A Distributional Memory for German

Sebastian Padó
Institut für Computerlinguistik
Universität Heidelberg
pado@cl.uni-heidelberg.de

Jason Utt
Institut für maschinelle Sprachverarbeitung
Universität Stuttgart
uttjn@ims.uni-stuttgart.de

Abstract

This paper describes the creation of a Distributional Memory (Baroni and Lenci 2010) resource for German. Distributional Memory is a generalized distributional resource for lexical semantics that does not have to commit to a particular vector space at the time of creation. We induce a resource from a German corpus, following the original design decisions as closely as possible, and discuss the steps necessary for a new language. We evaluate the German DM model on a synonym selection task, finding that it can compete with existing models.

1 Introduction

Distributional semantics is a paradigm for determining a word's meaning by observing its occurrence in large corpora. It builds on the Distributional Hypothesis (Harris, 1954; Miller and Charles, 1991) which states that words occurring in similar contexts are similar in meaning. Vector space models, the most widely used incarnation of distributional semantics, represent words as vectors in a high-dimensional space whose dimensions correspond to features of the words' contexts (Turney and Pantel, 2010). These features can be chosen in different ways; popular choices are context words (Schütze, 1992), dependency relations or paths (Lin, 1998; Padó and Lapata, 2007), or subcategorization frames (Schulte im Walde, 2006).

The notion of graded semantic similarity that vector space models provide has been used in various applications, including word sense disambiguation (McCarthy et al., 2004), representation of selectional preferences (Erk et al., 2010), verb class induction (Schulte im Walde, 2006), analogical reasoning (Turney, 2006), or alternation discovery (Joanis et al., 2006).

Such different applications however tend to require different types of semantic spaces. In traditional cases like WSD, the objects whose similarity we are interested in are words. In analogical reasoning, however, we need to compare word pairs, and for alternation discovery we need to compare verbal argument positions. Baroni and Lenci (2010) addressed this fragmentation by proposing a model called Distributional Memory (DM). It captures distributional information at a more abstract level and can be mapped onto various vector spaces to perform different tasks. Baroni and Lenci show that DM is competitive with state-of-the-art models with dedicated spaces on many NLP tasks.

Baroni and Lenci's original work only considered English. In this paper, we describe ongoing work on the construction and evaluation of a corresponding Distributional Memory resource for German. Section 2 provides background on Distributional Memory. Section 3 assesses potential strategies for inducing a German DM-style resource. Section 4 describes the concrete steps, and discusses properties of German that require particular attention. Section 5 evaluates the German DM on a synonym selection task, and Section 6 concludes.

2 Distributional Memory

Distributional Memory (DM, Baroni and Lenci 2010) is a recent multi-purpose framework in distributional semantics. Contrary to traditional studies, which directly constructed task-specific vector spaces, DM extracts a tensor, i.e. a three-dimensional matrix, of weighted *word-link-word* tuples. Each tuple is mapped onto a number by a

Proceedings of KONVENS 2012 (LexSem 2012 workshop), Vienna, September 21, 2012

scoring function $\sigma\colon W \times L \times W \to \mathbb{R}^+$. For example, $\langle$*pencil obj use*$\rangle$ is assigned a higher weight than $\langle$*elephant obj use*$\rangle$.

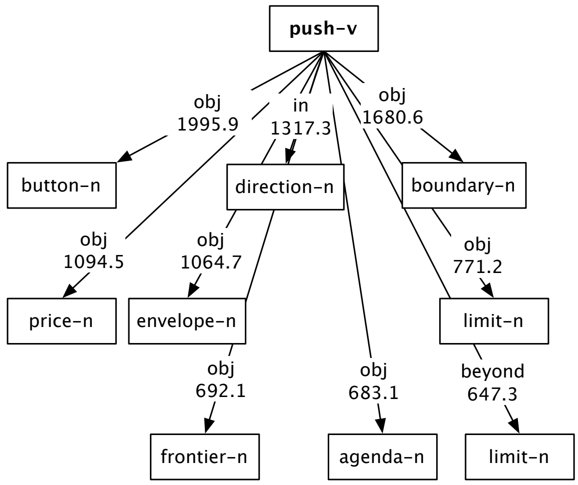

Figure 1: The DepDM tensor as labeled graph: *push* with its nine highest-scoring co-occurrents

