemmer must be applied. 0 for none.
           Default: -1
         --remove-accents
           Remove accents before query?
         --lower
           Lowercase words before query?
         --replace-numbers
           Replace numbers for <NUMBER>.
         --min
           Min length of each word
           Default: -1
         --max
           Max length of each word
           Default: -1
         --stop-words
           File containing the set of stop-words to be removed.
         --multi-word-tokens
           File containing the set of multi-words tokens.
```

FIGURE 4.2: Command-line of pre-processing components.

guarantee consistence. For instance, ensuring that a given model was generated by applying stemming and converting tokens to lowercase. Figure 4.4 shows an example of a metadata file.

```
$ java -cp indra-index.jar IndraIndexerCommandLine

Usage: Indra Indexer [options] [command] [command options]
 Options:
   --help
     You know this...
 Commands:
   index      Generate Models.
     Usage: index [options]
       Options:
       * -m, --model
           Input name of the model.
       * -c, --corpus-dir
           A directory in which there are two files. The first is
           'copus.metadata' containing the metadata information and the
           second is 'corpus.txt' contaning the data it self.
           'corpus.metadata' file is generated automatically during the
           preprocess step. In the case that your data was not preprocessed
           by Indra, please generate the metadata file before starting the
           model generation.
       * -d, --dimensions
           The number of dimensions.
           Default: -1
       * -o, --output
           The output directory.
         -w, --windows-size
           Window Size.
           Default: 5
         -f, --min-word-frequency
           Min word frequency.
           Default: 5
```

FIGURE 4.3: Command-line of model generation components.

INDRA loads the generated models into two types of data sources: ANNOY indexes[2] (for dense vectors models), Lucene indexes[3] (for sparse vectors models). The ANNOY[4] is a library to search for points in space that are close to a given query point. In addition to the identification of term's neighbours, the function also accepts a vector as input. ANNOY library has different features such as *using static files as indexes* it means index creation is separate from look-up, and it leads to sharing index across processes. ANNOY library works well for vector space models which have low dimensions (like <100). Therefore, for sparse vectors models, such as ESA does not work well. For this reason in INDRA (Sales et al., 2018), LUCENE has been used for indexing and searching capability.

[2] https://github.com/spotify/annoy
[3] https://lucene.apache.org
[4] (Approximate Nearest Neighbours Oh Yeah)

```
{
   "windowSize" : 5,
   "minWordFrequency" : 5,
   "corpusMetadata" : {
      "corpusName" : "wiki-2014",
      "stopWords" : ["been", "don't", ...],
      "replaceNumbers" : false,
      "applyStemmer" : 3,
      "removeAccents" : true,
      "maxTokenLength" : 100,
      "minTokenLength" : 3,
      "description" : null,
      "language" : "en",
      "encoding" : "UTF-8",
      "applyLowercase" : true
   },
   "vocabSize" : 1181258,
   "sparse"  :false,
   "model" : "w2v",
   "dimensions" : 300
}
```


FIGURE 4.4: Example of metadata file generated by INDRAINDEXER which describes how the user configured both the preprocessor and the Word embedding model generator.

4.2 Indra

INDRA's service exposes the functions as POST[5] methods, whose data are passed as a JSON[6] payload. For simplicity, we suppress the request headers to concentrate our attention in the payload itself. Every request has at least three mandatory fields: i) *language*, ii)*model* and iii)*corpus*. The first naturally corresponds to the language of the request to API. The second field corresponds to the model algorithm which third specifies the corpus from which the word embedding model was generated. These three fields correspond to the model's unique identifier.

4.2.1 Vector Look-up

Figure 4.5 shows a payload to the endpoint `/vectors` (Figure 4.1(F)) which returns the respective word embedding vectors of the `terms`. In case term is composed of more than one token, `termComposition` (See more details on 4.2.2) is applied, With `termComposition` component we can find a vector for a phrase/sentence.Figure 4.6 also shows a public endpoint of requesting the word embedding of the terms *love*, *mother* and the expression *santa claus*.

[5]https://en.wikipedia.org/wiki/POST_(HTTP)
[6]JavaScript Object Notation

```
{
  "corpus": "googlenews300neg",
  "model": "W2V",
  "language": "EN",
  "terms": ["love", "best of you"],
  "termComposition" : "AVERAGE"
}
```

FIGURE 4.5: Payload to request the word embedding of the term *love* and the expression *best of you*.

```
curl -X POST -H "Content-Type: application/json" -d '{
  "corpus": "wiki-2018",
  "model": "W2V",
  "language": "EN",
  "terms": ["love", "mother", "santa claus"],
  "termComposition" : "SUM"
}' "http://indra.lambda3.org/vectors"
```

FIGURE 4.6: Public Endpoint for requesting the word embedding of the terms *love*, *mother* and the expression *santa claus*.

```
{
    "corpus":"wiki-2018",
    "model":"W2V",
    "language":"EN",
    "terms":{
        "love":[2.2182834,-1.7239775, ..., -0.9383560],
        "mother":[0.3326052,-0.5156235, ..., -0.8851330],
        "santa claus":[0.4912644,0.8073116, ..., 0.2683054]
    }
}
```

FIGURE 4.7: Result of requesting the word embedding of the terms *love*, *mother* and the expression *santa claus*.

4.2.2 Vector Compositions

The VECTOR COMPOSITION component (Figure 4.1(E)) aims at generating a vector representation of phrases and sentences from the combination of individual vectors of its compound terms. For example, the vector representation of `santa claus` (Figure 4.6) is generated by the composition of the corresponding vectors of the two compounding terms `santa` and `claus`. Currently, INDRA implements three composition methods (*Sum*, *Normalised Sum* and *Average*). The VECTOR COMPOSITION function is automatically associated to the semantic relatedness

function (Section 4.2.3) or the retrieval of vectors (Section 4.2.1). Whenever an expression containing more than one token is submitted, INDRA composes their corresponding vectors before executing the required function.

4.2.3 Semantic Relatedness

INDRA implements two semantic relatedness methods (Figure 4.1(C)). The **first** is the pair-wise semantic relatedness in which the user provides pairs of terms to calculate their semantic relatedness. The **second** option is integrate the nearest neighbours function (Section 4.2.4), which returns the relatedness of the k closest terms.

4.2.3.1 Pair Relatedness:

The endpoint `/relatedness` returns the semantic relatedness of the `pairs`. The relatedness operation is defined by the field `scoreFunction` as shown in Figure 4.8. Currently INDRA supports more than ten different distance and correlation functions, including *Cosine*, *Alphaskew*, *Chebyshev*, *CityBlock*, *Pearson*, *Dice*, *Euclidean*, *Jaccard*, *Jaccard2*, *JensenShannon* and *Spearman Correlation* (More details at Chapter 3, Table 2.3) In the case that `termComposition` is not defined, the default function which is *SUM* is used.

```
{
    "corpus": "wiki-2018",
    "model": "ESA",
    "language": "PT",
    "scoreFunction": "COSINE",
    "pairs": [{
        "t1": "economia",
        "t2": "Rio de Janeiro"
        },
        {
        "t1": "economia",
        "t2": "soja"
    }]
}
```

FIGURE 4.8: Payload to request the cosine relatedness of two pairs of terms in Portuguese.

Figure 4.9 shows the public endpoint of requesting the cosine relatedness of two pairs of terms in English from *W2V* model, generated from *Wikipedia 2018*. Figure 4.10 shows the output of the requested command at Figure 4.9.

```
curl -X POST -H "Content-Type: application/json" -d '{
  "corpus": "wiki-2018",
  "model": "W2V",
  "language": "EN",
  "scoreFunction": "COSINE",
  "pairs": [{
    "t2": "love",
    "t1": "mother"
  },
  {
    "t2": "love",
    "t1": "santa claus"
  }]
}' "http://indra.lambda3.org/relatedness"
```

FIGURE 4.9: Public Endpoint of requesting the cosine relatedness of two pairs of terms in English.


```
{
    "corpus":"wiki-2018",
    "model":"W2V",
    "language":"EN",
    "pairs":[{
        "t1":"santa claus",
        "t2":"love",
        "score":0.07129745797653386
    },
    {
        "t1":"mother",
        "t2":"love",
        "score":0.31881560236682005}
    ],
    "scoreFunction":"COSINE"
}
```


FIGURE 4.10: Result of requesting the cosine relatedness of two pairs of terms in English.

4.2.3.2 One-to-many Relatedness:

The endpoint `/relatedness/otm` returns the semantic relatedness of `one` term against a set of `many` terms. Similarly to the previous operation, the relatedness operation is defined by the field `scoreFunction` as shown in Figure 4.11.

```
{
    "corpus": "wiki-2018",
    "model": "ESA",
    "language": "EN",
    "scoreFunction": "JACCARD",
    "one": "Germany",
    "many" : ["France", "China", "Brazil"]
}
```

FIGURE 4.11: Payload to request the Jaccard relatedness of the implicit pairs [Germany, France], [Germany, China] and [Germany, Brazil].

```
curl -X POST -H "Content-Type: application/json" -d '{
  "corpus": "wiki-2018",
    "model": "ESA",
    "language": "EN",
    "scoreFunction": "JACCARD",
    "one": "Germany",
    "many" : ["France", "China", "Brazil"]
}' "http://indra.lambda3.org/relatedness/otm"
```

FIGURE 4.12: Public Endpoint of requesting the Jaccard relatedness of the implicit pairs [Germany, France], [Germany, China] and [Germany, Brazil]

Figure 4.12 shows the public endpoint of requesting the Jaccard relatedness of the implicit pairs [Germany, France], [Germany, China] and [Germany, Brazil] in English from *ESA* model which generated based on *Wikipedia 2018*. Figure 4.13 shows the output of the requested commend at Figure 4.12.

```
{
    "corpus":"wiki-2018",
    "model":"ESA",
    "language":"EN",
    "one":"Germany",
    "many":{
        "France":0.00985682774033495,
        "China":0.002684318455401879,
        "Brazil":0.0021421896656059897
    },
    "scoreFunction":"JACCARD"
}
```


FIGURE 4.13: Result of requesting the Jaccard relatedness of the implicit pairs [Germany, France], [Germany, China] and [Germany, Brazil].

4.2.4 K-Nearest Neighbours

The Nearest Neighbours component (Figure 4.1(D)) is exposed in two methods. The endpoint `/neighbors/relatedness` (Section 4.2.3) returns the semantic relatedness score between the target terms and their top-k neighbours (Figure 4.17), according to the payload depicted in Figure 4.14.


```
{
  "corpus": "wiki-2018",
  "model": "GLOVE",
  "language": "SV",
  "topk": 10,
  "scoreFunction": "COSINE",
  "terms": ["ekonomi", "flicka", "frihet"]
}
```


FIGURE 4.14: Payload to request the 10 most related terms individually to *ekonomi*, *flicka* and *frihet*. This call returns three set of 10 terms, each one corresponding to one of the terms.

```
$curl -X POST -H "Content-Type: application/json" -d '{
  "corpus": "googlenews",
  "model": "W2V",
  "language": "EN",
  "topk": 9,
  "scoreFunction": "COSINE",
  "terms": ["love", "mother"]
}' "http://indra.lambda3.org/neighbors/relatedness"
```

```
${
    "corpus":"googlenews",
    "model":"W2V",
    "language":"EN",
    "topk":10,
    "terms":{
        "love":{
            "loved":0.6907791930663534,
            "loves":0.6618633482720202,
            "loving":0.5886635736527813,
            "absolutely_adore":0.5536840696305002,
            "cherish":0.5405279722821222,
            "romance":0.5208202320566526,
            "undying_passion":0.512841160945286,
            "smitten":0.5122916703806815,
            "friendship":0.510301842100534},
        "mother":{
            "daughter":0.8706234116051772,
            "father":0.7901483043348325,
            "son":0.7683206236942329,
            "sister":0.7633353128000974,
            "wife":0.7550681994420887,
            "stepmother":0.7531879825136341,
            "maternal_grandmother":0.7467763015862634,
            "daughters":0.7415745621301421,
            "husband":0.741405003066323}
        }
 }
```

FIGURE 4.15: Public endpoint for requesting the 10 most related terms individually to *love*, *mother* and its result which is the relatedness score between the target terms and their top-k neighbours.

When submitting the same payload to the endpoint `/neighbors/vectors` (Section 4.2.1), the service returns the list of the neighbours and their respective vectors (Figure 4.17).

```
$curl -X POST -H "Content-Type: application/json" -d '{
  "corpus": "googlenews",
  "model": "W2V",
  "language": "EN",
  "topk": 9,
  "terms": ["love", "mother"]
}' "http://indra.lambda3.org/neighbors/vectors"

${
    "corpus":"googlenews",
    "model":"W2V",
    "language":"EN",
    "topk":10,
    "terms":{
        "love":{
            "love":[0.10302734,-0.1523437,..., 0.0649414],
            "loving":[0.1396484,0.1142578,..., -0.0483398],
            "romance":[0.2373046,-0.1875,... ,0.3476562],
            "smitten":[0.1337890,0.1923828,..., 0.2255859],
            "absolutely_adore":[-0.0334472,-0.0834960,...,0.1127929],
            "friendship":[0.1835937,-0.0805664,...0.1015625],
            "loved":[0.0405273,-0.09619140,...-0.1186523],
            "undying_passion":[0.3261718,-0.046875,..., 0.1513671],
            "loves":[0.2197265,-0.0537109,... ,0.0045776],
            "cherish":[0.1191406,0.1269531,... -0.0037078]},
        "mother":{
            "daughters":[-0.0183105,-0.0024414,...,0.1030273],
            "son":[0.1079101,-0.0300292,... ,0.1757812],
            "maternal_grandmother":[0.0023956,-0.1835937,... ,-0.1611328],
            "wife":[0.0393066,-0.1962890,... , -0.0839843],
            "father":[0.0473632,-0.0317382,... ,0.0883789],
            "husband":[-0.03735351,-0.2636718, ..., 0.0795898],
            "stepmother":[0.0169677,-0.3105468,..., 0.0932617],
            "daughter":[0.0296630,-0.1279296, ..., 0.0786132],
            "sister":[-0.2363281,-0.07177734,... , 0.0534667]}}}
```

FIGURE 4.16: Public endpoint and its result for requesting the 10 most related terms individually to *love*, *mother* and *frihet*. The result which is a list of their neighbours and their respective vectors.

4.2.5 Translation-based Word Embeddings

These requests support the translated-based function (Figure 4.1(H)), in which the vectors is extracted from the corresponding English corpus after translating the terms from the original query. The translation-based function is activated

by appending "mt"=true in the payload. INDRA offers seven pre-computed light-weight translation models.

```
$curl -X POST -H "Content-Type: application/json" -d '{
  "corpus": "wiki-2018",
  "model": "W2V",
  "language": "PT",
  "scoreFunction": "COSINE",
  "mt" : true,
  "pairs": [{
    "t2": "amor",
    "t1": "mãe"
  }]
}' "http://indra.lambda3.org/relatedness"

