Proceedings of the 2007 Workshop on

Contextual Information in Semantic Space Models

Beyond Words and Documents

Marco Baroni
Alessandro Lenci
Magnus Sahlgren
(Editors)

AUGUST 2007 ROSKILDE UNIVERSITY COMPUTER SCIENCE RESEARCH REPORT #116

Marco Baroni, Alessandro Lenci, and Magnus Sahlgren

Computer Science
Roskilde University
P. O. Box 260
DK–4000 Roskilde
Denmark

Telephone: +45 4674 3839
Telefax: +45 4674 3072
Internet: http://www.ruc.dk/dat/
E-mail: datalogi@ruc.dk

ISSN 0109–9779

This research report constitutes the proceedings of the *2007 Workshop on Contextual Information in Semantic Space Models: Beyond Words and Documents* which is held in conjunction with the 6th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT 2007), Roskilde University, Denmark, August 2007.

Research reports are available electronically from:

http://www.ruc.dk/dat/

Sixth International and Interdisciplinary Conference on Modeling and Using Context

Workshop on Contextual Information in Semantic Space Models

Beyond Words and Documents

Roskilde (Denmark)

21st August 2007

Preface

Some variation of the so-called *distributional hypothesis* – i.e. that words with similar distributional properties have similar semantic properties – lies at the heart of a number of computational approaches that share the assumption that it is possible to build semantic space models through the statistical analysis of the *contexts* in which words occur. From the epistemological point of view, semantic space models raise the twofold question of the extent to which (lexico-)semantic properties can be reduced to usage patterns derived from texts, and of the role of context in determining the properties of the word space. In fact, computational approaches to meaning have been claiming to provide not only a different way to investigate the semantic properties of words through "text-mining" processes, but also the opportunity to design radically new styles of usage-anchored, context-sensitive semantic representations. Semantic space models have been applied to extract different types of properties, e.g. synonymic (Landauer & Dumais 1997) and analogical relations (Turney 2006), and have been used to automatically build thesauri for NLP applications such as Question Answering, Word Sense Disambiguation, etc. (Sahlgren 2006). In the cognitive sciences, many researchers have strongly argued for the psychological validity of distributional semantics. For instance, corpus-derived measures of semantic similarity have been assessed in a variety of psychological tasks ranging from similarity judgments to simulations of semantic and associative priming, etc. (McDonald 2000). Distributional techniques have also been applied to model child lexical development as a bootstrapping process in which lexical and grammatical categories are extracted from the statistical distributions in the input from adults (Li *et al.* 2004).

This wide cross-disciplinary consensus notwithstanding, the very notion of context on which semantic spaces rely on gives rise to various crucial issues both at the theoretical and at the computational level, which in turn determine a large space of parametric variations. The aim of this workshop is to foster a fully cross-disciplinary debate around the major open questions pertaining to the definition and usage of context in context-based semantic modeling. These essentially concern the assumptions about the context structure and can be briefly described as follows:

1. *the context type* – in classical semantic space models, such as LSA (Landauer & Dumais 1997) or HAL (Burgess & Lund 1997), context is represented either by a document (or document section) or by a window of words surrounding a target item. In turn, word-defined context can be variously represented either by raw token co-occurrences or by linguistically pre-processed contexts. More recently,

some interesting experiments have been carried out to define context in terms of grammatical dependency structures (Padó & Lapata 2003). In other approaches, the context is represented by a number of pre-selected human-identified linguistic patterns, that are regarded as more directly conducive to particular semantic relations (cf. Widdows & Dorow 2002). Thus, there is a trend going towards more sophisticated and abstract definition of context. This also raises the issue of how the different types of context relate to the type of extracted lexical relations and to the global quality and properties of semantic spaces. Furthermore, to the extent that different types of contexts lead to models that are better at different semantic similarity tasks (Sahlgren 2006), the issue arises of how to combine different contexts for an integrated approach to semantic space.

2. *the context source* – in state of the art models, context is essentially defined as "co-text", i.e. by the other words with which a target word occurs within a certain textually or linguistically defined unit (document, sentence, adjacent surrounding words, phrase etc.). However, neither the distributional hypothesis nor the mathematical models that implement it prevents us from exploring new types of context sources which might be useful to determine the lexical semantic space. For instance, the word context could also include extra-linguistic features of word denotata or of the communicative situation in which words are used. It is in fact important to evaluate the extent to which semantic space models are doomed to give us a purely "language internal" characterization of meaning, or could instead move on to capture the role of extralinguistic context in shaping meaning. This would also allow these models to meet those arguments that point to their purely text-based character as a major limit of their real capacity to provide a realistic models of meaning, which is ultimately regarded to be externally grounded (cf. Glenberg & Robertson 2000). Even within the realm of language-internal evidence, many alternative sources of contexts remain to be explored, such as co-occurrence with multi-word units instead of or in addition to single words, co-occurrence with abstract linguistic objects such as construction types, etc.

3. *modes of context processing* – Semantic space models differ also regarding the way context-based distributions are processed from text sources. A number of approaches (e,g, LSA, HAL, etc.) use a global representation of contextual distributions: that is to say, the position of each word in the semantic space is determined only after the whole document collection has been processed. On the other hand, the Random Indexing model (Sahlgren 2006) adopts an incremental strategy of contextual representation construction. This latter solution has the attractive feature of being closer to a realistic model of human processing of context. However, even within incremental approaches, the full implications of

incrementality have not been fully explored: for example, modeling how specific contextual information might temporarily override the more common interpretation of a word. A related issue is the role and the actual need for mathematical techniques of semantic space dimension reduction, such as PCA or SVD that operate on the full co-occurrence matrix. Their effectiveness in building accurate semantic spaces notwithstanding, the usage of non-incremental dimensionality reduction techniques, again, negatively impacts on the cognitive plausibility of the relevant models.

In the call of papers we solicited papers focusing on different aspects and open issues related to context modelling:

- using new linguistic and extra-linguistic contexts in semantic space modeling;

- integration of different context sources (linguistic, extra-linguistic, etc.) to bootstrap semantic spaces;

- plausible contexts for cognitively and neurally realistic semantic space models;

- effect of context choice on the performance of semantic space models in applicative settings;

- critical analysis of the limits of current context-based models of meaning and future perspectives.

Actually, most of these topics lie at the heart of the papers that were accepted to the workshop.

We would like to thank all the authors who submitted papers, as well as the members of the programme Committee for the time and effort they contributed in reviewing the papers. Our thanks go also to the organizers of the CONTEXT 07 Conference, and to the Workshop Chair, Stefan Schultz.

July, 2007

Marco Baroni, Alessandro Lenci, Magnus Sahlgren

Essential references

Burgess, C. & Lund, K. (1997), "Modelling parsing constraints with high-dimensional context space", *Language and Cognitive Processes*, 12: 1-34.

Glenberg, A. M. & Robertson, D. A. (2000), "Symbol grounding and meaning: A comparison of high-dimensional and embodied theories of meaning", *Journal of Memory & Language*, Vol 43(3): 379-401.

Landauer, T. K. and Dumais, S. T. (1997), "A solution to Plato's problem: the Latent Semantic Analysis theory of acquisition, induction and representation of knowledge", *Psychological Review*, 104(2): 211-240.

Li, P., Farkas, I., & MacWhinney, B. (2004), "Early lexical development in a self-organizing neural networks", *Neural Networks*, 17: 1345-1362.

McDonald, S. (2000), *Environmental Determinants of Lexical Processing Effort*, PhD dissertation, University of Edinburgh.

Padó S. & Lapata, M., (2003), "Constructing Semantic Space Models from Parsed Corpora", in *Proceedings of the 41st Meeting of the Association of Computational Linguistics*, 128-135.

Sahlgren, M. (2006), *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. Ph.D. dissertation, Department of Linguistics, Stockholm University.

Turney, P.D. (2006), "Similarity of semantic relations", *Computational Linguistics*, 32 (3), 379-416.

Widdows, D., & Dorow, B. (2002), "A Graph Model for Unsupervised Lexical Acquisition", in *19th International Conference on Computational Linguistics*, Taipei, August 2002: 1093-1099.

Workshop organization

Organizing Committee

Marco Baroni (University of Trento)
Alessandro Lenci (University of Pisa)
Magnus Sahlgren (Swedish Institute of Computer Science)

Programme Committee

Marco Baroni (University of Trento, co-chair)
Gemma Boleda (Pompeu Fabra University, Barcelona)
Paul Buitelaar (DFKI)
John Bullinaria (University of Birmingham)
Curt Burgess (University of California, Riverside)
Stefan Evert (University of Osnabrück)
Pentti Kanerva (CSLI, Stanford)
Jussi Karlgren (Swedish Institute of Computer Science)
Mirella Lapata (University of Edinburgh)
Alessandro Lenci (University of Pisa, co-chair)
Simonetta Montemagni (ILC-CNR)
Vito Pirrelli (ILC-CNR)
Massimo Poesio (University of Trento)
Reinhard Rapp (University of Mainz)
Magnus Sahlgren (Swedish Institute of Computer Science, co-chair)
Fabrizio Sebastiani (ISTI-CNR)
Peter Turney (National Research Council of Canada)
Gabriella Vigliocco (University College, London)

Table of Contents

Effectiveness of Indirect Dependency for Automatic Synonym Acquisition

Masato HAGIWARA, Yasuhiro OGAWA, and Katsuhiko TOYAMA

Nagoya University, Furo-cho, Chikusa-ku, Nagoya, Japan
{hagiwara,yasuhiro,toyama}@kl.i.is.nagoya-u.ac.jp

Abstract. Since synonyms are important lexical knowledge, various methods have been proposed for automatic synonym acquisition. Whereas most of the methods are based on the *distributional hypothesis* and utilize contextual clues, little attention has been paid to what kind of contextual information is useful for the purpose. As one of the ways to augment contextual information, we propose the use of *indirect dependency*, i.e. relation between two words related via two contiguous dependency relations. The evaluation has shown that the performance improvement over normal direct dependency is dramatic, yielding comparable results with surrounding words as context, even with smaller co-occurrence data.

1 Introduction

Lexical knowledge is one of the most fundamental but important resources for natural language processing. Among various kinds of lexical relations, synonyms are used in a broad range of applications such as query expansion for information retrieval [8] and automatic thesaurus construction [9].

Various methods [7, 10] have been proposed for automatic synonym acquisition. They are often based on the distributional hypothesis [6], which states that semantically similar words share similar contexts, and they can be roughly viewed as the combinations of these two steps: context extraction and similarity calculation. The former extracts useful information such as dependency relations of words from corpora. The latter calculates how semantically similar two given words are, based on the co-occurrence counts or frequency distributions acquired in the first step, using similarity models such as mutual information.

However, whereas many methods employ the context-based similarity calculation, almost no attention has been paid to what kind of contextual information is useful for word featuring in terms of synonym acquisition.

For example, Ruge [13] proposed the use of dependency structure of sentences to detect term similarities for automatic thesaurus construction and showed the evaluation result to be encouraging, but neither the further investigation of dependency selection nor the comparison with other kinds of contextual information is provided. Lin [10] used a broad-coverage parser to extract wider range of grammatical relationship and showed the possibility that other kind of dependency relations in addition to subject and object was contributing, although it is still not clear what kind of relations affects the performance, or to what extent.

1

Few exceptions include Curran's [3], where they compared context extractors such as window extractor and shallow- and deep-parsing extractor. Their observation, however, doesn't accompany discussion concerning the qualitative difference of the context extractors and its causes. Because the choice of useful contextual information has a critical importance on the performance, further investigations on which types of contexts are essentially contributing are required.

As one of the ways to augment the contextual information, this paper proposes the use of *indirect dependency*, and shows its effectiveness for automatic synonym acquisition. We firstly extract *direct dependency* using RASP parser [1] from three different corpora, then extend it to indirect dependency which includes the relations composed from two or more contiguous dependency relations. The contexts corresponding direct and indirect dependency are extracted, and co-occurrences of words and their contexts are obtained. Because the details of similarity calculation is not the scope of this paper, widely used vector space model, tf.idf weighting, and cosine measure are adopted. The acquisition performance is evaluated using two automatic evaluation measures: average precision (AP) and correlation coefficient (CC) based on three existing thesauri.

This paper is organized as follows: in Section 2 we mention the preliminary experiment result of contextual information selection, along with the background of how we get to choose the indirect dependency. Sections 3 and 4 detail the formalization and the context extraction for indirect dependency. Section 5 briefly describes the synonym acquisition model we used, and in the following Section 6 the evaluation method is detailed. Section 7 provides the experimental conditions and results, followed by Section 8 which concludes this paper.

2 Context Selection

In this section, we show the result of the preliminary experiment of contextual information selection, and describe how we came up with the idea that the extension of normal direct dependency could be beneficial. Here we focused on the following three kinds of contextual information for comparison:

- `dep`: direct dependency; contexts extracted from the grammatical relations computed by RASP parser.
- `prox`: word proximity; surrounding words, i.e. words which locate within the window centered at a target word, and their relative positions. For example, a context having "the" on the left is represented as `L1:the`. We set the window radius to 3 in this paper.
- `sent`: sentence co-occurrence; sentence id in which the words occur. The underlying assumption of using this information is that words which occur in the same sentence are likely to share similar topics.