This DM tensor can be visualized as a directed graph whose nodes are labeled with lemmas and whose edges are labeled with links and scores. As an example, Figure 1 shows the nine highest-scoring context co-occurrents for the verb *push* together with their scores. All of them happen to be arguments and adjuncts of *push*, although DM also models 'inverse' relations. Seven of the nine co-occurrents are objects, and two are prepositional adjuncts. The benefit of this tensor representation is that it is applicable to many tasks in computational linguistics. Once a task is selected, a dedicated semantic space for this task can be generated efficiently from the tensor by matricization. For example, the *word by link-word* space ($W \times LW$) contains vectors for words w and its dimensions are labeled with pairs $\langle l, w \rangle$ of a link and a context word. This space models similarity among words, e.g. for thesaurus construction (Lin, 1998). Another space is the *word-word by link* space ($WW \times L$). It contains co-occurrence vectors for word pairs $\langle w_1, w_2 \rangle$. Its dimensions are labeled with links l. This space can be used to model semantic relations.

DM does not assume any specific source for the tuples. However, since dependency parses are the most obvious source of such relational information, DM falls into the category of dependency-based semantic space models. DM does not presuppose a particular vocabulary of link (relation) types. Baroni and Lenci present three variants of DM that differ in the relations that they assume.

The simplest variant is Dependency DM (DepDM), as shown in Figure 1. It uses a set of relations among verbs, nouns, and adjectives most of which are adopted directly from the dependency parses. (*obj, iobj*). The subject relation is extended with subcategorization information (*subj_tr, subj_intr*) to better handle diathesis alternations, and prepositions are turned directly into links ($\langle$*walk on street*$\rangle$).

The second variant, Lexicalized DM (LexDM), uses more complex links. The links encode rich information about the two words in the triple, such as their part of speech, definiteness, modification, as well as lexicalized information about the dependency path between the words. An example is the tuple $\langle$*soldier use+n+the+n-a gun*$\rangle$ which is constructed from the sentence *The soldier used a gun* and indicates the verb linking the two nouns, their POS tags and articles. The definition of the LexDM links is highly language-specific and is based on WordNet information as well as semantic classes extracted from corpora with the help of various patterns.

Finally, the third variant, TypeDM, builds on LexDM but modified the scoring function, following the intuition that the frequency of a link is less informative (since it is influenced by a large number of factors) than the number of its surface varieties: links with a high number of varieties are likely to express prominent semantic relations.

3 Strategies for Creating a German DM

In situations like ours, where some resource is available in language A, but not in another language B, two strategies can be followed. The first one is *cross-lingual transfer*: The existing resource in language A is ported onto language B. The second one is *parallel induction*: The schema for creating the resource in language A is replicated, as closely as possible, for language B.

Cross-lingual transfer is an attractive proposition since it takes optimal advantage of the existing resource. It falls into two subcases: If the resource mainly contains information at the token (instance) level, this can be achieved via annotation projection in parallel corpora (Yarowsky and Ngai, 2001; Bentivogli and Pianta, 2005). If the resource concentrates on the type (lemma) level, bilingual dictionaries can be used to simply translate the resource (Fung and Chen, 2004; Peirsman and Padó, 2011).

The translation-based strategy would be applicable in the case of DM which records and scores lemma-link-lemma triples. For the language pair English–German, it is also the case that reliable, high-coverage bilingual dictionaries are publicly available. However, we decided against adopting this strategy. The reason is that simple translation runs into serious ambiguity problems. The problem can be visualized easily in terms of the labeled graph view on DM (Figure 1). Translating this graph involves replacing the original node labels, English lemmas, with German lemmas.[1] This is unproblematic for one-to-one translations where nodes are just relabeled (*sitzen* $\rightarrow$ *sit*). There is also a solution for cases where two English lemmas correspond to a single German lemma (*cup, mug* $\rightarrow$ *Tasse*): the two English nodes can be collapsed. However, a significant number of English words have more than one translation in German. Where these translations are not just synonyms but express different senses of the English word (*wood* $\rightarrow$ *Holz, Wald*), the English node would need to be split and all its incoming and outgoing edges assigned to either of the two new German nodes. This, however, is a full-fledged sense disambiguation problem which appears difficult to solve, in particular given the very limited amount of context information available in bilingual dictionaries.

