Towards Dynamic Word Sense Discrimination with Random Indexing

Hans Moen, Erwin Marsi, Björn Gambäck
Norwegian University of Science and Technology
Department of Computer and Information and Science
Sem Sælands vei 7-9
NO-7491 Trondheim, Norway
{hansmoe,emarsi,gamback}@idi.ntnu.no

Abstract

Most distributional models of word similarity represent a word type by a single vector of contextual features, even though, words commonly have more than one sense. The multiple senses can be captured by employing several vectors per word in a multi-prototype distributional model, prototypes that can be obtained by first constructing all the context vectors for the word and then clustering similar vectors to create sense vectors. Storing and clustering context vectors can be expensive though. As an alternative, we introduce Multi-Sense Random Indexing, which performs on-the-fly (incremental) clustering. To evaluate the method, a number of measures for word similarity are proposed, both contextual and non-contextual, including new measures based on optimal alignment of word senses. Experimental results on the task of predicting semantic textual similarity do, however, not show a systematic difference between single-prototype and multi-prototype models.

1 Introduction

Many terms have more than one meaning, or sense. Some of these senses are static and can be listed in dictionaries and thesauri, while other senses are dynamic and determined by the contexts the terms occur in. Work in *Word Sense Disambiguation* often concentrate on the static word senses, making the task of distinguishing between them one of classification into a predefined set of classes (i.e., the given word senses); see, e.g., Erk et al. (2013; Navigli (2009) for overviews of current work in the area. The idea of fixed generic word senses has received a fair amount of criticism in the literature (Kilgarriff, 2000).

This paper instead primarily investigates dynamically appearing word senses, word senses that depend on the actual usage of a term in a corpus or a domain. This task is often referred to as Word Sense Induction or *Word Sense Discrimination* (Schütze, 1998). This is, in contrast, essentially a categorisation problem, distinguished by different senses being more or less similar to each other at a given time, given some input data. The dividing line between Word Sense Disambiguation and Discrimination is not necessarily razor sharp though: also different senses of a term listed in a dictionary tend to have some level of overlap.

In recent years, distributional models have been widely used to infer word similarity. Most such models represent a word type by a single vector of contextual features obtained from co-occurrence counts in large textual corpora. By assigning a single vector to each term in the corpus, the resulting model assumes that each term has a fixed semantic meaning (relative to all the other terms). However, due to homonomy and polysemy, word semantics cannot be adequately represented by a single-prototype vector.

Multi-prototype distributional models in contrast employ different vectors to represent different senses of a word (Reisinger and Mooney, 2010). Multiple prototypes can be obtained by first constructing context vectors for all words and then clustering similar context vectors to create a sense vector. This may be expensive, as vectors need to stored and clustered. As an alternative, we propose a new method called Multi-Sense Random Indexing (MSRI), which is based on Random Indexing (Kanerva et al., 2000) and performs an on-the-fly (incremental) clustering.

MSRI is a method for building a multi-prototype / multi-sense vector space model, which attempts to capture one or more senses per unique term in an unsupervised manner, where each sense is represented as a separate vector in the model.

Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality, pages 83–90,
Sofia, Bulgaria, August 9 2013.

This differs from the classical Random Indexing (RI) method which assumes a static sense inventory by restricting each term to have only one vector (sense) per term, as described in Section 2. The MSRI method is introduced in Section 3.

Since the induced dynamic senses do not necessarily correspond to the traditional senses distinguished by humans, we perform an extrinsic evaluation by applying the resulting models to data from the Semantic Textual Similarity shared task (Agirre et al., 2013), in order to compare MSRI to the classical RI method. The experimental setup is the topic of Section 4, while the results of the experiments are given in Section 5. Section 6 then sums up the discussion and points to ways in which the present work could be continued.

2 Vector Space Models

With the introduction of LSA, Latent Semantic Analysis (Deerwester et al., 1990), distributed models of lexical semantics, built from unlabelled free text data, became a popular sub-field within the language processing research community. Methods for building such semantic models rely primarily on term co-occurrence information, and attempt to capture latent relations from analysing large amounts of text. Most of these methods represent semantic models as multi-dimensional vectors in a vector space model.

