Unsupervised Compound Splitting With Distributional Semantics Rivals Supervised Methods

Martin Riedl and **Chris Biemann**
Language Technology
Computer Science Department, Technische Universität Darmstadt
Hochschulstrasse 10, D-64289 Darmstadt, Germany
{riedl,biem}@cs.tu-darmstadt.de

Abstract

In this paper we present a word decompounding method that is based on distributional semantics. Our method does not require any linguistic knowledge and is initialized using a large monolingual corpus. The core idea of our approach is that parts of compounds (like "candle" and "stick") are semantically similar to the entire compound, which helps to exclude spurious splits (like "candles" and "tick"). We report results for German and Dutch: For German, our unsupervised method comes on par with the performance of a rule-based and a supervised method and significantly outperforms two unsupervised baselines. For Dutch, our method performs only slightly below a rule-based optimized compound splitter.

1 Introduction

Germanic and agglutinative languages (e.g. German, Swedish, Finnish, Korean) have a productive morphology that allows the formation of not space-separated compounds in a much larger extent than e.g. in English. The task of separating such compounds into their corresponding single word (sub-) units is called compound splitting or decompounding.

Decompounding showed impact in several NLP applications, e.g. ASR (Adda-Decker and Adda, 2000), MT (Koehn and Knight, 2003) or IR (Monz and de Rijke, 2001), and is generally perceived as a crucial component for the processing of respective languages. However, most existing systems rely on dictionaries or are trained in a supervised fashion. Both approaches require substantial manual work and do not adapt to vocabulary change. In this paper we introduce an unsupervised method for decompounding that relies on distributional semantics. For the computation of the semantic model we solely rely on a tokenized monolingual corpus and do not require any further linguistic processing. Most previous research on compound splitting concentrates on the detection of lemmas that form the compound. Whereas this is important for several tasks, in this work we focus on the splitting of a compound into its word units without any base form reduction, arguing that lemmatization is either part of the task pipeline anyways (e.g. IR) or not required (e.g. for ASR).

2 Related Work

Approaches to automatic decompounding can be classified into corpus-driven approaches and supervised approaches. Corpus-driven approaches are usually informed by a frequency list (Koehn and Knight, 2003), by a probabilistic model (Schiller, 2005), by parallel corpora (Koehn and Knight, 2003; Macherey et al., 2011) or by the existence of periphrases (i.e. reformulations) in large monolingual corpora (Holz and Biemann, 2008). As with other tasks, supervised approaches are usually superior to unsupervised approaches if sufficient training material is provided. A straightforward yet effective supervised decompounding system is contained in the ASV Toolbox (Biemann et al., 2008), which uses trie-based datastructures for recursively splitting compounds based on learned splits. Alfonseca

Proceedings of NAACL-HLT 2016, pages 617–622,
San Diego, California, June 12-17, 2016.

et al. (2008) combine several signals, including web anchor text, in an SVM-based supervised splitter. A widely used German decompounder is JWordSplitter[1], which is based on word lists of compound parts as well as manually crafted blacklists and whitelists. The NL Splitter[2] uses similar technology for Dutch compound decomposition. An unsupervised approach is presented in (Koehn and Knight, 2003): out of several splits as given by matching parts of the compound to a vocabulary list, they pick the split with the highest geometric mean of word frequencies, which is entirely corpus-driven but ignores semantic relations between the compound and its parts. Another unsupervised system is proposed by Daiber et al. (2015). They propose an analogy-based approach, which relies on word embeddings.

Decompounding is evaluated either intrinsically or in a task that benefits from it, e.g. IR (Monz and de Rijke, 2001), MT (Koehn and Knight, 2003; Macherey et al., 2011) or ASR (Adda-Decker and Adda, 2000; Ordelman et al., 2003).

3 Method

The introduced method, called SECOS (SEmantic COmpound Splitter)[3], is based on the hypothesis that compounds are similar to their constituting word units. Our method is based on a distributional thesaurus (DT) that is computed, based on the distributional hypothesis (Harris, 1951), using a monolingual background corpus and does not require any language-specific rules or preprocessing. We exemplify the method based on the compound noun *Bundesfinanzministerium* (*federal finance ministry*), which is assembled of the words *Bundes* (*federal*), *Finanz* (*finance*) and *Ministerium* (*ministry*).