For this reason, we decided to adopt the second strategy, parallel induction, and create a DM resource from German text. In the case of distributional semantic models, this choice is made possible by the fact of the relatively good corpus situation for German: there is a very large web corpus available for German, as well as fairly good dependency parsers. The remainder of this section discusses the different steps involved in this task and the difficulties that arise.

[1]We assume for this discussion that syntactic relations show a high degree of parallelism for the language pair English–German, which we found to be a reasonable assumption in previous experiments (Peirsman and Padó, 2011).

4 Inducing a DM Resource from German corpora

4.1 DepDM vs. LexDM vs. TypeDM

Our first decision was which of the three DM variants to implement for German. We observed above that the patterns in LexDM and TypeDM are more complex than those in DepDM. A number of (semi-)manual annotations were utilized to construct the former, such as a list of high-frequency verbs selected as part of the lexicalized edge labels. While the more elaborately designed TypeDM nearly always performs best (Baroni and Lenci, 2010), the much simpler DepDM performs at a similar level for many tasks. We therefore opted to implement DepDM.

4.2 Defining patterns to extract links

The types of German links we use correspond fairly directly to the simple syntactic patterns of DepDM. We obtained these patterns by inspecting the most frequent syntactic configurations in a large German corpus (cf. Section 4.3). The patterns can be categorized into two groups: unlexicalized and lexicalized patterns.

Unlexicalized patterns. We use 7 patterns which extract information at the verb phrase and sentence levels. subjects for transitive and intransitive verbs (`sbj_tr`, `sbj_intr`); direct objects, indirect objects, and phrasal complements of verbs (`obj, iobj, vcomp`); noun modification (`nmod`) and the relation between a subject and an object of a verb (`verb`). These patterns are unlexicalized, that is, the patterns correspond directly to link types.

Lexicalized patterns. Three more patterns are lexicalized. This means two things: (a), they may contain fixed lexical material (marked in boldface); (b), they may incorporate some of the lexical information on the dependency paths that they match into the resulting link (marked in square brackets below). These patterns apply mostly to NPs and PPs. The first pattern is n_1 $[prep]$ n_2, which captures phrases like *Recht auf Auskunft* which is turned into the triple ⟨*Recht auf Auskunft*⟩. The second pattern is adj n_1 **von** $[n_2]$ which extracts the second linking noun as a link and captures such phrases as *heutige Größe von der Sonne* which results in the triple ⟨*heutige Sonne Größe*⟩. The third lexicalized pattern is n_1 $[verb]$ n_2 where the particular verb combining subject n_1 and object n_2 is used as the link. For example, the sentence *Hochtief sieht Aufwind* results in the tuple ⟨*Hochtief sehen Aufwind*⟩. Due to the presence of lexical material in the links, these patterns give rise to a very large number of link types.

Note that we currently ignore prepositional arguments of verbs, adverbs, and relative clauses. This is partly a frequency-based decision (the patterns above account for the majority of links in the corpus), but also one motivated by minimalism: we wanted to find out how well such a minimal set of patterns is able to perform.

We also found that as opposed to English, German offers a number of difficulties when extracting word relations from text. Particle verbs for example often possess a detachable prefix e.g. *mit|geben, weiter|reichen*, which at the surface level can be realized at a large distance from the verb stem, e.g. *Er gab ihr das Buch, nach dem sie am Vortag gefragt hatte, nur ungern mit*. We reconstruct such verbs from the parser output by looking for words with the `PTKVZ` (verbal particle) POS tag which stand in a `SVP` (verbal particle) or `MO` (modifier) relation to a full verb.

Additional issues are the very productive compounding (e.g. *Wasserstoffbetankungseinrichtung*) and derivation (e.g. *Pappkärtchen*) of nouns. This gives rise to many more noun types that need to be integrated into the system than in English. In the design of the English DM, Baroni and Lenci set a limit for nouns, including only the 20k most frequency nouns into their DM. In German this would give too sparse a model (cf. Section 5). We thus chose to not use a cutoff for any part of speech even though this makes the tensor and its resulting matrices even sparser than usual in dependency-based representations. We considered splitting compound nouns, but left this problem for future work, given the problem of deciding whether compounds are semantically transparent or not (cf. *Schule – Baumschule*).

