Graph-based Clustering of Synonym Senses for German Particle Verbs

Moritz Wittmann and **Marion Weller-Di Marco** and **Sabine Schulte im Walde**
Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart
Pfaffenwaldring 5B, 70569 Stuttgart, Germany
{wittmamz,wellermn,schulte}@ims.uni-stuttgart.de

Abstract

In this paper, we address the automatic induction of synonym paraphrases for the empirically challenging class of German particle verbs. Similarly to Cocos and Callison-Burch (2016), we incorporate a graph-based clustering approach for word sense discrimination into an existing paraphrase extraction system, (i) to improve the precision of synonym identification and ranking, and (ii) to enlarge the diversity of synonym senses. Our approach significantly improves over the standard system, but does not outperform an extended baseline integrating a simple distributional similarity measure.

1 Introduction

Alignments in parallel corpora provide a straightforward basis for the extraction of paraphrases by means of re-translating pivots and then ranking the obtained set of candidates. For example, if the German verb *aufsteigen* is aligned with the English pivot verbs *rise* and *climb up*, and the two English verbs are in turn aligned with the German verbs *aufsteigen, ansteigen* and *hochklettern*, then *ansteigen* and *hochklettern* represent two paraphrase candidates for the German verb *aufsteigen*. Bannard and Callison-Burch (2005) were the first to apply this method to gather paraphrases for individual words and multi-word expressions, using translation probabilities as criteria for ranking the obtained paraphrase candidates.

This standard re-translation approach however suffers from a major *re-translation sense problem*, because the paraphrase candidates cannot distinguish between the various senses of the target word or phrase. Consequently, (i) the different senses of the original word or phrase are merged, when the back translations of all pivot words are collected within one set of paraphrase candidates; and (ii) the ranking step does not guarantee that all senses of a target are covered by the top-ranked candidates, as more frequent senses amass higher translation probabilities and are favoured.

paraphrase	valid	sense	gloss
richten	-		*to direct*
abzielen	+	1	*to concentrate*
konzentrieren	+	1	*to concentrate*
orientieren	-		*to orientate*
organisieren	+	2	*to organize*
beruhen	-		*to rely*
anstreben	-		*to strive*
lenken	-		*to steer*
zielen	-		*to aim*
erreichen	+	3	*to achieve*

Table 1: Top-ranked paraphrases for *ausrichten*.

Recently, Cocos and Callison-Burch (2016) proposed two approaches to distinguish between paraphrase senses (i.e., aiming to solve problem (i) above). In this paper, we address both facets (i) and (ii) of the *re-translation sense problem*, while focusing on an emprically challenging class of multi-word expressions, i.e., German particle verbs (PVs). German PVs can appear morphologically joint or separated (such as *steigt ... auf*), and are often highly ambiguous. For example, the 138 PVs we use in this paper have an average number of 5.3 senses according to the *Duden*[1] dictionary.

Table 1 illustrates the re-translation sense problem for German PVs. It lists the 10 top-ranked paraphrases for the target verb *ausrichten* obtained with the standard method. Four synonyms in the 10 top-ranked candidates were judged valid according to the *Duden*, covering three out of five senses listed in the *Duden*. Synonyms for a fourth sense "*to tell*" (*sagen, übermitteln, weitergeben*) existed in the candidate list, but were ranked low.

[1] www.duden.de

Proceedings of the 12th Workshop on Multiword Expressions, pages 38–43,
Berlin, Germany, August 7-12, 2016.

Our approach to incorporate word senses into the standard paraphrase extraction applies a *graph-based clustering* to the set of paraphrase candidates, based on a method described in (Apidianaki and He, 2010; Apidianaki et al., 2014). It divides the set of candidates into clusters by reducing edges in an originally fully-connected graph to those exceeding a dynamic similarity threshold. The resulting clusters are taken as paraphrase senses, and different parameters from the graphical clustering (such as connectedness in clusters; cluster centroid positions; etc.) are supposed to enhance the paraphrase ranking step. With this setting, we aim to achieve higher precision in the top-ranked candidates, and to cover a wider range of senses as the original re-translation method.

2 Related Work

Bannard and Callison-Burch (2005) introduced the idea of extracting paraphrases with the re-translation method. Their work controls for word senses regarding specific test sentences, but not on the type level. Subsequent approaches improved the basic re-translation method, including Callison-Burch (2008) who restrict paraphrases by syntactic type; and Wittmann et al. (2014) who add distributional similarity between paraphrase candidate and target word as a ranking feature. Approaches that applied extracted paraphrases relying on the re-translation method include the evaluation of SMT (Zhou et al., 2006) and query expansion in Q-A systems (Riezler et al., 2007).