Our method consists of three stages: First we extract a candidate word set that defines the possible word units of compounds. We present several approaches to generate such candidates. Second, we use a general method that splits the compound based on a candidate word set. Using the different candidate sets, we obtain different compound splits. Finally, we define a mechanism that ranks these splits and returns the top-ranked one.

3.1 Candidate Extraction

For the extraction of all candidates in C, we use a distributional thesaurus (DT) that is computed on a background corpus. We present three approaches for the generation of candidate sets.

When we retrieve the l most similar terms for a word w from a DT, we observe well-suited candidates that are nested in w. For example *Bundesfinanzministerium* is similar to *Bund*, *Bundes* and *Finanzministerium*. Extracting the most similar terms that are nested in w results in the first split candidate set, called *similar candidate units*.

However, only for few terms we observe nested candidates in the most similar words. Thus, we require methods to generate "back-off" candidates.

First, we introduce the *extended similar candidate units*. Here, we extract the l most similar terms for w and then grow this set by again adding their respective l most similar words. Based on these terms, we extract all words that are nested in w. This results into more but less precise decompounding candidates.

As the coverage might still be insufficient to decompound all words (e.g. entirely unseen compounds), we propose a method to generate a global dictionary of single atomic word units. For this, we iterate over the entire vocabulary of the background corpus, apply the compound splitter (see Section 3.2) to all words where we find *similar candidate units*. Then, we add these detected units to the dictionary. Finally, for word w subject to decompounding, we first extract all nested words NW from this dictionary. Then, we remove all words in NW that are nested itself in NW, resulting in the candidate set we call *generated dictionary*.

3.2 Compound Splitting

Here, we introduce the decompounding algorithm for a given candidate set. For decompounding the word w, we require a set of candidate words C. Each word in the candidate set needs to be a substring of w. We do not include candidates in C that have less than ml characters. Additionally, we apply a frequency threshold of wc. These mechanisms are intended to rule out spurious parts and 'words'

[1] https://github.com/danielnaber/jwordsplitter

[2] http://ilps.science.uva.nl/resources/compound-splitter-nl/

[3] An implementation and models for German and Dutch are available at: https://github.com/riedlma/SECOS

word w	Bundesfinanzministerium
candidates C w. ml=3	Finanzministerium, Ministerium, Bunde, Bund, Bundes, Minister
split possibilities	Bund-e-s-finanz-minister-ium
Merging character n-grams	
suffix-prefix	Bundes-finanz-ministerium
prefix-suffix	Bund-esfinanz-ministerium

Table 1: Examples of the output of our algorithms for the example term *Bundesfinanzministerium*.

that are in fact short abbreviations. We show candidates, extracted from the *similar candidate unit*, with $ml = 3$ for the example term in Table 1. Then, we iterate over each candidate $c_i \in C$ and add its beginning and ending position within w to the set S. This set is then used to identify possible split positions of w. For this, we iterate from left to right and add all split possibilities to the word w. This approach over-generates split points, as can be observed for the example word, which is split into 6 units: *Bund-e-s-finanz-minister-ium*.

To merge character n-grams, we use a suffix- and prefix-based method. The suffix merging method appends all character n-grams with n below ms to the left word. The prefix method merges all character n-grams with n below mp to the word on the right side. To avoid remaining prefixes/suffixes, we apply the opposite method afterwards. For the German language, the suffix-prefix ordering mostly yields the best output. The suffix-prefix-based approach results to *Bundes-finanz-ministerium* and the prefix-suffix method to *Bund-esfinanz-ministerium*. However for some words, the prefix-suffix generates the correct compound split, e.g. for the word *Zuschauer-er-wartung* (*audience* + *he* + *service*), which is correctly decompounded as *Zuschauer-erwartung* (*audience+expectation*).