${
    "corpus":"wiki-2018",
    "model":"W2V",
    "language":"PT",
    "pairs":[{
        "t1":"amor",
        "t2":"mãe",
        "score":0.4818156543722112
    }],
    "scoreFunction":"COSINE"
}
```

FIGURE 4.17: Public endpoint for requesting the 10 most related terms individually to *love*, *mother* and its result which is the relatedness score between the target terms and their top-k neighbours.

For a complete description of the methods and parameters, please refer to the project documentation[7].

4.3 Python Client

Our project also offers a client to access the service from Python application. The `pyindra` package is available in the `pip` repository. The client source code is at https://github.com/Lambda-3/pyindra.

[7] https://github.com/Lambda-3/Indra/wiki

Chapter 5

Evaluation of Language and Domain Transportability

5.1 Introduction

This chapter evaluates the transportability of the distributional semantic architecture across different languages (Section 5.2) and domains (Section 5.3).

5.2 Multilingual Analysis

5.2.1 Introduction

This chapter investigates how different distributional semantic models built from corpora in different languages and with different sizes perform in computing semantic similarity and relatedness tasks. Additionally, we analyse the role of heavyweight Google that is based on GNMT (RNN Models) (Wu et al., 2016) and Bing[1] machine translation approaches to support the construction of better-performing distributional vectors in the context of computing semantic similarity and relatedness measures for other languages. Also, this chapter aims at providing an analysis of the impact of a lightweight machine translation over an English DSM. It also reports on the underlying MT quality necessary to deliver word vector models with quality comparable to English.

This chapter addresses the following research questions:

- **RQ 1** How do different distributional semantic models built from corpora in different languages of varied corpus size perform in computing both semantic similarity and relatedness tasks?
- **RQ 2** Does machine translation of non English input to English perform better than the word vectors in the original language (for which languages and for which distributional semantic models)?

[1] http://bing.com/translator

- **RQ 3** Which DSMs and languages benefit more and less from the translation?
- **RQ 4** What is the quality of state-of-the-art machine translation approaches for automatically translating word pairs (and per language pair)?
- **RQ 5** Can a lightweight MT model over an English DSM provide higher quality word vectors compared to native word vectors?
- **RQ 6** How does a lightweight MT model compares with state-of-the-art MT models?
- **RQ 7** Are there DSMs which are more/less robust with respect to the quality of the MT?

5.2.2 Evaluation Methodology

The evaluation methodology focuses on making explicit the steps necessary to evaluate the research hypotheses. It consists of the following steps:

1. **Creation of multi-lingual test collections.**
2. **Development of a lightweight machine translation system.**
3. **Generating language-specific DSMs.**
4. **Experimental set-up and evaluation.**
5. **Evaluation of the full machine translations as baselines for the multi-lingual aspect.**

5.2.3 Creating a Multi-Lingual Gold Standard for Evaluation

Currently, the majority of the existing gold-standards for evaluating semantic similarity and relatedness have tended to focus on the English language, with some initiatives providing initial gold-standards for few other languages (Faruqui and Dyer, 2014a). Camacho-Collados et al. (2017) developed a multi-lingual gold-standard which includes 518 word pairs for five languages (English, German, Italian, Spanish and Persian). It is composed of nominal pairs of multi-word expressions, domain-specific terms and named entities that are manually scored between 0 to 4 where 0 indicates that they are completely dissimilar and 4 denotes that the two words are synonymous. The dataset developed by (Camacho-Collados et al., 2017) focuses on semantic similarity. Bruni, Tran, and Baroni (2014) introduced a test collection containing 3000 word pairs.

The MEN dataset was obtained by crowdsourcing using Amazon Mechanical Turk [2] via the CrowdFlower[3] interface. The dataset focuses on semantic relatedness pairs on the English language (similarly to the WS-353 dataset (Finkelstein et al., 2001)). It was specifically developed to test multimodal Distributional Semantic Models [4]. Compared to *WS-353*, `MEN` is sufficiently large enough and the human judgements are relative rather than absolute[5]. At Bruni, Tran, and Baroni (2014), each rater chose the word pair that was more similar out of two random pairs of words. They used this technique to have a comparative judgement rather than absolute scores for single pairs, which was used in the *WS-353*.

In order to conduct a comprehensive evaluation over several languages, we need a multi-lingual gold standard, to fill this gap we created *SemR-11*, which is a multilingual dataset for evaluating semantic similarity and relatedness for 11 languages (*German*, *French*, *Russian*, *Italian*, *Dutch*, *Chinese*, *Portuguese*, *Swedish*, *Spanish*, *Arabic* and *Persian*). Semantic similarity and relatedness gold standards have been initially used to support the evaluation of semantic distance measures in the context of linguistic and knowledge resources and distributional semantic models. SemR-11 builds upon the English gold-standards of Miller & Charles - MC (Miller and Charles, 1991), Rubenstein & Goodenough - RG (Rubenstein and Goodenough, 1965), WordSimilarity 353 - WS-353 (Finkelstein et al., 2001), and SIMLEX (Leviant and Reichart, 2015), providing a canonical translation for them. The final dataset consists of 15,917 word pairs and can be used to support the construction and evaluation of semantic similarity/relatedness and distributional semantic models. Table 5.1 quantifies the vocabulary and token distribution for each language. Since some of words are multi-word expressions they are counted once as *vocabulary* i.e multi tokens. For example *maison de fous* is one count of *vocabulary* but it contains *three tokens*. The *SemR-11* test collection was used to evaluate how different distributional semantic models built from corpora in multiple languages and of varied corpus size perform in computing semantic relatedness similarity and relatedness tasks.

The process of creating *SemR-11* (Table 5.2) consisted in the translation of the three gold-standards *WS-353*, *MC*, *RG* for *eleven* languages (*German*, *French*, *Russian*, *Italian*, *Dutch*, *Chinese*, *Portuguese*, *Swedish*, *Spanish*, *Arabic* and *Persian*) and of the *SIMLEX* for *seven* European languages (*German*, *French*, *Italian*, *Dutch*, *Portuguese*, *Swedish* and *Spanish*). Also *SemR-11* has been compared with a most recent multi-lingual gold standard that only covers four languages

[2] https://www.mturk.com/mturk/welcome
[3] http://crowdflower.com/
[4] Based on textual and visual information.
[5] The authors rather than ask annotators to give an absolute score to show how much a word pair is semantically related, they requested them to make comparative judgements on two pair examples at a time.

TABLE 5.1: The vocabulary and token distribution for each language of four gold-standards

Language	Parameters	MC	RG	WS-353	SIMLEX
German	# of Tokens	40	52	431	1094
	Vocabulary Size	40	52	431	1094
French	# of Tokens	37	45	430	1106
	Vocabulary Size	37	43	424	1097
Russian	# of Tokens	38	48	435	N/A
	Vocabulary Size	36	46	426	N/A
Italian	# of Tokens	34	43	426	1051
	Vocabulary Size	34	43	424	1051
Dutch	# of Tokens	37	45	426	1025
	Vocabulary Size	37	45	426	1018
Chinese	# of Tokens	37	51	471	N/A
	Vocabulary Size	37	51	471	N/A
Portuguese	# of Tokens	37	46	434	1149
	Vocabulary Size	37	46	434	1141
Swedish	# of Tokens	35	44	430	1002
	Vocabulary Size	35	44	430	995
Spanish	# of Tokens	35	44	437	993
	Vocabulary Size	35	44	437	991
Arabic	# of Tokens	38	54	448	N/A
	Vocabulary Size	36	49	448	N/A
Persian	# of Tokens	34	43	456	N/A
	Vocabulary Size	34	43	436	N/A

(Camacho-Collados et al., 2017). The word pairs were translated by paid professional translators[6], skilled in data localisation tasks. The datasets are available on the Web[7].

All translated pairs followed the protocol below:

1. Given a pair of words, translators should assume the most similar senses associated with the pair.

2. Translators should preserve the lexical category of the sense identified for that word.

[6] Lionbridge Natural Language Solutions

[7] https://github.com/Lambda-3/Gold-Standards/tree/master/SemR-11

TABLE 5.2: SemR-11 and its relation to existing multi-lingual gold standards.

Language	SemR-11				SE17- T2
	MC	RG	WS 353	SIMLEX	
German	✓	✓	✓	✓	✓
French	✓	✓	✓	✓	
Russian	✓	✓	✓		
Italian	✓	✓	✓	✓	✓
Dutch	✓	✓	✓	✓	
Chinese	✓	✓	✓		
Portuguese	✓	✓	✓	✓	
Swedish	✓	✓	✓	✓	
Spanish	✓	✓	✓	✓	✓
Arabic	✓	✓	✓		
Persian	✓	✓	✓		✓

The SEMR-11 gold-standard assumes that the translations are preserving the similarity and relatedness scores of their original English human annotation. The target task was described to the human translators, who had access to the word pairs and scores. Tables 5.3 and 5.4 show examples of translated pairs of SIMLEX9 test collection (with the associated average similarity score) into Portuguese and French languages, respectively, while Table 5.5 provides example of word-pairs for each language and dataset.

TABLE 5.3: Comparison between English and Portuguese gold standards in SEMR-11.

English	Portuguese
hard;difficult;9.69	difícil;difícil;10
apparent;obvious; 9.08	visível;óbvio;9.15
disease;infection;7.08	doença;infecção;4.46

TABLE 5.4: Comparison between English and French gold standards in *SemR-11*

English	French
woman;wife;283	Femme;femme;10
girl;child;4.77	Fille;enfant;5
understand;know;5.69	Comprendre;connaître;6.92

TABLE 5.5: Examples with all the languages for each of four datasets

Word-Pairs	MC	RG	WS-353	SIMLEX
English	food;rooster	monk;oracle	closet;clothes	clothes;closet
German	nahrung;hahn	mönch;orakel	Wandschrank;Kleidung	Kleider;Schrank
French	nourriture;coq	moine;oracle	cabinet;vêtements	vêtements;placard
Russian	еда;петух	монах;оракул	стенной шкаф;одежда	N/A
Italian	cibo;gallo	monaco;oracolo	ripostiglio;vestiti	vestiti;armadio
Dutch	voedsel;haan	monnik;orakel	kast;kleren	kleding;kast
Chinese	食物;公鸡	僧侣;甲骨文	壁橱;衣服	N/A
Portuguese	comida;galo	monge;oráculo	armário;roupas	roupas;roupeiro
Swedish	mat;tupp	munk;orakel	garderob;kläder	kläder;förråd
Spanish	comida;gallo	monje;oráculo	armario;ropa	ropa;armario
Arabic	طعام;ديك	راهب;وحي	خزانة;ملابس	N/A
Persian	غذا;خروس	راهب;وحی	گنجه;لباس	N/A

5.2.4 Experimental Setup

The experimental setup consists of the instantiation of four distributional semantic models (Explicit Semantic Analysis - *ESA* (Gabrilovich and Markovitch, 2007a), Latent Semantic Analysis - *LSA* (Landauer, Foltz, and Laham, 1998), Word2Vector - *W2V* (Mikolov et al., 2013a) and Global Vectors - *GloVe* (Pennington, Socher, and Manning, 2014a)) in 11 different languages - *English*, *German*, *French*, *Italian*, *Spanish*, *Portuguese*, *Dutch*, *Russian*, *Swedish*, *Arabic* and *Farsi*.

DSMs were generated from *Wikipedia* dumps (January 2015), which were pre-processed by converting word to lower case as well as stemming and removing stopwords. For *LSA* and *ESA*, the models were generated using the *SSpace Package* (Jurgens and Stevens, 2010b), while *W2V* and *GloVe* were generated using the open source code developed by each of the respective model's authors. For the experiment, the vector dimensions for *LSA*, *W2V* and *GloVe* were set to 300 while *ESA* was defined with 1500 dimensions. The difference in size occurs because *ESA* is composed of sparse vectors. All models used default parameters defined their respective implementation in the generation process. .

Each distributional model was evaluated for the task of computing semantic similarity and relatedness measures using four human-annotated gold standard datasets: *Miller & Charles - MC* (Miller and Charles, 1991), *Rubenstein & Goodenough - RG* (Rubenstein and Goodenough, 1965), *WordSimilarity 353 - WS353* (Finkelstein et al., 2001) and SIMLEX (Hill, Reichart, and Korhonen, 2016). Except for a few languages in previous works (Faruqui and Dyer, 2014a; Camacho-Collados, Pilehvar, and Navigli, 2015; Hill, Reichart, and Korhonen, 2016), all the four mentioned word-pair gold-standards were originally in English. These four gold-standards were translated and reviewed with the help of paid professional translators.

In the word-pair translation task, in case of word sense ambiguity, the translators were instructed to select the senses which are most related to the other word. In order to support reproducibility and comparability, the datasets (Barzegar et al., 2018c) are available on the web[8].

As baselines for the machine translation approach (Figure 3.1 *(i)*), we used the Google Translate Service and the Microsoft Bing Translation Service to compare with our lightweight machine translation (As explained on Chapter 3, Section 3.2.2). The lightweight MT was generated using three combined parallel corpora: Europarl, DGT and OpenSubtitle2016 (Tiedemann, 2012). Table 5.6 shows details of the parallel corpora size.

[8]https://github.com/Lambda-3/Gold-Standards/tree/master/SemR-11

TABLE 5.6: Details of Parallel Corpora Size (scale of 10^6).

Parallel Corpora	Parameters	Europarl	DGT	OpenSubtitle2016	All
Source = German Target = English	Sentence Alignments	2	3.2	13.9	19.1
	Source Tokens	45.4	48.4	84.7	178.5
	Target Tokens	53.1	53.1	88.3	194.5
Source = French Target = English	Sentence Alignments	2	3	33.8	38.8
	Source Tokens	53.6	57.7	214.6	325.9
	Target Tokens	51.3	52.8	221.7	325.8
Source = Spanish Target = English	Sentence Alignments	2	3.2	49.9	55.1
	Source Tokens	52.7	60.4	297.4	410.5
	Target Tokens	50.2	52.9	320	423.1
Source = Portuguese Target = English	Sentence Alignments	2	3.2	24.9	30.1
	Source Tokens	51	56.5	147.7	255.2
	Target Tokens	50.3	52.6	160	262.9
Source = Italian Target = English	Sentence Alignments	1.9	3.2	26.3	31.4
	Source Tokens	49	54.6	161.1	264.7
	Target Tokens	50.7	53	172.2	275.9
Source = Swedish Target = English	Sentence Alignments	1.9	3.2	11.9	17
	Source Tokens	42.2	47.1	69.4	158.7
	Target Tokens	46.7	53	81.2	180.9
Source = Dutch Target = English	Sentence Alignments	2	3.2	28.8	34
	Source Tokens	51.2	53.4	182.8	287.4
	Target Tokens	50.6	52.8	197.4	300.8

The lightweight MT over DSMs was implemented over the Indra service [9] (Barzegar et al., 2015; Sales et al., 2018).

5.2.5 Spearman Correlation and Corpus Size

Table 5.8 shows the correlation between the average Spearman correlation values for each DSM and two indicators of corpus size: # of tokens and # of unique tokens. *W2V* is consistently more robust (on average 61%) than the other models in relation to the corpus size due the fact that *W2V* caters a large context windows (10) in comparison to the other distributional models (except of *ESA* that its density is 0.59), which captures syntagmatic relations. These captured relations are appropriate for computing semantic relatedness measure in the multi-lingual scenario. While *ESA* considers the whole document as its context window, the other models are restricted to five (*LSA*) and ten (*Word2Vec* and *GloVe*) words.

[9] https://github.com/Lambda-3/Indra

Another observation is that the evaluation of the *WS-353* dataset is more dependent on the corpus size, which can be explained by the broader number of semantic relations expressed under the semantic relatedness umbrella.

Table 5.7 shows the size of each corpus in different languages regarding the number of unique tokens and the number of tokens.

TABLE 5.7: The sizes of the corpora in terms of the number of unique tokens(types) and tokens (scale of 10^6).

lang	unique tokens	tokens
en (English)	4.238	902.044
de (German)	4.233	312.380
fr (France)	1.749	247.492
ru (Russian)	1.766	202.163
it (Italian)	1.411	178.378
nl (Dutch)	2.021	105.224
pt (Portuguese)	0.873	96.712
sv (Swedish)	1.730	82.376
es (Spanish)	0.829	76.587
ar (Arabic)	1.653	46.481
fa (Farsi)	0.925	32.557

TABLE 5.8: Correlation between corpus size and different models.

ρ Density	MC		RG		WS353	
	/unique tokens	/tokens	/unique tokens	/tokens	/unique tokens	/tokens
ESA	0.40	0.34	0.51	0.65	0.17	0.41
LSA	0.51	0.62	0.52	0.67	0.51	0.69
W2V	0.42	0.56	0.54	0.65	0.64	0.84
Glove	0.29	0.28	0.36	0.45	0.50	0.69

5.2.6 Language-Specific DSMs

In the first part of the experiment, the Spearman correlations (ρ) between the human assessments and the computation of the semantic similarity and relatedness for all DSMs instantiated for all languages were evaluated (Figure 3.1 *(ii)*). Table 5.9 shows the Spearman correlation for each DSM using language-specific corpora (without machine translation), for the *four* word-pair datasets. The comparative language-specific analysis indicates that for *English* is the best-performing language (0.70), followed by *German* (0.62). The lowest Spearman correlation was observed in *Arabic* [10] (0.35). From the tested DSMs, *W2V* is consistently the best-performing DSM (0.52 *With SIMLEX*, 0.61 *Without SIMLEX*). The

[10] Based on three datasets (MC, RG, WS353)

language-specific DSMs achieved higher correlations for *RG* and *MC* (0.61 and 0.60, respectively), in comparison to 0.44 for *WS-353* and 0.21 for *SIMLEX*.

TABLE 5.9: Spearman correlation for the *language-specific* models. (M. AVG represents the average of the models and DS. AVG represents the average of the datasets)

DS	DSMs	en	de	fr	it	nl	pt	sv	es	ru	ar	fa	Model AVG.	DS. AVG
MC	ESA	0.69	0.67	0.54	0.54	0.58	0.78	0.54	0.65	0.66	0.37	0.56	0.60	0.60
	LSA	0.79	0.70	0.55	0.69	0.60	0.50	0.69	0.72	0.63	0.46	0.45	0.62	
	W2V	0.89	0.69	0.54	0.65	0.62	0.46	0.57	0.78	0.64	0.38	0.68	0.62	
	GLOVE	0.69	0.64	0.64	0.38	0.38	0.69	0.56	0.71	0.76	0.38	0.45	0.57	
RG	ESA	0.80	0.68	0.58	0.63	0.64	0.63	0.63	0.69	0.63	0.36	0.57	0.62	0.61
	LSA	0.72	0.65	0.47	0.61	0.58	0.44	0.65	0.51	0.51	0.35	0.46	0.54	
	W2V	0.85	0.78	0.66	0.71	0.67	0.54	0.69	0.73	0.64	0.36	0.55	0.65	
	GLOVE	0.74	0.69	0.62	0.57	0.55	0.62	0.61	0.71	0.70	0.32	0.59	0.61	
WS353	ESA	0.50	0.42	0.36	0.40	0.55	0.44	0.48	0.38	0.44	0.26	0.37	0.42	0.44