The overall experimental framework and evaluation scheme are same as the ones mentioned in the later sections. AP is the precision of acquired synonyms and CC is how similar the obtained similarity is correlated with WordNet's. The result, shown in Figure 1, suggests the superiority of `prox` over `dep` although the window range to capture the surrounding words is rather limited. This result

2

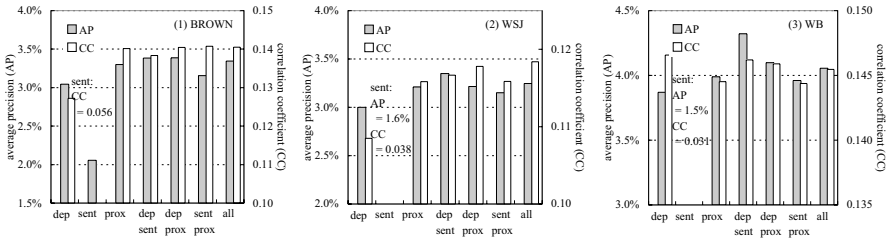


Fig. 1. Contextual information selection performances

makes us wonder what types of contextual information other than dependency are contained in the "difference" of two sets, and we suspect this remainder causes the significant improvement on the performance. In other words, there should be some useful contextual information contained in `prox` but not in `dep`.

We notice here that the word relations in `dep` are limited only to two words which have *direct* dependency between them, but there may be some words within the proximity window that indirectly have relations not captured by `dep`, e.g. a subject and an object sharing the same verb in a sentence. To capture this, we utilize this indirect dependency, which is detailed in the following section.

3 Indirect Dependency

This section describes the formalization of indirect dependency we adopted. Here we consider the dependency relations in a certain sentence s as a binary relation D over $W = \{w_1, ..., w_n\}$ i.e. $D \subset W \times W$, where $w_1, ..., w_n$ are the words in s. Since no words can be dependent or modifier of itself, D is irreflexive.

We define the composition of dependency $D^2 = D \circ D$ as *indirect dependency* where two words are related via two dependency relation edges. Each edge has labels assigned such as `subj` and `dobj` which specify what kind of syntactic relations the head and modifier possess. When an indirectly related pair $r_i \in D^2$ is composed from $r_j \in D$ with a label l_j and $r_k \in D$ with a label l_k, the label of r_i is also composed from l_j and l_k. We also define multiple composition of dependency recursively: $D^1 = D, \forall n > 1.\ D^n = D^{n-1} \circ D$. These are also indirect dependency relations in a broad sense. Notice here that D^n $(n > 1)$ can generally include reflexive relations, but it is clear that such relations don't serve as useful word features, so we re-define the composition operation so that the composed relation doesn't include any reflexive edges, i.e. $D \circ D - \{(w, w)|w \in W\}$.

4 Context Extraction

This section describes how to extract the contexts corresponding to direct and indirect dependency relations. First, the direct dependency is computed for each sentence, then the corresponding direct and indirect contexts are constructed from the dependency. As the extraction of comprehensive grammatical relations is a difficult task, RASP Toolkit was utilized to extract this kind of word relations. RASP analyzes sentences and extracts the dependency structure called grammatical relations (GRs). Take the following sentence for example:

3

(ncsubj be Shipment _)
(aux be have)
(xcomp _ be level)
(ncmod _ be relatively)
(ccomp _ level note)
(ncmod _ note since)
(ncsubj note Department _)
(det Department the)
(ncmod _ Department Commerce)
(dobj since January)

Fig. 2. Examples of extracted GRs

Shipment - (ncsubj be * _)
have - (aux be *)
be - (ncsubj * Shipment _)
be - (aux * have)
be - (xcomp _ * level)
be - (ncmod _ * relatively)
relatively - (ncmod _ be *)
⋮
since - (ncmod _ note *)
January - (dobj since *)
⋮

Fig. 3. Examples of contexts.

Shipments have been relatively level since January, the Commerce Department noted.

RASP extract GRs as n-ary relations as shown in Figure 2. While the RASP outputs are n-ary relations in general, what we need here is pairs of words and contexts, so we extract co-occurrences of words and direct contexts C^1 corresponding to D^1, by extracting the target word from the relation and replacing the slot by an asterisk "*", as shown in Figure 3. This operation corresponds to creating word-context pairs by converting a pair $r \in D^1$ of a head h and a dependent d with a label l_i into the pair $(h, l_i{:}d)$. If $(h, l_i{:}d) \in C^1$, then $(d, l_j{:}h) \in C^1$ also holds, where the label l_j is the *inverse* of l_i, as the two pairs `have - (aux be *)` and `be - (aux * have)` show in the figure. We treated all the slots except for head and modifier as the extra information and included them as the labels.

The co-occurrence of words and indirect contexts, C^2, which corresponds to indirect dependency D^2 is generated from C^1. For example, D^2 contains the indirect relation `Shipment - be - level` composed from `(ncsubj be Shipment _)` and `(xcomp _ be level)`. The context of `Shipment` extracted from this indirect relation is then formed by embedding the context of `be`: `(xcomp _ * level)` into the slot `be` of the context of `Shipment`: `(ncsubj be * _)`, which yields `Shipment - (ncsubj (xcomp _ * level) * _)`. Similarly, the indirect relation "`January` is the direct object of `since`, which in turn is modifying the verb `note`" is expressed as: `January - (dobj (ncmod _ note *) *)`.

Co-occurrences of indirect contexts C^n $(n \geq 3)$ corresponding to the multiple composition D^n are derived analogously. C^3, for example, is yielded just by embedding C^1 contexts into C^2 contexts shown in the previous example.

5 Synonym Acquisition Method

The purpose of the current study is to investigate the effectiveness of indirect dependency relations, not the language or acquisition model itself, we simply employed one of the most commonly used method: vector space model (VSM)

4

and tf.idf weighting scheme, although they might not be the best choice according to the past studies. In this framework, each word w_i is represented as a vector $\boldsymbol{w}_i$ whose elements are given by tf.idf, i.e. co-occurrence frequencies of words and contexts, weighted by normalized idf. That is, letting the number of distinct words and contexts in the corpus be N and M, co-occurrence frequency of word w_i and context c_j be $\mathrm{tf}(w_i, c_j)$,

$$\boldsymbol{w}_i = {}^t[\mathrm{tf}(w_i, c_1) \cdot \mathrm{idf}(c_1) \ ... \ \mathrm{tf}(w_i, c_M) \cdot \mathrm{idf}(c_M)], \tag{1}$$

$$\mathrm{idf}(c_j) = \frac{\log(N/\mathrm{df}(c_j))}{\max_k \log(N/\mathrm{df}(c_k))}, \tag{2}$$

where $\mathrm{df}(c_j)$ is the number of distinct words that co-occur with context c_j. The similarity between two words are then calculated using cosine of two vectors.

6 Evaluation

This section describes the two evaluation methods we employed — average precision (AP) and correlation coefficient (CC).

6.1 Average Precision

The first evaluation measure, average precision (AP), is a common evaluation scheme for information retrieval, which evaluates how accurately the methods are able to extract synonyms. We first prepare a set of *query words*, for which synonyms are obtained to evaluate the precision. We adopted the Longman Defining Vocabulary (LDV) [1] as the candidate set of query words. For each query word in LDV, three existing thesauri are consulted: Roget's Thesaurus [4], Collins COBUILD Thesaurus [2], and WordNet. The union of synonyms obtained when the query word is looked up as a noun is used as the reference set, except for words marked as "idiom," "informal," "slang" and phrases comprised of two or more words. The query words for which no noun synonyms are found in any of the reference thesauri are omitted. For each of the remaining query words, the number of which turned out to be 771, the eleven precision values at 0%, 10%, ..., and 100% recall levels are averaged to calculate the final AP value.

6.2 Correlation Coefficient

The second evaluation measure is correlation coefficient (CC) between the target similarity and the *reference similarity*, i.e. the answer value of similarity for word pairs. The reference similarity is calculated based on the closeness of two words in the tree structure of WordNet. More specifically, the similarity between word w with senses $w_1, ..., w_{m_1}$ and word v with senses $v_1, ..., v_{m_2}$ is obtained as follows. Let the depth of node w_i and v_j be d_i and d_j, and the maximum depth of the common ancestors of both nodes be d_{dca}. The similarity is then

$$sim(w, v) = \max_{i,j} sim(w_i, v_j) = \max_{i,j} \frac{2 \cdot d_{\mathrm{dca}}}{d_i + d_j}, \tag{3}$$

[1] http://www.cs.utexas.edu/users/kbarker/working_notes/ldoce-vocab.html.

5

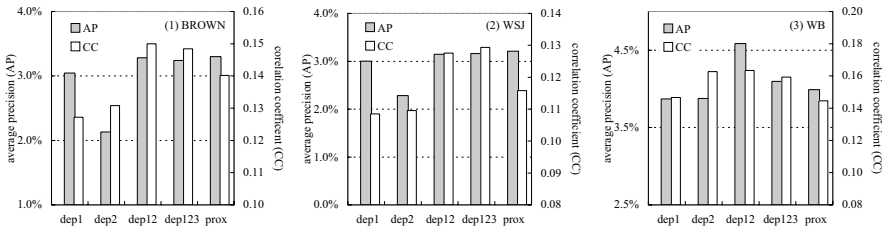


Fig. 4. Performance of the direct and indirect dependency relations

which takes the value between 0.0 and 1.0. Then, the value of CC is calculated as the correlation coefficient of reference similarities $\boldsymbol{r} = (r_1, r_2, ..., r_n)$ and target similarities $\boldsymbol{s} = (s_1, s_2, ..., s_n)$ over the word pairs in sample set P_s, which is created by choosing the most similar 2,000 word pairs from 4,000 random pairs. Every CC value in this paper is the average of 10 executions using 10 randomly created test sets to avoid the test-set dependency.

7 Experiments

Now we describe the evaluation results for indirect dependency.

7.1 Condition

We extracted contextual information from these three corpora: (1) Wall Street Journal (WSJ) (approx. 68,000 sentences, 1.4 million tokens), (2) Brown Corpus (BROWN) (approx. 60,000 sentences, 1.3 million tokens), both of which are contained in Treebank 3 [11], and (3) written sentences in WordBank (WB) (approx. 190,000 sentences, 3.5 million words) [2]. No additional annotation such as POS tags provided for Treebank was used. As shown in Sections 2 and 3, only relations (positions for `prox`) and word stems were used as context.

Since our purpose here is the automatic extraction of synonymous nouns, only the contexts for nouns are extracted. To distinguish nouns, using POS tags annotated by RASP, any words with POS tags APP, ND, NN, NP, PN, PP were labeled as nouns. We set a threshold t_f on occurrence frequency to filter out any words or contexts with low frequency and to reduce computational cost. More specifically, any words w such that $\sum_c \mathrm{tf}(w, c) < t_f$ and any contexts c such that $\sum_w \mathrm{tf}(w, c) < t_f$ were removed from the co-occurrence data. t_f was set to $t_f = 5$ for WSJ and BROWN, and $t_f = 15$ for WB.

7.2 Performance of Indirect Dependency

In this section, we experimented to confirm the effectiveness of indirect dependency. The performances of the following categories and combinations are evaluated: `prox`, C^1 (`dep1`), C^2 (`dep2`), $C^1 \cup C^2$ (`dep12`), and $C^1 \cup C^2 \cup C^3$ (`dep123`).

The evaluation result for three corpora is shown in Figure 4. We observe that whereas `prox` was better than the direct dependency `dep1` as shown in Section 2, the performance of the combination of direct and indirect dependency `dep12`

6

Table 1. Examples of acquired synonyms and their similarity for word "legislation".

dep1		dep12	
word	similarity	word	similarity
law	0.328	law	0.280
circumstance	0.242	money	0.229
auspices	0.239	plan	0.227
rule	0.225	issue	0.227
supervision	0.227	rule	0.225
pressure	0.226	change	0.222
condition	0.224	system	0.218
control	0.225	project	0.216
microscope	0.218	company	0.214
violence	0.209	power	0.212

was comparable to or even better than `prox`, and the improvement over `dep1` was dramatic. Table 1 shows the examples of extracted synonyms. It is seen that using `dep12` improves the result, and instead of less relevant words such as "microscope" and "violence", more relevant words like "plan" and "system" come up as the ten most similar words. Adding C^3 to `dep12`, on the other hand, didn't further improve the result, from which we can conclude that extending and augmenting C^1 just one step is sufficient in practice.

As for the data size, the numbers of distinct co-occurrences of `prox` and `dep12` extracted from BROWN corpus were 899,385 and 686,782, respectively. These numbers are rough approximations of the computational costs of calculating similarities, which means that `dep12` is a good-quality context because it achieves better performance with smaller co-occurrence data than `prox`. On the other hand, the numbers of distinct contexts of `prox` and `dep12` were 10,624 and 30,985, suggesting that the more diverse the contexts are, the better the performance is likely to be. This result was observed for other corpora as well, and is consistent with the one that we have previously shown [5], that is, what is essential to the performance is not the quality or the quantity of the context, but its diversity.

It is thus concluded that we can attribute the superiority of `dep12` to its potential to greatly increase the contextual information variety, and although the extraction of dependency is itself a costly task, adding the extra `dep2` is a very reasonable augmentation which requires little extra computational cost, aside from the marginal increase of the resultant co-occurrence data.

8 Conclusion

In this study, we proposed the use of indirect dependency composed from direct dependency to enhance the contextual information for automatic synonym acquisition. The indirect contexts were constructed from the direct dependency extracted from three corpora, and the acquisition result was evaluated based on two evaluation measures, AP and CC using the existing reference thesauri.

We showed that the performance improvement of indirect dependency over the direct dependency was dramatic. Also, the indirect contexts showed better

7

results when compared to surrounding words even with smaller co-occurrence data, which means that the indirect context is effective in terms of quality as well as computational cost. The use of indirect dependency is an very efficient way to increase the context variety, taking into consideration the fact that the diversity of contexts is likely to be essential to the acquisition performance.