4.3 Step 2: Corpus counts

The corpus we used to extract the co-occurrence counts was the SDEWAC web corpus (Faaß et al., 2010) parsed with the MATE German dependency parser (Bohnet, 2010). SDEWAC is based on the DEWAC corpus (Baroni and Kilgarriff, 2006), a large corpus collected from the German WWW (`.de` top-level domain). SDEWAC performs various preprocessing steps on DEWAC such as identifying sentences and then filtering out duplicate sentences from the same URL. It contains 9M different word types and 884M word tokens.

The advantage of using a web corpus is size and availability. At the same time, it includes text from many different domains, in contrast to most other corpora which contain mostly news text. On the other hand, it contains much noise such as spelling and grammatical errors that part-of-speech taggers and parsers must deal with. At the current stage, we are more interested in preserving the information in the data. While this means we include nouns such as *Kurz-vor-Ladenschluss-noch-schnell-einkaufenden-Kassen-Warteschlange*, we intend to address these issues in future work.

4.4 Step 3: Scoring

The weighting in our German DM follows the scheme for LexDM and DepDM, which is Local Mutual Information (LMI), Evert (2005)). The following equation defines the LMI of a word-link-word combination (i, j, k) via its observed frequency in the corpus O_{ijk} and its expected frequency E_{ijk}:

$$\mathrm{LMI}(i, j, k) = O_{ijk} \cdot log \frac{O_{ijk}}{E_{ijk}}.$$

The logarithmic term stems from the definition of pointwise mutual information: It captures the degree to which the observed frequency of the word-link-word combination differs from the expected frequency under an independence assumption, i.e. $P(i, j, k) = P(i) \cdot P(j) \cdot P(k)$. This means E_{ijk} assumes that (i, j, k) occurs due to chance rather than a latent relatedness. Then, if $O_{ijk} = E_{ijk}$, the mutual information will be equal to 0, since i, j, and k are statistically independent of one another.

In contrast to PMI, though, LMI contains the raw frequency O_{ijk} as a factor that discounts infrequent combinations, addressing PMI's well-known tendency to overestimate the informativeness of very infrequent combinations. We follow the construction of the English DM and set negative LocalMI values to zero, which is equivalent to excluding them from the tensor.

5 Evaluation

5.1 Statistics of German DM

The German DM resulting from these steps contains over 78M links for 3,5M words (nouns, verbs and adjectives). It contains about 220K link types, almost all of which stem from surface patterns. On average, a lemma has 22 links. This makes our German DM considerably more sparse than its English counterpart, which contains about 131M links for just 31K lemmas and has just over 25K link types.

Table 1 shows information about the link types. As described in Section 4.2, there are 7 unlexicalized link types and more than 220K lexicalized link types. Table 2 shows an example of the extracted co-occurrents for the verb *sehen*.

The current version of our German DM (DM.de) can be obtained at `http://goo.gl/GNbMb`.

5.2 Task-based evaluation: Synonym selection

As we have discussed above, the main benefit of the DM model is its ability to inform various tasks in lexical semantics. Baroni and Lenci (2010) evaluate their English DM on a number of tasks, including semantic similarity, relational similarity, and relation extraction. A comprehensive evaluation is outside the scope of this article; we focus on synonym selection.

Data. Our evaluation is based on the German Reader's Digest Word Power (RDWP) dataset (Wallace and Wallace, 2005) which contains 984 word choice problems.[2] This dataset is similar to the English TOEFL dataset (Landauer and Dumais, 1997). Each problem consists of one target word and a set of four possible candidates, which are either a synonym or a phrase defining the word. For example, for the word *Prozedur*, the candidates are *Gerichtsverfahren*, *Vorbild*, <u>*Vorgang*</u>, and *Demonstration*.

Model. As this is a word similarity task, we base our model on the word-by-link-word $W \times LW$ matricization of the DM tensor which represents words in a space whose dimensions are labeled with pairs of a link and a context word. For each problem, we compute the cosine between its vector and the candidates' vectors, predicting the most similar candidate.

Our base model ('base') excludes problems that include phrases as candidates, since "plain" DM focuses on representing the meaning of individual words. Given the high number of phrasal candidates in the dataset, however, we also also experiment with a very simplistic model for phrase meaning ('phrases') which defines the similarity between a target word and a phrasal candidate as the maximum similarity between the target and any of the constituent words of the phrase. This corresponds to the oversimplification that the meaning of a phrase is determined by its most informative word. Finally, we combine the two models ('combined'), which is trivial since they make predictions for disjoint sets of problems.