	LSA	0.54	0.48	0.40	0.38	0.49	0.39	0.44	0.37	0.40	0.28	0.43	0.42	
	W2V	0.69	0.57	0.54	0.53	0.60	0.54	0.49	0.54	0.53	0.44	0.53	0.54	
	GLOVE	0.49	0.44	0.38	0.40	0.43	0.38	0.38	0.33	0.42	0.26	0.36	0.39	
	Lang AVG. (above DS)	0.70	0.62	0.52	0.54	0.56	0.53	0.56	0.59	0.58	0.35	0.50	0.55	
SIMLEX	ESA	0.17	0.16	0.19	0.15	0.15	0.19	0.19	0.22	N/A	N/A	N/A	0.18	0.21
	LSA	0.18	0.16	0.22	0.14	0.19	0.20	0.19	0.24	N/A	N/A	N/A	0.19	
	W2V	0.25	0.23	0.27	0.24	0.23	0.24	0.25	0.30	N/A	N/A	N/A	0.25	
	GLOVE	0.27	0.19	0.24	0.19	0.17	0.22	0.20	0.28	N/A	N/A	N/A	0.21	
	Lang AVG. (all DS)	0.58	0.51	0.45	0.45	0.46	0.45	0.47	0.51					

5.2.7 Google and Bing MT vs. Language-Specific DSMs

In the second step of the evaluation, the results[11] for the language-specific DSMs were contrasted to the machine translation (MT) approach, according to the diagram depicted in Figure 3.1 *(i)*. The Spearman correlation for the MT-mediated approach by Bing and Google MT are shown in Tables 5.10 and 5.12, respectively. Using the *Bing* MT model, *W2V* is consistently the best performing DSM (average 0.50 *with SIMLEX*, 0.59 *without SIMLEX*), while *ESA* and *LSA* are consistently the worst performing models (average 0.42). We can interpret this result by stating that the benefit of using machine translation for *ESA* does not introduce significant performance improvements in comparison to the language-specific baselines. The best performing language is *French* ($\rho = 0.64$). The Spearman correlation variance across languages in the *Bing* MT models is low, as the impact of the use of the *English* corpus on the DSM model has a higher positive impact on the results in comparison to the variation of the quality of the *Bing* machine translation. The results for all languages achieve very similar correlation values.

[11] Because of not existing *SIMLEX* dataset for all eleven languages, the mentioned scores without any extra information — With/Without *SIMLEX* dataset — are counted with *SIMLEX* dataset

TABLE 5.10: Spearman correlation for the *Bing* machine translation models over the English corpus (M. AVG represents the average of the models and DS. AVG represents the average of the datasets).

DS	DSMs	de	fr	it	nl	pt	sv	es	ru	ar	fa	Model AVG.	DS. AVG
MC	ESA	0.46	0.67	0.38	0.72	0.48	0.70	0.64	0.60	0.48	0.14	0.53	0.58
	LSA	0.59	0.69	0.33	0.75	0.64	0.72	0.64	0.76	0.48	0.21	0.58	
	W2V	0.55	0.76	0.46	0.68	0.70	0.79	0.75	0.79	0.69	0.20	0.64	
	GLOVE	0.48	0.74	0.45	0.67	0.51	0.71	0.71	0.74	0.52	0.17	0.57	
RG	ESA	0.54	0.74	0.53	0.72	0.58	0.62	0.69	0.49	0.36	0.24	0.55	0.56
	LSA	0.56	0.59	0.48	0.66	0.59	0.63	0.68	0.43	0.19	0.26	0.51	
	W2V	0.62	0.79	0.56	0.69	0.71	0.70	0.79	0.52	0.26	0.27	0.59	
	GLOVE	0.56	0.77	0.58	0.69	0.60	0.64	0.74	0.56	0.35	0.26	0.57	
WS353	ESA	0.37	0.44	0.37	0.44	0.37	0.42	0.41	0.39	0.25	0.31	0.38	0.42
	LSA	0.41	0.47	0.40	0.49	0.39	0.46	0.43	0.44	0.28	0.30	0.41	
	W2V	0.57	0.63	0.55	0.62	0.57	0.60	0.55	0.56	0.42	0.34	0.54	
	GLOVE	0.35	0.41	0.36	0.43	0.36	0.41	0.42	0.41	0.22	0.25	0.36	
	Lang AVG. (above DS)	0.51	0.64	0.45	0.63	0.54	0.62	0.62	0.56	0.37	0.25	0.52	
SIMLEX	ESA	0.18	0.24	0.17	0.21	0.22	0.24	0.21	N/A	N/A	N/A	0.21	0.22
	LSA	0.16	0.23	0.19	0.18	0.21	0.24	0.20	N/A	N/A	N/A	0.20	
	W2V	0.17	0.25	0.22	0.23	0.24	0.28	0.23	N/A	N/A	N/A	0.23	
	GLOVE	0.21	0.28	0.21	0.24	0.25	0.28	0.25	N/A	N/A	N/A	0.25	
	Lang AVG. (all DS)	0.42	0.54	0.39	0.53	0.46	0.53	0.52					

The impact of the *Bing* MT model can be better interpreted by examining the difference between the *Bing* machine translation and the *domain-specific* models (depicted in Table 5.11). Including the *SIMLEX* dataset for seven languages, *GLOVE* accounts for the largest average percent improvement (3.68%), while *ESA* accounts for the lowest value (−1.52%). Without taking into consideration the *SIMLEX* dataset, *W2V* accounts for the largest average percent improvement (2.26%), while *ESA* accounts for the lowest value (−7.81%).

This can be explained by the sensitivity of these models to the corpus size due to the dimensional reduction strategy (*W2V, GLOVE*) or the broader context window (*ESA*). The remaining models are ($LSA = 0.79\%$, $W2V = -0.35\%$). *French* and *Dutch* achieved the highest percent gains (19.0% and 15.6%, respectively), while *German* accounts for worst results (−13.4%). The score per language is consistent with its respective corpus size (Table 5.1). For *German*, the result shows that the corpus volume of the *German* Wikipedia crossed a threshold size (34% of the English corpus) above which improvements for computing semantic similarity for the target word-pairs dataset might be marginally relevant, while the translation error accounts negatively in the final result. The average improvement for the *Bing* MT over the language specific models for each word-pairs dataset is: $MC = 1.8\%$, $RG = -5.7\%$, $WS353 = -2.1\%$ and $SIMLEX = 8.7\%$. *SIMLEX* works better than other because only seven European language have been accounted (*Bing* MT in Farsi and Arabic works Worst, have not been considered).

TABLE 5.11: Difference (%) between the *Bing* machine translation model and the *language-specific* (M. AVG represents the average of the models and DS. AVG represents the average of the datasets).

DS	DSMs	de	fr	it	nl	pt	sv	es	ru	ar	fa	Model AVG.	DS. AVG
MC	ESA	-31.3	24.4	-29.4	24.0	-38.6	29.5	-1.9	-8.7	30.0	-75.0	-7.7	1.8
	LSA	-15.0	24.0	-52.6	25.3	28.1	3.7	-10.3	20.8	3.0	-53.7	-2.7	
	W2V	-20.2	40.1	-29.8	10.2	53.0	38.1	-3.9	24.1	81.5	-70.0	12.3	
	GLOVE	-25.5	14.7	17.3	73.2	-26.8	25.6	0.1	-2.8	38.8	-63.5	5.1	
RG	ESA	-21.2	27.2	-16.1	12.6	-6.9	-2.1	0.4	-22.6	-1.7	-57.8	-8.8	-5.7
	LSA	-13.2	24.7	-21.5	13.6	36.6	-2.7	32.0	-16.9	-44.9	-42.4	-3.5	
	W2V	-19.9	19.5	-20.5	3.1	30.7	2.2	8.1	-19.4	-28.5	-51.4	-7.6	
	GLOVE	-18.4	24.3	0.3	26.1	-3.7	3.4	4.6	-20.1	9.3	-56.2	-3.0	
WS353	ESA	-12.3	23.0	-7.4	-19.3	-16.0	-12.6	8.4	-10.4	-5.8	-16.8	-6.9	-2.1
	LSA	-14.7	16.6	3.4	0.0	-0.3	4.6	14.7	10.6	0.8	-28.8	0.7	
	W2V	1.4	16.7	4.5	3.7	4.9	22.1	2.5	5.6	-6.0	-34.8	2.1	
	GLOVE	-19.9	9.6	-9.8	-0.7	-4.0	6.7	25.9	-2.2	-16.6	-31.7	-4.3	
	Lang AVG. (Above DS)	-17.5	22.1	-13.5	14.3	4.7	9.9	6.7	-3.5	5.0	-48.5	-2.0	
SIMLEX	ESA	13.6	24.7	10.0	43.0	16.0	22.5	-8.2	N/A	N/A	N/A	17.4	8.7
	LSA	2.8	5.6	39.6	-3.0	4.0	28.5	-17.6	N/A	N/A	N/A	8.6	
	W2V	-27.7	-7.9	-8.0	-1.6	-0.4	12.0	-23.7	N/A	N/A	N/A	-8.2	
	GLOVE	6.9	16.8	14.1	39.2	14.5	38.7	-11.7	N/A	N/A	N/A	16.9	
	Lang AVG. (All DS)	-13.4	19.0	-6.6	15.6	5.7	13.8	1.2				0.6	

Using the *Google* MT model, *W2V* is consistently the best performing DSM (average 0.58 *with SIMLEX*, 0.70 *without SIMLEX*), while *ESA* is consistently the worst performing model (average 0.47 *with SIMLEX*). We can interpret this result by stating that the benefit of using machine translation for *ESA* does not introduces significant performance improvements in comparison to the language-specific baselines. The best performing languages are *French* and *Spanish* ($\rho =$ 0.68). The Spearman correlation variance across languages for the *Google* MT models is low, as the impact of the use of the *English* corpus on the DSM model has a higher positive impact on the results in comparison to the variation of the quality of the *Google* machine translation. The results for all languages achieve very similar correlation values.

TABLE 5.12: Spearman correlation for the *Google* machine translation models over English corpus (M. AVG represents the average of the models and DS. AVG represents the average of the datasets).

DS	DSMs	de	fr	it	nl	pt	sv	es	ru	ar	fa	Model AVG.	DS. AVG
MC	ESA	0.57	0.69	0.71	0.68	0.56	0.67	0.82	0.42	0.31	0.58	0.60	0.70
	LSA	0.68	0.73	0.72	0.70	0.72	0.73	0.81	0.65	0.69	0.75	0.72	
	W2V	0.72	0.82	0.82	0.71	0.81	0.82	0.87	0.68	0.71	0.74	0.77	
	GLOVE	0.57	0.78	0.73	0.65	0.60	0.65	0.85	0.67	0.69	0.79	0.70	
RG	ESA	0.61	0.75	0.74	0.76	0.68	0.67	0.75	0.52	0.38	0.71	0.66	0.70
	LSA	0.66	0.65	0.74	0.70	0.67	0.68	0.65	0.59	0.55	0.70	0.66	
	W2V	0.72	0.82	0.81	0.77	0.78	0.76	0.75	0.69	0.57	0.79	0.75	
	GLOVE	0.65	0.79	0.79	0.76	0.69	0.68	0.74	0.71	0.65	0.80	0.73	
WS353	ESA	0.44	0.48	0.44	0.46	0.31	0.45	0.42	0.41	0.42	0.32	0.42	0.47
	LSA	0.50	0.51	0.49	0.52	0.38	0.51	0.45	0.47	0.37	0.43	0.46	
	W2V	0.63	0.63	0.60	0.66	0.56	0.61	0.58	0.57	0.50	0.52	0.59	
	GLOVE	0.44	0.47	0.45	0.46	0.32	0.43	0.42	0.42	0.33	0.37	0.41	
	Lang AVG. (above DS)	0.60	0.68	0.67	0.65	0.59	0.64	0.68	0.57	0.52	0.63	0.62	
SIMLEX	ESA	0.19	0.20	0.16	0.20	0.22	0.22	0.19	N/A	N/A	N/A	0.20	0.22
	LSA	0.18	0.20	0.18	0.19	0.20	0.23	0.19	N/A	N/A	N/A	0.20	
	W2V	0.20	0.24	0.23	0.24	0.24	0.25	0.24	N/A	N/A	N/A	0.23	
	GLOVE	0.24	0.25	0.23	0.25	0.26	0.28	0.25	N/A	N/A	N/A	0.25	
	Lang AVG. (all DS)	0.50	0.56	0.55	0.54	0.50	0.54	0.56					

The impact of the *Google* MT model can be better interpreted by examining the difference between the *Google* machine translation and the *domain-specific* models (depicted in Table 5.13). Taking into account the *SIMLEX* dataset for seven languages, *GLOVE* accounts for the largest average percent improvement (22.57%), while *ESA* accounts for the lowest value (7.22%). Without taking into account *SIMLEX* dataset, *GLOVE* accounts for the largest average percent improvement (23.55%), while *ESA* accounts for the lowest value (5.59%).

This can be explained by the sensitivity of these models to the corpus size due to the dimensional reduction strategy (*GLOVE*) or the broader context window (*ESA*). The remaining models are (*LSA* = 18.27%, *W2V* = 14.72%). Without taking into account the *SIMLEX* dataset, *French* and *Italian* achieved the highest percent gains (29.4% and 18.2%, respectively), while *German* accounts for worst results (−2.1%). These numbers are consistent with the corpus size. As previously noticed for *German*, the result shows that the corpus volume of the *German* Wikipedia crossed a threshold size (34% of the English corpus) above which improvements for computing semantic similarity for the target word-pairs dataset might be marginally relevant, while the translation error accounts negatively in the final result. The average improvement of *Google* MT over the language specific model for each word-pairs dataset is: *MC* = 23.5%, *RG* = 21.5%, *WS353* = 10.3% and *SIMLEX* = 7.5%. In summary, machine translation to English consistently performs better for all languages. The lightweight MT approach provides an average improvement of 15.12% over language-specific distributional semantic models.

TABLE 5.13: Difference (%) between the *Google* machine translation model and the *language-specific* (Model. AVG represents the average of the models and DS. AVG represents the average of the datasets).

DS	DSMs	de	fr	it	nl	pt	sv	es	ru	ar	fa	Model AVG.	DS. AVG
MC	ESA	-15.3	27.4	31.1	16.9	-27.8	25.2	26.4	-36.3	-16.4	3.2	3.4	23.5
	LSA	-2.8	31.1	4.3	17.0	43.1	6.0	12.6	4.3	50.3	67.8	23.4	
	W2V	4.6	51.5	24.7	15.4	76.8	43.9	11.4	6.7	87.5	9.4	33.2	
	GLOVE	-11.8	22.1	92.3	68.8	-13.3	14.7	19.0	-11.5	84.5	74.8	34.0	
RG	ESA	-10.2	29.1	18.3	19.8	8.3	5.9	9.3	-17.6	5.8	25.2	9.4	21.5
	LSA	2.4	36.0	20.5	20.3	53.5	4.2	26.6	14.3	57.5	51.5	28.7	
	W2V	-7.4	24.1	14.6	13.8	44.0	10.4	2.2	8.2	58.7	43.6	21.2	
	GLOVE	-5.3	27.7	37.1	39.4	12.1	10.7	4.2	1.0	103.8	35.5	26.6	
WS353	ESA	5.4	33.6	11.6	-15.3	-28.6	-6.3	10.9	-7.2	59.5	-13.5	5.0	10.3
	LSA	4.0	27.8	29.0	5.7	-1.4	14.8	19.2	18.9	34.2	1.3	15.3	
	W2V	11.4	16.3	14.6	10.1	3.7	25.1	7.3	9.1	12.0	-0.5	10.9	
	GLOVE	0.1	25.8	11.8	7.1	-15.4	13.0	26.8	0.0	27.5	3.6	10.0	
	Lang AVG. (above DS)	-2.1	29.4	25.8	18.2	12.9	14.0	14.7	-0.8	47.1	25.2	18.4	
SIMLEX	ESA	18.0	8.5	5.9	31.9	12.7	14.0	-13.7	N/A	N/A	N/A	11.0	7.5
	LSA	16.0	-10.4	31.4	2.9	0.0	20.9	-21.2	N/A	N/A	N/A	5.7	
	W2V	-13.1	-11.5	-3.8	2.8	-1.1	2.3	-20.6	N/A	N/A	N/A	-6.4	
	GLOVE	25.1	6.1	23.4	42.1	15.6	37.0	-11.6	N/A	N/A	N/A	19.7	
	Lang AVG. (all DS)	1.3	21.6	22.9	18.7	11.4	15.1	6.8					

5.2.8 Lightweight Machine Translation vs. Language-Specific DSMs

In the third part of the experiment we evaluate how the semantic similarity supported by the lightweight MT model performs in comparison to DSMs built over native language corpora. The Spearman Correlation (ρ) between human assessments was calculated for all native-language DSMs and English lookups supported by lightweight MT (Chapter 3, section 3.2.2).

The impact of the MT model can be better interpreted by examining the difference between the lightweight machine translation and the language-specific models (depicted in Table 5.15). *GLOVE* accounts for the largest average percent improvement (19.76%) using the lightweight MT model, while *LSA* accounts for the lowest value (12.96%). The remaining models accounted for substantial improvements ($W2V = 13.91\%$, $ESA = 13.84\%$). In terms of improvement per language, *Portuguese* achieved the highest percent gains (22.30%), while *Dutch* accounts for lower results (9.45%). The average improvement for the MT over the *language specific* model for each word-pair dataset is consistently significant: $MC = 26.90\%$, $RG = 16.73\%$, $WS353 = 6.58\%$ and $SIMLEX = 10.26\%$. The results shows in overall the results of *lightweight* MT outperforms the results of the *language-specific* models.

Another aspect that we can observe is with regard to which language benefited more from the application of the MT model. The comparative analysis between the models (Table 5.14) indicates that *Spanish* is the best-performing language (0.59), followed by *Swedish* (0.57). The lowest Spearman correlation was observed in *Dutch* (0.50). From the tested DSMs, *W2V* is consistently the best-performing DSM (0.62).

TABLE 5.14: Spearman correlation for the *lightweight* machine translation models over the *English* corpus.

DS	Model	de	fr	it	nl	pt	sv	es	Model AVG.	DS AVG.
MC	ESA	0.80	0.72	0.70	0.63	0.80	0.72	0.80	0.74	0.76
	LSA	0.72	0.71	0.67	0.65	0.67	0.80	0.78	0.72	
	W2V	0.80	0.86	0.75	0.72	0.82	0.89	0.87	0.82	
	GLOVE	0.79	0.78	0.70	0.61	0.80	0.78	0.82	0.75	
RG	ESA	0.71	0.77	0.68	0.68	0.79	0.73	0.81	0.74	0.72
	LSA	0.60	0.60	0.63	0.62	0.66	0.75	0.72	0.66	
	W2V	0.75	0.78	0.70	0.75	0.78	0.78	0.86	0.77	
	GLOVE	0.69	0.75	0.70	0.63	0.78	0.76	0.80	0.73	
WS353	ESA	0.46	0.41	0.39	0.44	0.44	0.42	0.41	0.42	0.47
	LSA	0.52	0.43	0.45	0.47	0.45	0.47	0.45	0.46	
	W2V	0.66	0.59	0.58	0.61	0.59	0.59	0.60	0.60	
	GLOVE	0.45	0.39	0.37	0.41	0.42	0.41	0.42	0.41	
SIMLEX	ESA	0.21	0.16	0.22	0.19	0.23	0.23	0.24	0.21	0.22
	LSA	0.20	0.16	0.20	0.18	0.21	0.23	0.23	0.20	
	W2V	0.21	0.20	0.23	0.22	0.24	0.27	0.27	0.24	
	GLOVE	0.25	0.20	0.26	0.23	0.27	0.27	0.29	0.25	
	Lang AVG.	0.55	0.53	0.51	0.50	0.56	0.57	0.59	0.54	

TABLE 5.15: Difference (%) between the *lightweight* machine translation model and the *language-specific*.

DS	Model	de	fr	it	nl	pt	sv	es	Model AVG.	DS AVG.
MC	ESA	19.35	33.02	29.18	7.98	3.34	34.04	23.45	21.48	26.90
	LSA	2.78	28.21	-3.27	10.03	33.76	15.04	9.49	13.72	
	W2V	15.70	59.27	14.92	17.32	78.78	55.52	10.72	36.03	
	GLOVE	23.27	21.41	82.90	57.73	15.72	37.98	15.53	36.36	
RG	ESA	5.01	31.75	8.79	6.35	25.71	15.62	17.52	15.82	16.73
	LSA	-7.72	26.71	3.03	6.93	51.64	16.14	40.81	19.65	
	W2V	-3.79	18.15	-0.48	10.91	42.95	13.70	17.46	14.13	
	GLOVE	1.07	21.74	21.24	15.24	26.02	23.70	12.39	17.34	
WS353	ESA	8.06	12.65	-1.78	-19.20	0.80	-11.18	7.06	-0.51	6.58
	LSA	7.82	6.24	17.73	-5.02	14.40	6.61	21.27	9.87	
	W2V	15.96	10.05	10.70	2.32	10.07	20.19	11.34	11.52	
	GLOVE	3.71	2.61	-6.86	-4.47	9.68	6.53	26.94	5.45	
SIMLEX	ESA	30.30	-14.49	43.77	25.28	18.70	17.78	8.57	18.56	10.26
	LSA	25.98	-28.37	47.90	-6.55	4.97	21.53	-5.14	8.62	
	W2V	-9.17	-26.35	-0.58	-4.82	-0.84	7.44	-7.86	-6.03	
	GLOVE	28.13	-15.10	37.04	31.12	21.10	32.99	3.85	19.88	
	Lang AVG.	10.41	11.72	19.01	9.45	22.30	19.60	13.34	15.12	

In terms of impact of the *lightweight* model for computing the Spearman correlation for different gold-standards: MC, RG and SIMLEX showed higher percentage improvements when compared to WS-353. The explanation can be found in the fact that the three former datasets focus on similarity computations (thus requiring more sensitive and informative semantic models) while WS-353 targets semantic relatedness.

On average the results show that the lightweight Machine translation consistently performs better for all languages.

5.2.9 Lightweight vs. Google and Bing Machine Translation

This section provides a comparative analysis of the *lightweight* MT model and the *Google* and *Bing* Services MT baselines. The Spearman correlation for the *lightweight* MT approach and their difference in relation to *Google* & *Bing* are shown in Tables 5.14, 5.16 and 5.17, respectively.

TABLE 5.16: Difference (%) between the *lightweight* machine translation model and the *Google* machine translation service.

DS	Model	de	fr	it	nl	pt	sv	es	Model AVG.	DS AVG.
MC	ESA	40.85	4.43	-1.45	-7.64	43.10	7.06	-2.31	12.01	6.08
	LSA	5.79	-2.18	-7.26	-5.98	-6.55	8.58	-2.78	-1.48	
	W2V	10.58	5.11	-7.83	1.69	1.11	8.04	-0.60	2.59	
	GLOVE	39.72	-0.57	-4.91	-6.56	33.49	20.25	-2.91	11.22	
RG	ESA	16.87	2.05	-8.06	-11.24	16.12	9.17	7.54	4.63	0.62
	LSA	-9.88	-6.85	-14.51	-11.08	-1.21	11.41	11.21	-2.99	
	W2V	3.95	-4.81	-13.18	-2.51	-0.69	2.99	14.89	0.09	
	GLOVE	6.78	-4.69	-11.56	-17.35	12.42	11.77	7.89	0.75	
WS353	ESA	2.55	-15.69	-12.00	-4.63	41.20	-5.20	-3.48	0.39	-1.53
	LSA	3.70	-16.90	-8.72	-10.11	16.04	-7.16	1.72	-3.06	
	W2V	4.06	-5.39	-3.42	-7.07	6.12	-3.90	3.80	-0.83	
	GLOVE	3.59	-18.42	-16.70	-10.77	29.65	-5.76	0.10	-2.61	
SIMLEX	ESA	10.38	-21.15	35.73	-5.00	5.34	3.30	25.74	7.76	2.93
	LSA	8.56	-20.09	12.54	-9.17	4.96	0.51	20.30	2.51	
	W2V	4.52	-16.82	3.36	-7.40	0.31	5.02	16.02	0.72	
	GLOVE	2.45	-19.97	11.06	-7.71	4.72	-2.90	17.52	0.74	
	Lang AVG.	9.65	-8.87	-2.93	-7.66	12.88	3.95	7.17	2.03	

TABLE 5.17: Difference (%) between the *lightweight* machine translation model and the *Bing* machine translation service.

DS	Model	de	fr	it	nl	pt	sv	es	Model AVG.	DS AVG.
MC	ESA	73.62	6.91	82.99	-12.92	68.40	3.49	25.83	35.47	22.75
	LSA	20.90	3.39	104.06	-12.21	4.45	10.89	22.03	21.93	
	W2V	45.04	13.66	63.67	6.42	16.84	12.61	15.27	24.79	
	GLOVE	65.57	5.81	55.90	-8.95	58.01	9.84	15.37	28.79	
RG	ESA	33.29	3.61	29.67	-5.56	35.02	18.10	17.07	18.74	13.38
	LSA	6.27	1.58	31.33	-5.83	11.02	19.31	6.69	10.05	
	W2V	20.05	-1.14	25.17	7.63	9.38	11.20	8.71	11.57	
	GLOVE	23.92	-2.04	20.84	-8.58	30.83	19.57	7.47	13.14	
WS353	ESA	23.22	-8.38	6.09	0.09	19.99	1.57	-1.28	5.90	5.45
	LSA	26.40	-8.92	13.83	-5.05	14.80	1.94	5.69	6.96	
	W2V	14.31	-5.67	5.89	-1.37	4.88	-1.55	8.58	3.58	
	GLOVE	29.45	-6.35	3.27	-3.79	14.25	-0.17	0.84	5.36	
