An Eigenvalue-Based Measure for Word-Sense Disambiguation

An Eigenvalue-Based Measure for Word-Sense Disambiguation

Ioana Hulpuş, Conor Hayes and **Marcel Karnstedt**
Digital Enterprise Research Institute
National University of Ireland, Galway, Ireland
{ioana.hulpus, conor.hayes, marcel.karnstedt}@deri.org

Derek Greene
School of Computer Science and Informatics
University College Dublin, Dublin, Ireland
derek.greene@ucd.ie

Abstract

Current approaches for word-sense disambiguation (WSD) try to relate the senses of the target words by optimizing a score for each sense in the context of all other words' senses. However, by scoring each sense separately, they often fail to optimize the relations between the resulting senses. We address this problem by proposing a HITS-inspired method that attempts to optimize the score for the entire sense combination rather than one-word-at-a-time. We also exploit word-sense disambiguation via topic-models, when retrieving senses from heterogeneous sense inventories. Although this entails the relaxation of several assumptions behind current WSD algorithms, we show that our proposed method E-WSD achieves better results than current state-of-the-art approaches, without the need for additional background knowledge.

Introduction

Word-sense disambiguation (WSD) chooses the correct sense of a word from a set of different possible senses. The set of possible senses are hereby usually obtained from one knowledge base, and the choice of the right sense is based on the observed context of the word. WSD plays an important role for a wide set of applications, such as information retrieval & Web search, entity resolution, concept recognition, etc. WSD is formally defined as follows (Navigli 2009). Given n related words $\{w_1, w_2, ..., w_n\}$, called **target words**, each target word w_i has m_i possible senses, denoted $w_i\#j, i \in \{1, ..., n\}$ and $j \in \{1, ..., m_i\}$, such that $w_i\#j$ denotes the j^{th} sense of the target w_i. The problem is to find, for each target word w_i, the appropriate sense $w_i\#*$, in the context of all the other target words $w_j, j \neq i$. This general WSD problem is usually extended by assuming that, for each sense $w_i\#j$, we also have a bag of auxiliary **related words** ω_{ij}^k, where ω_{ij}^k denotes the k^{th} word related to the sense $w_i\#j$.

Current WSD algorithms agree upon the intuition that, for a particular context, the correct senses of all target words should be related. The sense of a target word is usually chosen based on the connectivity of that sense to *all* the possible senses of all other target words and no measure of relatedness among the final chosen senses is computed. As we illustrate in this work, this results in problems in certain cases. In addition, modern web technologies make more and more heterogeneous sense inventories publicly available. This availability of diverse background knowledge is not exploited by most WSD algorithms that are highly-tailored for single inventories.

The contribution of this paper is three-fold. First, we propose an eigenvalue-based measure, E-WSD, inspired by HITS (Kleinberg 1999), which computes the relatedness within combinations of word senses to overcome the aforementioned drawbacks. We show that this method is suited to identify the optimal combination of senses without the need for any additional background knowledge. This results in an improved accuracy of WSD compared to current state of the art algorithms that rely on similar background knowledge, but select the right sense for one word at a time. Second, we base our approach for WSD on two different inventories: WordNet(Miller 1995) and DBPedia[1]. We show that the inclusion of DBPedia results in a significant gain in accuracy compared to the traditionally exclusively used WordNet.

The third novel aspect of our work is that we exploit word-sense disambiguation via topic-models (Steyvers and Griffiths 2007). This helps to reduce the complexity of WSD tasks for large text corpora. Although topic models have been used for document representation and extraction of related words for many years, word-sense disambiguation focuses mainly on disambiguation of words in the context of phrases. We show that the context created by a topic can be used to improve disambiguation performance, by amplifying the relatedness between target words appearing in the same context. Our evaluations are performed on ground truth data collected via a two week crowd-sourcing experiment.

The use of topics and heterogeneous sense sources entails the relaxation of several assumptions behind current WSD algorithms, and consequently the emergence of new challenges. First, there is no unified structure to connect the senses. Second, within topics it is not clear what part of speech the words have been used with. Third, the sense source might be intractable or require many resources for pre-processing. We believe that these challenges have to be faced by modern WSD algorithms in the current Web context, where more and more data and data sources are made available.