In order to select the correct split, we compute the geometric mean of the joint probability for each split variation. For this we use word counts from a background corpus. In addition to the geometric mean formula introduced in (Koehn and Knight, 2003), we apply a smoothing factor[4] ϵ to each frequency in order to assign non-zero values to unknown units. This yields the following formula for a compound w, which is decomposed into the units $w_i, \ldots, w_N$:

$$p(w) = \left(\prod_{i}^{N} \frac{wordcount(w_i) + \epsilon}{total\_wordcount + \epsilon * \#words} \right)^{\frac{1}{N}} \quad (1)$$

Here, $\#word$ denotes the total number of words in the background corpus and $total\_wordcount$ is the sum of all word counts. Then, we select the split variation with the highest geometric mean.[5] In our example, this is the prefix-suffix-merged candidate *Bundes-finanz-ministerium*.

3.3 Split Ranking

We have examined schemes of priority ordering for integrating information from different candidate sets, e.g. using the *similar candidate units* first and only apply the other candidate sets if no split was found. However, preliminary experiments revealed that it was always beneficial to generate splits based on all three candidate sets and use the geometric mean scoring as outlined above to select the best split as decomposition of a word.

4 Datasets

For testing the performance of our method, we chose four datasets. The first dataset was manually labeled by Holz and Biemann (2008) and consists of 700 German nouns from different frequency bands. The second dataset consists of 158,653 nouns from the German newspaper magazine c't[6] and was created by Marek (2006). As third dataset we use a noun compound dataset of 54,571 nouns from GermaNet[7], which has been constructed by Henrich and Hinrichs (2011).[8] While converting these datasets for the task of compound splitting, we do not separate words in the gold standard, which comprise of prepositions, e.g. the word *Abgang (outflow)* is not split into *Ab-gang (off walk)*. To show the language independency of our method, we apply it to a

[4] We set $\epsilon = 0.01$. Using values in the range of $\epsilon = [0.0001, 1]$ we observe marginally higher scores using smaller values.

[5] Whereas our method mostly does not assume language knowledge, we uppercase the first letter of each w_i, when we apply our method on German texts.

[6] http://heise.de/ct

[7] available at: http://www.sfs.uni-tuebingen.de/lsd/documents/compounds/split_compounds_from_GermaNet10.0.txt

[8] We follow Schiller (2005) and remove all words including dashs. This only affects the GermaNet dataset and reduces the effective test set to 53,118 nouns.

Dutch compound dataset proposed by van Zaanen et al. (2014). This dataset comprises of 21,997 nouns.

5 Experimental Setting

The corpus-based DT is computed following the approach by Biemann and Riedl (2013). For each word, we use the left and the right neighboring word as context representation to compute the DT. For the generation of the split candidates we rely on the $l = 200$ most similar entries for each word.

The German DT is computed based on 70 million newspaper sentences, which are extracted from the Leipzig Corpora Collection (LCC) (Richter et al., 2006). For the generation of the Dutch DT, we use the Dutch web corpus (Schäfer and Bildhauer, 2013), which is composed from 259 million sentences.[9]

We evaluate the performance of the algorithms using precision and recall as defined by Koehn and Knight (2003). As unsupervised baselines we use the split ranking by (Koehn and Knight, 2003), called KK, and the semantic analogy-based splitter (SAS) from Daiber et al. (2015).[10] As advanced systems we apply the lexicon- and rule-based JWordSplitter (JWS) and the supervised decompounding algorithm (ASV), introduced by Holz and Biemann (2008).[11] For all algorithms, we converted the splits to capture all characters in the words, reverting base forms to full forms. For Dutch, we apply the KK baseline and the NL Splitter.

6 Method Tuning

We use the small dataset with the 700 German nouns to find the best parameter settings of our method. The highest F1-scores are obtained using candidates with a frequency above 50 (wc=50) and that have more than 4 characters (ml=5). Further we append only prefixes and suffixes equal or shorter than 3 characters (ms=3 and mp=3).

The highest precision is achieved with the similar candidate units (see Table 2). However, the recall is lowest as for many words no information is available. Using the extended similarities, the precision decreases and the recall increases. The best overall performance is achieved with the generated dictionary, which yields an F1-measure of 0.9384. The selection mechanism using the geometric mean scoring brings F1-measure up to 0.9515 on this dataset.