SIMLEX	ESA	14.74	-31.42	30.73	-12.37	2.34	-3.83	18.25	2.63	2.65
	LSA	22.53	-32.17	5.93	-3.67	0.90	-5.46	15.10	0.45	
	W2V	25.55	-20.04	8.06	-3.31	-0.40	-4.06	20.74	3.79	
	GLOVE	19.84	-27.34	20.14	-5.77	5.76	-4.13	17.66	3.74	
	Lang AVG.	29.04	-6.78	31.72	-4.70	18.53	5.58	12.75	12.31	

In the analysis, word pairs were sent to the baseline machine translation services which translated them to *English*. The translated words were then used to compute the semantic relatedness using the *native English* DSMs and their Spearman

correlations with the translated pairs were computed.

The *lightweight* MT on average performs equivalently or better than *Google* and *Bing* MT (with the exception of *WS353* for *Google*): *Google* (*MC* = 6.08%, *RG* = 0.62%, *WS353* = −1.53% and *SIMLEX* = 2.93%), *Bing* (*MC* = 27.75%, *RG* = 13.38%, *WS353* = 5.45% and *SIMLEX* = 2.65%). A possible explanation for this observed behavior is that the baselines are MT models supported by language models which target the translation of sentences instead of word pairs.

On average the results show that using *lightweight* MT is equivalent or slightly better to more sophisticated machine translation. However, there were significant individual variations across languages and the baseline MT services. *Portuguese* and *German* achieved the highest percent gains (12.88% and 9.65%, respectively), *Google* MT outperformed the *lightweight* MT for *French*, *Dutch* and *Italian* (−8.87%, −7.66% and −2.93%, respectively). But compared with the *Bing* MT, *Italian* and *German* achieved the highest percentage gains (31.72% and 29.04%, respectively), while *Bing* MT outperforms the lightweight MT for *French* and *Dutch* (−6.78% and −4.70%, respectively).

Google MT currently uses Google Neural Machine Translation (GNMT) (Wu et al., 2016) which is an end-to-end learning approach for translation sentences. Our approach (Lightweight MT) differs in that the model which accesses the unigram-level source-target probabilities which can be directly computed from the parallel corpora. To our knowledge Google does not offer such approach in that none has been disseminated openly for academic peer review or analysis.

5.2.10 Word-pair Machine Translation Quality

In order to verify the hypothesis that the translation accuracy of the *lightweight* model is equivalent or superior to the *baseline* MT models, the quality of the MT was evaluated in isolation. Tables 5.20, 5.18 and 5.19 show the accuracy of all MT approaches using the translated gold-standard, and Tables 5.21 and 5.22 show the translation accuracy of *lightweight* MT and their difference in relation to *Google* and *Bing* MT, respectively.

At the accuracy or the translation for the *Google* MT and eleven language (Table 5.18), *WS353* has the best-performing translations with an average accuracy of 75% (with maximum 87% and minimum 43%) following by *SIMLEX* 0.74% accuracy (for Seven European language). This value dropped significantly for *Farsi* (average 39%). For *MC* and *RG*, the average translation accuracy for the semantic similarity pairs are 57% and 55%, respectively. This difference may be a result of a deficit of contextual information during the machine translation process. For these word-pairs datasets, the difference between best translation

performers and lower performers (across languages) is smaller. Additionally, the final translation accuracy for all languages and all word-pairs datasets is 66%. *Dutch*, *French* and *Spanish* are the languages with best automatic translations.

At the translation accuracy for the *Bing* MT for Seven language (Table 5.19), *WS353* has the set of best-performing translations with an average accuracy of 82% (with maximum 86% and minimum 78%) following by *SIMLEX* 0.76% accuracy. For *MC* and *RG*, the average translation accuracy for the semantic similarity pairs are 55% and 54%, respectively. This difference may be a result of a deficit of contextual information during the machine translation process. For these word-pairs datasets, the difference between best translation performers and lower performers (across languages) is smaller. Additionally, the final translation accuracy for all languages and all word-pairs datasets is 67%. *Dutch*, *Swedish* and *France* are the languages with best automatic translations.

The accuracy of the translation of the *lightweight* MT significantly outperforms the *Bing* and *Google* MT, except for 3 languages, especially for *German* (−7.89%).

TABLE 5.18: Translation accuracy for the *Google* MT.

Dataset/Lang	**de**	**fr**	**it**	**nl**	**pt**	**sv**	**es**	**ru**	**ar**	**fa**	**DS AVG.**
MC	0.48	0.58	0.65	0.60	0.58	0.68	0.67	0.58	0.53	0.38	0.57
RG	0.42	0.65	0.58	0.65	0.53	0.70	0.66	0.53	0.43	0.36	0.55
WS353	0.82	0.85	0.81	0.87	0.75	0.87	0.78	0.76	0.57	0.43	0.75
SIMLEX	0.69	0.75	0.75	0.77	0.71	0.81	0.73	N/A	N/A	N/A	0.74
Lang AVG	0.60	0.71	0.70	0.72	0.64	0.77	0.71	0.69	0.51	0.39	0.66

TABLE 5.19: Translation accuracy for the *Bing* MT.

Dataset/Lang	**de**	**fr**	**it**	**nl**	**pt**	**sv**	**es**	**DS AVG.**
MC	0.48	0.47	0.42	0.58	0.60	0.60	0.60	0.54
RG	0.45	0.65	0.41	0.60	0.51	0.62	0.59	0.55
WS353	0.80	0.86	0.78	0.86	0.82	0.83	0.80	0.82
SIMLEX	0.72	0.77	0.75	0.79	0.79	0.77	0.73	0.76
Lang AVG	0.61	0.69	0.59	0.71	0.68	0.70	0.68	0.67

TABLE 5.20: Translation accuracy for the *lightweight* MT.

Dataset/Lang	de	fr	it	nl	pt	sv	es	DS AVG.
MC	0.54	0.60	0.61	0.68	0.54	0.66	0.65	0.61
RG	0.48	0.61	0.56	0.61	0.47	0.68	0.64	0.58
WS353	0.83	0.82	0.75	0.86	0.82	0.83	0.80	0.82
SIMLEX	0.74	0.71	0.75	0.78	0.75	0.74	0.76	0.75
Lang AVG	0.65	0.68	0.67	0.73	0.65	0.73	0.71	0.69

TABLE 5.21: Difference (%) in translation accuracy between lightweight MT and Google MT.

Dataset/Lang	de	fr	it	nl	pt	sv	es	GS AVG.
MC	-10.77	-2.10	6.12	-12.20	7.69	3.14	1.91	-0.88
RG	-12.35	7.59	3.75	6.33	12.65	3.70	3.30	3.57
WS353	-0.94	3.58	7.50	0.90	-7.72	4.43	-2.13	0.80
SIMLEX	-7.50	4.87	-0.45	-1.45	-6.06	9.01	-4.93	-0.93
Lang AVG	-7.89	3.49	4.23	-1.60	1.64	5.07	-0.46	0.64

TABLE 5.22: Difference (%) in translation accuracy between lightweight MT and Bing MT.

Dataset/Lang	de	fr	it	nl	pt	sv	es	GS AVG.
MC	12.07	27.68	47.00	17.14	-9.72	10.42	9.03	16.23
RG	8.19	-7.06	38.21	1.28	-7.20	9.69	8.12	7.32
WS353	3.08	-4.57	-2.74	0.58	0.09	0.17	0.18	-0.46
SIMLEX	2.87	-7.98	-0.15	-0.33	-4.30	-2.94	4.89	-1.13
Lang AVG	6.55	2.02	20.58	4.67	-5.28	4.33	5.55	5.49

5.2.11 Parallel Corpora Size & MT Quality

Our last analysis focuses on the correlation between the size of the supporting parallel corpora (Table 5.1) used to built the *lightweight* MT model and the *Spearman correlation* for each gold standard, averaged for all models (Figure 5.1). As the *lightweight* MT model works over a word-based lexical table, the model is more dependent on a parallel corpora with a representative set of unigram translations instead of a language model which is able to model phrasal (above bigram) translations. This shows that the lightweight MT can be potentially transported to languages with smaller parallel corpora.

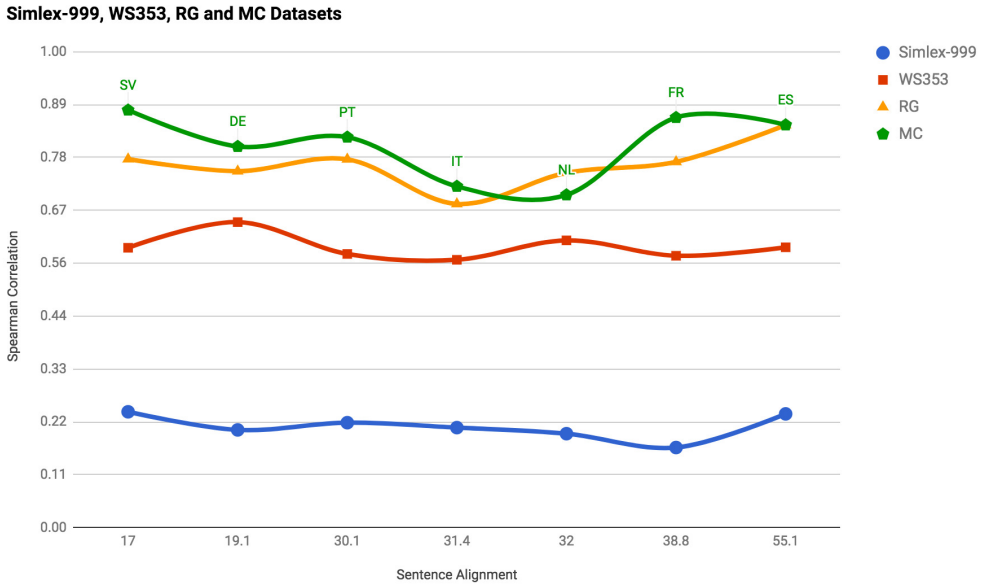

FIGURE 5.1: Correlation between the Spearman correlation values evaluated by lightweight MT over English-DSM and size of parallel corpora that the *lightweight* MT is learned over them.

5.2.12 Summary

Below, the interpretation of the results are summarised as the core research questions which we aim to answer in this Section:

Question 1: Does machine translation of non English input to English perform better than the word vectors in the original language (for which languages and for which distributional semantic models)? Machine translation to English consistently performs better for all languages. The *lightweight* MT approach provides an average improvement of 15.12% over language-specific distributional semantic models.

Question 2: How does a lightweight MT model compares with more complex MT models? The *lightweight* Machine translation to English consistently performs better for all languages. The lightweight MT approach provides an average improvement of 12.31% over *Bing* MT and 2.03% over *Google* MT.

Question 3: Which DSMs or MT-DSMs work best for the set of analysed languages? W2V-MT consistently performs as the best model for all word-pairs datasets and languages.

Question 4: What is the quality of machine translation approaches for word-pairs? The average translation accuracy for all languages and all word-pairs

datasets is 69%. Translation quality varies according to the nature of the word-pair (better translations are provided for word pairs which are semantically related compared to semantically similar word pairs), reaching a maximum of 85% and a minimum of 36% across different languages.

For the distributional semantics user/practitioner, as a general practice, we recommend using *W2V* built over an *English* corpus, supported by machine translation. Additionally, the accuracy of machine translation approaches work better for translating semantically related word pairs (in contrast to semantically similar word pairs).

5.3 Specific-Domain Analysis

5.3.1 Introduction

The evaluation aims to find which DSM configurations provide better performance for assessing semantic relatedness for smaller, domain-specific corpora. Another question that we aim to answer is how much information is needed to build effective dsDSMs.

This section answers the following research questions:

- **RQ 1** Which types of Distributional Semantic Models are most cost-effective with respect to smaller, domain-specific corpora?
- **RQ 2** What type of discourse expressed in small-scale corpora leads to better domain-specific Distributional Semantic Models (dsDSMs)?
- **RQ 3** How to quantify the factors which influence the performance of ds-DSMs?

5.3.2 Evaluation Methodology

The evaluation methodology focuses on making explicit the steps necessary to evaluate the research hypotheses. It consists of the following steps:

1. Creation of two test collections in the financial domain for the evaluation of domain-specific DSMs.
2. The instantiation of dsDSMs associated with the different discourse types.
3. The evaluation of the Spearman correlation for semantic relatedness tasks against the domain-specific test collections.

5.3.3 Corpus Acquisition

5.3.3.1 Financial Corpora

In order to create a diversified dataset covering a wide range of financial terms, we employed a twofold crawling approach. We focused on trusted finance sources for two categories of discourse: *financial glossaries* and *curated encyclopaedias*.

We crawled definitions from three authoritative Web sources to create the *Glossary* subcorpus:

- The Bloomberg financial Glossary[12] (8324 definitions; 212,421 tokens).
- SGM Glossary[13] (1007 definitions; 43,638 tokens).
- Investopedia definitions[14] (15476 definitions; 2,462,801 tokens).

For the the *Encyclopedic* subcorpus two sources were used:

- Investopedia articles[15] (5890 articles; 5,129,793 tokens).
- Wikipedia (articles on Investment[16] and Finance[17]) (8306 articles; 6,714,129 tokens).

The combination of these two sub-corpora, *Finance-all*, consists of a total of 14,562,782 tokens.

5.3.3.2 Generic Corpus

In order to compare DSMs trained on domain-specific versus generic corpora, we generated DSMs from Wikipedia dumps[18] (as of September 2017), which were preprocessed by lowercasing and removing stopwords (without applying stemming approach), resulting in a final corpus size of 900 million tokens. Table 5.23 shows an overview of the corpora used in this work.

5.3.3.3 Enriched Finance Corpora

Due to the reduced size, the finance corpus suffers from scarcity of lexical associations. We enriched the finance corpus in order to compare DSMs trained on

[12] http://www.isotranslations.com/resources/Bloomberg%20Financial%20Glossary.pdf
[13] http://www.sapient.com/content/dam/sapient/sapientglobalmarkets/pdf/thought-leadership/SGM_Glossary_2014_final.pdf
[14] http://www.investopedia.com/terms/a/
[15] http://www.investopedia.com/articles/pf/
[16] https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Investment
[17] https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Finance
[18] https://dumps.wikimedia.org/enwiki/

domain-specific and generic corpora versus enriched financial corpus, we enriched the *Finance corpora* by injecting it with the definitions (glosses) of each word on *Finance corpora* from WordNet[19], Wiktionary dump[20], if the definition is related to the finance domain. [21]. Table 5.23 shows an overview of the corpora used in this work.

TABLE 5.23: Corpora used for the DSMs

Corpus	Sources	# tokens
Glossary	Bloomberg glossary, SGM glossary, Investopedia definitions	2,718,860
Encyclopedic	Investopedia articles, Wikipedia-Finance, Wikipedia-Investment	11,843,922
Finance-all	all of the above corpora(Glossary, Encyclopedic)	14,562,782
Enriched Finance Corpora	Finance-all, Wordnet, Wiktionary	911 million
Wikipedia-generic	Wikipedia dumps	900 million

5.3.4 Creation of the Test Collection for Evaluation

The evaluation methodology inherits from the existing methodologies for the evaluation of semantic similarity and relatedness, which consists of comparing the Spearman correlation between the semantic similarity scores of word pairs assigned by the DSM with the semantic similarity and relatedness of the gold standard (Resnik, 1999; Finkelstein et al., 2001).

5.3.4.1 Word pair creation

After the creation of the financial corpus, we selected *syntagmatic* word pairs from the *Glossary* and *Encyclopedic* financial corpora according to the following methodology: splitting the corpus into sentences, the first word of the pair was randomly selected amongst the tokens in the sentence, with the only constraint that it was listed in one of the three financial glossaries. Then, the second word was manually selected, ensuring that various degrees of relatedness were represented. This process resulted in a set of 422 word pairs. Also we selected *paradigmatic* word pairs from FinNet, a third-party financial vocabulary in RDF (which includes a taxonomy) (Maia et al., 2018; Davis et al., 2016).

5.3.4.2 Human annotation

The *Syntagmatic* word pairs were annotated for semantic relatedness by two domain experts with an economic and financial background and native or near-native proficiency in English. The annotators were instructed to assign a score

[19] https://wordnet.princeton.edu/
[20] https://dumps.wikimedia.org/enwiktionary/
[21] We find out it by using the vocabulary-based disambiguation

between 0 (unrelated) and 10 (highly related or identical) to each pair. Examples from general-domain language were used to illustrate relatedness, instantiating relations such as synonymy (*pretty* - *attractive*), antonymy (*dead* - *alive*), meronymy (*hand* - *body*), functional relation (*electricity* - *oven*) and association (*ocean* - *cruise*). Fractional scores such as 6.5 were allowed. In case no score could be assigned with confidence, word pairs could be skipped. The experts could also use dictionaries in case of doubts regarding the meaning of a specific word. For the final version of the dataset, the average of the individual scores was taken as relatedness value of the pair.

Table 5.24 shows some examples of *Syntagmatic* word pairs (materialised within the SFWP-422 dataset) with their human scores.

TABLE 5.24: Some example of *syntagmatic* pairs

Word1	Word2	Score
stock	exchange	8.75
policy	understanding	2
cover	repair	5
without	intangible	2.5
long range	advisor	2
purchase	payment	8
leading	service	1.5
fund	chairman	6

Figure 5.2 shows the distribution of scores in the *syntagmatic* dataset. The scores are evenly distributed (Figure 5.2) with a mean score of 5.54, and a standard deviation of 2.43. The Spearman correlation between the two human annotators is 0.571, defining an upper limit for the maximum level of correlation that can be achieved within this dataset.

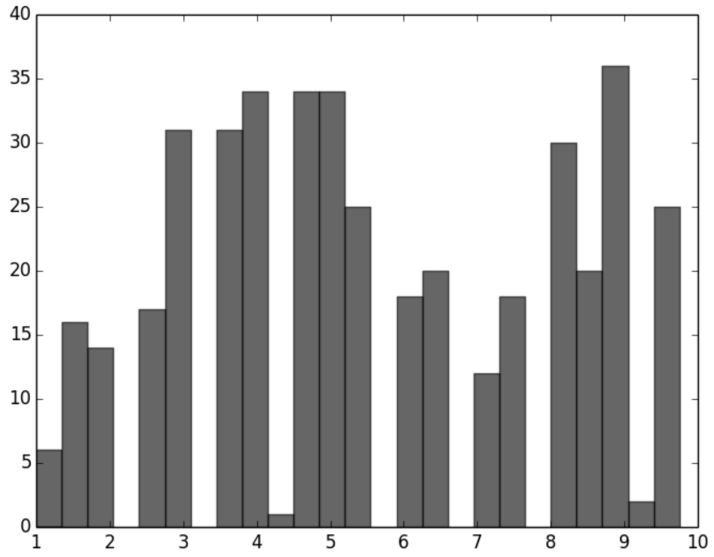

FIGURE 5.2: Distribution of human assigned semantic relatedness scores for the *SFWP-422* dataset.

The *paradigmatic* word pairs were annotated automatically according to the following methodology: The sibling subclasses have the highest scores (8), with the scores decreasing proportionally to the distance within the taxonomy. The lowest score is 1.

Table 5.25 shows examples of paradigmatic word pairs with their associated human annotated average score. This was materialised into the PFWP dataset.

TABLE 5.25: Some example of *paradigmatic* pairs

Word1	Word2	Score
uptrend	price pattern	5
unusual low volume	event	3
descending triangle	tripletop	8
overselling	corporate event	5
bank	investor	8
breakout	through	8
hanging man	harami	8
buyside	fundamentals	8
hikkake	intraday pattern	8

Figure 5.3 shows the distribution of scores in the paradigmatic dataset. The scores are evenly distributed (Figure 5.3) with a mean score of 6.734, and a standard deviation of 1.696.

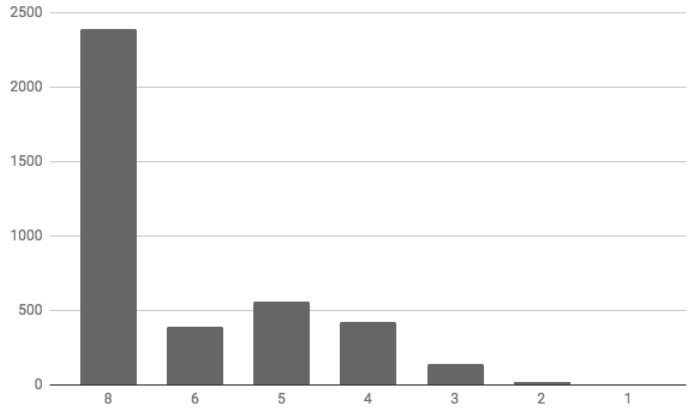

FIGURE 5.3: Distribution of human assigned semantic relatedness scores for the *PFWP-3920* dataset.

5.3.5 Vector Space Model Parameterisation

With regard to the model parameterization, initial parameters were chosen based on configurations which are common best practices (considered optimal configurations) in the literature (Kiela and Clark, 2014; Barzegar et al., 2015). While *ESA* uses the whole document as its context window, the other models restrict this to word windows of *five* (*LSA*), and *ten* words (*Word2Vec* and *GloVe*). With regard to the dimensionality of the vector space (number of latent dimensions), vector sizes of 300 for *Word2vec*, *Glove*, and *LSA* were selected. Since the *ESA* model is based on a sparse concept vector, each dimension in the *ESA* vector is the identifier for a document. In this version of the experiment, we use *Cosine* similarity between vectors to predict the relatedness between word pairs. *Cosine* similarity was used as the distance measure for all the models.

As a fifth model, we used the post-processing approach proposed by (Speer, Chin, and Havasi, 2017) (*ConceptNet-Numberbatch*), which injects semantic constraints into existing distributional vector spaces to combine the trained *W2V* model on the *finance corpora* to the commonsense graph resource ConceptNet Version 5.5 (*FinS-CN*).