[1] http://dbpedia.org

Related Work

The unsupervised WSD systems can be split into two categories: The first type of system usually extracts a lexical knowledge base (LKB) from a sense inventory containing semantic representations of all the word senses *before* the actual task of WSD (Agirre and Soroa 2009; Navigli and Velardi 2005; Mendes et al. 2011). These approaches often suffer from lack of portability, as LKB creation is dependent on the structure, availability and scale of the sense inventory.

Our approach is closer to the second type of system, which uses on-the-fly queries on sense inventories. One of the most popular approaches is based on maximizing the number of overlapping words in the definitions (Lesk 1986). Recently, Anaya-Sánchez *et al.* (2009) represented all the possible senses as topic signatures, and then applied an extended star clustering algorithm to find groups of similar senses. Then, the clusters are iteratively scored and ordered based on the number of words they disambiguate, context overlap and WordNet sense frequency. We envisage a similar workflow where the clusters are scored using the eigenvalue-based measure proposed in this paper.

Closer to our approach, Tsatsaronis *et al.* (2007) used a strategy based on applying spreading activation to a weighted graph where the target words are initially linked to their possible senses in WordNet. However, where multiple sense inventories are used, the activation network may be disconnected or related senses from different inventories may never activate each other. The approach also relies on the assumption that the sense inventory provides relations among senses. Our approach relaxes these requirements.

Work by Sinha & Mihalcea (2007) is most closely aligned with our proposed approach. The authors present a graph-based WSD framework, which can accommodate any type of measure of relatedness between senses and any centrality measure to determine the most connected senses. However, this approach optimizes each individual sense selection, not the joint choice of senses. Consequently, there is the risk that selected senses will not be related to each other, breaking a main assumption behind WSD.

Most unsupervised WSD approaches naturally tend to prefer senses with longer definitions, as there are more words with which they can overlap. We overcome this bias by weighting related words based on the number of senses they occur in. Thus, a term occurring in several definitions will receive more emphasis than a term that occurs in two definitions alone. This idea strongly resonates with the HITS algorithm. Also, we propose a novel weighting scheme penalizing senses that relate to many named entities without sharing them with other senses.

Motivation

We illustrate the main issues addressed in this paper in the toy example shown in Table 1, where we want to disambiguate three target words: {*web, internet, page*}. It is not hard for humans to determine that the correct meanings for the two ambiguous targets *web* and *page* are *web#1* and *page#1* respectively. One important evidence is that they share the related word *computer*. However, when this knowledge alone is available, current graph-based WSD algorithms from (Sinha and Mihalcea 2007) will generate the graph in Figure 1. Originally, the edges are weighted with the number of overlapping words between the senses. In our toy example all these weights equal to 1, so we omit them from the figure. The score for each sense is computed as the degree of the corresponding node, and appears in the figure under the sense representation.

Target	Candidate	Related words
web	*web#1*	computer, network, application
	web#2	feather, net, flat, part
	web#3	world, english, bible, work, public
internet	*internet#1*	computer, connected, http, net
page	*page#1*	computer, media, public
	page#2	paper, flat, side
	page#3	boy, work, knight, part

Table 1: A toy example for three target words, with a set of candidate meanings and their respective related words.

This example helps us to illustrate three drawbacks of this and most other approaches that handle WSD in one-word-at-a-time manner. Firstly, inspecting the disambiguation of the word *web*: the score for *web#2* is 3, as it accumulates the degree score from the sense *internet#1*, *page#2*, and *page#3*. But *page#2* and *page#3* are mutually exclusive, since an instance of the word *page* cannot have both senses at the same time. Therefore, their connection to the other target words' senses should be treated separately. The same problem causes *page#3* to achieve a score of 2.

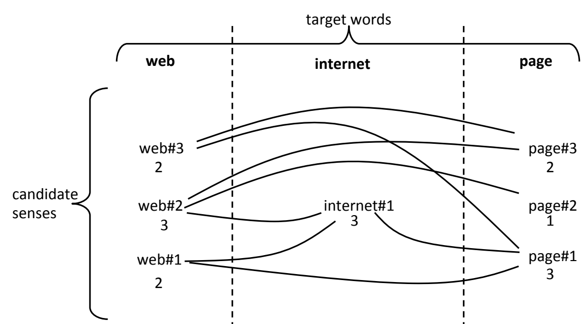


Figure 1: Graph generated by current graph-based WSD algorithms on the toy disambiguation example in Table 1.