After LSA, other methods for building semantic models have been proposed, one of them being Random Indexing (Kanerva et al., 2000). Common to these methods is that they generate a *context vector* for each unique term in the training data which represents the term's "contextual" meaning in the vector space. By assigning a single context vector to each term in the corpus, the resulting model assumes that each term has a fixed semantic meaning (relative to all other terms).

Random Indexing incrementally builds a co-occurrence matrix of reduced dimensionality, by first assigning *index vectors* to each unique term. The vectors are of a predefined size (typically around 1000), and consist of a few randomly placed 1s and -1s. Context vectors of the same size are also assigned to each term, initially consisting of only zeros. When traversing a document corpus using a sliding window of a fixed size, the context vectors are continuously updated: the term in the centre of the window (the target term), has the index vectors of its neighbouring terms (the ones in the window) added to its context vector using vector summation. Then the *cosine similarity measure* can be used on term pairs to calculate their similarity (or "contextual similarity").

Random Indexing has achieved promising results in various experiments, for example, on the TOEFL test ("Test of English as a Foreign Language") (Kanerva et al., 2000). However, it is evident that many terms have more than one meaning or sense, some being static and some dynamic, that is, determined by the contexts the terms occur in. Schütze (1998) proposed a method for clustering the contextual occurrences of terms into individual "prototype" vectors, where one term can have multiple prototype vectors representing separate senses of the term. Others have adopted the same underlying idea, using alternative methods and techniques (Reisinger and Mooney, 2010; Huang et al., 2012; Van de Cruys et al., 2011; Dinu and Lapata, 2010).

3 Multi-Sense Random Indexing, MSRI

Inspired by the work of Schütze (1998) and Reisinger and Mooney (2010), this paper introduces a novel variant of Random Indexing, which we have called "Multi-Sense Random Indexing". MSRI attempts to capture one or more senses per unique term in an unsupervised and incremental manner, each sense represented as an separate vector in the model. The method is similar to classical sliding window RI, but each term can have multiple context vectors (referred to as *sense vectors* here) which are updated separately.

When updating a term vector, instead of directly adding the index vectors of the neighbouring terms in the window to its context vector, the system first computes a separate *window vector* consisting of the sum of the index vectors. The similarity between the window vector and each of the term's sense vectors is calculated. Each similarity score is then compared to a pre-set *similarity threshold*:

- if no score exceeds the threshold, the window vector becomes a new separate sense vector for the term,
- if exactly one score is above the threshold, the window vector is added to that sense vector, and
- if multiple scores are above the threshold, all the involved senses are merged into one sense vector, together with the window vector.

Algorithm 1 MSRI training

```
for all terms t in a document D do
    generate window vector w⃗in from the neighbouring words' index vectors
    for all sense vectors s⃗_i of t do
        sim(s_i) = CosSim(w⃗in, s⃗_i)
    end for
    if sim(s_{i..k}) ≥ τ then
        Merge s⃗_{i..k} and w⃗in through summing
    else
        if sim(s_i) ≥ τ then
            s⃗_i+ = w⃗in
        end if
    else
        if sim(s_{i..n}) < τ then
            Assign w⃗in as new sense vector of t
        end if
    end if
end for
```

See Algorithm 1 for a pseudo code version. Here τ represents the similarity threshold.

This accomplishes an incremental (on-line) clustering of senses in an unsupervised manner, while retaining the other properties of classical RI. Even though the algorithm has a slightly higher complexity than classical RI, this is mainly a matter of optimisation, which is not the focus of this paper. The incremental clustering that we apply is somewhat similar to what is used by Lughofer (2008), although we are storing in memory only one element (i.e., vector) for each "cluster" (i.e., sense) at any given time.

When looking up a term in the vector space, a pre-set *sense-frequency threshold* is applied to filter out "noisy" senses. Hence, senses that have occurred less than the threshold are not included when looking up a term and its senses for, for example, similarity calculations.

As an example of what the resulting models contain in terms of senses, Table 1 shows four different senses of the term 'round' produced by the MSRI model. Note that these senses do not necessarily correspond to human-determined senses. The idea is only that using multiple prototype vectors facilitates better modelling of a term's meaning than a single prototype (Reisinger and Mooney, 2010).

round$_1$	**round**$_2$	**round**$_3$	**round**$_4$
finish	camping	inch	launcher
final	restricted	bundt	grenade
match	budget	dough	propel
half	fare	thick	antitank
third	adventure	cake	antiaircraft

Table 1: Top-5 most similar terms for four different senses of 'round' using the *Max* similarity measure to the other terms in the model.