Because we started from the "difference" of dependency relations and word proximity, the investigation of other kinds of useful contextual information should be conducted in the future. There are also some studies including Pado's [12] that make the most of dependency paths in the sentence, but their model does not take into account the dependency label. This increases the granularity of contexts and its effect is an open issue which we should bring up in another article. The application to other categories of words or the extraction of semantic relations other than synonyms is the future work.

References

1. Briscoe, T., Carroll, J., Watson, R.: The Second Release of the RASP System. Proc. COLING/ACL 2006 Interactive Presentation Sessions (2006) 77–80.
2. Collins.: Collins COBUILD Major New Edition CD-ROM. HyperCollins (2002).
3. Curran, James R., Moens, M.: Improvements in Automatic Thesaurus Extraction. Proc. SIGLEX (2002) 59–66.
4. Editors of the American Heritage Dictionary: Roget's II: The New Thesaurus, 3rd ed. Boston: Houghton Mifflin (1995).
5. Hagiwara, M., Ogawa, Y., Toyama, K.: Selection of Effective Contextual Information for Automatic Synonym Acquisition. Proc. COLING/ACL (2006) 353–360.
6. Harris, Z.: Distributional Structure. Katz, J. J. (ed.): The Philosophy of Linguistics, Oxford University Press (1985) 26–47.
7. Hindle, D.: Noun classification from predicate-argument structures. Proc. ACL (1990) 268–275.
8. Jing, Y., Croft, B.: An association thesaurus for information retrieval. Proc. RIAO (1994) 146–160.
9. Kojima, H., Ito, A.: Adaptive Scaling of a Semantic Space. IPSJ SIGNotes Natural Language, NL108-13, (1995) 81–88. (in Japanese)
10. Lin, D.: Automatic retrieval and clustering of similar words. Proc. COLING/ACL (1998) 786–774.
11. Marcus, M. P., Santorini, B., Marcinkiewicz, M. A.: Building a large annotated corpus of English: The Penn treebank. Computational Linguistics, 19(2) (1994) 313–330.
12. Pado, S., Lapata, M.: Constructing semantic space models from parsed corpora. Proc. ACL (2003) 128–135.
13. Ruge, G.: Automatic detection of thesaurus relations for information retrieval applications. Foundations of Computer Science: Potential - Theory - Cognition, LNCS, vol. 1337 (1997) 499–506.

8

Finding semantically related words in Dutch. Co-occurrences versus syntactic contexts.

Yves Peirsman[1,2], Kris Heylen[1] and Dirk Speelman[1]

[1] Quantitative Lexicology and Variational Linguistics (QLVL),
University of Leuven, Belgium
[2] Research Foundation – Flanders
yves.peirsman@arts.kuleuven.be
kris.heylen@arts.kuleuven.be
dirk.speelman@arts.kuleuven.be

Abstract. Despite the growing interest in distributional approaches to semantic similarity, the linguistic differences between their results are still unclear. In this paper we use six vector-based techniques to retrieve semantically related nouns from a corpus of Dutch, and investigate them from a computational as well as a linguistic perspective. In particular, we compare the results of a bag-of-word model with a syntactic model, and experiment with different context sizes and means of reducing the dimensionality of the vector space. We find that a full syntactic context model clearly outperforms all other approaches, both in its overall performance as in the proportion of synonyms it discovers.

1 Introduction

The automatic discovery of semantic similarity between nouns on the basis of a corpus has been a hot topic in recent years. It has applications in question answering, information retrieval, thesaurus extraction, parsing, and many other computational-linguistic tasks. Although in theory, electronic resources can be used for the extraction of semantically similar words, these are often incomplete or even unavailable for the language or domain at hand. The most popular automatic approaches are vector-based algorithms that rely on a large corpus to construct a vector with information about the contexts a word occurs in [1, 2]. They are based on the so-called *distributional hypothesis* [3], which states that words that appear in similar contexts will have related meanings. The semantic distance or similarity between two words can then be captured by the distance between their respective vectors.

Those computational-linguistic techniques can also be fruitfully exploited in theoretically oriented research. This paper reports on research carried out within the sem·metrix project, which aims to quantify the differences in word use between different varieties of Dutch. In particular, it investigates what lexemes are used to refer to a large number of concepts and how frequently those occur in the different varieties. In previous research [4], the sets of synonyms for each concept were constructed manually. As a result, the scope of the investigation

9

had to be rather limited. In order to extend this research to a larger number of concepts, the sem·metrix project now makes use of NLP approaches to retrieve semantically related words automatically.

The distributional methods described above allow for many different specific implementations. While their performance on a particular task is being investigated extensively, much less attention goes to the linguistic characteristics of their results. We therefore complement a study of the overall performance of the models with an investigation of the types of semantic relation they discover. With this goal in mind, we varied three of the models' main parameters: (1) the type of context features retrieved from the corpus, (2) the size of the context window and (3) the use of a dimensionality reduction technique.

After a discussion of the setup of our experiments in section 2, section 3 is concerned with the interpretation of the results. Section 4, finally, draws the main conclusions and wraps up with perspectives for future work.

2 Experimental setup

2.1 Types of context features

Broadly speaking, vector-based approaches to semantics witness a competition between two types of context features. First, in a *bag-of-word* or *co-occurrence* model, the vector of a target word records what words occur within a context window of n words on either side of the target. It often contains the frequencies of such co-occurrences, or a figure that expresses if the context word is found more often than expected. Such a bag-of-word model was used by Schütze [1] in his landmark paper in the field.[3] It is the first type of context model that we will investigate.

Second, in a *syntactic model*, the vector of a target word contains information about the words with which the target occurs in a syntactic relationship. This approach was taken by Lin [5] and Curran and Moens [6] for English, and by Van der Plas and Bouma [7] and Van de Cruys [8] for Dutch. Padó and Lapata [2] compared the results of such a syntactic model with a bag-of-word approach, and found the syntactic model to be superior.

These syntactic models still allow much freedom in the type of syntactic relations that are considered. Our implementation took into account eight different types of syntactic dependency relations, in which the target noun could be

- subject of verb v,
- direct object of verb v,
- prepositional complement of verb v introduced by preposition p,
- the head of an adverbial prepositional phrase (PP) of verb v introduced by preposition p,
- modified by adjective a,

[3] It should be noted, however, that Schütze worked with so-called *second-order* co-occurrences, which model a word in terms of the context words of its context words.

10

– postmodified by a PP with head n, introduced by preposition p,
– modified by an apposition with head n, or
– coordinated with head n.

Each specific instantiation of the variables v, p, a, or n was responsible for a new context feature.

2.2 Size of the context window

In a bag-of-word model, it is also possible to vary the size of the context window around the target word. While a technique like LSA [9] looks at words in the entire paragraph, Schütze [1] defined a context as twenty-five words on either side of the target, and Lund and Burgess [10] worked with fewer than ten words. We experimented with two context sizes: one with five words on either side of the target, one with fifty. For the fifty-word context window, we kept the dimensionality of the vectors relatively low, by only treating the 2,000 most frequent words in the corpus as possible context words, similar to Padó and Lapata [2]. For the five-word window, we experimented with this setup as well as with the full dimensionality.

2.3 Random Indexing

The feature vectors that can be obtained from a 300 million word corpus are massive, and the resulting computation cost is obviously very high. A number of techniques have been developed to deal with this computational inefficiency. One of those is Random Indexing (RI) [11]. RI may be less popular than Singular Value Decomposition (SVD) [12], but it has the advantage of bypassing the construction of an enormous co-occurrency matrix. While SVD reduces the dimensionality of the matrix after it has been made, RI does this during its construction.

Random Indexing creates a so-called *index vector* for each contextual feature, whose length is much smaller than the total number of contextual features. This vector contains a large number of 0s, and a small number of randomly distributed $+1$s and -1s. The context vector of a word is then constructed by summing the index vectors of all its contextual features. During this process, each index vector can optionally be weighted according to the statistical behaviour of its feature and the current target [13]. For our experiments, however, we implemented basic Random Indexing, which simply weights each index vector by the frequency of its feature in the context of the target word. We used index vectors of length 1,800 with eight non-zero elements.

2.4 Other parameters and evaluation

The rest of the setup was left identical for all our experiments. As our data we used the parsed and lemmatized Twente Nieuws Corpus (TwNC), a 300 million word corpus of Dutch newspaper articles from 1999 to 2002. Our word vectors

11

did not contain the frequency of the context features, but their point-wise mutual information (PMI) with the target word. This statistic quantifies if the combination *(target word, context feature)* appears more often than expected on the basis of their individual frequencies in the corpus. Informative context features thus receive a higher value. The usefulness of this statistic was demonstrated by Van der Plas and Bouma [7]. Semantically empty words were automatically ignored (on the basis of a stoplist), as were co-occurrences with a frequency of less than five. For the syntactic features, we did not use a frequency cutoff.

For the evaluation of the algorithms, we randomly sampled 1,000 nouns from the corpus, making sure that they had an absolute frequency of at least 50 and also appeared in Dutch EuroWordNet. The former requirement was meant to sidestep data sparseness, while the latter ensured the possibility of automatic evaluation. Twenty-four of the 1,000 sampled words turned out not to have any features with a frequency of five or more for at least one of our models. This brought the final number of test words down to 976. Like Schütze [1], we used the cosine measure to find for each word its ten most similar words in the corpus. Here, too, we only looked at words with an absolute frequency of at least fifty.

3 Results and discussion

We compared a total of six models. First, we looked at the overall performance of each of the approaches, by investigating if the ten words that they returned are indeed semantically related to their targets. Second, we homed in on the semantic relations that the algorithms discover — synonyms, hypernyms, hyponyms or co-hyponyms — in order to find whether a certain model has a preference for a specific syntactic relation. Both types of evaluation used Dutch EuroWordNet [14] as a gold standard.[4]

3.1 Performance

For each of our 976 target words, the algorithm returned the ten most similar words in the corpus. On the basis of Dutch EuroWordNet, we computed the Wu & Palmer similarity score between each of these words and its target. This score captures the similarity between two words w_1 and w_2 on the basis of their relation in a lexical hierarchy [15]. In particular, it finds their lowest shared hypernym, h_l, and divides twice the depth of this hypernym in the hierarchy by the sum of the depths of w_1 and w_2:

$$s_{WP}(w_1, w_2) = \frac{2 \times depth(h_l)}{len(w_1, h_l) + len(w_2, h_l) + 2 \times depth(h_l)} \quad (1)$$

[4] We should add a word of caution here, since the coverage of Dutch EuroWordNet is not as high as that of its English counterpart. On average, EuroWordNet contains between 5.7 and 7.4 of the ten most related words that the algorithm finds. Despite this coverage problem, EuroWordNet is still the most common and straightforward means of evaluation for distributional algorithms.

12

	average Wu & Palmer scores		
	syntactic model	bag-of-word model	
		5-word window	50-word window
all features	0.48	0.34	—
Random Indexing	0.31	0.26	—
2,000 most frequent features	—	0.28	0.23

Table 1. Average Wu & Palmer similarity score for the ten most related words.

If a word in our output did not occur in EuroWordNet, it was simply ignored. If it appeared several times, the maximum Wu & Palmer score was taken.[5]

Table 1 sums up the average Wu & Palmer scores for the six models. These results show three important patterns. First, the syntactic context model performs better than the bag-of-word approach. Syntactic context features thus allow us to find words that are more closely related to the target word. Second, drastically reducing the dimensionality of the vector space (either by Random Indexing or by a cutoff at the 2,000 most frequent words) brings down performance considerably. This indicates that the algorithm had better take into account as much information contained in the corpus as possible. Third, with 2,000 features, the fifty-word context model performs less well than the five-word one. We believe this is the case because the fifty-word model finds more *loose* semantic associations (e.g. *car – road*), while the five-word model discovers more *tight* semantic relations (e.g. *car – vehicle*). This hypothesis follows from the fact that a narrow context will contain a higher proportion of words that are also syntactically related to the target word, and hence, more linguistic information about the target. This deserves more careful investigation, however.

In order to see if the full syntactic and bag-of-word models score well on the same words, we calculated the correlation between the average Wu & Palmer scores for the best two models. In order to make the figures more robust, the rest of this section looks only at target words with five or more of their ten nearest neighbours in EuroWordNet (552 targets). Spearman's correlation statistic between the results was 69.9%. This suggests that the difficulty of a word is indeed fairly independent of the type of context features that the algorithm uses.

Still, there are differences between the models' behaviour. In particular, the full syntactic context model gives a correlation of 38.8% between the average Wu & Palmer similarity of the ten nearest neighbours and the depth of the target in the EuroWordNet hierarchy.[6] For the bag-of-word model, the same statistic is only 28.3%. Both models thus work better for words deeper in the hierarchy, but the performance of the bag-of-word model decays less with decreasing depth.

[5] In the case of a polysemous word, we are only interested in the meaning that is most related to the target word.

[6] If a word appeared several times in EuroWordNet, we used its minimum depth.

13

3.2 Semantic relations

The ultimate goal of our project is to retrieve words that have a specific semantic relation to the target word. We would thus like to gain more insight in the types of semantic relations that the algorithms find. We therefore checked against EuroWordNet what relation, if any, each of the 9760 retrieved words has with its target word. We took the following semantic relations into account:[7]

- synonymy: the retrieved word co-occurs in a synset together with the target.
- hyponymy: the retrieved word occurs in a synset that is a direct daughter of (one of) the target's synset(s).
- hypernymy: the retrieved word occurs in a synset that is the direct mother of (one of) the target's synset(s).
- cohyponymy: the retrieved word occurs in a synset that is a direct daughter of one of the target's hypernym synsets as defined above.