The second problem is encountered when bringing together the local disambiguation solutions. For the word *web*, it will return the sense *web#2*, while for the word *page* it will return *page#1*. While the two senses accumulate the highest scores due to their connectivity to other senses, they are not related to each other by any connection. But the main assumption behind word-sense disambiguation is that the target words should have been used with related senses. Therefore, we argue for a disambiguation approach that aims to optimize the overall score among all disambiguated senses, rather than optimize a local score for each sense.

The third problem that we identify is that, while the sense *web#1* shares the term "computer" with both *internet#1* and

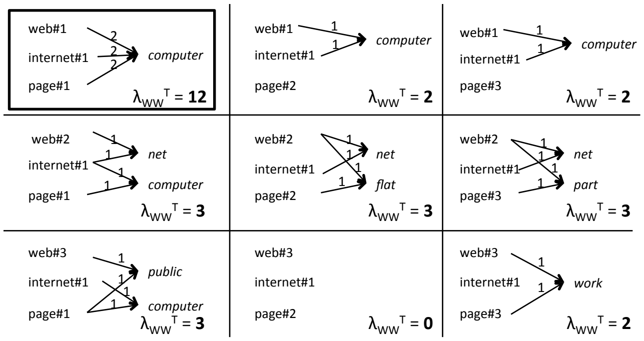


Figure 2: Bipartite graphs generated by each combination of meanings in the toy disambiguation example in Table 1, with the corresponding eigenvalue scores (λ).

page#1, this ternary relation is ignored. This is due to the fact that current approaches do not represent the related words as nodes of the graph, therefore ignoring important knowledge. Indeed, in this case the system would choose the correct meaning for word *page*, but due to a lucky coincidence, and it would fail to give a reasonable explanation to the user for the choice.

These three problems can be avoided by (i) considering the related words as components of the graph structure, therefore being able to weight them accordingly; (ii) treating each meaning combination separately, therefore avoiding problems due to score accumulation from mutually exclusive senses and problems due to a poor local optima.

Eigenvalue-Based WSD (E-WSD)

We propose an approach inspired by the weighted variation of the HITS algorithm (Kleinberg 1999; Bharat and Henzinger 1998), which uses the principal eigenvalue of the hubs matrix[2] as a score for each possible sense combination. By scoring the whole disambiguation graph with this eigenvalue, our proposed approach solves the WSD task by concurrent disambiguation of all words in the topic at the same time, rather than one-word-at-a-time disambiguation.

To avoid the problems identified in the previous section, our approach evaluates each possible joint disambiguation, and selects the one with the highest score. This ensures that the selected senses will be strongly inter-related. The approach makes use of the related words not just to compute the similarity between senses, but as stand alone entities. By considering the relations between the senses and the related words, a bipartite graph can be built. Since related words belonging to a single meaning cannot be used in a measure of similarity between meanings, we only represent target words which belong to at least two meanings. In this way, each combination of meanings can be represented by a bipartite graph, as shown in the example in Figure 2.

Let W be the adjacency matrix corresponding to the bipartite weighted graph of the current sense combination, where the entry W_{ij} is the weight of the edge going from the current sense of word w_i to the related word ω_j. The strength of the relation between two senses can be measured by computing their inner product. Mathematically, this is expressed by $R_{ij} = \sum_k W_{ik}W_{jk} = (WW^{\intercal})_{ij}, i \neq j$, where $WW^{\intercal}$ is called the hubs matrix. As for the case when $i = j$, the importance of a sense to the joint disambiguation graph can be interpreted as the magnitude of its vector $m_i = \sqrt{\sum_k W_{ik}}$. Then, we can define the diagonal matrix $M = diag({m_1}^2, {m_2}^2, ..., {m_n}^2)$, resulting in $WW^{\intercal} = R + M$. Therefore the hubs matrix represents a very simple relation between the relatedness of two senses and their magnitudes. A similar intuition can be built for the authorities matrix $W^{\intercal}W$.