3.1 Term Similarity Measures

Unlike classical RI, which only has a single context vector per term and thus calculates similarity between two terms directly using cosine similarity, there are multiple ways of calculating the similarity between two terms in MSRI. Some alternatives are described in Reisinger and Mooney (2010). In the experiment in this paper, we test four ways of calculating similarity between two terms t and t' in isolation, with the Average and Max methods stemming from Reisinger and Mooney (2010).

Let $\vec{s}_{i..n}$ and $\vec{s'}_{j..m}$ be the sets of sense vectors corresponding to the terms t and t' respectively. Term similarity measures are then defined as:

Centroid

For term t, compute its centroid vector by summing its sense vectors $\vec{s}_{i..n}$. The same is done for t' with its sense vectors $\vec{s'}_{j..m}$. These centroids are in turn used to calculate the cosine similarity between t and t'.

Average

For all $\vec{s_{i..n}}$ in t, find the pair $\vec{s}_i, \vec{s'}_j$ with highest cosine similarity:

$$\frac{1}{n}\sum_{i=1}^{n} CosSim_{max}(\vec{s}_i, \vec{s'}_j)$$

Then do the same for all $\vec{s'}_{j..m}$ in t':

$$\frac{1}{m}\sum_{j=1}^{m} CosSim_{max}(\vec{s'}_j, \vec{s}_i)$$

The similarity between t and t' is computed as the average of these two similarity scores.

Max

The similarity between t_i and t'_i equals the similarity of their most similar sense:

$$Sim(t, t') = CosSim_{max_{ij}}(\vec{s}_i, \vec{s'}_i)$$

Hungarian Algorithm

First cosine similarity is computed for each possible pair of sense vectors $\vec{s}_{i..n}$ and $\vec{s'}_{j..m}$, resulting in a matrix of similarity scores. Finding the optimal matching from senses $\vec{s}_i$ to $\vec{s'}_j$ that maximises the sum of similarities is known as the *assignment problem*. This combinatorial optimisation problem can be solved in polynomial time through the Hungarian Algorithm (Kuhn, 1955). The overall similarity between terms t and t' is then defined as the average of the similarities between their aligned senses.

All measures defined so far calculate similarity between terms in isolation. In many applications, however, terms occur in a particular context that can be exploited to determine their most likely sense. Narrowing down their possible meaning to a subset of senses, or a single sense, can be expected to yield a more adequate estimation of their similarity. Hence a context-sensitive measure of term similarity is defined as:

Contextual similarity

Let $\vec{C}$ and $\vec{C'}$ be vectors representing the contexts of terms t and t' respectively. These context vectors are constructed by summing the index vectors of the neighbouring terms within a window, following the same procedure as used when training the MSRI model. We then find $\hat{s}$ and $\hat{s}'$ as the sense vectors best matching the context vectors:

$$\hat{s} = \arg\max_i \ CosSim(\vec{s}_i, \vec{C})$$

$$\hat{s}' = \arg\max_j \ CosSim(\vec{s}_j, \vec{C'})$$

Finally, contextual similarity is defined as the similarity between these sense vectors:

$$Sim_{context}(t, t') = CosSim(\hat{s}, \hat{s}')$$

3.2 Sentence Similarity Features

In the experiments reported on below, a range of different ways to represent sentences were tested. Sentence similarity was generally calculated by the average of the maximum similarity between pairs of terms from both sentences, respectively. The different ways of representing the data in combination with some sentence similarity measure will here be referred to as similarity *features*.

1. `MSRI-TermCentroid`:
   In each sentence, each term is represented as the sum of its sense vectors. This is similar to having one context vector, as in classical RI, but due to the sense-frequency filtering, potentially "noisy" senses are not included.

2. `MSRI-TermMaxSense`:
   For each bipartite term pair in the two sentences, their sense-pairs with maximum cosine similarity are used, one sense per term.