An instance of each of the four models was instantiated on the finance-all corpora in Table 5.23, yielding four different configurations. The experiment consists in the construction of four DSMs based on the collection of definitional (*glossaries*) and *encyclopedic* (*Investopedia, Wikipedia-Finance, Wikipedia-Investment*) corpora within the financial domain. Additionally DSMs built over the full *Wikipedia* and the enriched *Finance corpora* were also used in the evaluation. The distributional models were used to calculate the semantic relatedness measures for the word pairs. The model scores were then compared against the human gold standard average using Spearman correlation.

5.3.6 Results

Table 5.26 shows the Spearman correlation of the *syntagmatic* dataset for each domain-specific corpus for the *four* instantiated DSMs. It can be observed that for the *FinS-CN* model, which is based on a predictive-based model, provides the configuration with the highest Spearman correlation. The nature of the predictive-based approach, which aims at using a backpropagation-based model to predict the relationship between words and target context windows supports the construction of models which generalise with *syntagmatic* samples. This is different from weighting schemes such as raw word counts and *tf-idf* (present in *LSA* and *ESA*) which are dependent on their comparative distributions over the corpora.

TABLE 5.26: Spearman correlation on *Syntagmatic* dataset (cosine) for different models by *Glossary*, *Encyclopedic*, *Financial-all* and *FinS-CN*

Model	**Glossary**	**Encyclopedic**	**Finance-all**	**FinS-CN**	**Model AVG.**
LSA	0.2621	0.2470	0.2641	N/A	0.2577
ESA	0.2610	0.1044	0.3586	N/A	0.2413
W2V	0.4093	0.4141	0.4822	0.5783	0.4710
GLOVE	0.0497	-0.0013	0.3776	N/A	0.1420
Source AVG.	0.2455	0.1911	0.3706	0.5783	

At the paradigmatic dataset (Table 5.27), *ESA* is the best performing model on *Finance-all* corpus ($\rho = 0.1945$), which still has a better *spearman correlation* score compare to *ESA* model for *SIMLEX* dataset on the wikipedia corpus, as a generic domain (Freitas et al., 2016). Although *ESA* has a weighting scheme based on *tf-idf*, it uses the whole document as a context window, in this case, the definitions. We enriched the *Finance-all* corpora by injection the definitions of each word on *Finance corpora* from *WordNet* and *Wiktionary* if the definition related to the finance domain by using the vocabulary-based disambiguation[22]. Based on the result the disambiguation approach does not work well.

[22]If it exists in *Glossary*

TABLE 5.27: Spearman correlation on *Paradigmatic* dataset (cosine) for different models by *Glossary*, *Encyclopedic*, *Financial-all* and *Enriched Finance-all/*

Model	**Glossary**	**Encyclopedic**	**Finance-all**	**Enriched Finance-all**	**Model AVG.**
LSA	-0.0748	-0.1321	-0.1177	0.1471	-0.0444
ESA	-0.1358	0.0039	0.1946	0.1837	0.0616
W2V	-0.0294	-0.1280	-0.0140	0.00812	-0.03376
GLOVE	0.0272	0.0311	0.1577	0.1037	0.0799
Source AVG.	0.0621	-0.0563	0.1387	0.1131	

The financial-specific DSMs were contrasted with the instantiation of the *W2V* model using *Wikipedia* as an open domain corpus. Table 5.30 shows the Spearman correlation for the *Wikipedia-generic*, *Finance-all* corpora (the latter combining the two financial corpora *Glossary* and *Encyclopedic*) and *FinS-CN*. While the open-domain based approaches are less efficient than their domain-specific counterparts, the level of association provided by a larger corpus, in particular on the Syntagmatic dataset, capturing associations in which one of the words is not domain-specific (e.g. revenue, *increase*), significantly affects the result (*Wikipedia-generic* 9.77% improvement and *FinS-CN* 19.93% improvement).

Table 5.28 and 5.29 show the correlation between the average Spearman correlation values (based on *Sytagmatic* and *Paradigmatic* datasets, respectively) for each DSM and two indicator of corpus size: # of tokens and # of unique tokens. On Table 5.28 *W2V* is consistently more robust that the other models in relation to the corpus size.

TABLE 5.28: Correlation between corpus size and different models based on the *Syntagmatic* dataset (Scale of 10^5 for Unique words, Scale of 10^7 for Tokens).

ρ Density	Glossary		Article		Finance All	
	/ unique-words	/ tokens	/ unique-words	/ tokens	/ unique-words	per tokens
LSA	0.65	0.96	0.14	0.21	0.14	14.20
ESA	0.64	0.96	0.06	0.09	0.19	0.25
W2V	1.01	1.50	0.24	0.35	0.26	0.33
GLOVE	0.12	0.18	0.00	0.00	0.20	0.26

TABLE 5.29: Correlation between corpus size and different models based on the *Paradigmatic* dataset (Scale of 10^5 for Unique words, Scale of 10^7 for Tokens).

ρ Density	Glossary		Article		Finance All	
	/ unique-words	/ tokens	/ unique-words	/ tokens	/ unique-words	/ tokens
LSA	-0.18	-0.28	-0.08	-0.11	-0.06	-0.08
ESA	-0.33	-0.50	0.00	0.00	0.10	0.13
W2V	-0.16	-0.24	-0.07	-0.11	-0.05	-0.06
GLOVE	0.07	0.10	0.02	0.03	0.08	0.11

TABLE 5.30: Spearman correlation on *Syntagmatic* and *Paradigmatic* datasets (cosine) for *Wikipedia-generic*, *FinS-CN* and *Finance-all* corpora based on *W2V* DSMs

Dataset	**Finance-all**	**FinS-CN**	**Wikipedia-generic**	**Model AVG.**
Syntagmatic	0.4822	0.5783	0.5293	0.5299
Paradigmatic	-0.0846	0.0137	0.1549	0.0191

5.3.7 Correlation by Semantic Relation Type

In order to determine which types of semantic relations were most costly to compute, the term pairs were annotated with regard to a set of semantic relations extracted from the *DOLCE* ontology (Masolo et al., 2003). First, we assigned a *DOLCE* class to each term using *WordNet-DOLCE* alignment (Gangemi et al., 2003). In a second step, we searched for the most suitable relation between them, i.e. a property from *DOLCE* having the classes assigned to the concepts as domain and range. We were able to assign direct *DOLCE* relations[23] to 77 term pairs.

[23]For this evaluation, we ignore composite relations with several intermediate concept pair relations linking the two terms.

TABLE 5.31: Spearman correlation for different categories of relation types.

DOLCE relation	Average Spearman	DOLCE relation	Average Spearman
patient	0.455	descriptive-place	0.161
product	0.420	used-by	0.151
parametrized-by	0.414	result	0.151
involves	0.376	referenced-by	0.128
product-of	0.366	uses	0.102
performs	0.337	abstract-location-of	0.100
descriptive-place-of	0.306	result-of	0.087
instrument	0.259	target	0.075
use-of	0.241	expresses	0.066
participant-place	0.222	patient-of	0.056
target-of	0.219	about	0.000
made-by	0.192	happens-at	0.000
part-of	0.178	used-in	-0.003
references	0.177	unit-of	-0.023
component-of	0.172	performed-by	-0.113
weak-connection	0.167		

We thus segmented the word pair dataset into 31 distinct *DOLCE* categories and calculated the Spearman correlation for each category of semantic relation (based on a DSM which was trained on the *Wikipedia* corpus using *W2V*). The *DOLCE* categories with higher agreement in terms of semantic relatedness are *patient* (0.46), *product* (0.42), *parameterised-by* (0.41), *involves* (0.38), and *product-of* (0.37). Mentions including passive involvement (*patient*), production (*product*, *product-of*), parameterizations (*parameterized-by*) and descriptions (*involves*) are easier to capture for DSMs, possibly because these stem from a limited/finite repository of lexical expressions and lexical contexts. The most costly relations are *performed-by* (-0.11), *unit-of* (-0.02), *used-in* (-0.002), *about* (0), and *happens-at* (0). The associations between actions and both agents (*performed-by*) and endurants (*used-in*) are difficult for DSMs, presumably due to the variety of possible associations and ways in which they can be expressed. *Unit-of*, *about* and *happens-at* could simply be due to sparsity in the underlying corpus. Table 5.31 shows the distribution of Spearman correlation value for the *W2V* model over the full *Wikipedia* for each type of *DOLCE* relation.

5.3.8 Summary

This chapter provides a detailed analysis of the performance of existing distributional semantic models (DSMs) under a domain-specific task. As an experimental setting, the financial domain of discourse was chosen. One central factor in the construction of domain specific DSMs include the scarcity of available corpora in comparison with open domain scenarios, and the demand for more fine-grained

semantic relatedness measures.

The experimental analysis, performed using two financial word pair gold standard the first containing 422 manually annotated word pairs (*SFWP-422*) and the second containing 3920 automatically annotated word pairs (*PFWP-3920*), analysed *domain-specific* discourse (*encyclopedic, glossary*) and *open domain (encyclopedic)* for the construction of the distributional models.

Below, the interpretation of the results are summarised as the core research questions which we aim to answer in this Section:

- **Question 1:** Which types of Distributional Semantic Models are most cost-effective with respect to smaller, domain-specific corpora?

  The results show that for the *syntagmatic* dataset, the best performing model is *W2V*, which was generated using the *Finance-all* ,*open domain encyclopedic* corpora (*FinS-CN*) ($\rho = 0.5783$). For the *paradigmatic* dataset, the combination of *ESA* and domain-specific discourse (encyclopedic, glossary) provide the best performing model ($\rho = 0.1946$).

- **Question 2:** What type of discourse expressed in small-scale corpora leads to better domain-specific Distributional Semantic Models (dsDSMs)?

  Encylopedic corpora that is included from *Investopedia* and *wikipedia articles on Investment and Finance* leads to a better domain-specific Distributional Semantic Models, specially for *W2V* model.

- **Question 3:** How to quantify the factors which influence the performance of dsDSMs?

  Based on the results, the predictive word embedding models (*W2V* and *GLOVE*) can capture better semantic information than count-based WEMs from a small-scale corpora. Furthermore, injecting semantic constraints into existing distributional vector spaces, thus combining trained *W2V* model on *finance corpora* with a commonsense knowledge graph resource ConceptNet Version 5.5 - (*FinS-CN*), leads to best performance for dsDSMs.

Chapter 6

Evaluation of the Composite Semantic Relation Classification

6.1 Introduction

As discussed in Chapter 3, this thesis aims to investigate two main conceptual gaps in the development of a distributional semantics architecture: (i) language and domain transportability and (ii) a transition point from distributional to fine-grained semantics. The latter (ii) led to the development of a supporting composite relation classification model (Described in Chapter 3, Section 3.4) which complements the coarse grained semantics of distributional semantics and the associated semantic relatedness/similarity scores with a set of relations linking the concepts.

This Chapter provides an evaluation of the composite semantic relation classification (Section 3.4) developed within the proposed architecture and it aims at addressing the following research questions:

- **RQ 1:** How distributional semantic relatedness models can be complemented with composite semantic relations?
- **RQ 2:** How to evaluate composite semantic relation classification?

6.2 Evaluation Methodology

The evaluation methodology focuses on making explicit the steps necessary to evaluate the research hypotheses. It consists of the following steps:

- Bring a systematic way to evaluate composite semantic relation classification.
- Creation of a test collection for the composite relation classification task.

- Evaluation of the results (using the *measures: precision, recall,* f_1 *score and accuracy*).
- Post-processing the distributional semantic models on specific semantic relations and lexical categories.

6.3 Training and Test Dataset

Two evaluation sets were generated by collecting all pairs of entity mentions in the *SemEval* 2010 task 8 (Hendrickx et al., 2009). The first dataset consists of entity pairs that have no attached semantic relation classes (i.e. which contained the relation label `OTHER`) while the second dataset contains *ALL* relations including relations labelled with `OTHER`[1].

For all entities, we performed a *ConceptNet* lookup (Speer and Havasi, 2012), where we generated all paths from lengths 1, 2 and 3 (number of relations) occurring between both entities(e_1 and e_2) and their relations (R). For example:

$\mathbf{e1} - R1_i - \mathbf{e2}$
$\mathbf{e1} - R1_i - \mathbf{X1}_n - R2_j - \mathbf{e2}$
$\mathbf{e1} - R1_i - \mathbf{X1}_n - R2_j - \mathbf{X2}_m - R3_k - \mathbf{e2}$

where X contains the intermediate entities between the target entity mentions **e1** and **e2**. Obviously, between two entities there may be different paths expressed with different intermediary entities and relations. For instance, for the paths between *silver* and *ring* entities we have:

- **silver**/UsedFor/**jewelry**/MadeOf/**gold**/AtLocation/**ring**
- **silver**/Antonym/**gold**/AtLocation/**ring**
- **silver**/Antonym/**bronze**/Antonym/**gold**/AtLocation/**ring**
- **silver**/AtLocation/**jewelry**/MadeOf/**gold**/AtLocation/**ring**

In next step, the *Distributional Navigational Algorithm (DNA)* is applied over the entity paths(Freitas et al., 2014; Silva, Handschuh, and Freitas, 2018a). In the final step of generating the training & test datasets from $3,728$ entity pairs assigned to the `OTHER` relation label in *SemEval* (`OTHER` dataset), we found $20,261$ paths. From $21,434$ entity pairs assigned to `ALL` relations in *SemEval* (`ALL` data set), we found $111,526$ paths in *ConceptNet*.

[1] Called `OTHER` and `ALL` sets, respectively.

6.4 Baseline Models

The performance of baselines is measured using the test dataset, as defined in Section 6.3 where we hold out the last relation and rate a system by its ability to infer this relation. As baselines, we use language models which define the conditional probabilities between a sequence of semantic relations r after the observation of entities e, i.e. $P(r \mid e)$.

- **Random Model:** This is the simplest baseline, which outputs randomly selected relation pairs.
- **Unigram Model:** Predicts the next relation based on unigram probability of each relation which was calculated from the training set. In this model, relations are assumed to occur independently.
- **Bigram Model:**

  The Bigram model is defined by (Jans et al., 2012):

  $$P(r \mid e) = \frac{P(r, e)}{P(e)} \tag{6.1}$$

  where $P(r \mid e)$ is the probability of seeing e and r, in order. Let A be an ordered list of relations and entities, $|A|$ is the length of R, For $i = 1, .., |A|$, define a_i to be the ith element of A. We rank candidate relations r by maximising P(r,a), defined as

  $$P(r, A) = \sum_{i=1}^{|A|-1} logP(r \mid a_i) \tag{6.2}$$

  where the conditional probabilities $P(r \mid a_i)$ calculated using Equation (6.1).
- **Random Forest:** This is an ensemble learning method for classification and other tasks, that operates by constructing a multitude of decision trees at training time. Random decision forests correct for the decision trees' limitation of overfitting to their training set.

6.5 Prediction Task

The Neural Entity/Relation Model (`NERM`) predicts composite relations between two given entities ($\mathbf{e}_1$ and $\mathbf{e}_2$). Given a sequence of source and target entities and a sequence of relations between them, the task consists of the prediction of the next relation X_i.

A semantically relevant path $\mathbf{e}_1 - R1_i - \mathbf{X1}_n - R2_j - \mathbf{X2}_m - R3_k - \mathbf{e}_2$ is converted into the following formats (Tables 6.1, 6.2 and 6.3) for the classification

task[2] (for different path lengths):

TABLE 6.1: First evaluation dataset for Neural Entity/Relation Model.

Input	Prediction
$\mathbf{e}_1$ $\mathbf{e}_2$ $\mathbf{X1}_n$	$\mathbf{R1}_i$
$\mathbf{e}_1$ $\mathbf{e}_2$ $\mathbf{X2}_m$ $\mathbf{X1}_n$ $\mathbf{R1}_i$	$\mathbf{R2}_i$
$\mathbf{e}_1$ $\mathbf{e}_2$ $\mathbf{X2}_m$ $\mathbf{R2}_i$ $\mathbf{X1}_n$ $\mathbf{R1}_i$	$\mathbf{R3}_i$

TABLE 6.2: Second evaluation dataset for Neural Entity/Relation Model.

Input	Prediction
$\mathbf{e}_1$ $\mathbf{X1}_n$	$\mathbf{R1}_i$
$\mathbf{e}_1$ $\mathbf{R1}_i$ $\mathbf{X1}_n$ $\mathbf{X2}_m$	$\mathbf{R2}_i$
$\mathbf{e}_1$ $\mathbf{R1}_i$ $\mathbf{X1}_n$ $\mathbf{R2}_i$ $\mathbf{X2}_m$ $\mathbf{e}_2$	$\mathbf{R3}_i$

TABLE 6.3: Third evaluation dataset for Neural Entity/Relation Model.

Input	Prediction
$\mathbf{e}_2$ $\mathbf{e}_1$ $\mathbf{X1}_n$	$\mathbf{R1}_i$
$\mathbf{e}_2$ $\mathbf{e}_1$ $\mathbf{R1}_i$ $\mathbf{X1}_n$	$\mathbf{R2}_i$
$\mathbf{e}_2$ $\mathbf{e}_1$ $\mathbf{R1}_i$ $\mathbf{X1}_n$ $\mathbf{R2}_i$ $\mathbf{X2}_m$	$\mathbf{R3}_i$

We provide statistics for the generated datasets in the Table 6.4, whereby our dataset is divided into a training set and a test set with scale $(80-20\%)$. Also we used 20% of the training set for cross-validation. For the `OTHER` dataset, we have $16,166$ examples for training, $4,042$ for validation and $5,952$ for testing and for `ALL` set we have $90,493$ examples for training, $22,624$ for validation and $28,280$ for testing.

[2]The best format based on our experiments is Table 6.1

TABLE 6.4: Distribution of instances used to train the LSTM model.

Dataset	#Train	#Dev	#Test
`OTHER`	16,166	5,052	4,042
`ALL`	90,493	28,280	22,624

6.6 Word Embeddings

The experimental setup consists of the instantiation of the W2V distributional semantic model (*Word2Vector (W2V)*(Mikolov et al., 2013a)). Indra (Chapter 3, Section 3.5) provides a software infrastructure which facilitates the experimentation and customisation of multilingual Word Embedding Models (Barzegar et al., 2018b; Barzegar et al., 2018c), allowing end-users and applications to consume and operate over multiple word embedding spaces as a service. In the experimental setup, we used `INDRA` as a service to get word embeddings for our classification model.

6.7 Results

To achieve the classification goal, we generated a Neural Entity/Relation Model for the composite relation classification task. In our experiments, a final batch size 25, with 50 epochs was used. An embedding layer was defined using Word2Vec pre-trained vectors (Freitas et al., 2016; Sales et al., 2018).