Table 2 gives the absolute and relative frequencies of the relations found with the different models, summing over all the target words. Because of the limited coverage of EuroWordNet, not all retrieved top ten similar words could be evaluated against the thesaurus. The percentages of related words are therefore relative to the number of retrieved words present in EuroWordNet (last column). As above, the syntactic context model outperforms all other approaches, both in the overall proportion of semantically related words (31.4%) as in the percentage of retrieved words with a specific semantic relation (e.g. 6.3% synonyms[8]). The other relations between the context models mirror those in the previous section.

Moreover, a chi-square test shows that there is an interaction between the type of context model and the frequency of the semantic relations ($\chi^2 = 49.65$, $df = 15$, $p < 0.001$). Table 3 gives the differences between the observed frequencies of the semantic relations and their expected frequencies. It shows that the full syntactic model returns significantly more synonyms and hyponyms than expected under independence, but far fewer cohyponyms. Interestingly, all models with a reduced dimensionality find fewer synonyms than expected, but more cohyponyms. In three of the four cases, this number of retrieved cohyponyms is significantly higher than expected. Severely reducing the dimensionality of the word vectors thus leads to a retrieval of more loosely related words. One of the reasons for this finding may lie in the informativeness of relatively low-frequent context features that are highly correlated with the occurrence of the target word. While these features receive a high PMI value in the full models, they are weighted by their frequency in the Random Indexing setup, or even simply

[7] Since our system did not disambiguate between the different senses of a word, all synsets in which a retrieved word or a target word appeared were taken into consideration. When a word had several relations with the target, only the closest relation was added to the tally, with synonymy > hyponymy > hypernymy > cohyponymy.

[8] The percentage of synonyms is rather low for all systems, but this might be partly due to the cut-off at ten most similar words. Not many words have ten synonyms, while the number of potential (co)hyponyms is often far larger.

14

	syno.	hypo.	hyper.	cohyp.	all 4	in EWN
syn	392 (6.3)	249 (4.0)	262 (4.2)	1050 (16.9)	1953 (31.4)	6215
bow, 5w	236 (4.2)	150 (2.7)	156 (2.8)	686 (12.2)	1228 (21.8)	5645
syn, RI	154 (2.1)	94 (1.3)	141 (1.9)	624 (8.6)	1013 (13.9)	7275
bow, 5w, 2000	144 (2.5)	106 (1.8)	99 (1.7)	561 (9.8)	910 (15.9)	5739
bow, 5w, RI	135 (2.4)	75 (1.3)	118 (2.1)	467 (8.3)	795 (14.2)	5598
bow, 50w, 2000	106 (1.6)	60 (0.9)	76 (1.1)	416 (6.2)	658 (9.8)	6682

Table 2. Relations found among the ten most related words, if present in EuroWordNet, summed over all target words (percentage of row totals between brackets).

syn, bow: syntactic/bag-of-word context model
5w,50w: 5/50-word context window
RI: Random Indexing
2000: 2,000 most frequent words in corpus as context words

	syno.	hypo.	hyper.	cohyp.
syn	**44.41**	**30.38**	8.23	**-83.02**
bow,5w	17.44	12.54	-3.56	-26.42
syn,RI	**-26.29**	**-19.40**	9.37	**36.31**
bow,5w,2000	-17.96	4.13	**-19.24**	**33.07**
bow,5w,RI	-6.49	-13.99	14.70	5.79
bow,50w,2000	-11.11	-13.66	-9.50	**34.27**

Table 3. Differences between observed frequencies and expected frequencies on the basis of a chi-square test for Table 2. Significant differences are shown in bold.

ignored when only the 2,000 most frequent corpus words are taken into account. When the algorithm is to find tight semantic relations like synonymy, the statistical relationship between a target word and each of its features may thus play an important role.

4 Conclusions and future work

In this paper, we have compared six distributional approaches that find the most similar words for any given target on the basis of a corpus. In particular, we have contrasted the impact of three parameters: (1) the type of context feature (co-occurrences vs. syntactic features), (2) the size of the context window (five vs. fifty context words) and (3) the reduction of the dimensionality, either through Random Indexing or a frequency cutoff.

The full syntactic context model outperformed all other combinations, both in overall performance as in the number of synonyms it finds. Drastically reducing the dimensionality of the vector space brings down performance substantially, as does enlarging the context window of our bag-of-word model from five to fifty words. An analysis of the retrieved semantic relations showed that the

15

full models are more sensitive to synonymy relations, while those with reduced dimensionality are biased towards co-hyponyms.

In the future, we would like to experiment with more parameter settings and more advanced context models. In particular, we would like to investigate second-order co-occurrences [1] or indirect syntactic relations [2]. For all of these possible extensions, we are particularly interested in the linguistic implications of the different models: the types of words they are suited for, and the kind of semantic relations that they discover.

References

1. Schütze, H.: Automatic word sense discrimination. Computational Linguistics **24**(1) (1998) 97–124
2. Padó, S., Lapata, M.: Dependency-based construction of semantic space models. Computational Linguistics **33**(2) (2007) 161–199
3. Harris, Z., ed.: Mathematical Structures of Language. New York: Wiley (1968)
4. Geeraerts, D., Grondelaers, S., Speelman, D.: Convergentie en divergentie in de Nederlandse woordenschat. Een onderzoek naar kleding- en voetbaltermen. Meertens Instituut, Amsterdam (1999)
5. Lin, D.: Automatic retrieval and clustering of similar words. In: Proceedings of COLING-ACL98, Montreal, Canada (1998) 768–774
6. Curran, J.R., Moens, M.: Improvements in automatic thesaurus extraction. In: Proceedings of the Workshop on Unsupervised Lexical Acquisition (SIGLEX), Philadelphia, PA, USA (2002)
7. Van der Plas, L., Bouma, G.: Syntactic contexts for finding semantically related words. In: Proceedings of Computational Linguistics in the Netherlands 15. (2005) 173–186
8. Van de Cruys, T.: The application of Singular Value Decomposition to Dutch noun-adjective matrices. In: Actes de la 13e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), Leuven, Belgium (2006) 767–772
9. Landauer, T.K., Dumais, S.T.: A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. Psychological Review **104** (1997) 211–240
10. Lund, K., Burgess, C.: Producing high-dimensional semantic spaces from lexical co-occurrence. Behavior Research Methods, Instruments and Computers **27** (1996) 203–208
11. Kanerva, P., Kristoferson, J., Holst, A.: Random Indexing of text samples for Latent Semantic Analysis. In: Proceedings of the 22nd Annual Conference of the Cognitive Science Society, Philadelphia, PA, USA (2000) 1036
12. Golub, G.H., Van Loan, C.F.: Matrix Computations. London: The Johns Hopkins University Press (1989)
13. Gorman, J., Curran, J.R.: Random indexing using statistical weight functions. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP), Sydney, Australia (2006) 457–464
14. Vossen, P., ed.: EuroWordNet: a multilingual database with lexical semantic networks for European Languages. Kluwer, Dordrecht (1998)
15. Wu, Z., Palmer, M.: Verb semantics and lexical selection. In: Proceedings of the 32nd Annual meeting of the Association for Computational Linguistics, Las Cruces, NM (1994) 133–138

16

Finding Translations for Low-Frequency Words in Comparable Corpora

Viktor Pekar[1], Ruslan Mitkov[1] Dimitar Blagoev[2], and Andrea Mulloni[1]

[1] ILP, University of Wolverhampton, WV1 1SB, United Kingdom
{V.Pekar,R.Mitkov,Andrea2}@wlv.ac.uk
[2] University of Plovdiv, Department of Informatics, 4003 Plovdiv, Bulgaria
gefix@pu.acad.bg

Abstract. The paper proposes a method to improve the extraction of low-frequency translation equivalents from comparable corpora. Prior to performing the mapping between vector spaces of different languages, the method models context vectors of rare words using their distributional similarity to words of the same language to predict unseen co-occurrences as well as to smooth rare, unreliable ones. Our evaluation shows that the proposed method delivers a consistent and significant improvement on the conventional approach to this task.

1 Introduction

The distributional hypothesis, the idea that words with similar meaning tend to occur in similar contexts, can be extended to the bilingual scenario, so that distributional similarity between words of different languages is used to discover translationally equivalent words. The assumption underlies a growing body of research into automated compilation of bilingual lexicons from bilingual comparable corpora [3, 4, 7, 9, among others]. The general procedure implementing this idea begins by collecting co-occurrence data on words of potential interest from monolingual corpora and representing them as context vectors. After that context vectors of different languages are mapped onto a single vector space using a bilingual dictionary. Translation equivalents are then retrieved as pairs of words from different languages that have the greatest similarity of their vectors.

A well-known limitation of this approach is that it performs quite unreliably on all but the most frequent words. Even if one ensures that the bilingual dictionary has sufficient coverage so that every occurrence pattern is matched with its equivalent in the other language, a lot of evidence on the words is still lost because of the many-to-many mapping between the two sets of co-occurrence features, resulting from polysemy and synonymy in both languages. As a consequence, only frequent words remain relatively robust against the noise introduced during translation.

In this paper we aim to improve the accuracy of retrieval of translation equivalents for rare words from comparable corpora. We describe an extension of the similarity-based method for estimating word co-occurrence probabilities [2] to the problem of modeling context features of rare words prior to translating their vectors into the vector space of a different language.

17

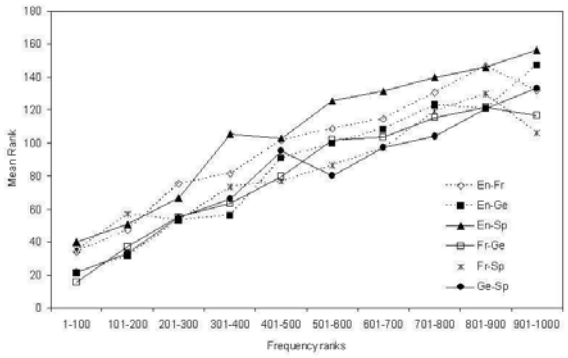


Fig. 1. The performance of the standard algorithm with respect to words with different corpus frequencies. The x-axis shows frequency ranks of source words, the y-axis – the mean rank of their correct translations as assigned by the algorithm.

2 Dealing with Data Sparseness

To verify the effect of word frequency on the standard algorithm, we run a pilot experiment on six pairs of comparable corpora in different languages. We extracted a sample of 1000 pairs of translation equivalents from each pair of corpora and divided them into 10 equal-size bands according to their frequency (Section 5 contains a detailed description of this experimental setup). Figure 1 depicts the mean rank achieved by the algorithm for each language pair, within each frequency band.

For all the language pairs, we indeed find large differences in the algorithm's performance in relation to words belonging to different frequency ranges. For example, for the most frequent words in the sample (frequency ranks 1 to 100), the correct equivalent typically appears at ranks between 20 and 40, while for the least frequent ones, one can expect to find it only between ranks 100 and 160. The shapes of the performance functions are also very much the same across the language pairs.

This observation calls for a certain way to estimate the probability of occurrence of rare words in contexts where they failed to occur or occurred too few times. Overcoming data sparseness by smoothing corpus frequencies is a familiar problem in NLP, with some techniques, such as Good-Turing and Katz's back-off, being the standard approaches. Comparative studies of methods for smoothing bigrams [1, 2, 5] suggest that class-based smoothing, and distance-based averaging, and methods for reconstructing word frequencies from the web are among the best choices. Class-based smoothing [8] relies on a broad-coverage taxonomy of semantic classes. Such resources may not be readily available for a given language, and dependence on them would greatly limit the portability of the overall approach. Web-based estimation of bigram counts [5] appears impractical for a large-scale smoothing exercise. Therefore in this study, we opt for distance-based averaging techniques.

18

3 Distance-Based Averaging

In the distance-based averaging framework [2], the probability of co-occurrence of two words is modeled by analogy with other words that are distributionally similar to the given ones. In this study we employ the nearest neighbor variety of the approach, where the set of neighbors, i.e. distributionally similar words, is created ad hoc for each bigram, rather than using fixed sets of similar words obtained by clustering.

If the probability of a word n appearing with a context word v cannot be estimated because of a zero co-occurrence count, the nearest neighbor method computes the estimate $p^*(v|n)$ as a weighted average of known probabilities $p(v'|n)$, where each n' is a close neighbor of n. The weight with which each neighbor influences the average is determined by its similarity to n:

$$w(n, n') = 10^{-\beta \cdot SimScore(n,n')} \tag{1}$$

where β is a parameter that diminishes the effect of distant neighbors (in our experiments fine-tuned to .13). The probability estimate is calculated based on K nearest neighbors as follows (K is set experimentally):

$$p^*(v|n) = \sum_{n' \in K} \frac{p(v|n') \cdot w(n, n')}{norm(n)} \tag{2}$$

where $norm(n) = \sum_{n' \in K} w(n,n')$ is a normalization factor used to ensure that the conditional probabilities for n sum to 1.

4 Constructing Smoothed Context Vectors

We wish not only to predict probabilities for unseen co-occurrences, but also to smooth known, but unreliable probabilities for low frequency words. In the latter case, the corpus-estimated probability $p(v|n)$ participates in the calculation of the average p^*, with the weight γ:

$$p^*(v|n) = \gamma \cdot p(v|n) + (1 - \gamma) \cdot \sum_{n' \in K} \frac{p(v|n') \cdot w(n, n')}{norm(n)} \tag{3}$$

Here, γ controls the amount by which the corpus-estimated probability is smoothed. We believe that γ should be a function of the frequency of n: the less frequent is n, the more its corpus-estimated probabilities should be smoothed with data from its neighbors. We propose and evaluate two ways to estimate this function.