3. `MSRI-TermInContext`:
   A $5 + 5$ window around each (target) term is used as context for selecting one sense of the term. A window vector is calculated by summing the index vectors of the other terms in the window (i.e., except for the target term itself). The sense of the target term which is most similar to the window vector is used as the representation of the term.

4. `MSRI-TermHASenses`:
   Calculating similarity between two terms is done by applying the Hungarian Algorithm to all their bipartite sense pairs.

5. `RI-TermAvg`:
   Classical Random Indexing — each term is represented as a single context vector.

6. `RI-TermHA`:
   Similarity between two sentences is calculated by applying the Hungarian Algorithm to the context vectors of each constituent term.

The parameters were selected based on a combination of surveying previous work on RI (e.g., Sokolov (2012)), and by analysing how sense counts evolved during training. For MSRI, we used a similarity threshold of 0.2, a vector dimensionality of 800, a non-zero count of 6, and a window size of $5 + 5$. Sense vectors resulting from less than 50 observations were removed. For classical RI, we used the same parameters as for MSRI (except for a similarity threshold).

4 Experimental Setup

In order to explore the potential of the MSRI model and the textual similarity measures proposed here, experiments were carried out on data from the Semantic Textual Similarity (STS) shared task (Agirre et al., 2012; Agirre et al., 2013).

Given a pair of sentences, systems participating in this task shall compute how semantically similar the two sentences are, returning a similarity score between zero (completely unrelated) and five (completely semantically equivalent). Gold standard scores are obtained by averaging multiple scores obtained from human annotators. System performance is then evaluated using the Pearson product-moment correlation coefficient (ρ) between the system scores and the human scores.

The goal of the experiments reported here was not to build a competitive STS system, but rather to investigate whether MSRI can outperform classical Random Indexing on a concrete task such as computing textual similarity, as well as to identify which similarity measures and meaning representations appear to be most suitable for such a task. The system is therefore quite rudimentary: a simple linear regression model is fitted on the training data, using a single sentence similarity measure as input and the similarity score as the dependent variable. The implementations of RI and MSRI are based on JavaSDM (Hassel, 2004).

As data for training random indexing models, we used the CLEF 2004–2008 English corpus, consisting of approximately 130M words of newspaper articles (Peters et al., 2004). All text was tokenized and lemmatized using the TreeTagger for English (Schmid, 1994). Stopwords were removed using a customized version of the stoplist provided by the Lucene project (Apache, 2005).

Data for fitting and evaluating the linear regression models came from the STS development and test data, consisting of sentence pairs with a gold standard similarity score. The STS 2012 development data stems from the Microsoft Research Paraphrase corpus (`MSRpar`, 750 pairs), the Microsoft Research Video Description corpus (`MSvid`, 750 pairs), and statistical machine translation output based on the Europarl corpus (`SMTeuroparl`, 734 pairs). Test data for STS 2012 consists of more data from the same sources: `MSRpar` (750 pairs), `MSRvid` (750 pairs) and `SMTeuroparl` (459 pairs). In addition, different test data comes from translation data in the news domain (`SMTnews`, 399 pairs) and ontology mappings between OntoNotes and WordNet (`OnWN`, 750 pairs). When testing on the STS 2012 data, we used the corresponding development data from the same domain for training, except for `OnWN` where we used all development data combined.

The development data for STS 2013 consisted of all development and test data from STS 2012 combined, whereas test data comprised machine translation output (`SMT`, 750 pairs), ontology mappings both between WordNet and OntoNotes (`OnWN`, 561 pairs) and between WordNet and FrameNet (`FNWN`, 189 pairs), as well as news article headlines (`HeadLine`, 750 pairs). For simplicity, all development data combined were used for fitting the linear regression model, even though careful matching of development and test data sets may improve performance.

5 Results and Discussion

Table 2 shows Pearson correlation scores per feature on the STS 2012 test data using simple linear regression. The most useful features for each data set are marked in bold. For reference, the scores of the best performing STS systems for each data set are also shown, as well as baseline scores obtained with a simple normalized token overlap measure.