In our experiment, we optimised the hyper-parameters of the LSTM model. After several preliminary experiments, the best model was achieved with the following set of parameters:

- Input length and dimension are 6 and 300, respectively.
- Three hidden layers with 450, 200 and 100 nodes and *Tanh* activation,
- Dropout technique (0.5),
- *Adam* optimiser.

We configured our Neural Entity/Relation Model and conducted experiments with three different pre-trained embedding settings:

- *Word2Vec (Google News)* with 300 dimensions
- *Word2Vec (Wikipedia 2016)* with 300 dimensions

- No pre-trained word embedding

The accuracy for the configuration above after 50 epochs is shown in the Table 6.5. Table 6.6 contains the *Precision*, *Recall*, *F1-Score* and *Accuracy* metrics.

Table 6.5: Validation Accuracy

CRSC	**W2V Google News**	**W2V Wikipedia**	**No Pre-Training**
Accuracy	0.4208	0.3841	0.2196

Table 6.6 shows the comparative analysis between NERM and the existing baselines.

Table 6.6: Evaluation results on the baseline models compared to the proposed approach.

Method	**Recall**	**Precision**	**F1 Score**	**Accuracy**
Random	0.0160	0.0220	0.0144	0.0234
Unigram	0.0270	0.0043	0.0074	0.1606
Bigram	0.2613	0.2944	0.2502	0.3793
Random Forest	0.2476	0.3663	0.2766	0.3299
Neural Entity/Relation model	0.3073	0.3281	0.3119	0.4208

Between the evaluated models, the Neural Entity/Relation Model achieved the highest F1 Score and Accuracy. The Bigram model achieved the second highest accuracy 0.3793 followed by Random forest model 0.3299. The `NERM` approach using *LSTM* provides an improvement of 9.86% on accuracy over the baselines, and 11.31% improvement on F1-score. Random Forest achieved the highest precision, while the Neural Entity/Relation Model achieved the highest recall.

TABLE 6.7: Extracted information from the Confusion Matrix - Part 1.

Relation	#Correctly Predicted	Rate of Correct Predictions[3]
NotIsA	2	1
AtLocation	172	0.67
NotDesires	6	0.666
Similar	5	0.625
Desires	36	0.593
HasPrerequest	23	0.547
CausesDesire	17	0.548
IsA	147	0.492
Antonym	68	0.492
InstanceOf	46	0.479
UsedFor	47	0.475
DesireOf	5	0.5
HasContext	2	0.5
HasLastSubevent	2	0.5
NotHasA	1	0.5
MemberOf	1	0.5
HasA	24	0.393
HasSubEvent	12	0.378
PartOf	16	0.374
HasPropertry	12	0.375
Synonym	54	0.312
DerivedFrom	20	0.307
EtymologicallyDerivedFrom	6	0.3
CapableOf	13	0.26
MotivationByGoal	3	0.25
ReceiveAction	5	0.238
CreatedBy	4	0.2
MadeOf	3	0.16
Causes	3	0.15
Genre	1	0.11

The confusion matrix is shown in Tables 6.7 and 6.8. These two tables are calculated based on the first version of the evaluation, in which we have $3,120$ examples for training, 551 for validation and $1,124$ for testing. In Table 6.7 the *'Correctly Predicted'* column indicates the proportion of relations that are predicted correctly, and *'Correct Prediction Rate'* column indicates the rate at which the relations correctly predicted. For instance, our model predicts the relation *'NotIsA'* correctly in 100% of the cases.

Table 6.8 shows the relations which are wrongly predicted (*'Wrongly Predicted'* columns). Based on the results, the most incorrectly predicted relation is *'IsA'*, which accounts for a large proportion of relations of the dataset (around 150 out

of 550). In the second place is the *'AtLocation'* relation (172 out of 550). In the third place is the *'antonym'* relation. On the other hand, some relations which are not correctly predicted can be treated as semantically equivalent to their prediction, whereby their correct assignment depends on modelling decisions in the relation schema. The same situation occurs for specialisation relations (e.g. *'EtymologicallyDerivedFrom'* and *'DerivedFrom'*). Another issue is the low occurrence of certain relations expressed in the dataset.

TABLE 6.8: Extracted information from the Confusion Matrix - Part 2.

Relation	# Correct Predicted	Rate	Wrong Relation 1	# Falsely Predicted for Relation 1	Wrong Relation 2	# Falsely Predicted for Relation 2	Wrong Relation 3	# Falsely Predicted for Relation 3
AtLocation	172	0.67	Antonym	20	UsedFor	17		
Desire	36	0.593	IsA	6	CapableOf	6	UsedFor	5
HasPrerequest	23	0.547	Synonym	4	Antonym	3	AtLocation	2
CausesDesire	17	0.548	UsedFor	7				
IsA	147	0.492	AtLocation	26	Antonym	22	InstanceOf	22
Antonym	68	0.492	IsA	17	AtLocation	9		
InstanceOf	46	0.479	IsA	27	AtLocation	8		
UsedFor	47	0.475	AtLocation	26	IsA	18		
HasA	24	0.393	Antonym	11	UsedFor	6		
HasSubEvent	12	0.378	Causes	5	Antonym	4		
PartOf	16	0.374	Synonym	12	Antonym	3	HasProperty	3
HasProperty	12	0.375	IsA	8				
Synonym	54	0.312	IsA	31	HasProperty	17	AtLocation	12
DerivedFrom	20	0.307	IsA	10	Synonym	8	Etymologically-DerivedFrom	8
Etymologically-DerivedFrom	6	0.3	DerivedFrom	6				
CapableOf	13	0.26	UsedFor	13	IsA	7		
MotivatedByGoal	3	0.25	Causes	3	HasSubEvent	2		
ReceiveAction	5	0.238	AtLocation	9	UsedFor	3		
CreatedBy	4	0.2	Antonym	6	IsA	5		
MadeOf	3	0.16	IsA	7	Antonym	3	HasA	2
Causes	3	0.15	CausesDesire	6	HasSubEvent	4	DerivedFrom	3

6.8 Enriching Relationships

Based on the results at Table 6.8, some relations which are not correctly predicted can be treated as semantically equivalent to their prediction. Table 6.9 contains a description of a set of merged relations (merging more specific relations into more abstract categories with similar semantic function).

Also to keep the datasets coherent, we eliminate vague relations, such as (*'RelatedTo'*. *'DistinctFrom'*, *'EtymologicallyRelatedTo'*) and relations implying negation, such as (*'Antonym'* and *'Not'*).
Table 6.10 shows the accuracy of `CSRC` (Composite Semantic Relation Classification) after merging relations that are semantically equivalent. Before merging similar relations, `OTHER` and `ALL` datasets contained 41 and 44 relations while after merging we have 18 and 21 relations, respectively.

TABLE 6.9: Merging similar relations with a more abstract relation.

Main Relation	Similar Relations
HasSubevent	HasFirstSubevent, HasLastSubevent, HasPrerequisite, Entails, MannerOf
Causes	MotivatedByGoal, CausesDesire
DerivedFrom	FormOf
SimilarTo	Synonym
IsA	InstanceOf, DefinedAs
LocatedNear	AtLocation, HasA, MadeOf, PartOf

TABLE 6.10: Accuracy before and after merging similar relations.

Word Embedding	Without Merging Relations at OTHER Set	Merged Similar Relations at OTHER Set	Merged Similar Relations at ALL Set
W2V Google News	0.42	0.64	0.73

6.9 Knowledge Base (KB) Embeddings

The last part of the evaluation concentrates on assessing the impact of Knowledge Base (KB) embeddings into the ERM (Entity-Relation Model).

6.9.1 Post-Processing Word Embeddings

Faruqui et al. (2014) proposed a graph-based learning technique to obtain higher quality word embeddings by using lexical relational resources such as *Wordnet* (Fellbaum, 2005), *Freebase* (Bollacker et al., 2008). This technique known as *retrofitting*, brings semantically similar words close together while keeping them (relatively) close to their initial distributional vectors. It is a post-processing approach, whereby we inject semantic constraints[4] into existing distributional vector spaces.

Speer, Chin, and Havasi (2017) introduced an ensemble method known as *ConceptNet-Numberbatch*, which combines data from pre-trained word embeddings and knowledge graphs, using a variation on retrofitting (Faruqui et al., 2014) to produce a high-quality word embeddings. They achieve this goal by applying the following method:

- Expanding the retrofitting algorithm Faruqui et al. (2014) to benefit from structured links outside the original vocabulary.
- Using *ConceptNet* (Speer and Havasi, 2012) as a resource of structured connections between words.

[4] A semantic constraint is an element that describes a concept or a relation between two concepts

- Merging two pre-trained DSMs (Word2Vec (Mikolov et al., 2013a) and Glove (Pennington, Socher, and Manning, 2014a)) using a local linear interpolation. This combination performs better than each of the models separately.
- Applying expanded retrofitting method on the combined vector space model by using *ConceptNet* as a lexical relational resource.

Speer, Chin, and Havasi (2017) called their proposed word embedding *ConceptNet-Numberbatch*, showing that the combined embedding outperforms *W2V* on word-similarity evaluations.

In this work, we used *ConceptNet-Numberbatch* DSM as a pre-trained embedding variation instead of W2V model. Table 6.11 shows the accuracy of the CSRC classification using the *ConceptNet-Numberbatch* word embedding.

TABLE 6.11: Applying *ConceptNet-Numberbatch* as a pre-trained embedding vector space model in the CSRC classification model.

Word Embedding	**Merged Similar Relations at `OTHER` Set**	**Merged Similar Relations at `ALL` Set**
ConceptNet Numberbatch	0.66	0.74

Speer et al. consider the data in ConceptNet as a symmetric matrix of association between words to apply the expanded retrofitting method. Therefore, they eliminate non-symmetric relations in ConceptNet and disregard these relation types to generate new word embeddings. We argue that in order to achieve a high quality semantic relation classification, all relations must be taken into account. Hence a more comprehensive approach is needed which includes knowledge about how both asymmetric and symmetric allowing us to inject all semantic constraints into existing word embeddings for completeness.

6.9.2 Embedding Entities and Relations

As a second KB embedding model, we experimented with *translation embedding methods* as a pre-trained word embedding method.
Bordes et al. (2013) proposed an energy-based model for learning low-dimensional embeddings of entities which is materialised into the $TransE$ model. Relationships are represented as translations in the embedding space. In other words, the basic idea behind behind the model is, in a triple set (h, r, t) that composes two entities $h, t \in E$ the set of entities and a relationship $r \in L$ (the set of relationships), the embedding of the entity t should be close to the embedding of the head entity h plus some vector that depends on the relationship r.

$$h + r \simeq t$$

To learn such embeddings, they minimise a margin-based ranking criterion over the training set (Yang et al., 2014), where the scoring function of $TransE$ is

$$-(2g_r^a(y_h, y_t) - 2g_r^b(y_h, y_t) + \left\|V_r\right\|_2^2)$$

where:

$$g_r^a(y_h, y_t) = A_r^T \begin{pmatrix} y_h \\ y_t \end{pmatrix} \quad and \quad g_r^b(y_h, y_t) = y_h^T B_r y_t$$

and A_r^T, B_r are relation-specific parameters and equal to $(V_r^T \quad -V_r^T)$ and I, respectively.

The main motivation of the translation-based parametrisation is the structure of the hierarchical relationships that are very common in KBs; therefore translations are the best and natural transformations for representing them. Their model relies on a reduced set of parameters as it learns only one low-dimensional vector for each entity and each relationship. The optimisation is carried out by stochastic gradient descent (using *minibatches*), and also the embedding vectors of the entities are normalised. $TransE$ has fewer parameters compared to other approaches, leading to a simplification of the training process and preventing under-fitting.

A new word embedding called *CTransNet* was built by applying *STransE* (Nguyen et al., 2016) on the *ConceptNet* semantic network (Speer and Havasi, 2012). We trained *STransE* with *ConceptNet-Numberbatch* pre-trained word vectors, size=300, l1 norm, margin=5 and learning rate=0.0005, nepoch=2000 using *ConceptNet V*5.5. Table 6.12 shows the accuracy of the CSRC classification using the *CTransNet* word embedding.

TABLE 6.12: Use of CTransNet as a pre-trained embedding vector space model in the CSRC classification model.

Word Embedding	**Merged Similar Relations at `OTHER` Set**	**Merged Similar Relations at `ALL` Set**
CTransNet	0.73	0.80

6.10 Final Results

In the previous sections, we investigated the influence of different types of embedding-based models for the task of composite semantic relation classification in the context of the Neural Entity-Relation Model (NERM). Three models were analysed: (1) traditional Word Vector embeddings (W2V), (2) Post-Processing Word Embeddings and (3) Embedding Models of Entities and Relations.

The results (Table 6.13) show that using the *CTransNet* word embedding outperforms the *W2V - Google News* and *ConceptNet-Numberbatch* word embedding on the composite semantic relation classification task.

TABLE 6.13: Comparison of accuracy scores of three types of Word Embeddings in our classification model (NERM).

Word Embedding	Without Merging Relations at OTHER Set	Merged Similar Relations at OTHER Set	Merged Similar Relations at ALL Set
W2V - Google News	0.42	0.64	0.73
ConceptNet Numberbatch	N/A	0.66	0.74
CTransNet	N/A	0.73	0.80

6.11 Summary

Below, the interpretation of the results are summarised as the core research question which we aim to answer with this section:

- **Question 1**: How distributional semantic relatedness models can be complemented with composite semantic relations?

  The Distributional semantic based filter (*Distributional Navigational Algorithm - DNA*) is one of components of the composite semantic relation classification. `DNA` (Described in Chapter 3, Section 3.4.3.3) method provides a semantic selection mechanism by using semantic relatedness models for eliminate non-relevant and meaningless facts and also coping with information incompleteness in large KBs. These meaningful facts can be used in the sequence machine learning model to address composite semantic relation classification.

- **Question 2**: How to evaluate composite semantic relation classification?

  For evaluating a classification model, a *test* set is needed besides *training* and *development* sets for training the sequence machine learning model. These datasets are extracted from the *SemEval* 2010 task 8 (Hendrickx et al., 2009) and *ConceptNet* (Speer and Havasi, 2012). Four measures *precision, recall,* f_1 *score and accuracy* have been applied for the evaluation of the results. These measures have been compared to other baseline models

such as *Random Forest* and Language Models. The results show that the `NERM` reached to accuracy 80 % for `All` set and 73 % for `OTHER` set by *CTransNet* word embedding.

Chapter 7

Epilogue

7.1 Summary & Conclusion

TABLE 7.1: Research Question related to Language Transportability

#	Research Question
1	How different distributional semantic models built from corpora in different languages and with different sizes perform in computing semantic relatedness similarity and relatedness tasks?
2	Does machine translation of non English to English perform better than the word vectors in the original language (for which languages and for which distributional semantic models)?
3	Which DSMs and languages benefit more and less from the translation
4	What is the quality of state-of-the-art machine translation approaches for word pairs (for each language)?
5	Can a lightweight MT model over an English DSM provide higher quality word vectors compared to native word vectors?
6	How does a lightweight MT model compares with state-of-the-art MT models?
7	Are there DSMs which are more/less robust with respect to the quality of the MT?

Chapter 3, Section 3.2 provides a comparative analysis of the performance of four state-of-the-art distributional semantic models over 11 languages (**RQ 1.1**), contrasting the native language-specific models with the use of machine translation over English-based DSMs. The experimental results show that there is a significant improvement (average of 18.4% for the Spearman correlation) by using off-the-shelf machine translation approaches (**RQ 1.2**) and that the benefit of using a more informative (English) corpus outweighs the possible errors introduced by the machine translation approach (**RQ 1.4**). The average accuracy of the machine translation approach (*Google*) is 62. Moreover, for all languages, *W2V* (**RQ 1.7**) showed consistently better results. For all languages , the combination of machine translation over the *W2V* English distributional model (**RQ 1.3**) provided the best results consistently (average Spearman correlation of 0.58). Section 3.2 also proposed the use of a *lightweight* Machine Translation (MT) model over an English Distributional Semantic Model (DSM) as an intermediate layer for the creation of high-quality multi-lingual distributional word vectors.

The results (**RQ 1.5**) show that the proposed model consistently outperforms native language DSMs for word pair similarity evaluation settings: *MC* (26.90%) , *RG* (16.73%), *WS353* (6.58%) and *SIMLEX* (10.26%). Additionally, the paper shows that the *lightweight* MT model is in the worst case equivalent and in some cases outperforms *state-of-the-art* MT systems for the translation of word pairs (**RQ 1.6**).

TABLE 7.2: Research Question related to Domain Transportability

#	Research Question
1	Which types of Distributional Semantic Models are most cost-effective with respect to smaller, domain-specific corpora?
2	What type of discourse expressed in small-scale corpora leads to better domain-specific Distributional Semantic Models (dsDSMs)?
3	How to quantify the factors which influence the performance of dsDSMs?

Chapter 3, Section 3.3 provides a detailed analysis of the performance of existing distributional semantic models (DSMs) under a domain-specific task. As an experimental setting, the financial domain of discourse was chosen. One central factor in the construction of domain specific DSMs include the scarcity of available corpora in comparison with open domain scenarios, and the demand for more fine-grained semantic relatedness measures.

The experimental analysis, performed using two financial word pair gold standard 1- containing 422 manually annotated word pairs (*SFWP-422*) 2- containing 3920 automatically annotated word pairs (*PFWP-3920*), analysed *domain-specific* discourse (*encyclopedic, glossary*) and *open domain (encyclopedic)* for the construction of the distributional models. The results show that for the *syntagmatic* dataset the best performing model (**RQ 1**) is *W2V* which generated by using *Finance-all* and *open domain encyclopedic* corpora (*FinS-CN*) ($\rho = 0.5783$). For the *paradigmatic* dataset, the combination of *ESA* and domain-specific discourse (encyclopedic, glossary) provide the best performing model ($\rho = 0.1946$). Encyclopedic corpora (**RQ 2**) generated from *Investopedia* and *wikipedia articles on Investment and Finance* leads to a better domain-specific Distributional Semantic Models, specially for *W2V* model. Based on the results, the predictive word embedding models (*W2V* and *GLOVE*) can capture better semantic information from a small-scale corpora. Furthermore by injecting semantic constraints[1] into existing distributional vector spaces, the combination of the trained *W2V* model