We compare two versions of our DM model that differ in the amount of sparsity that they tolerate. The first model (DM.de$^{\alpha}$) excludes items for which at least one candidate has a zero similarity to the target (i.e. there is not one single context in which both words were observed). This model is conservative and abstains from any experiments items where sparsity problems can be expected. The second model, DM.de$^{\beta}$, excludes only those items where all candidates have a zero similarity to the target; this generally indicates that the target was seen never or almost never.

Evaluation. We evaluate analogously to Mohammad et al. (2007), defining the Score as a weighted sum of correct predictions, while discounting ties: Score $= A + .5 \cdot B + .\bar{3} \cdot C + .25 \cdot D$, where A is the number of correctly predicted items with no ties, B those with a 2-way tie, C with a 3-way tie, and D with a 4-way tie. Note that Score does not take the number of covered problems into account. For this reason, Mohammad et al. also define a precision-oriented Accuracy measure which is defined as the average Score per covered problem: Accuracy = Score/Covered. Note that a random baseline would perform at an accuracy of .25.

We compare our results with those reported by

[2] The dataset is available from: `http://goo.gl/PN42E`

Unlexicalized links		
7,833,635	n (i)`obj` v	noun n is (in)direct object of verb v
7,478,550	n `sbj_(in)tr` v	noun n is subject of verb v in an (in)transitive construction
677,397	v_1 `vcomp` v_2	subcategorization of verb v_1 by a verb v_2 (excluding modals and auxiliaries)
1,575,516	x `nmod` n_2	noun n_2 modified by $x \in \{adj, n\}$
3,304,045	n_1 `verb` n_2	noun n_1 is subject and n_2 object of a verb
Lexicalized links		
220,269	link types	e.g. `in, von, an, als, vor, um, gegen, stellen, machen, bieten, geben`
18,180,604	links	2,462,927 n_1 `in` n_2; 1,424,398 n_1 `von` n_2; 1,170,609 n_1 `mit` n_2; ...

Table 1: Statistics for the German DM tensor.

`sbj_intr`		`sbj_tr`		`obj`		`iobj`	
Realität	2,339.3	*Entwurf*	4,455.5	*Chance*	18,805.5	*Mensch*	512.7
Wirklichkeit	1,343.0	*Mensch*	3,555.2	*Film*	14,978.0	*Tatsache*	439.3
Sache	1,204.1	*Senat*	3,244.1	*Bild*	14,925.5	*Wahrheit*	382.7
Welt	802.4	*Zuschauer*	3,147.2	*Möglichkeit*	12,429.9	*Zukunft*	324.8

Table 2: Highest-LMI co-occurrents for the verb *sehen*. The intransitive subjects and indirect objects are mainly due to misparses (*'die X sieht ... aus'*) and idiomatic expressions, respectively (*'der Y ins Gesicht sehen'*).

Mohammad et al. (2007) who evaluate a number of monolingual models based on GermaNet, a large lexical resource for German. They fall into two categories: (a) gloss-based methods which are variants of the Lesk (1986) algorithm applied to GermaNet glosses[3]; and (b), hierarchical methods which compute similarity with various graph-based similarity measures in GermaNet. In contrast, our DM-based model is purely distributional and does not use any structured semantic knowledge for German.

Results. Table 3 shows the results. In terms of accuracy, Mohammad et al.'s gloss-based models are the clear winners: the very specific "context" provided by a definition provides an excellent basis for synonym detection where it is available; but these models at the same time have a low coverage of below 30%. Hierarchical GermaNet models show a higher coverage (up to 40%), but also a substantially lower precision and accuracies of around .5.

Our 'base' models already perform at the same level as the hierarchical models, with accuracies of .5 and above. The 'combined' models (both DM.de$^\alpha$ and DM.de$^\beta$) show the highest coverage of all models considered: the more cautious DM.de$^\alpha$ covers more than 40%, and the DM.de$^\beta$ even over 80% of all items. This result underscores the benefit of distributional models in terms of coverage.

Somewhat surprisingly, the 'phrases' model reliably outperforms the 'base' model even though it uses a simplistic heuristic. The reason can be found in the properties of the phrasal candidates. As Table 4 shows, they often resemble definitory phrases or glosses. Consequently, our model basically attempts a simplified Lesk-style disambiguation task, similar to Mohammad et al.'s gloss-based models. In contrast, the single-word candidates treated by the 'base' model are often rare or otherwise difficult synonyms of the target. For instance,

[3]The Lesk algorithm is a word sense disambiguation method which uses dictionary glosses for a word's different senses and checks their overlap with the given context. The sense whose gloss has the highest overlap is taken to be the correct sense.