Most recently, Cocos and Callison-Burch (2016) proposed two clustering algorithms to address one of the sense problems: They discriminate between target word senses, exploiting hierarchical graph factorization clustering and spectral clustering. The approaches cluster all words in the Paraphrase Database (Ganitkevitch et al., 2013) and focus on English nouns in their evaluation.

A different line of research on synonym extraction has exploited distributional models, by relying on the contextual similarity of two words or phrases, e.g. Sahlgren (2006), van der Plas and Tiedemann (2006), Padó and Lapata (2007), Erk and Padó (2008). Typically, these methods do not incorporate word sense discrimination.

3 Synonym Extraction Pipeline

This section lays out the process of extracting, clustering and ranking synonym candidates.

3.1 Synonym Candidate Extraction

Following the basic approach for synonym extraction outlined by Bannard and Callison-Burch (2005), we gather all translations (i.e., pivots) of an input particle verb, and then re-translate the pivots. The back translations constitute the set of synonym candidates for the target particle verb.

In order to rank the candidates according to how likely they represent synonyms, each candidate is assigned a probability. The *synonym probability* $p(e_2|e_1)_{e2 \neq e1}$ for a synonym candidate verb e_2 given a target particle verb e_1 is calculated as the product of two translation probabilities: the *pivot probability* $p(f_i|e_1)$, i.e. the probability of the English pivot f_i being a translation of the particle verb e_1, and the *return probability* $p(e_2|f_i)$, i.e. the probability that the synonym candidate e_2 is a translation of the English pivot f_i. The final synonym score for e_2 is the sum over all pivots $f_{1..n}$ that re-translate into the candidate:

$$p(e_2|e_1)_{e2 \neq e1} = \sum_{i=1}^{n} p(f_i|e_1)p(e_2|f_i) \quad (1)$$

The translation probabilities are based on relative frequencies of the counts in a parallel corpus, cf. section 4.1.

Filtering We apply filtering heuristics at the *pivot probability step* and the *return probability step*: obviously useless pivots containing only stop-words (e.g. articles) or punctuation are discarded. In the back-translation step, synonym candidates that did not include a verb are removed. Furthermore, we removed pivots (*pivot probability step*) and synonym candidates (*return probability step*) consisting only of light verbs, due to their lack of semantic content and tendency to be part of multi-word expressions. If left unfiltered, light verbs often become super-nodes in the graphs later on (see section 3.2) due to their high distributional similarity with a large number of other synonym candidates. This makes it difficult to partition the graphs into meaningful clusters with the algorithm used here.

Distributional Similarity We add distributional information as an additional feature for the ranking of synonym candidates, because weighting the score from equation (1) by simple multiplication with the distributional similarity between the candidate and the target (as obtained from large corpus data, cf. section 4.1), has been found to improve the ranking (Wittmann et al., 2014).

Properties of the clusters:	
C(#(cand))	number of synonym candidates in a cluster
C(av-sim(cand,c))	average distributional similarity between synonym candidates in a cluster and the cluster centroid
C(av(#(e)))	average number of edges in the clusters of the cluster analyses
C(#(e))	total number of edges in a cluster
C(av-sim(cand,v))	average distributional similariy between synonym candidates in a cluster and the target PV
C(av-sim(cand,gc))	average distributional similariy between all synonym candidates and the global centroid
C(sim(c,v))	distributional similarity between a cluster centroid and the target PV
C(con)	connectedness of a cluster
Properties of the synonym candidates:	
S(tr)	translation probability of a synonym candidate
S(#(e))	number of edges of a synonym candidate
S(cl%(#(e)))	proportion of cluster edges for a synonym candidate
S(sim(cand,v))	distributional similarity between a synonym candidate and the target PV
S(sim(cand,c))	distributional similarity between a synonym candidate and the cluster centroid
S(sim(cand,gc))	distributional similarity between a synonym candidate and the global centroid

Table 2: Properties of synonym candidates and clusters.