[1] A semantic constraint is an element that describes a concept or a relation between two concepts

on the *finance corpora* to the commonsense graph resource ConceptNet Version 5.5 (*FinS-CN*) led to best performance with respect dsDSMs (**RQ 3**).

TABLE 7.3: Research Question related to Composite Semantic Relation Classification

#	Research Question
1	How do we complement distributional semantic relatedness models with composite semantic relations?
2	How to evaluate composite semantic relation classification?

In Chapter 3, Section 3.4, we introduced the task of composite semantic relation classification. The paper proposes a composite semantic relation classification model which combines *commonsense KB lookup*, a *distributional semantics based filter* and the application of a *sequence-based machine learning model* to address the task.

The Distributional semantic based filter (*Distributional Navigational Algorithm - DNA*) is one of components of the composite semantic relation classification. *DNA* method provides a semantic selection mechanism by using semantic relatedness models to eliminate non-relevant and meaningless facts and also coping with information incompleteness in large KBs. These meaningful facts can be used in the sequence machine learning model to address composite semantic relation classification (**RQ 1**). For evaluating (**RQ 2**) a classification model, a *test* set is needed besides *training* and *development* sets for training the sequence machine learning model. Also the accuracy of *test* set has been compared to some other baseline models such as *Random Forest* and *Language Models*. The highest accuracy for the task of composite semantic relation classification was achieved by using Long Short-Term Memory - `LSTM` (Sak, Senior, and Beaufays, 2014) as the sequence-based models and translation-based embeddings via the (Neural Entity-Relation Model - `NERM` (described at Section 3, Section 3.4.3.4)). The proposed approach achieved 0.80 accuracy for the task at hand.

Many applications of word embedding models require the customisation of the models in the direction of domain-specific vocabularies, specific languages or specific semantic approximation behaviour (e.g. paradigmatic vs syntagmatic behaviour), distance measures as well as compositional models. In addition, currently available frameworks miss important aspects of the word embedding pipelines. To overcome these limitation, we proposed the transportable distributional semantic architecture (`INDRA` framework) which manages the complexity of experimenting and using word embedding models in exploratory scenarios and

production environments. INDRA also shares more than 65 pre-computed models and is available as an open-source software.

7.2 Future work

Future work on language transportability (Chapter 3, Section 3.2) will concentrate on the analysis of the suitability of lightweight MT approaches for computing compositional-distributional over phrasal elements. The current version only works on single words, but it is important to have language transportabilty for full sentences.

On domain trnasportability (Chapter 3, Section 3.3), future work will include 1) the investigation of the role of different parameterization strategies (e.g. context window sizes, weighting schemes and distance measures) in the performance of the models. 2) the study of generating domain specific relations embeddings. Furthermore, Multimodal - joint image (such as diagrams) and language applications are beginning to appear at NLP conferences; therefore one research question would be, how can we improve patent retrieval using domain specific distributional semantics and images?

ConceptNet is built from nodes representing words or short phrases of natural language and *abstract* relationships between them. Future work will focus on enriching the relations with syntactic information. One example is the Syntactic Ngrams dataset, which contains dependency tree fragments extracted from of the Google Books corpus (Goldberg and Orwant, 2013). It contains a diverse set of relations, with maximal significance on relations between words. The dataset corpus is based on 3.5 million English books (Over 10 billion distinct items). A Syntactic Ngram is a rooted connected dependency tree over n words. For each n words in a sentence, a POS-tag[2] and a basic dependency label for a given head word are provided. With this information, we can collect all SPO — Subject, Predicate, and Object — relations for each given word pairs for training our prediction model. Also we have a plan to compare our proposed LSTM model with other models such CNN. Finally additional baseline models such as SVM[3] beyond the existing baselines (Such as *Random, Unigram, Bigram, Random Forest*) will be also included.

[2] Penn-Treebank part of speech tag.
[3] Support Vector Machines.

Bibliography

Agirre, Eneko et al. (2009). "A study on similarity and relatedness using distributional and wordnet-based approaches". In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 19–27.

Al-Rfou, Rami, Bryan Perozzi, and Steven Skiena (2013). "Polyglot: Distributed Word Representations for Multilingual NLP". In: *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 183–192. URL: http://www.aclweb.org/anthology/W13-3520.

Aloni, Maria and Paul Dekker (2016). *The Cambridge Handbook of Formal Semantics*. Cambridge University Press.

Anastasakos, Tasos, Young-Bum Kim, and Anoop Deoras (2014). "Task specific continuous word representations for mono and multi-lingual spoken language understanding". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 3246–3250.

Atzori, Maurizio, Simone Balloccu, and Andrea Bellanti (2018). "Unsupervised Singleton Expansion from Free Tex". In: *2018 IEEE Twelfth International Conference on Semantic Computing (ICSC)*.

Auer, Sören et al. (2007). "Dbpedia: A nucleus for a web of open data". In: *The semantic web*, pp. 722–735.

Baroni, Marco et al. (2012). "Entailment above the word level in distributional semantics". In: *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 23–32.

Barzegar, Siamak et al. (2015). "DINFRA: A One Stop Shop for Computing Multilingual Semantic Relatedness". In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '15. Santiago, Chile: ACM, pp. 1027–1028. ISBN: 978-1-4503-3621-5. DOI: 10.1145/2766462.2767870. URL: http://doi.acm.org/10.1145/2766462.2767870.

Barzegar, Siamak et al. (2017). "Composite Semantic Relation Classification". In: *International Conference on Applications of Natural Language to Information Systems*. Springer, pp. 406–417.

Barzegar, Siamak et al. (2018a). "Multilingual Semantic Relatedness using lightweight machine translation". In: *IEEE International Conference on Semantic Computing*. IEEE.

Barzegar, Siamak et al. (2018b). "Multilingual Semantic Relatedness using lightweight machine translation". In: *Semantic Computing (ICSC), 2018 IEEE 12th International Conference on*. IEEE, pp. 108–114.

Barzegar, Siamak et al. (2018c). "SemR-11: A Multi-Lingual Gold-Standard for Semantic Similarity and Relatedness for Eleven Languages". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA).

Beyer, Kevin et al. (1999). "When is "nearest neighbor" meaningful?" In: *International conference on database theory*. Springer, pp. 217–235.

Biemann et al. (2013). "Text: Now in 2D! A Framework for Lexical Expansion with Contextual Similarity". In: *Journal of Language Modelling* 1.1, pp. 55–95.

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan, pp. 993–1022.

Boleda, Gemma and Aurélie Herbelot (2016). "Formal distributional semantics: Introduction to the special issue". In: *Computational Linguistics* 42.4, pp. 619–635.

Bollacker, Kurt et al. (2008). "Freebase: a collaboratively created graph database for structuring human knowledge". In: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. AcM, pp. 1247–1250.

Bond, Francis and Ryan Foster (2013). "Linking and Extending an Open Multilingual Wordnet." In: *ACL (1)*, pp. 1352–1362.

Bordes, Antoine et al. (2013). "Translating embeddings for modeling multi-relational data". In: *Advances in Neural Information Processing Systems*, pp. 2787–2795.

Bruni, Elia, Nam-Khanh Tran, and Marco Baroni (2014). "Multimodal distributional semantics." In: *J. Artif. Intell. Res.(JAIR)* 49.2014, pp. 1–47.

Camacho-Collados, José, Mohammad Taher Pilehvar, and Roberto Navigli (2015). "A framework for the construction of monolingual and cross-lingual word similarity datasets". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*. Citeseer, pp. 1–7.

Camacho-Collados, Jose et al. (2017). "Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity". In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017). Vancouver, Canada*.

Carvalho, Danilo S. et al. (2014). "EasyESA: A Low-effort Infrastructure for Explicit Semantic Analysis". In: *Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web*

Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014. Pp. 177–180. URL: http://ceur-ws.org/Vol-1272/paper_137.pdf.

Chen, Jinxiu et al. (2005). "Unsupervised feature selection for relation extraction". In: *Proceedings of IJCNLP*.

Clark, Stephen (2015). "Vector space models of lexical meaning". In: *The Handbook of Contemporary semantic theory*, pp. 493–522.

Clark, Stephen and Stephen Pulman (2007). "Combining symbolic and distributional models of meaning." In: *AAAI Spring Symposium: Quantum Interaction*, pp. 52–55.

Cortis, Keith et al. (2017). "Semeval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news". In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 519–535.

Curran, James Richard (2004). "From distributional to semantic similarity". In:

Dai, Andrew M and Quoc V Le (2015). "Semi-supervised sequence learning". In: *Advances in neural information processing systems*, pp. 3079–3087.

Davis, Brian et al. (2016). "Social sentiment indices powered by x-scores". In: *ALLDATA 2016*, p. 21.

Davis, Mark and Laurențiu Iancu (2017). *UNICODE TEXT SEGMENTATION*. Tech. rep. The Unicode Consortium. URL: http://unicode.org/reports/tr29/.

De Marneffe, Marie-Catherine, Bill MacCartney, Christopher D Manning, et al. (2006). "Generating typed dependency parses from phrase structure parses". In: *Proceedings of LREC*. Vol. 6. 2006. Genoa Italy, pp. 449–454.

Deerwester, Scott et al. (1990). "Indexing by latent semantic analysis". In: *Journal of the American society for information science* 41.6, pp. 391–407.

Dinu, Georgiana, Marco Baroni, et al. (2013). "Dissect-distributional semantics composition toolkit". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 31–36.

Dinu, Georgiana, Nghia The Pham, and Marco Baroni (2013). "DISSECT - DIStributional SEmantics Composition Toolkit". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 31–36. URL: http://www.aclweb.org/anthology/P13-4006.

Dumais, S. T. et al. (1988). "Using Latent Semantic Analysis to Improve Access to Textual Information". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '88. Washington, D.C., USA: ACM, pp. 281–285. ISBN: 0-201-14237-6. DOI: 10.1145/57167.57214. URL: http://doi.acm.org/10.1145/57167.57214.

Elman, Jeffrey L (1990). "Finding structure in time". In: *Cognitive science* 14.2, pp. 179–211.

Faruqui, Manaal and Chris Dyer (2014a). "Community Evaluation and Exchange of Word Vectors at wordvectors.org". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Baltimore, USA: Association for Computational Linguistics.

— (2014b). "Improving Vector Space Word Representations Using Multilingual Correlation". In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Gothenburg, Sweden: Association for Computational Linguistics, pp. 462–471. URL: http://www.aclweb.org/anthology/E14-1049.

Faruqui, Manaal et al. (2014). "Retrofitting word vectors to semantic lexicons". In: *arXiv preprint arXiv:1411.4166*.

Fellbaum, Christiane (2005). "WordNet and wordnets". In:

Finkelstein, Lev et al. (2001). "Placing search in context: The concept revisited". In: *Proceedings of the 10th international conference on World Wide Web*. ACM, pp. 406–414.

Freifeld, Clark C et al. (2008). "HealthMap: global infectious disease monitoring through automated classification and visualization of Internet media reports". In: *Journal of the American Medical Informatics Association* 15.2, pp. 150–157.

Freitas, A (2015a). "Schema-agnositc queries over large-schema databases: a distributional semantics approach". PhD thesis. PhD Thesis.

Freitas, André (2015b). "Schema-agnositc queries over large-schema databases: a distributional semantics approach." PhD thesis. Digital Enterprise Research Institute (DERI), National University of Ireland, Galway.

Freitas, Andre and Edward Curry (2014). "Natural language queries over heterogeneous linked data graphs: A distributional-compositional semantics approach". In: *Proceedings of the 19th international conference on Intelligent User Interfaces*. ACM, pp. 279–288.

Freitas, André, Edward Curry, and Siegfried Handschuh (2014). "Towards a Distributional Semantic Web Stack." In: *URSW*. Citeseer, pp. 49–52.

Freitas, André, Edward Curry, and Seán O'Riain (2012a). "A distributional approach for terminological semantic search on the linked data web". In: *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. ACM, pp. 384–391.

Freitas, Andre, Edward Curry, and Sean O'Riain (2012b). "Distributional Approach for Terminological Semantic Search on the Linked Data Web". In: *27th ACM Symposium On Applied Computing (SAC), Semantic Web and Applications (SWA)*.

Freitas, André, Siegfried Handschuh, and Edward Curry (2015). "Distributional-relational models: Scalable semantics for databases". In: *2015 AAAI Spring Symposium Series*.

Freitas, André and Joao C Pereira da Silva (2014). "Semantics at scale: When distributional semantics meets logic programming". In: *ALP Newsletter*.

Freitas, André et al. (2012). "Querying heterogeneous datasets on the linked data web: Challenges, approaches, and trends". In: *IEEE Internet Computing* 16.1, pp. 24–33.

Freitas, André et al. (2013a). "Distributional relational networks". In: *AAAI Fall Symposium Series*.

Freitas, André et al. (2013b). "Querying linked data graphs using semantic relatedness: A vocabulary independent approach". In: *Data & Knowledge Engineering* 88, pp. 126–141.

Freitas, André et al. (2013c). "Querying linked data graphs using semantic relatedness: A vocabulary independent approach". In: *Data Knowl. Eng.* 88, pp. 126–141. DOI: 10.1016/j.datak.2013.08.003. URL: https://doi.org/10.1016/j.datak.2013.08.003.

Freitas, André et al. (2014). "A distributional semantics approach for selective reasoning on commonsense graph knowledge bases". In: *Natural Language Processing and Information Systems*. Springer, pp. 21–32.

Freitas, André et al. (2016). "Semantic Relatedness for All (Languages): A Comparative Analysis of Multilingual Semantic Relatedness Using Machine Translation". In: *Knowledge Engineering and Knowledge Management: 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings 20*. Springer, pp. 212–222.

Furnas, George W. et al. (1987). "The vocabulary problem in human-system communication". In: *Communications of the ACM* 30.11, pp. 964–971.

Gabrilovich, Evgeniy and Shaul Markovitch (2007a). "Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis". In: *Proceedings of the 20th International Joint Conference on Artifical Intelligence*. IJCAI'07. Hyderabad, India: Morgan Kaufmann Publishers Inc., pp. 1606–1611. URL: http://dl.acm.org/citation.cfm?id=1625275.1625535.

— (2007b). "Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis". In: *Proceedings of the 20th International Joint Conference on Artifical Intelligence*. IJCAI'07. Hyderabad, India: Morgan Kaufmann Publishers Inc., pp. 1606–1611. URL: http://dl.acm.org/citation.cfm?id=1625275.1625535.

Gal, Yarin and Phil Blunsom (2013). "A systematic Bayesian treatment of the IBM alignment models". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 969–977.

Gangemi, Aldo et al. (2003). "Sweetening wordnet with dolce". In: *AI magazine* 24.3, p. 13.

Garcia-Duran, Alberto et al. (2016). "Combining Two and Three-Way Embedding Models for Link Prediction in Knowledge Bases". In: *Journal of Artificial Intelligence Research* 55, pp. 715–742.

Ghosh, Saurav et al. (2016). "Designing Domain Specific Word Embeddings: Applications to Disease Surveillance". In: *arXiv preprint arXiv:1603.00106*.

Girju, Roxana (2008). "Semantic relation extraction and its applications". In: *ESSLLI, Hamburg*.

Goldberg, Yoav and Jon Orwant (2013). "A Dataset of Syntactic-Ngrams over Time from a Very Large Corpus of English Books." In: ** SEM@ NAACL-HLT*, pp. 241–247.

Gouws, Stephan, Yoshua Bengio, and Greg Corrado (2015). "Bilbowa: Fast bilingual distributed representations without word alignments". In: *International Conference on Machine Learning*, pp. 748–756.

Grosz, Barbara J et al. (1987). "TEAM: an experiment in the design of transportable natural-language interfaces". In: *Artificial Intelligence* 32.2, pp. 173–243.

Harrell, Frank E (2015). "Ordinal logistic regression". In: *Regression modeling strategies*. Springer, pp. 311–325.

Harris, Zellig S (1954). "Distributional structure". In: *Word* 10.2-3, pp. 146–162.

Hasegawa, Takaaki, Satoshi Sekine, and Ralph Grishman (2004). "Discovering relations among named entities from large corpora". In: *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, p. 415.

Hearst, Marti A. et al. (1998). "Support vector machines". In: *IEEE Intelligent Systems and their applications* 13.4, pp. 18–28.

Hendrickx, Iris et al. (2009). "Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals". In: *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*. Association for Computational Linguistics, pp. 94–99.