Measure		**Covered**	**Correct**	**Ties**	**Score**	**Accuracy**
		gloss-based (Mohammad et al. 2007)				
HPG		222	174	11	171.5	**.77**
RPG		266	188	15	184.7	.69
		distributional (our work)				
	(base)	174	96	0	96	.55
DM.de$^\alpha$	(phrases)	228	135	4	132	.58
	(combined)	402	231	4	**228**	**.57**
	(base)	358	178	0	178	.50
DM.de$^\beta$	(phrases)	466	261	4	258	.55
	(combined)	824	439	4	**436**	**.53**
		hierarchical (Mohammad et al. 2007)				
JC		357	157	1	156.0	.44
Lin_{GN}		298	153	1	152.5	**.51**
Res		299	154	33	148.3	.50

Table 3: Performance on the Reader's Digest Word Power data set, DM.de$^\alpha$ includes only items for which all candidates have a non-zero similarity, while DM.de$^\beta$ only requires one similarity be non-zero.

Golfwägelchen is so uncommon that we get no similarity to the target *Caddie*, while *Golfschläger* is frequent enough that the shared topic leads to a certain degree of similarity. On the other hand, the common word *Witz*, while not an exact synonym, would be used in defining the target word *Kalauer*. The difference between the 'base' and 'phrases' models remains fairly small, though.

The non-zero ties are all phrasal items that include the same phrase with subtle variations, e.g. *Sommertag* as a target has the candidate phrases: *[Höchsttemperatur] von mindestens 25/20/28/30°C*. This clearly shows the limits of our simple phrasal comparison method which fails to distinguish between candidates when the most similar word is shared.

The 'combined' models, which combine all predictions of the 'base' and 'phrases' models, end up with the partial models' averaged accuracies (.53 and .57 for DM.de$^\alpha$ and DM.de$^\beta$, respectively). They outperform Mohammad et al.'s hierarchical models but not the gloss-based models. They show however the two highest score numbers, 228 and 436, which is a direct result of their high coverage. We see these results as encouraging and promising for an unoptimized and sparse representation like our current DepDM.de.

word	candidate	similarity
	Golfwägelchen	0
target:	*Golfschläger*	.008
Caddie	*Golfplatz*	0
	Golf-Abschlag	0
	billiger [Witz]	**.146**
target:	*leichte [Kutsche]*	.055
Kalauer	*steifer [Hut]*	.036
	[Merkspruch]	0

Table 4: Examples of 'base' items and 'phrase' items. Words in brackets are those with the highest similarity to the target.

The difference in accuracies between DM.de$^\alpha$ and DM.de$^\beta$ indicates that requiring all candidates to have non-zero similarities to the target, as DM.de$^\alpha$ does, does indeed reduce the number of sparsity-related problems. However, the modest increase in accuracy comes with a massive loss in coverage of roughly 50%. Thus, even the large SDEWAC appears not to be large enough to provide sufficiently rich representations for many infrequent targets and candidates.

Finally, we investigated the nature of the 160 items which were not covered by any of the DM.de models. We discovered that the majority of them are adjectives, which we attribute to the fewer number of link types associated with adjectives as opposed to nouns. In the current version of DM.de, adjectives are only "counted" when they occur in `nmod` contexts, i.e., when they are in an attributive construction. In the next iteration of DM.de, we will include predicative uses as well.

6 Conclusion

This paper has reported on ongoing work to create a German Distributional Memory resource. Using a fairly simple set of patterns and collecting counts from a German web corpus, we have been able to perform competitively on a German synonym selection task. Our work can provide a blueprint for the induction of similar resources in other languages and indicates that interesting results can be achieved with relatively little effort.

Nevertheless, we are aware of a number of shortcomings in the model. Some of them relate to preprocessing. In our web corpus, tagging errors are a frequent source of problems. The tagger has difficulties with tokens it was not trained on, e.g. '*[*' is tagged either as a verb and an adjective. One remedy would be to filter out likely tagging errors with simple heuristics such as excluding nouns that are lowercase. Misparses of NP-PP sequences form another problem: in *'während **lange Zeit** von **Säkularisierung** gesprochen wurde'*, the *von*-PP belongs to the phrasal verb *von etwas sprechen* but it is matched by the adjective - noun (*of*) - noun pattern.