The first one is a heuristic that computes γ as a ratio between the log-transformed counts of n and the most frequent word in the data. This has the effect that the most frequent word will not be smoothed at all, while the least frequent ones will be mainly estimated from the data on their neighbors:

$$\gamma = \frac{\log f(n)}{\log \max_{x \in N} f(x)} \tag{4}$$

19

The second method estimates γ based on the performance of the algorithm on a held-out set of translation equivalents. First, the held-out word pairs divided into a number of frequency ranges are used to find out the mean rank of the correct translation for each frequency range. Then, function $g(x)$ is interpolated along the points corresponding to the mean ranks in order to predict the mean rank for a certain novel word, given its frequency. γ is then determined as the ratio between the predicted rank of n and that of the most frequent word in the data:

$$\gamma = \frac{g(n)}{g(\max_{x \in N} f(x))} \tag{5}$$

Another modification of the standard algorithm we introduce aims to capture the intuition that infrequent neighbors are likely to decrease the quality of the smoothed vector, because of their unreliable corpus-estimated probabilities. We study the effect of discarding those neighbors that have a lower frequency than the word being smoothed.

5 Experimental Setup

Dictionary. We evaluate the proposed method on translationally equivalent nouns in six language pairs, all pairwise combinations between English, French, German and Spanish. As the gold standard, we use pairs of nouns extracted from synsets and the multilingual synset index in EuroWordNet (EWN)[3]. In a similar manner we extracted pairs of equivalent verbs from EWN for the six language pairs. These were used to construct the translation matrix necessary for mapping context vectors into different languages. If, during the translation, a context word that had multiple equivalents in the target language according to the dictionary, it was mapped into all its equivalents, with its original probability equally distributed among them.

Corpus Data. As comparable corpora, we use newspaper texts from the *Wall Street Journal* (1987-89) for English, *Le Monde* (1994-96) for French, *die tageszeitung* (1987-89 and 1994-98) for German, and *EFE* (1994-95) for Spanish. The English and Spanish corpora were processed with the Connexor FDG parser, French with Xerox Xelda, and German with Versley's parser. From the parsed corpora we extracted verb–direct object dependencies, where the noun is the head of the modifier phrase.

Evaluation Nouns. To ensure that the evaluation data for all the language pairs contain an equal number of nouns from similar frequency ranges, we used the following sampling procedure. For each language pair, we first created a list of all translation equivalents that are present both in EWN and in both monolingual corpora with frequency above 5. The pairs were then sorted according to the count of the noun which was less frequent of the two, on the assumption that the less frequent word is the better indicator of the difficulty of finding its equivalent. After that, 1000 pairs were selected from equidistant locations in this ordered list, and divided into 10 equal-size frequency bands, such that the first band included the top 100 most frequent pairs, the second one – pairs with frequency ranks between 101 and 200, and so on.

Assignment Algorithm. Once the similarities between the source word and the target words have been computed (we use the Jensen-Shannon Divergence to measure

[3] http://www.illc.uva.nl/EuroWordNet/

20

(dis)similarity of context vectors, for a discussion of the function, cf. [2]), the problem is to select the most likely translation for the source word. To determine optimal assignment for the entire set of source words, we employ the Hungarian (also known as Kuhn-Munkres) algorithm [6], which efficiently finds such matching of source and target words that maximizes the sum of similarity scores in the bipartite graph made up of the two sets of words.

Evaluation Measure. Following the evaluation procedure adopted in [10], we note the system-assigned rank of the correct translation for each source word and compute a mean rank over all the pairs in sample. A mean rank appears an intuitive evaluation measure, since it describes how soon a correct translation for a source word can be found by a lexicographer who revises translations proposed by the system.

Baseline. The baseline in our experiments is the standard algorithm without any smoothing of the data. Its performance achieved on different language pairs with respect to different frequency bands is shown in Figure 1. In the following sections, we report differences to the baseline attained by configurations of the extended algorithm.

6 Results

6.1 Nearest Neighbors Smoothing

We first examined how nearest neighbors smoothing affects the performance of the standard algorithm. The smoothing of the probability in the vector for each noun was carried out according to Equation 3, with γ set to 0 and the noun being smoothed was included into the nearest neighbor set. The nearest neighbors are determined from the entire set of nouns extracted from the monolingual corpus, not only from nouns included into the evaluation sample. In the experiment, we varied k, the number of nearest neighbors, between 1 and 1000. Table 2 shows the differences in the mean rank achieved by the most optimal values of k in comparison to the baseline algorithm.

Most of the time smoothing noun vectors with nearest neighbors resulted in a higher mean rank in comparison to the baseline, i.e., the performance degraded. While there are a few ranges for some language pairs where a lower mean rank was reached, the average over frequency ranges was higher than that of the baseline, with the exception of the German-Spanish pair where it was only slightly lower.

6.2 Ignoring Less Frequent Neighbors

Our next experiment consisted in smoothing vectors as in the previous experiment, but excluding those nouns from the set of nearest neighbors that had corpus frequency below that of the noun being smoothed. After infrequent nearest neighbors have been removed, the set of neighbors has been expanded accordingly. Table 3 shows the results.

The removal of infrequent neighbors resulted in a noticeably better performance in lower frequency ranges: for ranges 301-400 and above the reduction was generally more than 10 points for all language pairs. In the top two ranges, smoothing still often led to higher mean ranks.

Considering the performance on the entire sample (the last row in the table), discarding infrequent neighbors entailed a modest reduction of the mean rank wrt

21

Table 1. Changes of the mean rank of the correct translation with respect to the baseline after nearest-neighbor smoothing.

	En-Fr	En-Ge	En-Sp	Fr-Ge	Fr-Sp	Ge-Sp
1-100	14.6	8.7	13.4	6.1	4.9	6.6
101-200	10.5	11.3	7.3	1.9	-3.0	6.2
201-300	9.2	2.3	18.0	-5.7	-5.7	-7.7
301-400	14.5	3.8	8.7	-2.4	5.5	-12.2
401-500	16.3	14.3	13.4	2.7	10.9	-13.7
501-600	24.9	7.5	9.3	-0.6	4.4	1.4
601-700	9.4	2.4	6.6	14.2	9.5	12.2
701-800	25.9	12.6	13.2	17.2	-4.4	2.4
801-900	14.8	10.8	14.8	5.1	-3.8	4.7
901-1000	19.2	2.6	16.4	6.8	6.9	-2.0
Average	15.9	7.6	12.1	4.5	2.5	-0.2

the baseline for all the language pairs (between 0.7 and 15.1 points, .9% and 18%). According to a two-tailed paired t-test[4], the reduction was significant in three pairs at $p < .001$ (French-German, French-Spanish, German-Spanish), but in the other three pairs the test failed to show any significance of the improvement. In all the following experiments, less frequent neighbors were excluded from the set of nearest neighbors.

6.3 Heuristical Estimation of γ

We next examined the performance of the algorithm when γ in Equation 3 was set to be a function of the frequency of the noun being smoothed. Table 4 describes the mean ranks achieved when γ was calculated heuristically according to Equation 4.

We see that making γ dependent on the frequency of the word being smoothed leads to even better results. With the exception of the most frequent band, all frequency ranges for all language pairs demonstrate lower mean ranks compared with the baseline. In general, it seems that better improvement are achieved on words with lower frequencies: while for the 101-200 range the improvement is under 10 points, for the 201-300 range, it is between 10 and 20 points, and for ranges above 301 it is often over 30 points.

Comparing the mean rank on the entire sample against the one achieved with the baseline, we see improvement for all the language pairs. The improvement is statistically significant at $p < .001$ across for all pairs.

6.4 Performance-Based Estimation of γ

We then examined the alternative method to compute γ, based on a function estimated from the performance of the method on a held-out set of words (Equation 5). These results are similar to those obtained with the heuristical computation of γ: infrequent

[4] df = 1000 in all the tests reported below.

22

Table 2. Changes of the mean rank wrt the baseline, after the removal of infrequent neighbors.

	En-Fr	En-Ge	En-Sp	Fr-Ge	Fr-Sp	Ge-Sp
1-100	2.3	9.1	10.5	4.7	3.7	5.5
101-200	1.5	8.2	4.2	-7.3	-2.8	-2.4
201-300	-1.4	-4.7	4.7	-9.5	-10.6	-11.8
301-400	-11.1	-11.3	-10.0	-22.4	-7.6	-20.2
401-500	-18.7	-13.5	-10.2	-20.2	-7.0	-37.1
501-600	-9.1	-14.2	-9.1	-35.3	-16.5	-15.0
601-700	-0.2	-7.5	-25.9	-22.6	-21.1	-23.6
701-800	-5.1	-12.2	-6.4	-17.9	-34.4	-30.0
801-900	-10.4	-9.8	-4.7	-24.8	-25.7	-32.7
901-1000	-13.6	-26.7	-12.1	-15.6	-4.9	-27.4
Average	-1.0	-0.7	-1.6	-13.5	-8.9	-15.1

Table 3. Changes of the mean rank for the heuristical estimation of γ wrt the baseline.

	En-Fr	En-Ge	En-Sp	Fr-Ge	Fr-Sp	Ge-Sp
1-100	1.1	1.8	11.2	-0.3	-0.1	3.9
101-200	-4.2	-2.0	-3.1	-10.6	-6.6	-5.5
201-300	-13.4	-17.9	-6.9	-20.1	-15.0	-15.8
301-400	-24.0	-22.6	-23.4	-29.0	-15.9	-30.2
401-500	-36.9	-31.7	-25.0	-35.9	-17.0	-45.0
501-600	-38.7	-41.4	-30.2	-49.1	-29.6	-30.9
601-700	-36.0	-39.5	-39.5	-40.3	-33.3	-33.5
701-800	-39.2	-47.2	-30.1	-37.8	-41.3	-38.2
801-900	-39.4	-34.8	-20.4	-41.8	-31.3	-45.9
901-1000	-32.3	-47.8	-33.1	-32	-15.8	-34.6
Average	-23.3	-26.0	-16.9	-27.7	-18.4	-25.3

nouns tend to benefit more from this smoothing technique, and only in the topmost range the performance shows a slight degradation. Considering the mean rank for the entire sample, it is also significantly lower than the baseline, at $p < .001$ for all the language pairs. Comparing the two ways to compute γ, we find that the heuristical approach delivers consistently lower mean ranks, the difference being significant for all the language pairs at $p < .025$.

7 Conclusion

Our study was carried out in the framework which models translational equivalence between words via similarity of their occurrence patterns found in monolingual corpora. In order to improve the retrieval of equivalents for low-frequency words, which are particularly vulnerable to noise introduced during the cross-linguistic mapping

23

Table 4. Changes of the mean rank for the performance-based estimation of γ wrt the baseline.

	En-Fr	En-Ge	En-Sp	Fr-Ge	Fr-Sp	Ge-Sp
1-100	5.1	2.0	12.7	-2.8	0.2	3.4
101-200	-2.6	1.4	-2.2	-9.9	-6.5	-5.7
201-300	-11.6	-16.6	-5.0	-20.3	-15.3	-14.5
301-400	-24.2	-22.2	-23.1	-29.8	-14.9	-29.0
401-500	-38.2	-32.7	-24.6	-37.7	-17.3	-45.4
501-600	-37.7	-45.2	-29.7	-55.7	-29.2	-31.0
601-700	-33.0	-39.9	-40.4	-43.2	-34.3	-33.2
701-800	-32.6	-49.3	-25.8	-33.5	-40.1	-37.7
801-900	-31.4	-33.6	-13.6	-36.3	-29.8	-43.2
901-1000	-18.8	-46.7	-30.6	-27.7	-10.4	-35.9
Average	-17.2	-23.5	-14.2	-26.3	-16.7	-24.5

of context vectors, we proposed a method which models occurrence patterns of words on analogy with words that are distributionally similar to them. The method extends the distance-based averaging technique to predict not only unseen word co-occurrences, but also to obtain more reliable probability estimates for rare bigrams. Our experimental evaluation has showed that the method yields a significant improvement on the conventional approach in relation to low frequency words and has a considerable positive effect on the overall retrieval accuracy.

References

1. Carsten Brockmann and Mirella Lapata. Evaluating and combining approaches to selectional preference acquisition. In *Proc. EACL*, pages 27–34, 2003.
2. Ido Dagan, Lillian Lee, and Fernando C. N. Pereira. Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34(1-3):43–69, 1999.
3. Hervé Déjean, Éric Gaussier, and Fatiha Sadat. An approach based on multilingual thesauri and model combination for bilingual lexicon extraction. In *Proc. COLING*, 2002.
4. Pascale Fung and Kathleen McKeown. Finding terminology translations from non-parallel corpora. In *The 5th Annual Workshop on Very Large Corpora*, pages 192–202, 1997.
5. Frank Keller and Mirella Lapata. Using the web to obtain frequencies for unseen bigrams. *Computational Linguistics*, 29(3):459–484, 2003.
6. Harold W. Kuhn. The Hungarian Method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.
7. Reinhard Rapp. Automatic identification of word translations from unrelated English and German corpora. In *Proc. ACL*, 1999.
8. Phillip Resnik. *Selection and Information: A Class-Based Approach to Lexical Relationships*. PhD thesis, University of Pennsylvania, 1993.
9. Xavier Robitaille, Yasuhiro Sasaki, Masatsugu Tonoike, Satoshi Sato, and Takehito Utsuro. Compiling French-Japanese terminologies from the web. In *Proc. EACL*, 2006.
10. Takehito Utsuro, Takashi Horiuchi, Takeshi Hamamoto, Kohei Hino, and Takeaki Nakayama. Effect of cross-language IR in bilingual lexicon acquisition from comparable corpora. In *Proc. EACL*, pages 355–362, 2003.