3.2 Graph-Based Clustering of Candidates

The clustering algorithm suggested by Apidianaki et al. (2014) is adopted for clustering all extracted synonym candidates for a specific particle verb target. In a first step, a fully connected undirected graph of all synonym candidates is created as a starting point, with nodes corresponding to synonym candidates and edges connecting two candidates; edge weights are set according to their distributional similarity. In a second step, a similarity threshold is calculated, in order to delete edges with weights below the threshold. The threshold is initialized with the mean value between all edge weights in the fully connected graph. Subsequently, the threshold is updated iteratively:

1. The synonym candidate pairs are partitioned into two groups: P_1 contains pairs with similarities below the current threshold, and P_2 contains pairs with similarities above the current threshold **and** sharing at least one pivot.

2. A new threshold is set: $T = \frac{A_{P_1}+A_{P_2}}{2}$, where A_{P_i} is the mean over all similarities in P_i.

After convergence, the resulting graph consists of disconnected clusters of synonym candidates. Singleton clusters are ignored. The sub-graphs represent the cluster analysis to be used in the ranking of synonyms for the target particle verb.

Iterative Application of Clustering Algorithm Because the resulting clusterings of the synonym candidates typically contain one very large (and many small) clusters, we extend the original algorithm and iteratively re-apply the clustering: After one pass of the clustering algorithm as described above (T_1), the resulting set of connected synonym candidates becomes the input to another iteration of the algorithm ($T_{2...n}$). Each iteration of the algorithm results in a smaller and more strongly partitioned sub-graph of the initially fully connected graph because the similarity threshold for edges becomes successively higher.

3.3 Synonym Candidate Ranking

Assuming that clusters represent senses, we hypothesize that combining properties of individual synonym candidates with properties of the graph-based clusters of synonym candidates results in a ranking of the synonym candidates that overcomes both facets of the *re-translation sense problem*: Including synonym candidates from various clusters should ensure more senses of the target particle verbs in the top-ranked list; and identifying salient clusters should improve the ranking. Table 2 lists the properties of the individual synonym candidates S and the properties of the graph-based cluster analyses C that we consider potentially useful. For the experiments in section 4, we use all combinations of S and C properties.

4 Experiments, Results and Discussion

4.1 Data and Evaluation

For the extraction of synonym candidates, we use the German–English version of Europarl (1.5M parallel sentences) with GIZA++ word alignments for the extraction of synonym candidates. In the alignments, the German data is lemmatized and reordered in order to treat split occurrences of particle and verb as a single word (Schmid et al., 2004; Schmid, 2004; Fraser, 2009).

	system	ranking	prec. top 10	prec. top 20	no. of senses	prop. of senses
1	basic	S(tr)	34.57	25.76	1.99	45.59
2	basic + distr. sim.	S(tr) · sim(cand,v)	38.19	27.79	**2.04**	**46.89**
3	clustering + ranking (1)	S(tr) · S(sim(cand,v)) · C(#(e))	**38.41**	**27.90**	**2.04**	**46.89**
4	clustering + ranking (2)	S(tr) · S(sim(cand,v)) · C(av-sim(cand,gc))	38.26	**27.90**	**2.04**	**46.89**
5	clustering + ranking (3)	S(tr) · S(sim(cand,v))	38.19	**27.90**	**2.04**	**46.89**
6	clustering + ranking (4)	S(tr) · S(sim(cand,v)) · C(sim(cand,v))	38.12	**27.90**	**2.04**	**46.89**
7	clustering + ranking (5)	S(tr) · S(sim(cand,v)) · C(con)	37.97	27.83	2.03	46.65

Table 3: Evaluation of basic approaches and best five rankings: precision & no./proportion of senses.

The distributional similarity sim is determined by cosine similarities between vectors relying on co-occurrences in a window of 20 words. We use the German web corpus *DECOW14AX* (Schäfer and Bildhauer, 2012; Schäfer, 2015) containing 12 billion tokens, with the 10,000 most common nouns as vector dimensions. The feature values are calculated as *Local Mutual Information (LMI)*, cf. (Evert, 2005).

Our dataset contains the same 138 German particle verbs from Europarl as in previous work (Wittmann et al., 2014), all PVs with a frequency $f \geq 15$ and at least 30 synonyms listed in the *Duden* dictionary. For the evaluation, we also rely on the *Duden*, which provides synonyms for the target particle verbs and groups the synonyms by word sense. We consider four evaluation measures, and compare the ranking formulas by macro-averaging each of the evaluation measures over all 138 particle verbs:

- *Precision* among the *10/20 top-ranked* synonym candidates.
- *Number and proportion of senses* represented among the 10 top-ranked synonyms.