There is large variation in correlation scores, ranging from 0.77 down to 0.27. Part of this variation is due to the different nature of the data sets. For example, sentence similarity in the SMT domain seems harder to predict than in the video domain. Yet there is no single measure that obtains the highest score on all data sets. There is also no consistent difference in performance between the RI and MSRI measures, which seem to yield about equal scores on average. The `MSRI-TermInContext` measure has the lowest score on average, suggesting that word sense disambiguation in context is not beneficial in its current implementation.

The corresponding results on the STS 2013 test data are shown in Table 3. The same observations as for the STS 2012 data set can be made: again there was no consistent difference between the RI and MSRI features, and no single best measure.

All in all, these results do not provide any evidence that MSRI improves on standard RI for this particular task (sentence semantic similarity). Multi-sense distributional models have, however, been found to outperform single-sense models on other tasks. For example, Reisinger and Mooney (2010) report that multi-sense models significantly increase the correlation with human similarity judgements. Other multi-prototype distributional models may yield better results than their single-prototype counterparts on the STS task.

Features:	MSRpar	MSRvid	SMTeuroparl	SMTnews	OnWN	Mean
Best systems	0.73	0.88	0.57	0.61	0.71	0.70
Baseline	0.43	0.30	0.45	0.39	0.59	0.43
RI-TermAvg	0.44	0.71	**0.50**	**0.42**	0.65	**0.54**
RI-TermHA	0.41	0.72	0.44	0.35	0.56	0.49
MSRI-TermCentroid	**0.45**	0.73	**0.50**	0.33	0.64	0.53
MSRI-TermHASenses	0.40	**0.77**	0.47	0.39	**0.68**	**0.54**
MSRI-TermInContext	0.33	0.55	0.36	0.27	0.42	0.38
MSRI-TermMaxSense	0.44	0.71	**0.50**	0.32	0.64	0.52

Table 2: Pearson correlation scores per feature on STS 2012 test data using simple linear regression

Feature	Headlines	SMT	FNWN	OnWN	Mean
Best systems	0.78	0.40	0.58	0.84	0.65
Baseline	0.54	0.29	0.21	0.28	0.33
RI-TermAvg	0.60	**0.37**	0.21	0.52	0.42
RI-TermHA	**0.65**	0.36	0.27	0.52	0.45
MSRI-TermCentroid	0.60	0.35	**0.37**	0.45	0.44
MSRI-TermHASenses	0.63	0.35	0.33	**0.54**	**0.46**
MSRI-TermInContext	0.20	0.29	0.19	0.36	0.26
MSRI-TermMaxSense	0.58	0.35	0.31	0.45	0.42

Table 3: Pearson correlation scores per feature on STS 2013 test data using simple linear regression

Notably, the more advanced features used in our experiment, such as `MSRI-TermInContext`, gave very clearly inferior results when compared to `MSRI-TermHASenses`. This suggests that more research on MSRI is needed to understand how both training and retrieval can be fully utilized and optimized.

6 Conclusion and Future Work

The paper introduced a new method called Multi-Sense Random Indexing (MSRI), which is based on Random Indexing and performs on-the-fly clustering, as an efficient way to construct multi-prototype distributional models for word similarity. A number of alternative measures for word similarity were proposed, both context-dependent and context-independent, including new measures based on optimal alignment of word senses using the Hungarian algorithm. An extrinsic evaluation was carried out by applying the resulting models to the Semantic Textual Similarity task. Initial experimental results did not show a systematic difference between single-prototype and multi-prototype models in this task.

There are many questions left for future work. One of them is how the number of senses per word evolves during training and how the distribution of senses in the final model looks like. So far we only know that on average the number of senses keeps growing with more training material, currently resulting in about 5 senses per word at the end of training (after removing senses with frequency below the sense-frequency threshold). It is worth noting that this depends heavily on the similarity threshold for merging senses, as well as on the weighting schema used.

In addition there are a number of model parameters that have so far only been manually tuned on the development data, such as window size, number of non-zeros, vector dimensionality, and the sense frequency filtering threshold. A systematic exploration of the parameter space is clearly desirable. Another thing that would be worth looking into, is how to compose sentence vectors and document vectors from the multi-sense vector space in a proper way, focusing on how to pick the right senses and how to weight these. It would also be interesting to explore the possibilities for combining the MSRI method with the Reflective Random Indexing method by Cohen et al. (2010) in an attempt to model higher order co-occurrence relations on sense level.