24

Part of Speech Filtered Word Spaces

Klaus Rothenhäusler and Hinrich Schütze

Institute for NLP, University of Stuttgart,
Azenbergstr. 12, 70 174 Stuttgart, Germany

Abstract We present a novel way to use linguistic information in the construction of word spaces. In our current experiments we filter the space dimensions for part of speech constructing different subspaces for different part of speech groups. In this paper we evaluate the overlap of the subspaces and present results for tests with association norms and the TOEFL synonym test.

1 Introduction

Most attempts to incorporate part of speech information when constructing word spaces have met with little success or even failure. We believe that this is in part due to the fact that the systems that made use of this information did so in a rather unprincipled fashion. On the other hand we see it as a general flaw of the research in this field that a single representation is used for all purposes a word space can be put to. We propose to build multiple representations for a single word with the help of linguistically annotated corpora currently only using parts of speech. The different representations may be selected according to the task at hand and the representation that suits.

The motivation for constructing different word spaces according to parts of speech was that, depending on the word to be represented, the context that best captures its meaning will differ. A noun may be characterised most significantly by the adjectives that modify it and the nouns modified by an adjective may in turn be more important to describe its meaning than the verbs in its vicinity for instance. We use parts of speech instead of real head-modifier relations because they are more easily available and may still approximate the same information.

2 Related Research

In our use of linguistic information we are closest in spirit to the proposals of Hindle (1990) and the ones following in his wake. Hindle computes different object and subject similarities for a noun from automatically assigned predicate argument structures. He only computes similarities for nouns though. In the same vein follows Lin (1998) who uses dependency relations to characterise words. Similarities are computed over a single representation comprising all dependency relations. This differs from Hindle who computes different subject and object similarities and then combines these. Padó and Lapata (2003) can be seen as a

25

direct extension of Lin's approach that includes longer reaching dependencies via the definition of dependency paths. Additionally, their system allows for much finer grained control over the inclusion of different kinds of dependencies. While this absolutely concurs with our view, their interest lies more in integrating different representations into a single overall similarity measure while we rather try to select a single appropriate representation.

All these systems are much more sophisticated in their use of dependency relations in comparison to just using parts of speech to include linguistic information into word spaces. However, we see the different part of speech spaces as cheap approximations of more realistic modification relations as detailed above.

Karlgren and Sahlgren (2001), Widdows (2003) and Wiemer-Hastings and Zipitria (2001) share with us the use of part of speech tags in the construction of word spaces. But in contrast to our approach they use the tagged words to construct one overall word space effectively just inflating the vocabulary.

Whereas Rapp (2002) and Sahlgren (2006) made important contributions to shed light on the differences between word spaces constructed from syntagmatic and paradigmatic contexts we focus on different paradigmatic contexts and try to deliberately vary them to capture different information.

3 Experimental Setup

Word-by-word matrices were constructed from the British National Corpus where the words serving as dimensions were filtered for part of speech. For the experiments the word class annotation provided in the BNC was used. Filters for four different groups were identified: a noun space including words tagged as NN, a verb space (VV), an adjective space (AJ) and an adverb space (AV). Number and gender information was disregarded by cutting off the last character of the part of speech tags. For the construction of the word spaces the BNC was lemmatised.

For most of the experiments described in this paper relatively small word spaces were constructed comprising 20 000 word vectors with a dimensionality n of 1 000. The words for which vectors were computed were chosen as follows: First of all the words from the USF Free Association Norms (Nelson et al., 1998) and the TOEFL data set were added because they were to be used for evaluation as described in the next section. Together they comprise a vocabulary of 10 007 words of which 684 did not occur in the BNC and were therefore dropped. The remaining 9 323 terms were tagged and lemmatised according to the frequencies in the BNC. Having added these a list of approximately 500 stop words was removed from the vocabulary and the most frequent ones of the remaining were chosen until the 20 000 were full.

For the dimensions the most frequent words in the respective part of speech groups were selected. This relatively humble size seems justified for the exploratory nature of this study. We preferred index term selection via sheer frequency to more informed methods in order to avoid the construction of the whole word-by-word matrix for efficiency reasons.

26

So what we end up with are four different kinds of word spaces whose dimensions are filtered to include only one kind of word class. All these spaces contain vectors for the same words so that there is a different representation for every word in each of the spaces.

For comparison we also constructed two unfiltered word spaces: one with part of speech tagged lemmas (+pos) and one with plain lemmas (−pos). Thus, in the first space like in all the filtered spaces, for a token like '`suit`' there might be representations for (`suit`, NN) and (`suit`, VV) while in the untagged space there would only be a single representation for '`suit`'. The characteristics of the different spaces are summarized in Table 1.

Table 1. Characteristics of word spaces used in the experiments. All spaces have $n = 1000$ index words as dimensions. Set A contains 20 000 part of speech tagged lemmas selected as described in the text, set B contains the same number of untagged lemmas, which may of course differ apart from the evaluation vocabulary.

identifier	**index words (= dimensions)**	**set represented**
noun space	most frequent lemmatised nouns	A
verb space	most frequent lemmatised verbs	A
adjective space	most frequent lemmatised adjectives	A
adverb space	most frequent lemmatised adverbs	A
unfiltered space (+pos)	most frequent tagged lemmas of all parts of speech	A
unfiltered space (−pos)	most frequent untagged lemmas of all parts of speech	B

Term frequency counts tf were logarithmically dampened by

$$\text{dampen}(tf) = \log(tf + 1)$$

yielding weights w_i that were normalised by dividing through vector length computed as

$$||\boldsymbol{w}|| = \sqrt{\sum_{i=1}^{n} w_i^2}$$

Thus the computation of cosine similarities that were used throughout the experiments is reduced to the calculation of an inner product between word vectors.

We used symmetric context windows of small size, namely 1–10 words, in order to include only contexts that contain direct modification relations. To specify context windows we write pairs of the form "n+n" where n is the number of words to the left and to the right treated as context.

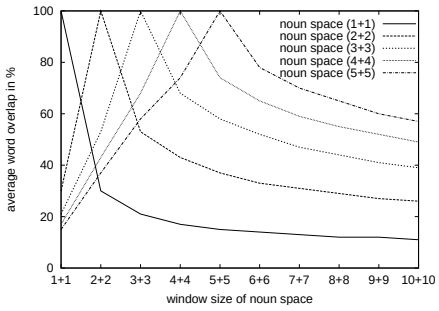
27

4 Evaluation

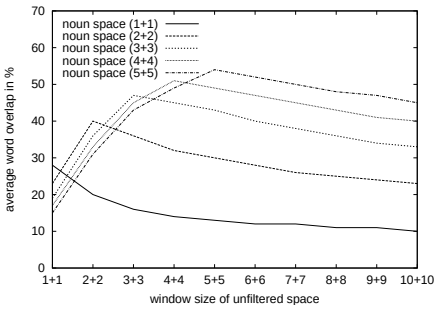
4.1 Overlap of Word Spaces

To evaluate whether the different subspaces contain different information the overlap between spaces was computed. Following Sahlgren (2006) the overlap between two word spaces was calculated as follows: For each word its k nearest neighbours are retrieved in the different word spaces and the percentage of shared nearest neighbours is determined. The figures are then averaged over the whole vocabulary.

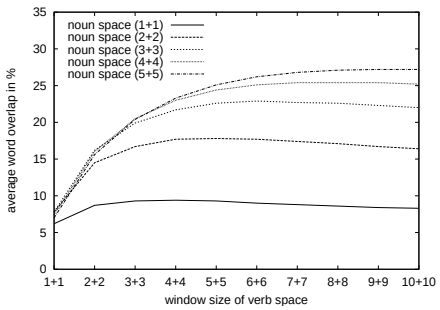
We calculated the overlap for $k = 10$. Figure 1(a) shows the overlap of five noun spaces differing in window size with other noun spaces of all window sizes as indicated on the x-axis. In 1(b) the overlap of the same noun spaces with all the unfiltered spaces is plotted and 1(c) is the analogous figure for verb spaces. So in 1(c) for example the intersection of the lowest curve with the y-axis corresponds to the overlap (6.2%) of the noun space (1+1) and the verb space (1+1) and the second point on that curve to the overlap (8.7%) of the same noun space with the verb space (2+2). The corresponding diagrams for other combinations of filtered spaces have similar characteristics.



(a) overlap with noun spaces



(b) overlap with unfiltered spaces (+pos)



(c) overlap with verb spaces

Figure 1. Overlap of five noun spaces of different window sizes.

28

The interpretation of the first diagram seems straight forward and it is mainly included for comparison purposes: The peaks of the different lines correspond to the overlap of one space with itself, which is always 100% of course. Looking at the second diagram this effect of equal window size determining the peak of overlap is still clearly visible. To both sides of the peaks the overlap decreases monotonically with growing difference in window size. For the noun and verb spaces there are no clear cut peaks visible anymore. And what is more, for the smallest window size the overlap drops noticeably for the differently filtered space of equal window size and in this case is even the smallest overall overlap. We interpret this as a clear effect of the filtering and conclude that at least for the smallest window size the filtered spaces capture different information for words.

Table 2 gives the overlap figures between the different filtered spaces for the minimal window size. These figures are somewhat larger than the ones Sahlgren (2006) reports for word spaces constructed from word-by-word and word-by-document matrices but still support the claim that there are clear differences between the spaces. Compare these figures to the overlap between the noun (1+1) and the unfiltered (1+1) space which amounts to 28.6%.

Table 2. Overlap for top 10 neighbours between all part of speech filtered spaces with a context window of one word to each side. The overlap between two spaces is only given once and figures for identical spaces are not printed.

	verb space (1+1)	adjective space (1+1)	adverb space (1+1)
noun space (1+1)	6.2%	6.8%	4.3%
verb space (1+1)		6.4%	5.3%
adjective space (1+1)			4.1%

4.2 Association Norms

The USF norms contain 5 018 cue words along with their associates as determined in a free association task with an overall of more than 6 000 participants. They were presented with the cue word and asked to write down the first meaningfully related word that came to their mind. Thus, 72 176 responses, called targets, were collected. They vary greatly according to the relation they have to the cue word as can be seen in the following extract for '`abdomen`':

```
stomach, body, muscle, sit ups, organ, pain, intestine
```

These targets range from obviously synonymous through hyper- and hyponymous to somewhat vaguely syntagmatically related words.

The broad notion of relatedness that is exposed in the lists serves the exploratory purpose of our experiments very well as we wanted to establish whether the constructed subspaces contain any sensible information at all. Not all cues were in the vocabulary of the BNC, a total of 107 were missing. This is mostly

29

due to differing spellings because of the provenance of the resources from either Britain or the USA. We did not use these cues in the evaluation.

The part of speech information provided with the USF association norms is incomplete. We hence chose the most frequent part of speech as counted in the BNC if there were different options for a cue.

For the evaluation for each cue the same number of nearest neighbours as there were targets in the USF norm was retrieved and the overlap between these two sets was computed. This measure corresponds to R-Precision or strict accuracy as used by Sahlgren (2006) and is averaged over all cues to yield the evaluation figures as presented in Table 3.

Table 3. Results for the association test for different spaces

	1+1	2+2	3+3	4+4	5+5	6+6	7+7	8+8	9+9	10+10
noun space	6.29	9.24	10.21	**10.26**	10.13	9.99	9.74	9.53	9.37	9.19
verb space	5.89	8.97	9.53	**9.55**	9.31	9.15	9.00	8.70	8.60	8.43
adjective space	7.02	8.95	9.43	**9.49**	9.35	9.18	8.94	8.77	8.57	8.47
adverb space	3.63	4.69	5.14	5.33	5.36	5.38	**5.39**	5.36	5.38	5.29
unfiltered space, +pos	9.17	10.55	**10.66**	10.37	10.02	9.72	9.45	9.11	8.86	8.70
unfiltered space, −pos	9.17	10.46	**10.50**	10.15	9.82	9.49	9.18	9.02	8.67	8.51

The best results for each kind of space are printed in boldface. The altogether best result is reached with an unfiltered word space using part of speech tagged lemmas and a context window of three words. Noticeably it is better than the untagged version. Except for the adverb space the filtered spaces all performed surprisingly well, reaching their best results at a slightly bigger context window of four words. The poor results for the adverb space are probably a sparsity effect because adverbs are the least frequent of the four word classes and so a lot of word vectors won't have any non-zero entries for small window sizes. Note, however, that for all the filtered word spaces including the adverb space the performance gain between window sizes of one and two is much bigger than for the unfiltered spaces. This seems to corroborate our finding in the previous subsection that these spaces contain more specific information.

Our numbers are not directly comparable to the ones in Sahlgren (2006) (figures ranging between 4.17% and 8.76%) because we use a rather small vocabulary. In order to determine how much easier our task becomes because of this we further reduced the size of our vocabulary so that it only included the set of words found in the USF norms roughly reducing the vocabulary by half. We only computed the R-precision values for the unfiltered spaces: for the tagged version we reached 10.92% and for the untagged 10.98%, so the untagged version is a bit better. Though performance increases slightly it is still close to our figures for the word spaces comprising 20 000 words. We hence conclude that the size of the vocabulary has only a minor influence on the results.

30

4.3 TOEFL Synonym Test

In the TOEFL synonym test[1] the right synonym for a given word among four candidates has to be chosen. In terms of the word space representation this corresponds to identifying the closest among the candidates. We tagged and lemmatised the test words according to the frequencies in the BNC whereas the answer candidates were lemmatised and assigned the part of speech of the respective test word.