When W represents the weighted adjacency matrix of a bipartite graph, the principal eigenvalue λ of $WW^{\intercal}$ corresponds to the most densely linked collection of hubs and authorities (Kleinberg 1999). Similarly, in the WSD setting, the principal eigenvalue will define the most densely linked network of meanings and words. Therefore, the magnitude of the principal eigenvalue of the $WW^{\intercal}$ matrix can be used as a score for a meaning-word network. We can then select the best word sense combination as the graph whose hub and authority matrices have the highest principal eigenvalue.

Assigning Weights to Edges

The edges in our disambiguation bipartite graph point from senses to their related words. The eigenvalue of the hubs matrix of the graph will depend only on the adjacency matrix, therefore on the edge weights. In order to weight the senses or the related words, the weights of the connecting edges must be adjusted accordingly.

Our disambiguation scoring scheme starts from the assumption that important related words should belong to the bag of words of many important senses. This is ensured by computing the eigenvalue based score as shown in the previous section. Furthermore, we consider named entities (Finkel, Grenager, and Manning 2005) as a key indicator for the importance of a sense in the joint disambiguation. If a sense definition refers to many named entities, and no other sense in the joint disambiguation relates to the same named entities, we consider that sense too specific and penalize it. A straightforward penalty computation is:

$$p_i = \begin{cases} \frac{|NE_i \cap \bigcup_{j \neq i} NE_j|}{|NE_i|} & \text{if } |NE_i| \geq t \\ 0 & \text{otherwise} \end{cases}$$

where NE_i denotes the set of named entities belonging to the bag of words of sense i, and t is a predefined threshold. In our experiments we set $t = 4$.

If we consider the weight of a sense as $w_i = 1 - p_i$, then we compute the weight of an edge as

$$w_{ik} = w_i \sum_{j \in S(k),\ j \neq i} w_j;$$

where w_{ik} denotes the weight of the edge going from sense i to related word k, and $S(k)$ represents the set of senses which are connected to the related word k.

[2] The principal eigenvalue of the hubs matrix is always equal to the principal eigenvalue of the authorities matrix. In this paper we will refer to it as the eigenvalue of the hubs matrix.

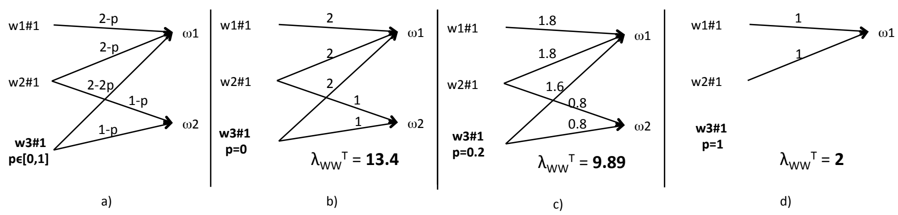


Figure 3: A set of four examples illustrating how sense penalties influence the weights of graph edges.

Figure 3 shows examples of how sense penalties influence the weights of graph edges. Figure 3(a) shows how the penalty for the sense $w3\#1$ is propagated to the edges in the graph. Figure 3(b) shows the case when there is no penalty, while in Figure 3(c) the sense is penalized with 0.2. Finally, Figure 3(d) shows the case when the penalty is 1, effectively removing the sense from the computation.

An important aspect which makes this strategy applicable in our approach is that a sense will only be penalized in a group of senses it shares no entities with. At the same time, in another combination of senses, the same sense might be better connected and not attract any penalty. Note that this egde weighting scheme is generally applicable with other heuristics for sense penalizing.

The E-WSD Algorithm

Given the target words to disambiguate and the bag of words representing each possible sense of all words, the proposed E-WSD algorithm involves the following steps:

1. Generate all possible combinations of senses. For each sense combination, execute steps 2 and 3 below.
2. Create the weighted bipartite graph, and the corresponding adjacency matrix W relating senses to the related words, where edges are weighted as described previously.
3. Compute the score of the combination as the principal eigenvalue of the square matrix WW^{T}.
4. Assign each target word to the corresponding sense in the combination with the highest score.