4.2 Results

The basic system (line 1 in table 3) only relies on the translation probabilities ($S(tr)$). It is extended by incorporating the distributional similarity between the target particle verb and the synonym candidates (line 2).

Our five best rankings with one iteration of graphical clustering (T_1) are shown in lines 3-7. All of these include the translation probability and the distributional similarity between candidate and particle verb; only one makes use of cluster information. Thus, the simple distributional extension is so powerful that additional cluster information cannot improve the system any further. The most relevant cluster measure is the number of edges

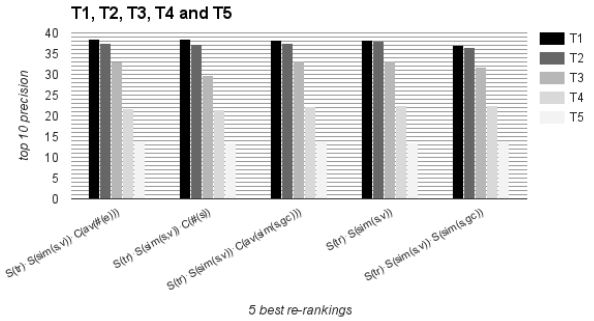

Figure 1: Evaluating an iterative application of the clustering algorithm ($T_{1...5}$).

of the cluster *C(#(e))*, an indication of cluster size and connectedness.

While the best three clustering systems[2] outperform the extended basic system (line 2) in terms of top-10/top-20 precision, none of the improvements is significant.[3]

Also, the number and proportion of senses remain the same as in the basic approach with distributional extension. Further iterations of the clustering step ($T_{2...n}$) up to $n = 8$ lead to increasingly worse precision scores and sense detection, cf. figure 1 for $T_{1...5}$.

4.3 Discussion

Overall, the distributional similarity between the target word and the synonym candidates represents the strongest extension of the basic retranslation approach, and the cluster graphs do not provide further useful information. A breakdown of the cluster analyses revealed that the cluster sizes are very unevenly distributed. Typically, there is one very large cluster and several considerably smaller clusters, as shown by the first part of table 4, which depicts the proportion of *synonym candidates* in the largest cluster vs. the average

[2]The systems in lines 2 and 5 use the same ranking information but have different results, due to the removal of singletons from the graphs in the clustering, see section 3.2.

[3]χ^2 without Yates' correction

candidates	$\mathbf{T}_1$	$\mathbf{T}_2$	$\mathbf{T}_3$	$\mathbf{T}_4$	$\mathbf{T}_5$
prop. largest [%]	99.95	97.09	59.69	8.19	7.15
avg. prop rest [%]	0.13	0.16	0.34	0.80	1.60
...					
synonyms	$\mathbf{T}_1$	$\mathbf{T}_2$	$\mathbf{T}_3$	$\mathbf{T}_4$	$\mathbf{T}_5$
prop. largest [%]	99.90	96.18	60.24	9.70	8.55
avg. prop. rest [%]	0.25	0.21	0.33	0.79	1.58
...					
senses	$\mathbf{T}_1$	$\mathbf{T}_2$	$\mathbf{T}_3$	$\mathbf{T}_4$	$\mathbf{T}_5$
prop. largest [%]	100.00	96.85	74.30	15.25	9.83
avg. prop. rest [%]	1.23	0.92	1.08	1.58	2.34

Table 4: Distribution of *candidates*, *synonyms* and *senses* in the largest cluster vs. all other clusters in the iterations T_1-T_5.

proportion of candidates in the remaining clusters. In addition, we found that most *correct synonyms* are also in the largest cluster (middle part of table 4). Accordingly, the cluster analyses do not represent partitions of the target verb senses, but most *senses* are in the largest cluster (bottom part of table 4).

Consequently, while the synonym features are useful for ranking the set of candidates, cluster-level features are ineffective as they are derived from effectively meaningless cluster analyses.[4] While re-applying the clustering step gradually overcomes the uneven cluster distribution (iterations T_2-T_5 in table 4), the sizes of the graphs decrease dramatically. For example (not depicted in table 4), on average there are only 169 candidates left in T_5 compared to 1,792 in T_1, with an average of 2.8 correct synonyms instead of 22.5, and an average of 1.7 senses instead of 4.5.

We assume that partitioning the candidate set according to senses in combination with the cluster-level measures is a valid approach to deal with the word sense problem, but based on our analysis we conclude that either (i) the context vectors are not suitable to differentiate between senses, or that (ii) the clustering algorithm is inapt for this scenario. A possible solution might be to apply the algorithms suggested in Cocos and Callison-Burch (2016). Finally, no weighting was applied to any of the properties listed in table 2. This could be improved by using a held-out data development set, and a greater number of particle verbs (we only use 138) would probably be needed as well.