The results for the TOEFL synonym test (cf. Table 4) are comparable to other results in the literature. But we were not so much interested in absolute figures here but rather wanted to put our findings from the previous experiments to the test, which suggested that filtered word spaces constructed from minimal windows contain more specific information. If our notion was correct that the filtered spaces approximate head-modifier relations, different spaces would be suitable to find neighbours for words from different word classes. We chose a very simple scheme for that: if a test word was a noun we looked for its synonym in the adjective space, if it was a verb we searched the adverb space and vice versa. The results are listed in the last row of the table under "pos selected". Strikingly, the results for the selection scheme are the best when the spaces constructed from minimal windows are used and they clearly beat the unfiltered spaces confirming our interpretation. The difference is not significant though (χ^2-test).

Table 4. Results for the TOEFL synonym test for different spaces

	1+1	2+2	3+3	4+4	5+5	6+6	7+7	8+8	9+9	10+10
noun space	50.00	**57.50**	55.00	48.75	47.50	47.50	43.75	45.00	45.00	43.75
verb space	50.00	**57.50**	53.75	55.00	58.75	50.00	53.75	48.75	50.00	46.25
adjective space	**63.75**	62.50	51.25	52.50	53.75	46.25	46.25	45.00	42.50	45.00
adverb space	56.25	52.50	**60.00**	53.75	52.50	50.00	43.75	38.75	41.25	40.00
unfiltered space, +pos	**67.50**	55.00	51.25	51.25	48.75	47.50	46.25	43.75	42.50	42.50
unfiltered space, −pos	**63.75**	57.50	52.50	45.00	45.00	45.00	41.25	40.00	38.75	41.25
pos selected	**70.00**	66.25	63.75	61.25	58.75	52.50	48.75	50.00	48.75	46.25

5 Discussion

We could not replicate the findings of other experiments (Widdows, 2003) that performance deteriorates with the addition of parts of speech. In general our results were at least marginally better even for unfiltered spaces when we added parts of speech. This is most probably due to the coarse distinction of parts of speech we used that disregarded number and gender features. The small size of

[1] We used the original 80 items of Landauer and Dumais (1997) that were kindly provided by Thomas Landauer.

31

the word spaces leaves some doubt to what extent the results will carry over to larger setups. Especially since only vectors for the most frequent words were computed, effects of sparsity were not an issue as they would be when covering a wider range of words. On the other hand our experiments with a reduced vocabulary seem to imply that vocabulary size does not have a dramatic effect on the performance.

We wanted to explore the idea of filtering the index terms of word spaces according to part of speech to deliberately construct spaces that capture more specific information than unfiltered ones. In a number of experiments we managed to show that for a window size of one word this seems to hold. By selecting one of the spaces based on part of speech of a cue in the TOEFL test we managed to improve over the result of unfiltered spaces. This result is encouraging especially since the decision strategy employed was so simple.

6 Future Work

One issue we are going to explore is the use of more sophisticated decision schemes. Another thing we would like to look into is the use of directional matrices, i.e. word vectors constructed from asymmetric context windows, which should yield even more specific representations while being less sensitive to window size. And finally our next steps will include the use of a dependency parser to yield more appropriate contexts and to implement more complex filters.

References

Hindle, D. (1990). Noun classification from predicate-argument structures. In *ACL 1990 Proceedings*, pages 268–275.

Karlgren, J. and Sahlgren, M. (2001). From words to understanding. In Uesaka, Y., Kanerva, P., and Asoh, H., editors, *Foundations of Real-World Intelligence*, pages 294–308. CSLI Publications, Stanford.

Landauer, T. K. and Dumais, S. T. (1997). A Solution to Plato's Problem. *Psychological Review*, 104(2):211–240.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *COLING-ACL Proceedings*, pages 768–774.

Nelson, D. L., McEvoy, C. L., and Schreiber, T. A. (1998). The university of south florida word association, rhyme, and word fragment norms.

Padó, S. and Lapata, M. (2003). Constructing semantic space models from parsed corpora. In *ACL '03 Proceedings*, pages 128–135.

Rapp, R. (2002). The computation of word associations: comparing syntagmatic and paradigmatic approaches. In *COLING 2002 Proceedings*, pages 1–7.

Sahlgren, M. (2006). *The Word Space Model*. PhD thesis, Department of Linguistics, Stockholm University.

Widdows, D. (2003). Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *NAACL '03*, pages 197–204.

Wiemer-Hastings, P. and Zipitria, I. (2001). Rules for syntax, vectors for semantics. In *CogSci '01 Proceedings*.

Exploring Three Way Contexts for Word Sense Discrimination

Tim Van de Cruys

University of Groningen
t.van.de.cruys@rug.nl

Abstract. In this paper, an extension of a dimensionality reduction algorithm called NON-NEGATIVE MATRIX FACTORIZATION is presented that combines 'bag of words' data and syntactic data, in order to find latent semantic dimensions according to which both words and syntactic relations can be classified. The use of three way data allows one to determine which dimension(s) are responsible for a certain sense of a word, and adapt the corresponding feature vector accordingly, 'subtracting' one sense to discover another one. The intuition in this is that the syntactic features of the syntax-based approach can be disambiguated by the latent semantic dimensions found by the bag of words approach.

1 Introduction

Automatically acquiring semantics from text is a subject that has gathered a lot of attention for quite some time now. As Manning and Schütze [1] point out, most work on acquiring semantic properties of words has focused on *semantic similarity*. 'Automatically acquiring a relative measure of how similar a word is to known words [...] is much easier than determining what the actual meaning is.' [1, §8.5]

Most work on semantic similarity relies on the distributional hypothesis [2]. This hypothesis states that words that occur in similar contexts tend to be similar. With regard to the context used, two basic approaches exist. One approach makes use of 'bag of words' co-occurrence data; in this approach, a certain window around a word is used for gathering co-occurrence information. The window may either be a fixed number of words, or the paragraph or document that a word appears in. The window-based method is often augmented with some form of dimensionality reduction, that is able to capture 'latent semantic dimensions' in the data.

The second approach uses a more fine grained distributional model, focusing on the syntactic relations that words appear with. Typically, a large text corpus is parsed, and dependency triples are extracted.[1] Words are considered similar if they appear with similar syntactic relations. Note that the former approach does not need any kind of linguistic annotation, whereas for the latter, some

[1] e.g. dependency relations that qualify *apple* might be 'object of *eat*' and 'adjective *red*'. This gives us dependency triples like $< apple, obj, eat >$.

33

form of syntactic annotation is needed. Also note that syntax-based approaches typically do not use any form of dimensionality reduction; using these seems much more cumbersome with syntax-based approaches, and does not seem to yield very sensible semantic dimensions.

Especially the syntax-based method has been adopted by many researchers in order to find semantically similar words. There is, however, one important problem with this kind of approach: the method is not able to cope with ambiguous words. Take the examples:

(1) een oneven nummer
a odd number
an odd number

(2) een steengoed nummer
a great number
'a great song'

The word *nummer* does not have the same meaning in these examples. In example (1), *nummer* is used in the sense of 'designator of quantity'. In example (2), it is used in the sense of 'musical performance'. Accordingly, we would like the word *nummer* to be disambiguated into two senses, the first sense being similar to words like *getal* 'number', *cijfer* 'digit' and the second to words like *liedje* 'song', *song* 'song'.

While it is relatively easy for a human language user to distinguish between the two senses, this is a difficult task for a computer. Even worse: the results get blurred because the attributes of both senses (in this example *oneven* and *steengoed*) are grouped together into one sense. This is the main drawback of the syntax-based method. On the other hand, methods that capture semantic dimensions are known to be useful in disambiguating different senses of a word. Particularly, PROBABILISTIC LATENT SEMANTIC ANALYSIS (PLSA) is known to simultaneously encode various senses of words according to latent semantic dimensions [3]. In this paper, we want to explore an approach that tries to remedy the shortcomings of the former, syntax-based approach with the benefits of the latter. The intuition in this is that the syntactic features of the syntax-based approach can be disambiguated by the 'latent semantic dimensions' found in the window-based approach.

2 Previous Work

2.1 Distributional Similarity

There have been numerous approaches for computing the similarity between words from distributional data. We mention some of the most important ones.

One of the best known techniques is LATENT SEMANTIC ANALYSIS (LSA) [4, 5]. In LSA, a term-document matrix is created, containing the frequency of each word in a specific document. This matrix is then decomposed into three other matrices with a mathematical technique called SINGULAR VALUE DECOMPOSITION. The

34

most important dimensions that come out of the SVD allegedly represent 'latent semantic dimensions', according to which nouns and documents can be presented more efficiently. Originally, LSA uses a term-document matrix, but subsequent researchers (e.g. [6]) have applied the same methods using bag of words co-occurrence information. In this view, LSA is an example of the window-based approach.[2]

LSA has been criticized for not being the most appropriate data reduction method for textual applications. The SVD underlying the method assumes normally-distributed data, whereas textual count data (such as the term-document matrix) can be more appropriately modeled by other distributional models such as Poisson [1, §15.4.3]. Successive methods such as PROBABILISTIC LATENT SEMANTIC ANALYSIS (PLSA) [3], try to remedy this shortcoming by imposing a proper latent variable model, according to which the values can be estimated. The method we adopt in our research – NON-NEGATIVE MATRIX FACTORIZATION – is similar to PLSA, and adequately remedies this problem as well.

The second approach – using syntactic relations – has been adopted by many researchers, in order to acquire semantically similar words. One of the most important is Lin's [8]. For Dutch, the approach has been applied by Van der Plas & Bouma [9].

2.2 Discriminating senses

Schütze [6] uses a disambiguation algorithm – called context-group discrimination – based on the clustering of the context of ambiguous words. The clustering is based on second-order co-occurrence: the contexts of the ambiguous word are similar if the words they in turn co-occur with are similar.

Pantel [10] presents a clustering algorithm – coined CLUSTERING BY COMMITTEE (CBC) – that automatically discovers word senses from text. The key idea is to first discover a set of tight, unambiguous clusters, to which possibly ambiguous words can be assigned. Once a word has been assigned to a cluster, the features associated with that particular cluster are stripped off the word's vector. This way, less frequent senses of the word can be discovered.

The former approach uses a window-based method; the latter uses syntactic data. But none of the algorithms developed so far have combined both sources in order to discriminate among different senses of a word.

3 Methodology

3.1 Non-negative Matrix Factorization

Theory Non-negative matrix factorization (NMF) [11] is a group of algorithms in which a matrix V is factorized into two other matrices, W and H.

[2] Note, however, that researchers have found substantial differences with regard to semantic similarity between document frequency and window-based co-occurrence frequency. See e.g. [7].

35

$$V_{n\times m} \approx W_{n\times r} H_{r\times m} \tag{1}$$

Typically r is much smaller than n, m so that both instances and features are expressed in terms of a few components.

Non-negative matrix factorization enforces the constraint that all three matrices must be non-negative, so all elements must be greater than or equal to zero. The factorization turns out to be particularly useful when one wants to find 'additive properties'.

Formally, the non-negative matrix factorization is carried out by minimizing an objective function. Two kinds of objective function exist: one that minimizes the Euclidian distance, and one that minimizes the Kullback-Leibler divergence. In this framework, we will adopt the latter, as – from our experience – entropy-based measures tend to work well for natural language. Thus, we want to find the matrices W and H for which the Kullback-Leibler divergence between V and WH (the multiplication of W and H) is the smallest.

Practically, the factorization is carried out through the iterative application of update rules. Matrices W and H are randomly initialized, and the rules in 2 and 3 are iteratively applied – alternating between them. In each iteration, each vector is adequately normalized, so that all dimension values sum to 1. The rules in 2 and 3 are guaranteed to converge to a local optimum in the minimization of the KL-divergence (for a proof, see [11]).

$$H_{a\mu} \leftarrow H_{a\mu} \frac{\sum_i W_{ia} \frac{V_{i\mu}}{(WH)_{i\mu}}}{\sum_k W_{ka}} \tag{2}$$

$$W_{ia} \leftarrow W_{ia} \frac{\sum_\mu H_{a\mu} \frac{V_{i\mu}}{(WH)_{i\mu}}}{\sum_v H_{av}} \tag{3}$$

Example We can now straightforwardly apply NMF to create semantic word models. NMF is applied to a frequency matrix, containing bag of words co-occurrence data. The additive property of NMF ensures that semantic dimensions emerge, according to which the various words can be classified. Two sample dimensions are shown in example (3). For each dimension, the words with the largest value on that dimension are given. Dimension (a) can be qualified as a 'transport' dimension, and dimension (b) as a 'cooking' dimension.

(3) a. *bus* 'bus', *taxi* 'taxi', *trein* 'train', *halte* 'stop', *reiziger* 'traveler', *perron* 'platform', *tram* 'tram', *station* 'station', *chauffeur* 'driver', *passagier* 'passenger'

b. *bouillon* 'broth', *slagroom* 'cream', *ui* 'onion', *eierdooier* 'egg yolk', *laurierblad* 'bay leaf', *zout* 'salt', *deciliter* 'decilitre', *boter* 'butter', *bleekselderij* 'celery', *saus* 'sauce'

36

3.2 Extending Non-negative Matrix Factorization

We now propose an extension of NMF that combines both the bag of words approach and the syntactic approach. The algorithm finds again latent semantic dimensions, according to which nouns, contexts and syntactic relations are classified.