Hengstler, Monika, Ellen Enkel, and Selina Duelli (2016). "Applied artificial intelligence and trust - The case of autonomous vehicles and medical assistance devices". In: *Technological Forecasting and Social Change* 105.Supplement C, pp. 105 –120. ISSN: 0040-1625. DOI: https://doi.org/10.1016/j.techfore.2015.12.014. URL: http://www.sciencedirect.com/science/article/pii/S0040162515004187.

Henriksson, Aron (2015). "Learning multiple distributed prototypes of semantic categories for named entity recognition". In: *International journal of data mining and bioinformatics* 13.4, pp. 395–411.

Henriksson, Aron et al. (2015). "Identifying adverse drug event information in clinical notes with distributional semantic representations of context". In: *Journal of biomedical informatics* 57, pp. 333–349.

Hill, Felix, Roi Reichart, and Anna Korhonen (2015). "Simlex-999: Evaluating semantic models with (genuine) similarity estimation". In: *Computational Linguistics*.

— (2016). "Simlex-999: Evaluating semantic models with (genuine) similarity estimation". In: *Computational Linguistics*.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.

Jans, Bram et al. (2012). "Skip n-grams and ranking functions for predicting script events". In: *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 336–344.

Jurgens, David and Keith Stevens (2010a). "The S-Space package: an open source package for word space models". In: *Proceedings of the ACL 2010 System Demonstrations*. Association for Computational Linguistics, pp. 30–35.

— (2010b). "The S-Space Package: An Open Source Package for Word Space Models". In: *Proceedings of the ACL 2010 System Demonstrations*. ACLDemos '10. Uppsala, Sweden: Association for Computational Linguistics, pp. 30–35. URL: http://dl.acm.org/citation.cfm?id=1858933.1858939.

Kambhatla, Nanda (2004). "Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations". In: *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*. Association for Computational Linguistics, p. 22.

Kartsaklis, Dimitri (2014). "Compositional Operators in Distributional Semantics". In: *Springer Science Reviews* 2.1, pp. 161–177. ISSN: 2213-7793. DOI: 10.1007/s40362-014-0017-z. URL: https://doi.org/10.1007/s40362-014-0017-z.

Katukuri, Jayasimhaa Reddy, Vijay Varadachari Raghavan, and Ying Xie (2013). *Semantic relationship extraction, text categorization and hypothesis generation*. US Patent 8,494,987.

Kiela, Douwe and Stephen Clark (2014). "A systematic study of semantic vector space model parameters". In: *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC) at EACL*, pp. 21–30.

Kim, Yoon (2014). "Convolutional neural networks for sentence classification". In: *arXiv preprint arXiv:1408.5882*.

Kong, Yun-Long et al. (2018). "Long Short-Term Memory Neural Networks for Online Disturbance Detection in Satellite Image Time Series". In: *Remote Sensing* 10.3, p. 452.

Lai, Siwei et al. (2015). "Recurrent Convolutional Neural Networks for Text Classification." In: *AAAI*. Vol. 333, pp. 2267–2273.

Lample, Guillaume et al. (2018). "Word translation without parallel data". In:

Landauer, Thomas K and Susan T Dumais (1997). "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge." In: *Psychological review* 104.2, p. 211.

Landauer, Thomas K, Peter W Foltz, and Darrell Laham (1998). "An introduction to latent semantic analysis". In: *Discourse processes* 25.2-3, pp. 259–284.

Lapesa, Gabriella and Stefan Evert (2014). "A large scale evaluation of distributional semantic models: Parameters, interactions and model selection". In: *Transactions of the Association of Computational Linguistics* 2.1, pp. 531–545.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *nature* 521.7553, p. 436.

Lehmann, Jens and Johanna Voelker (2014). "An introduction to ontology learning". In: *Perspectives on Ontology Learning. IOS Press, Amsterdam, The Netherlands*.

Lenci, Alessandro (2008). "Distributional semantics in linguistic and cognitive research". In: *Italian journal of linguistics* 20.1, pp. 1–31.

Leviant, Ira and Roi Reichart (2015). "Separated by an un-common language: Towards judgment language informed vector space modeling". In: *arXiv preprint arXiv:1508.00106*.

Lin, Jianhua (1991). "Divergence measures based on the Shannon entropy". In: *IEEE Transactions on Information theory* 37.1, pp. 145–151.

Lin, Yankai et al. (2015). "Learning entity and relation embeddings for knowledge graph completion." In: *AAAI*. Vol. 15, pp. 2181–2187.

Liu, Yang et al. (2015). "A dependency-based neural network for relation classification". In: *arXiv preprint arXiv:1507.04646*.

Loebbecke, Claudia and Arnold Picot (2015). "Reflections on societal and business model transformation arising from digitization and big data analytics: A research agenda". In: *The Journal of Strategic Information Systems* 24.3, pp. 149 –157. ISSN: 0963-8687. DOI: https://doi.org/10.1016/j.jsis.2015.08.002. URL: http://www.sciencedirect.com/science/article/pii/S0963868715000372.

Lund, Kevin and Curt Burgess (1996). "Producing high-dimensional semantic spaces from lexical co-occurrence". In: *Behavior research methods, instruments, & computers* 28.2, pp. 203–208.

Maia, Macedo et al. (2018). "WWW'18 Open Challenge: Financial Opinion Mining and Question Answering". In: *Companion of the The Web Conference 2018 on The Web Conference 2018*. International World Wide Web Conferences Steering Committee, pp. 1941–1942.

Malandrakis, Nikolaos et al. (2013). "Distributional semantic models for affective text analysis". In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.11, pp. 2379–2392.

Masolo, Claudio et al. (2003). "Wonderweb deliverable d18, ontology library (final)". In: *ICT project* 33052.

Mehdad, Yashar, Alessandro Moschitti, and Fabio Massimo Zanzotto (2010). "Syntactic/semantic structures for textual entailment recognition". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 1020–1028.

Mikolov, Tomáš et al. (2010). "Recurrent neural network based language model". In: *Eleventh Annual Conference of the International Speech Communication Association*.

Mikolov, Tomas et al. (2013a). "Efficient Estimation of Word Representations in Vector Space". In: *ICLR Workshop Papers*.

— (2013b). "Efficient estimation of word representations in vector space". In: *ICLR Workshop*.

Miller, George A (1995). "WordNet: a lexical database for English". In: *Communications of the ACM* 38.11, pp. 39–41.

Miller, George A and Walter G Charles (1991). "Contextual correlates of semantic similarity". In: *Language and cognitive processes* 6.1, pp. 1–28.

Miller, George A et al. (1990). "Introduction to WordNet: An on-line lexical database". In: *International journal of lexicography* 3.4, pp. 235–244.

Mitchell, Jeff and Mirella Lapata (2008). "Vector-based models of semantic composition". In: *proceedings of ACL-08: HLT*, pp. 236–244.

— (2010). "Composition in Distributional Models of Semantics". In: *Cognitive Science* 34.8, pp. 1388–1429. ISSN: 1551-6709. DOI: 10.1111/j.1551-6709.2010.01106.x. URL: http://dx.doi.org/10.1111/j.1551-6709.2010.01106.x.

Moen, Hans et al. (2015). "Care episode retrieval: distributional semantic models for information retrieval in the clinical domain". In: *BMC medical informatics and decision making* 15.2, p. 1.

Molino, Piero et al. (2012). "Exploiting distributional semantic models in question answering". In: *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*. IEEE, pp. 146–153.

Moro, Sérgio, Paulo Cortez, and Paulo Rita (2015). "Business intelligence in banking: A literature analysis from 2002 to 2013 using text mining and latent Dirichlet allocation". In: *Expert Systems with Applications* 42.3, pp. 1314 – 1324. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2014.09.024. URL: http://www.sciencedirect.com/science/article/pii/S0957417414005636.

Moussallem, Diego et al. (2017). "MAG: A Multilingual, Knowledge-base Agnostic and Deterministic Entity Linking Approach". In: *Proceedings of the Knowledge Capture Conference*. ACM, p. 9.

Mrkšić, Nikola et al. (2016). "Counter-fitting word vectors to linguistic constraints". In: *arXiv preprint arXiv:1603.00892*.

Nastase, Vivi et al. (2013). "Semantic relations between nominals". In: *Synthesis lectures on human language technologies* 6.1, pp. 1–119.

Navigli, Roberto and Simone Paolo Ponzetto (2012). "BabelRelate! A Joint Multilingual Approach to Computing Semantic Relatedness". In: *AAAI Conference on Artificial Intelligence*.

Nguyen, Dat Quoc et al. (2016). "STransE: a novel embedding model of entities and relationships in knowledge bases". In: *arXiv preprint arXiv:1606.08140*.

Nguyen, Thien Huu and Ralph Grishman (2015). "Combining Neural Networks and Log-linear Models to Improve Relation Extraction". In: *arXiv preprint arXiv:1511.05926*.

Pearl, Judea and Elias Bareinboim (2014). "External validity: From do-calculus to transportability across populations". In: *Statistical Science*, pp. 579–595.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning (2014b). "GloVe: Global Vectors for Word Representation". In: *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.

Pennington, Jeffrey, Richard Socher, and Christopher D Manning (2014a). "Glove: Global vectors for word representation". In: *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)* 12, pp. 1532–1543.

Pichotta, Karl and Raymond J Mooney (2016). "Using Sentence-Level LSTM Language Models for Script Inference". In: *arXiv preprint arXiv:1604.02993*.

Qin, Pengda, Weiran Xu, and Jun Guo (2016). "An Empirical Convolutional Neural Network approach for Semantic Relation Classification". In: *Neurocomputing*.

Quinlan, J. Ross (1986). "Induction of decision trees". In: *Machine learning* 1.1, pp. 81–106.

Ramkumar, A Sudha and B Poorna (2014). "Ontology Based Semantic Search: An Introduction and a Survey of Current Approaches". In: *Intelligent Computing Applications (ICICA), 2014 International Conference on*. IEEE, pp. 372–376.

Rehurek, R and Petr Sojka (2011). *Gensim–Statistical Semantics in Python*.

Rehurek, Radim (2014). *Performance shootout of nearest neighbours: Querying*.

Řehůřek, Radim (2014). *Performance Shootout of Nearest Neighbours: Querying*. https://rare-technologies.com/performance-shootout-of-nearest-neighbours-querying/.

Řehůřek, Radim and Petr Sojka (2010). "Software Framework for Topic Modelling with Large Corpora". English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. http://is.muni.cz/publication/884893/en. Valletta, Malta: ELRA, pp. 45–50.

Resnik, Philip (1995). "Using information content to evaluate semantic similarity in a taxonomy". In: *arXiv preprint cmp-lg/9511007*.

Resnik, Philip et al. (1999). "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language". In: *J. Artif. Intell. Res.(JAIR)* 11, pp. 95–130.

Ristoski, Petar et al. (2018). "RDF2Vec: RDF graph embeddings and their applications". In: *Semantic Web* Preprint, pp. 1–32.

Rubenstein, Herbert and John B Goodenough (1965). "Contextual correlates of synonymy". In: *Communications of the ACM* 8.10, pp. 627–633.

Ruder, Sebastian (2017). *Transfer learning: Machine learning's next frontier.*

Ruder, Sebastian, Ivan Vulić, and Anders Søgaard (2017). "A survey of cross-lingual word embedding models". In: *arXiv preprint arXiv:1706.04902.*

Sahlgren, Magnus (2005). "An introduction to random indexing". In: *Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering, TKE.* Vol. 5.

— (2006). "The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces". PhD thesis.

— (2008). "The distributional hypothesis". In: *Italian Journal of Disability Studies* 20, pp. 33–53.

Sak, Haşim, Andrew Senior, and Françoise Beaufays (2014). "Long short-term memory recurrent neural network architectures for large scale acoustic modeling". In: *Fifteenth annual conference of the international speech communication association.*

Sales, Juliano Efson, Andre Freitas, and Siegfried Handschuh (2018). "An Open Vocabulary Semantic Parser for End-User Programming using Natural Language". In: *2018 IEEE Twelfth International Conference on Semantic Computing (ICSC).*

Sales, Juliano Efson et al. (2016). "A Compositional-Distributional Semantic Model for Searching Complex Entity Categories". In: *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics (*SEM)*, pp. 199–208.

Sales, Juliano Efson et al. (2018). "Indra: A Word Embedding and Semantic Relatedness Server". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).* Miyazaki, Japan: European Language Resources Association (ELRA).

Santos, Cıcero Nogueira dos, Bing Xiang, and Bowen Zhou (2015a). "Classifying relations by ranking with convolutional neural networks". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing.* Vol. 1, pp. 626–634.

Santos, Cicero Nogueira dos, Bing Xiang, and Bowen Zhou (2015b). "Classifying relations by ranking with convolutional neural networks". In: *arXiv preprint arXiv:1504.06580.*

Schütze, Hinrich (1998). "Automatic word sense discrimination". In: *Computational linguistics* 24.1, pp. 97–123.

Schütze, Hinrich and Jan O Pedersen (1995). "Information retrieval based on word senses". In:

Shalaby, Walid et al. (2016). "Entity Type Recognition using an Ensemble of Distributional Semantic Models to Enhance Query Understanding". In: *arXiv preprint arXiv:1604.00933*.

Shi, Baoxu and Tim Weninger (2017). "Open-World Knowledge Graph Completion". In: *arXiv preprint arXiv:1711.03438*.

Silva, V, S Handschuh, and A Freitas (2018a). "Recognizing and Justifying Text Entailment through Distributional Navigation on Definition Graphs". In: AAAI.

Silva, Vivian S, Siegfried Handschuh, and Andre Freitas (2018b). "Recognizing and Justifying Text Entailment through Distributional Navigation on Definition Graphs". In: *Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI.

Silva, Vivian S et al. (2018). "Semantic relation classification: task formalisation and refinement". In: *arXiv preprint arXiv:1806.07721*.

Singh, Push et al. (2002). "The public acquisition of commonsense knowledge". In: *Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*.

Socher, Richard et al. (2012). "Semantic compositionality through recursive matrix-vector spaces". In: *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*. Association for Computational Linguistics, pp. 1201–1211.

Socher, Richard et al. (2013a). "Reasoning with neural tensor networks for knowledge base completion". In: *Advances in Neural Information Processing Systems*, pp. 926–934.

Socher, Richard et al. (2013b). "Recursive deep models for semantic compositionality over a sentiment treebank". In: *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642.

Soricut, Radu and Franz Och (2015). "Unsupervised morphology induction using word embeddings". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1627–1637.

Speer, Robert, Joshua Chin, and Catherine Havasi (2017). "ConceptNet 5.5: An Open Multilingual Graph of General Knowledge." In: *AAAI*, pp. 4444–4451.

Speer, Robert and Catherine Havasi (2012). "Representing General Relational Knowledge in ConceptNet 5." In: *LREC*, pp. 3679–3686.

Stevenson, Mark and Yorick Wilks (2003). "Word sense disambiguation". In: *The Oxford Handbook of Comp. Linguistics*, pp. 249–265.

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le (2014). "Sequence to sequence learning with neural networks". In: *Advances in neural information processing systems*, pp. 3104–3112.

Tang, Duyu, Bing Qin, and Ting Liu (2015). "Document modeling with gated recurrent neural network for sentiment classification". In: *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 1422–1432.

Team, DJD (2016). "Deeplearning4j: Open-source distributed deep learning for the jvm". In: *Apache Software Foundation License* 2.

Tiedemann, Jörg (2012). "Parallel Data, Tools and Interfaces in OPUS." In: *LREC*. Vol. 2012, pp. 2214–2218.

Turney, Peter D and Patrick Pantel (2010a). "From frequency to meaning: Vector space models of semantics". In: *Journal of artificial intelligence research* 37, pp. 141–188.

Turney, Peter D. and Patrick Pantel (2010b). "From Frequency to Meaning: Vector Space Models of Semantics". In: *J. Artif. Int. Res.* 37.1, pp. 141–188. ISSN: 1076-9757. URL: http://dl.acm.org/citation.cfm?id=1861751.1861756.

Utt, Jason and Sebastian Padó (2014). "Crosslingual and Multilingual Construction of Syntax-Based Vector Space Models". In: *Transactions of the Association of Computational Linguistics* 2, pp. 245–258.

Wang, Peng et al. (2016). "Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification". In: *Neurocomputing* 174, pp. 806–814.

Wang, Zhongyuan et al. (2015). "An inference approach to basic level of categorization". In: *Proceedings of the 24th acm international on conference on information and knowledge management*. ACM, pp. 653–662.

Wu, Yonghui et al. (2016). "Google's neural machine translation system: Bridging the gap between human and machine translation". In: *arXiv preprint arXiv:1609.08144*.

Wu, Zhibiao and Martha Palmer (1994). "Verbs semantics and lexical selection". In: *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 133–138.

Xu, Kun et al. (2015a). "Semantic relation classification via convolutional neural networks with simple negative sampling". In: *arXiv preprint arXiv:1506.07650*.

Xu, Yan et al. (2015b). "Classifying relations via long short term memory networks along shortest dependency paths". In: *Proceedings of Conference on Empirical Methods in Natural Language Processing (to appear)*.

Xu, Yan et al. (2016). "Improved Relation Classification by Deep Recurrent Neural Networks with Data Augmentation". In: *arXiv preprint arXiv:1601.03651*.

Yang, Bishan et al. (2014). "Embedding entities and relations for learning and inference in knowledge bases". In: *arXiv preprint arXiv:1412.6575*.

Zelenko, Dmitry, Chinatsu Aone, and Anthony Richardella (2003). "Kernel methods for relation extraction". In: *Journal of machine learning research* 3.Feb, pp. 1083–1106.

Zeng, Daojian et al. (2014). "Relation Classification via Convolutional Deep Neural Network." In: *COLING*, pp. 2335–2344.

Zesch, Torsten, Christof Müller, and Iryna Gurevych (2008). "Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary." In: *LREC*. Vol. 8. 2008, pp. 1646–1652.

Zou, Will Y et al. (2013). "Bilingual Word Embeddings for Phrase-Based Machine Translation." In: *EMNLP*, pp. 1393–1398.