[4]Intuitively, many of the smaller clusters are actually semantically coherent, but often not semantically related to the target verb and thus not helpful.

5 Summary

We hypothesized that graph-based clustering properties in addition to synonym candidate properties should improve the precision of synonym identification and ranking, and extend the diversity of synonym senses. Unfortunately, our extensions failed, and analyses of cluster properties revealed that future work should improve the vector representations and compare other clustering algorithms. One should keep in mind, however, that we focused on a specifically challenging class of multi-word expressions: highly ambiguous German particle verbs.

Acknowledgments

This work was funded by the DFG Research Project "Distributional Approaches to Semantic Relatedness" (Moritz Wittmann, Marion Weller-Di Marco) and the DFG Heisenberg Fellowship SCHU-2580/1-1 (Sabine Schulte im Walde).

References

Marianna Apidianaki and Yifan He. 2010. An Algorithm for Cross-Lingual Sense-Clustering tested in a MT Evaluation Setting. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 219–226, Paris, France.

Marianna Apidianaki, Emilia Verzeni, and Diana McCarthy. 2014. Semantic Clustering of Pivot Paraphrases. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 4270–4275, Reykjavik, Iceland.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with Bilingual Parallel Corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 597–604, Ann Arbor, MI, USA.

Chris Callison-Burch. 2008. Syntactic Constraints on Paraphrases Extracted from Parallel Corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 196–205, Honolulu, Hawaii, USA.

Anne Cocos and Chris Callison-Burch. 2016. Clustering Paraphrases by Word Sense. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1463–1472, San Diego, CA, USA.

Katrin Erk and Sebastian Padó. 2008. A Structured Vector Space Model for Word Meaning in Context. In *Proceedings of the joint Conference on Empirical*

Methods in Natural Language Processing and Computational Natural Language Learning, pages 897–906, Waikiki, Hawaii, USA.

Stefan Evert. 2005. *The Statistics of Word Co-Occurrences: Word Pairs and Collocations*. Ph.D. thesis, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Alexander Fraser. 2009. Experiments in Morphosyntactic Processing for Translating to and from German. In *Proceedings of the 4th Workshop on Statistical Machine Translation*, pages 115–119, Athens, Greece.

Juri Ganitkevitch, Benjamin van Durme, and Chris Callison-Burch. 2013. PPDB: The Paraphrase Database. In *Proceedings of the 14th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, GA, USA.

Sebastian Padó and Mirella Lapata. 2007. Dependency-based Construction of Semantic Space Models. *Computational Linguistics*, 33(2):161–199.

Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridid, Vibhu Mittal, and Yi Liu. 2007. Statistical Machine Translation for Query Expansion in Answer Retrieval. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 464–471, Prague, Czech Republic.

Magnus Sahlgren. 2006. *The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-Dimensional Vector Spaces*. Ph.D. thesis, Stockholm University.

Roland Schäfer and Felix Bildhauer. 2012. Building Large Corpora from the Web Using a New Efficient Tool Chain. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 486–493, Istanbul, Turkey.

Roland Schäfer. 2015. Processing and Querying Large Web Corpora with the COW14 Architecture. In *Proceedings of the 3rd Workshop on Challenges in the Management of Large Corpora*, pages 28–34, Mannheim, Germany.

Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. SMOR: A German Computational Morphology Covering Derivation, Composition and Inflection. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 1263–1266, Lisbon, Portugal.

Helmut Schmid. 2004. Efficient Parsing of Highly Ambiguous Context-Free Grammars with Bit Vectors. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 162–168, Geneva, Switzerland.

Lonneke van der Plas and Jörg Tiedemann. 2006. Finding Synonyms using Automatic Word Alignment and Measures of Distributional Similarity. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 866–873, Sydney, Australia.

Moritz Wittmann, Marion Weller, and Sabine Schulte im Walde. 2014. Automatic Extraction of Synonyms for German Particle Verbs from Parallel Data with Distributional Similarity as a Re-Ranking Feature. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 1430–1437, Reykjavik, Iceland.

Liang Zhou, Chin-Yew Lin, and Eduard Hovy. 2006. Re-evaluating Machine Translation Results with Paraphrase Support. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 77–84, Sydney, Australia.