Since we are interested in the classification of nouns according to both 'bag-of-words' context and syntactic context, we first construct three matrices that capture the co-occurrence frequency information for each mode. The first matrix contains co-occurrence frequencies of nouns cross-classified by dependency relations, the second matrix contains co-occurrence frequencies of nouns cross-classified by words that appear in the noun's context window, and the third matrix contains co-occurrence frequencies of dependency relations cross-classified by co-occurring context words.

We then apply NMF to the three matrices, but we interleave the separate factorizations: the results of the former factorization are used to initialize the factorization of the next matrix. This implies that we need to initialize only three matrices at random; the other three are initialized by calculations of the previous step. The process is represented graphically in figure 1.

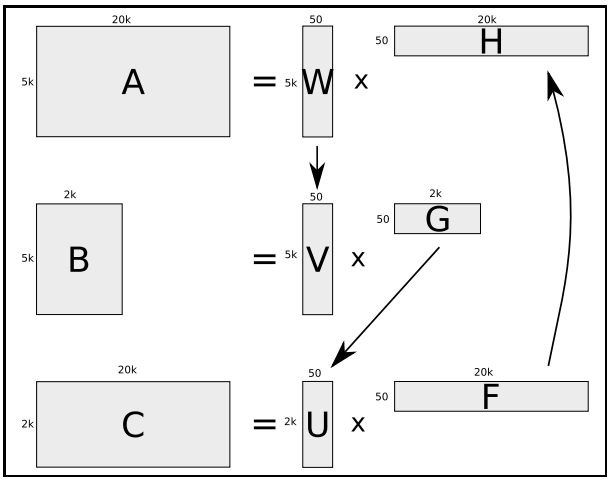


Fig. 1. A graphical representation of the extended NMF

When the factorization is finished, the three modes (nouns, dependency relations and context words) are classified according to latent semantic dimensions.

3.3 Sense Subtraction

Next, we want to use the factorization that has been created in the former step for word sense discrimination. The intuition is that we 'switch off' one dimension

37

of an ambiguous word, to reveal possible other senses of the word. This intuition is implemented as follows:

From matrix W, we know which dimensions are the most important ones for a certain word. Matrix H gives us the importance of each syntactic relation given a dimension. By applying the formula in equation (4), we can 'subtract' the syntactic relations that are responsible for a certain dimension, from the vector in our original matrix.

$$\overrightarrow{v}_{new} = \overrightarrow{v}_{orig}(\overrightarrow{1} - \overrightarrow{h}_{dim}) \tag{4}$$

This formula multiplies each feature (syntactic relation) of the original noun vector ($\overrightarrow{v}_{orig}$) with a scaling factor, according to the load of the feature on the subtracted dimension ($\overrightarrow{h}_{dim}$ – the vector of matrix H containing the dimension we want to subtract). $\overrightarrow{1}$ designates a vector of 1's the size of $\overrightarrow{h}_{dim}$.

4 Results

This section will show some exploratory results of three way contexts, and word sense discrimination using those contexts. It tends to give an idea of the kind of data produced, and the way word senses can be discriminated using these data. No formal evaluation of the method has been carried out yet.

4.1 Experimental Design

The interleaved NMF presented in section 3.2 has been applied to Dutch, using the CLEF corpus (containing Dutch newspaper text from 1994 and 1995). The corpus is consistently divided into paragraphs, which have been used as the context window for the bag-of-words mode. The corpus has been parsed by the Dutch dependency parser Alpino [12], and dependency triples have been extracted using XML-stylesheets. Next, the three matrices needed for our method have been constructed: one containing nouns by dependency relations (5K × 20K), one containing nouns by context words (5K × 2K) and one containing dependency relations by context words (20K × 2K). We did 50 iterations of the algorithm, factorizing the matrices into 50 dimensions. The NMF algorithm has been implemented in Python, using the NUMPY module for scientific computing.

4.2 Examples

In (4), an example is given of the kind of semantic dimensions found. This dimension may be called the 'transport' dimension, as is shown by the top 10 nouns (a), context words (b) and syntactic relations (c).

(4) a. *auto* 'car', *wagen* 'car', *tram* 'tram', *motor* 'motorbike', *bus* 'bus', *metro* 'subway', *automobilist* 'driver', *trein* 'trein', *stuur* 'steering wheel', *chauffeur* 'driver'

38

b. *auto* 'car', *trein* 'train', *motor* 'motorbike', *bus* 'bus', *rij* 'drive', *chauffeur* 'driver', *fiets* 'bike', *reiziger* 'reiziger', *passagier* 'passenger', *vervoer* 'transport'

c. viertraps$_{adj}$ 'four pedal', verplaats_met$_{obj}$ 'move with', toeter$_{adj}$ 'honk', tank_in_houd$_{obj}$ [parsing error], tank$_{subj}$ 'refuel', tank$_{obj}$ 'refuel', rij_voorbij$_{subj}$ 'pass by', rij_voorbij$_{adj}$ 'pass by', rij_af$_{subj}$ 'drive off', peperduur$_{adj}$ 'very expensive'

In what follows, we will talk about dimensions like this one as, e.g., the 'music' dimension or the 'city' dimension. In the vast majority of the cases, the dimensions are indeed as clear-cut as the transport dimension shown above, so that the dimensions can be rightfully labeled this way.

Next, two examples are given of how the semantic dimensions that have been found might be used for word sense discrimination. We will consider two ambiguous nouns: *pop*, which can mean 'pop music' as well as 'doll', and *Barcelona*, designating either the Spanish city or the Spanish football club.

First, we look up the top dimensions for each noun. Next, we subtract successively the highest and second highest dimension from the noun vector, as described in 3.3. This gives us three vectors for each noun: the original vector, and two vectors with one of the highest scoring dimensions eliminated. For each of these vectors, the top ten similar nouns are given, in order to compare the changes brought about.

(5) a. *pop*, *rock*, *jazz*, *meubilair* 'furniture', *popmuziek* 'pop music', *heks* 'witch', *speelgoed* 'toy', *kast* 'cupboard', *servies* '[tea] service', *vraagteken* 'question mark'

b. *pop*, *meubilair* 'furniture', *speelgoed* 'toy', *kast* 'cupboard', *servies* '[tea] service', *heks* 'witch', *vraagteken* 'question mark' *sieraad* 'jewel', *sculptuur* 'sculpture', *schoen* 'shoe'

c. *pop*, *rock*, *jazz*, *popmuziek* 'pop music', *heks* 'witch', *danseres* 'dancer', *servies* '[tea] service', *kopje* 'cup', *house* 'house music', *aap* 'monkey'

Example (5) shows the top similar words for the three vectors of *pop*. In (a), the most similar words to the original vector are shown. In (b), the top dimension (the 'music dimension') has been subtracted from (a), and in (c), the second highest dimension (a 'domestic items' dimension) has been subtracted from (a).

The differences between the three vectors are clear: in vector (a), both senses are mixed together, with 'pop music' and 'doll' items interleaved. In (b), no more music items are present. Only items related to the doll sense are among the top similar words. In (c), the music sense emerges much more clearly, with *rock*, *jazz* and *popmuziek* being the most similar, and a new music term (*house*) showing up among the top ten.

Admittedly, in vector (c), not all items related to the 'doll' sense are filtered out. We believe this is due to the fact that this sense cannot be adequately filtered out by one dimension (in this case, a dimension of 'domestic items' alone), whereas it is much easier to filter out the 'music' sense with only one 'music'

39

dimension. In future work, we want to investigate the possibility of subtracting multiple dimensions related to one sense.

A second example, the ambiguous proper noun *Barcelona*, is given in (6).

(6) a. *Barcelona*, *Arsenal*, *Inter*, *Juventus*, *Vitesse*, *Milaan* 'Milan', *Madrid*, *Parijs* 'Paris', *Wenen* 'Vienna', *München* 'Munich'
b. *Barcelona*, *Milaan* 'Milan', *München* 'Munich', *Wenen* 'Vienna', *Madrid*, *Parijs* 'Paris', *Bonn*, *Praag* 'Prague', *Berlijn* 'Berlin', *Londen* 'London'
c. *Barcelona*, *Arsenal*, *Inter*, *Juventus*, *Vitesse*, *Parma*, *Anderlecht*, *PSV*, *Feyenoord*, *Ajax*

In (a), the two senses of *Barcelona* are clearly mixed up, showing cities as well as football clubs among the most similar nouns. In (b), where the 'football dimension' has been subtracted, only cities show up. In (c), where the 'city dimension' has been subtracted, only football clubs remain.

5 Conclusion & Future Work

In this paper, an extension of NMF has been presented that combines both bag of words data and syntactic data in order to find latent semantic dimensions according to which both words and syntactic relations can be classified. The use of three way data allows one to determine which dimension(s) are responsible for a certain sense of a word, and adapt the corresponding feature vector accordingly, 'subtracting' one sense to discover another one. We believe that the use of three way distributional data is effectively able to disambiguate the features of a given word, and accordingly its word senses.

We conclude with some issues for future work. First of all, we'd like to test the method that has been explored in this paper in a proper evaluation framework, and compare the method to other methods that discriminate senses. Next, we'd like to work out a proper probabilistic framework for the 'subtraction' of dimensions. And finally, we'd like to combine the method with a clustering approach. Thus, one can determine which are the important dimensions for a given cluster, subtract these from the individual words, and see whether other senses of the word emerge.

References

1. Manning, C., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, Massachussets (2000)
2. Harris, Z.: Distributional structure. In Katz, J.J., ed.: The Philosophy of Linguistics. Oxford University Press (1985) 26–47
3. Hofmann, T.: Probabilistic latent semantic analysis. In: Proc. of Uncertainty in Artificial Intelligence, UAI'99, Stockholm (1999)
4. Landauer, T., Dumais, S.: A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. Psychology Review **104** (1997) 211–240

40

5. Landauer, T., Foltz, P., Laham, D.: An Introduction to Latent Semantic Analysis. Discourse Processes **25** (1998) 295–284
6. Schütze, H.: Automatic word sense discrimination. Computational Linguistics **24**(1) (1998) 97–123
7. Sahlgren, M.: The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces. PhD thesis, Stockholm University (2006)
8. Lin, D.: Automatic retrieval and clustering of similar words. In: Proceedings of COLING/ACL 98, Montreal, Canada (1998)
9. van der Plas, L., Bouma, G.: Syntactic contexts for finding semantically similar words. In van der Wouden, T., et al., eds.: Computational Linguistics in the Netherlands 2004. Selected Papers from the Fifteenth CLIN Meeting, Utrecht, LOT (2005) 173–184
10. Pantel, P., Lin, D.: Discovering word senses from text. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM Special Interest Group on Knowledge Discovery in Data, ACM Press (2002) 613–619
11. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: NIPS. (2000) 556–562
12. van Noord, G.: At Last Parsing Is Now Operational. In Mertens, P., Fairon, C., Dister, A., Watrin, P., eds.: TALN06. Verbum Ex Machina. Actes de la 13e conference sur le traitement automatique des langues naturelles, Leuven (2006) 20–42

41

RECENT RESEARCH REPORTS

#116 Marco Baroni, Alessandro Lenci, and Magnus Sahlgren, editors. *Proceedings of the 2007 Workshop on Contextual Information in Semantic Space Models: Beyond Words and Documents*, Roskilde, Denmark, August 2007.

#115 Paolo Bouquet, Jérôme Euzenat, Chiara Ghidini, Deborah L. McGuinness, Valeria de Paiva, Luciano Serafini, Pavel Shvaiko, and Holger Wache, editors. *Proceedings of the 2007 workshop on Contexts and Ontologies Representation and Reasoning (C&O:RR-2007)*, Roskilde, Denmark, August 2007.

#114 Bich-Liên Doan, Joemon Jose, and Massimo Melucci, editors. *Proceedings of the 2nd International Workshop on Context-Based Information Retrieval*, Roskilde, Denmark, August 2007.

#113 Henning Christiansen and Jørgen Villadsen, editors. *Proceedings of the 4th International Workshop on Constraints and Language Processing (CSLP 2007)*, Roskilde, Denmark, August 2007.

#112 Anders Kofod-Petersen, Jörg Cassens, David B. Leake, and Stefan Schulz, editors. *Proceedings of the 4th International Workshop on Modeling and Reasoning in Context (MRC 2007) with Special Session on the Role of Contextualization in Human Tasks (CHUT)*, Roskilde, Denmark, August 2007.

#111 Ioannis Hatzilygeroudis, Alvaro Ortigosa, and Maria D. Rodriguez-Moreno, editors. *Proceedings of the 2007 workshop on REpresentation models and Techniques for Improving e-Learning: Bringing Context into the Web-based Education (ReTIeL'07)*, Roskilde, Denmark, August 2007.

#110 Markus Rohde. *Integrated Organization and Technology Development (OTD) and the Impact of Socio-Cultural Concepts — A CSCW Perspective*. PhD thesis, Roskilde University, Roskilde, Denmark, 2007.

#109 Keld Helsgaun. An effective implementation of k-opt moves for the Lin-Kernighan TSP heuristic. 2006, Roskilde University, Roskilde, Denmark.

#108 Pernille Bjørn. *Virtual Project Teams — Distant Collaborative Practice and Groupware Adaptation*. PhD thesis, Roskilde University, Roskilde, Denmark, 2006.

#107 Henrik Bulskov Styltsvig. *Ontology-based Information Retrieval*. PhD thesis, Roskilde University, Roskilde, Denmark, 2006.

#106 Rasmus Knappe. *Measures of Semantic Similarity and Relatedness for Use in Ontology-based information Retrieval*. PhD thesis, Roskilde University, Roskilde, Denmark, 2006.

#105 Davide Martinenghi. *Advanced Techniques for Efficient Data Integrity Checking*. PhD thesis, Roskilde University, Roskilde, Denmark, 2005.