The fact that the induced dynamic word senses do not necessarily correspond to human-created senses makes evaluation in traditional word sense disambiguation tasks difficult. However, correla-

tion to human word similarity judgement may provide a way of intrinsic evaluation of the models (Reisinger and Mooney, 2010). The *Usim* bench mark data look promising for evaluation of word similarity in context (Erk et al., 2013).

It is also worth exploring ways to optimise the algorithm, as this has not been the focus of our work so far. This would also allow faster training and experimentation on larger text corpora, such as Wikipedia. In addition to the JavaSDM package (Hassel, 2004), Lucene (Apache, 2005) with the Semantic Vectors package (Widdows and Ferraro, 2008) would be an alternative framework for implementing the proposed MSRI algorithm.

Acknowledgements

This work was partly supported by the Research Council of Norway through the EviCare project (NFR project no. 193022) and by the European Community's Seventh Framework Programme (FP7/20072013) under grant agreement nr. 248307 (PRESEMT). Part of this work has been briefly described in our contribution to the STS shared task (Marsi et al., 2013).

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A pilot on semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*, volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, pages 385–393, Montreal, Canada, June. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, pages 32–43, Atlanta, Georgia, June. Association for Computational Linguistics.

Apache. 2005. Apache Lucene open source package. `http://lucene.apache.org/`.

Trevor Cohen, Roger Schvaneveldt, and Dominic Widdows. 2010. Reflective random indexing and indirect inference: A scalable method for discovery of implicit connections. *Journal of Biomedical Informatics*, 43(2):240–256, April.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Georgiana Dinu and Mirella Lapata. 2010. Measuring distributional similarity in context. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1162–1172, Cambridge, Massachusetts, October. Association for Computational Linguistics.

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. 2013. Measuring word meaning in context. *Computational Linguistics*, 39(3):501–544.

Martin Hassel. 2004. JavaSDM package. `http://www.nada.kth.se/~xmartin/java/`. School of Computer Science and Communication; Royal Institute of Technology (KTH); Stockholm, Sweden.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 873–882, Jeju Island, Korea. Association for Computational Linguistics.

Pentti Kanerva, Jan Kristoferson, and Anders Holst. 2000. Random indexing of text samples for latent semantic analysis. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, page 1036, Philadelphia, Pennsylvania. Erlbaum.

Adam Kilgarriff. 2000. I don't believe in word senses. *Computers and the Humanities*, 31(2):91–113.

Harold W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

Edwin Lughofer. 2008. Extensions of vector quantization for incremental clustering. *Pattern Recognition*, 41(3):995–1011, March.

Erwin Marsi, Hans Moen, Lars Bungum, Gleb Sizov, Björn Gambäck, and André Lynum. 2013. NTNU-CORE: Combining strong features for semantic similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, pages 66–73, Atlanta, Georgia, June. Association for Computational Linguistics.

Roberto Navigli. 2009. Word Sense Disambiguation: a survey. *ACM Computing Surveys*, 41(2):1–69.

Carol Peters, Paul Clough, Julio Gonzalo, Gareth J.F. Jones, Michael Kluck, and Bernardo Magnini, editors. 2004. *Multilingual Information Access for Text, Speech and Images, 5th Workshop of the Cross-Language Evaluation Forum, CLEF 2004*, volume 3491 of *Lecture Notes in Computer Science*. Springer-Verlag, Bath, England.

Joseph Reisinger and Raymond J. Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117, Los Angeles, California, June.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the 1st International Conference on New Methods in Natural Language Processing*, pages 44–49, University of Manchester Institute of Science and Technology, Manchester, England, September.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123, March.

Artem Sokolov. 2012. LIMSI: learning semantic similarity by selecting random word subsets. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*, volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, pages 543–546, Montreal, Canada, June. Association for Computational Linguistics.

Tim Van de Cruys, Thierry Poibeau, and Anna Korhonen. 2011. Latent vector weighting for word meaning in context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1012–1022, Edinburgh, Scotland, July. Association for Computational Linguistics.

Dominic Widdows and Kathleen Ferraro. 2008. Semantic vectors: a scalable open source package and online technology management application. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, pages 1183–1190, Marrakech, Morocco.