Another general problem that we have mentioned repeatedly is sparsity. Our German DM is considerably more sparse than the English DM. We plan to extend the pattern definitions to other syntactic constructions and will consider more sophisticated ways of computing similarity that take advantage of DM's graph structure, such as graph walks (Toutanova et al., 2004).

Finally, we want to revisit our original decision not to use a translation-based approach and plan to combine monolingual and cross-lingual evidence (Naseem et al., 2009) to bootstrap a more reliable DM for new languages. This will, however, require formulating a probabilistic framework for the induction task to weigh the relative evidence from both languages.

Acknowledgments. This work was partially supported by the EC-funded project EXCITEMENT (FP7 ICT-287923) and the German Research Foundation (SFB 732).

References

Marco Baroni and Adam Kilgarriff. 2006. Large Linguistically-Processed Web Corpora for Multiple Languages. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 87–90.

Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):1–49.

Luisa Bentivogli and Emanuele Pianta. 2005. Exploiting parallel texts in the creation of multilingual semantically annotated resources: the MultiSemCor Corpus. *Journal of Natural Language Engineering*, 11(3):247–261.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 89–97, Beijing, China.

Katrin Erk, Sebastian Padó, and Ulrike Padó. 2010. A Flexible, Corpus-Driven Model of Regular and Inverse Selectional Preferences. *Computational Linguistics*, 36(4):723–763, December.

Stefan Evert. 2005. *The statistics of word cooccurrences: word pairs and collocations*. Ph.D. thesis, IMS, Universität Stuttgart.

Gertrud Faaß, Ulrich Heid, and Helmut Schmid. 2010. Design and Application of a Gold Standard for Morphological Analysis: SMOR as an Example of Morphological Evaluation. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta.

Pascale Fung and Benfeng Chen. 2004. BiFrameNet: Bilingual Frame Semantics Resources Construction by Cross-Lingual Induction. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 931–935, Geneva, Switzerland.

Zelig S. Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Eric Joanis, Suzanne Stevenson, and David James. 2006. A general feature space for automatic verb classification. *Natural Language Engineering*, 14(03):337–367.

Thomas K Landauer and Susan T Dumais. 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.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, SIGDOC '86, pages 24–26, New York, NY, USA. ACM.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of COLING/ACL*, pages 768–774, Montreal, QC.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding Predominant Word Senses in Untagged Text. In *Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics*, pages 279–286, Barcelona, Spain.

George A. Miller and Walter G. Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

Saif Mohammad, Iryna Gurevych, Graeme Hirst, and Torsten Zesch. 2007. Cross-Lingual Distributional Profiles of Concepts for Measuring Semantic Distance. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 571–580, Prague, Czech Republic.

Tahira Naseem, Benjamin Snyder, Jacob Eisenstein, and Regina Barzilay. 2009. Multilingual Part-of-Speech Tagging : Two Unsupervised Approaches. *Journal of Artificial Intelligence Research*, 36:1–45.

Sebastian Padó and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

Yves Peirsman and Sebastian Padó. 2011. Semantic relations in bilingual vector spaces. *ACM Transactions in Speech and Language Processing*, 8(2):3:1–3:21.

Sabine Schulte im Walde. 2006. Experiments on the Automatic Induction of German Semantic Verb Classes. *Computational Linguistics*, 32(2):159–194.

Hinrich Schütze. 1992. Dimensions of meaning. In *Proceedings of Supercomputing '92*, pages 787–796, Minneapolis, MN.

Kristina Toutanova, Christopher D. Manning, and Andrew Y. Ng. 2004. Learning random walk models for inducing word dependency distributions. In *Proceedings of ICML*, Banff, BC. ACM Press.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Peter Turney. 2006. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416.

DeWitt Wallace and Lila Acheson Wallace. 2005. *Reader's Digest, das Beste für Deutschland*. Verlag Das Beste, Stuttgart.

David Yarowsky and Grace Ngai. 2001. Inducing Multilingual POS Taggers and NP Bracketers via Robust Projection across Aligned Corpora. In *Proceedings of the 2nd Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 200–207, Pittsburgh, PA.