The generated bipartite graphs are very small, so that steps 2 and 3 do not result in significant computational overhead. The overall quality improvements come with the overhead of processing all possible combinations. We come back to these issues and possible optimizations in the Discussion and Conclusions section.

Evaluation

To test E-WSD, we attempted to link the target words occurring in topics with either DBPedia concepts or WordNet concepts. In order to retrieve possible meanings from DBPedia, we used the DBPedia look-up service, YAGO(Hoffart et al. 2011) and the DBPedia SPARQL endpoint. For each retrieved DBPedia resource, we also collect the DBPedia categories and classes it belongs to. From WordNet we retrieve the definition, synonyms, hypernyms, and meronyms, if available.

All the data collected about each sense are then passed through a text processing phase consisting of stopword removal, word stemming, 2-gram and 3-gram phrase extraction, and named entity extraction (Finkel, Grenager, and Manning 2005). This results in a bag-of-words representation for each sense. We then applied an implementation of the E-WSD algorithm to our data.

As a state-of-the-art baseline we implemented the algorithm described in (Sinha and Mihalcea 2007), weighting the edges with the Lesk method (Lesk 1986) and computing the node centrality with in-degree node centrality. We also ran a random baseline for 10 times over the words and senses and we report the average of the obtained values.

Crowd-Sourcing Ground Truth Data

Since the Senseval[3] corpora for word-sense disambiguation do not contain annotations for DBPedia senses, we ran a crowd-sourcing experiment to gather ground truth benchmark data for WSD. In order to generate the data, we ran LDA (McCallum 2002) on three corpora: (i) British Academic Written English Corpus (BAWE) (Nesi et al. 2007), (ii) BBC (Greene and Cunningham 2006), and (iii) Semantic Web Corpus [4]. We extracted 500 topics, 250 topics, and 100 topics from these three corpora respectively. We randomly selected a subset of 130 topics to match the feasibility of a user study. Afterwards we automatically extracted the possible senses from WordNet and DBPedia as described above.

We created a web interface to gather the annotators' input. For each randomly selected topic, annotators were given seven related target words. For each word, the system listed the definitions of the possible senses extracted from DBPedia and WordNet. For each sense, they could assign a score on a scale from 5 to 1 where 5 represents "Perfect match", 3 represents "Acceptable", and 1 represents "Not related at all". They could also label a word as being too ambiguous in the given context, could label the whole topic as too ambiguous, or just skip the whole topic. In the final data set, each sense has been annotated by three different annotators.

	if at least 2 annotators label the sense with:		
Hit	3, 4, or 5	4 or 5	5
Miss	1 or 2	1,2, or 3	1,2,3, or 4
	Acceptable	**Good**	**Perfect**

Table 2: Algorithm performance assessment scheme.

77 annotators participated in the experiment. After finally removing topics and words marked as too ambiguous by the users, we resulted in 55 topics from the BAWE corpus, 28

[3] http://www.senseval.org/

[4] http://data.semanticweb.org

Disagreement Filter	Only CAW	Correctness Assessment	#Total Words	Fleiss Kappa	Figures
YES	YES	Acceptable	330	0.83	Figure 4
		Good	305	0.73	
		Perfect	261	0.56	
YES	NO	Acceptable	588	0.83	Figure 5
		Good	557	0.73	
		Perfect	490	0.56	
NO	NO	Acceptable	616	0.63	Figure 6
		Good	586	0.55	
		Perfect	515	0.48	

Table 3: Experiment setups; CAW stands for Computationally Ambiguous Words

from BBC, and 33 from the Semantic Web corpus. The final set contained a total of 116 topics, 500 unique words, 633 words occurrences to be disambiguated, and 2578 annotated senses.

Results

For accuracy assessment of the WSD algorithms, we collapsed the sense annotation categories into two categories (Hit and Miss) and assessed the performance of the algorithms as "Acceptable", "Good", and "Perfect" according to Table 2. For instance, when we report the accuracy for the "Acceptable" case, it means that we counted as a hit a word for which the algorithm returned a sense that achieved at least two human annotations of 3, 4, or 5.

Regarding the word senses, we consider a sense annotated as *right* if at least two users labelled it with 3, 4, or 5. We consider it annotated as *wrong* otherwise. For an improved inter-annotator agreement, we also ran the experiments after filtering senses that were annotated with 4 or 5 by some annotators and with 1 and 2 by other annotators. Thus, this filter removes highly polarizing senses and removes cases with high inter-annotator disagreement. Thus, we refer to it as *disagreement filter*.

Because we gathered senses from two sense inventories, there is a high chance that more senses of one word are annotated as right. This possibility is further amplified by the use of topics as contexts for the words as it is sometimes impossible to get a clear-cut between closely related senses. Thus, in our case, rather than distinguish between polysemous and monosemous words, it is of more importance to distinguish between (i) words that were annotated with both right and wrong senses, and (ii) words annotated only with right or only with wrong senses. Only words annotated with both wrong and right senses pose an actual disambiguation challenge to the WSD algorithms, so we simply call them *computationally ambiguous*. Words whose senses were all either annotated right or all wrong are consequently called *computationally non-ambiguous*. We report the main results corresponding to the cases summarized in Table 3. The tests were carried only on the words which had at least one word annotated as a possible hit for each of the three correctness assessment cases. This avoids penalizing the algorithms for not finding the right sense when in fact there is no such sense.

We now present the comparative results obtained by running the three approaches. Figure 4 presents the accuracy values obtained only on computationally ambiguous words, after applying the disagreement filter. This setting contains the words and senses which achieved the highest inter-annotator agreement, i.e., 0.83. Because we wanted to check whether the number of words in the topic influences the performance of the algorithms, we ran the same experiment with 7, 8, 9, and 10 words per topic. Although small fluctuations appear in the performance of both algorithms, no consistent trend was observed.

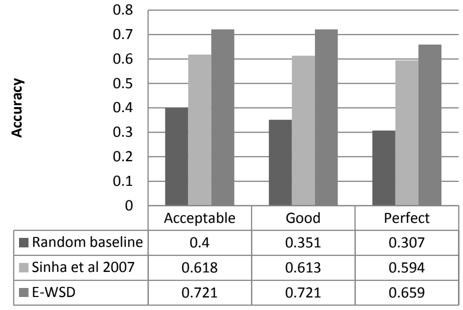


Figure 4: Accuracy measured only on computationally ambiguous words, after applying the disagreement filter

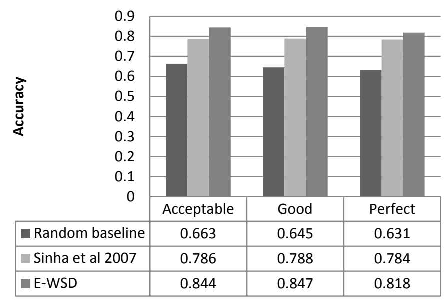


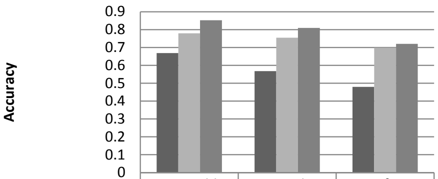
Figure 5: Accuracy measured on words after applying the disagreement filter

We also ran the algorithms on all words filtered only by the disagreement filter. The results are presented in Figure 5. As expected, this strongly improves the performance of all algorithms. This is due to the fact that most computationally non-ambiguous words had all their senses annotated as right. These results show that when run on WordNet and DBPedia, considering the context as topics, the unsupervised WSD algorithms can have a high accuracy. However, for comparing the two algorithms, the results in Figure 5 are not as conclusive as those in Figure 4.

Finally, we report on the results obtained when running the algorithms on all words and senses in Figure 6. Although causing the most disagreement among annotators, this setting has the highest number of words covered.

Discussion and Conclusions

All the tests we ran show that the E-WSD achieves significantly higher accuracy than other graph-based state-of-



	Acceptable	Good	Perfect
Random baseline	0.669	0.568	0.48
Sinha et al 2007	0.779	0.754	0.699
E-WSD	0.852	0.809	0.72

Figure 6: Accuracy measured on all words, without the disagreement filter

the-art WSD algorithms. The highest accuracy reported in (Sinha and Mihalcea 2007) is 0.611. In our setting, the same algorithm achieves an accuracy ranging from 0.59 to 0.78 on the "Perfect" assessment configuration. This suggests that using topics as contexts, the accuracy of unsupervised WSD algorithms can be significantly improved, as co-occurence in topics is due to higher relatedness between words. The use of DBPedia as a sense repository in addition to WordNet leads to the remarkable fact that out of the 633 words evaluated, only 17 had no sense annotated as right. Thus, querying these two repositories ensured a word coverage of 97%.

The improvements brought by our eigenvalue-based measure for WSD come at the cost of an increased computational overhead. This is due to the need to exhaustively search for the best sense combination. Nevertheless, the promising results encourage us to further investigate strategies for reducing the search space. Well-known techniques like simulated annealing, which has already been used with success for WSD (Cowie, Guthrie, and Guthrie 1992), are a promising direction to investigate.

The research on disambiguation against multiple knowledge bases including DBPedia, opens the way to text document linking to online data sources. This linking brings the opportunity to enhance text document retrieval, clustering and navigation in the current Web context. In the future, we will investigate performance issues and will work on establishing our newly proposed method for application in the aforementioned domains.

Acknowledgements

This work was supported by Science Foundation Ireland (SFI) partly under Grant No. 08/CE/I1380 (Lion-2), and partly under Grant No. 08/SRC/I1407 (Clique: Graph and Network Analysis Cluster).

References

Agirre, E., and Soroa, A. 2009. Personalizing pagerank for word sense disambiguation. In *Proc. 12th Conf. of the European Chapter of the Association for Computational Linguistics*.

Anaya-Sánchez, H.; Pons-Porrata, A.; and Berlanga-Llavori, R. 2009. Using sense clustering for the disambiguation of words.

Bharat, K., and Henzinger, M. R. 1998. Improved algorithms for topic distillation in a hyperlinked environment. In *Proc. 21st SIGIR conference on Research and development in information retrieval*, SIGIR '98, 104–111. ACM.

Cowie, J.; Guthrie, J.; and Guthrie, L. 1992. Lexical disambiguation using simulated annealing. In *Proc of COLING-92*, 359–365.

Finkel, J. R.; Grenager, T.; and Manning, C. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proc. 43nd Annual Meeting of the Association for Computational Linguistics*, 363 – 370.

Greene, D., and Cunningham, P. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd International Conference on Machine learning (ICML'06)*, 377–384. ACM Press.

Hoffart, J.; Suchanek, F. M.; Berberich, K.; Kelham, E. L.; de Melo, G.; and Weikum, G. 2011. Yago2: Exploring and querying world knowledge in time, space, context, and many languages. In *World Wide Web Conference 2011*.

Kleinberg, J. M. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM* 46:604–632.

Lesk, M. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proc. 5th International Conference on Systems Documentation*, SIGDOC '86, 24–26. ACM.

McCallum, A. K. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Mendes, P. N.; Jakob, M.; Garca-Silva, A.; and Bizer, C. 2011. Dbpedia spotlight: Shedding light on the web of documents. In *Proc. 7th Int. Conference on Semantic Systems*.

Miller, G. A. 1995. Wordnet: A lexical database for english. *Communications of the ACM* 38(11):39–41.

Navigli, R., and Velardi, P. 2005. Structural semantic interconnections: A knowledge-based approach to word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(7).

Navigli, R. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.* 41:10:1–10:69.

Nesi, H.; Gardner, S.; Thompson, P.; and Wickens, P. 2007. British academic written english (bawe) corpus. *Universities of Warwick, Reading and Oxford Brookes, under funding from ESRC (RES-000-23-0800)*.

Sinha, R., and Mihalcea, R. 2007. Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In *Proc. International Conference on Semantic Computing*, 363–369. IEEE Computer Society.

Steyvers, M., and Griffiths, T. 2007. *Probabilistic Topic Models*. Lawrence Erlbaum Associates.

Tsatsaronis, G.; Vazirgiannis, M.; and Androutsopoulos, I. 2007. Word sense disambiguation with spreading activation networks generated from thesauri. In *Proc. 20th Int. Joint Conf. in Artificial Intelligence*, 1725–1730.