	P	R	F1
similar cand.	**0.9880**	0.6798	0.8054
ext. sim. cand.	0.9617	0.7304	0.8303
gen. dictionary	0.9576	**0.9199**	**0.9384**
geom. mean scoring	0.9698	0.9338	0.9515

Table 2: Precision (P), Recall (R) and F1-Measure (F1) for the 700 compound nouns using different split candidates.

7 Results

In this section we compare the performance of our method against the unsupervised baselines and the knowledge-based systems (see Table 3).

		P	R	F1
700	JWS	0.9328	0.9037	0.9180
	ASV	0.9584	0.9238	0.9408
	SAS	0.8723	0.6224	0.7265
	KK	0.9532	0.7513	0.8403
	SECOS	**0.9698**	**0.9338**	**0.9515**
c't	JWS	0.9557	**0.9045**	**0.9294***
	ASV	0.9571	0.8980	0.9266
	SAS	0.9303	0.5428	0.6856
	KK	0.9432	0.8114	0.8723
	SECOS	**0.9606**	0.8809	0.9190
GermaNet	JWS	0.9248	0.8734	0.8983
	ASV	0.9346	**0.8866**	0.9100
	SAS	0.8861	0.6188	0.7287
	KK	0.9486	0.7361	0.8289
	SECOS	**0.9543**	0.8773	**0.9142***
Dutch	NL Splitter	**0.9706**	**0.8694**	**0.9172***
	KK	0.9579	0.7735	0.8559
	SECOS	0.9624	0.8272	0.8897

Table 3: Results for three German datasets and for one Dutch dataset. The significantly best results are marked with an asterisk (*).

For the 700 nouns we achieve the highest precision, recall and F1-measure using our method. However, we have tuned our parameters on this dataset. Our improvement in terms of F-score is not significant[12] with respect to the ASV system, but with re-

[9] available at: http://webcorpora.org/.

[10] https://github.com/jodaiber/semantic_compound_splitting

[11] http://wortschatz.uni-leipzig.de/~cbiemann/software/toolbox/.

[12] We perform a Wilcoxon signed-rank test between the F1-

spect to all other systems on this dataset. Nevertheless, JWS is based on a manually created dictionary and ASV uses a supervised algorithm. On this dataset, ASV outperforms JWS. Due to their low recall, both unsupervised baselines (SAS and KK) achieve significantly lower F1-scores than SECOS.

Using the c't dataset we observe a different trend. Here, the best results are observed by using JWS followed by ASV and our method. Nevertheless, our method yields the highest precision value. Again, SAS and KK score lowest.

For the GermaNet dataset, our method significantly outperforms all others. Similar to the evaluation with the 700 nouns, JWS performs lower than the decompounding method from the ASV toolbox. Whereas our method obtains lower recall than ASV and JWS, it still significantly outperforms the unsupervised baselines and yields the overall highest precision.

In a last experiment, we show the performance on the Dutch dataset. As no trained models for JWS and ASV are available, we did not use these tools but compare to NL splitter, achieving a competitive precision but lower recall. This is caused by many short split candidates that are not detected due to the ml parameter. However, our method still beats the KK baseline significantly.

8 Error Analysis

In order to understand the errors of our method, we analyzed the compounds that have been split incorrectly. Considering the 700 German compounds our method splits 12.17% incorrectly, for the Dutch dataset, we observe the highest percentage of 32.60% incorrectly split compounds (see Table 4).

In addition, we analyzed how many compounds have been split in fewer parts (*under-split*), more parts (*over-split*) than the gold data or have the same number of splits, which, however, are incorrect (*wrongly-split*). For all datasets we observe a general trend: our method tends to suppress splitting compounds, due to the parameters *ms* and *mp* that suppress very short parts. Compounds that are split at entirely incorrect positions constitute the lowest error class. We also analyzed for incorrectly split compounds how often our method missed a split, performed a wrong split and split correctly (see bottom three lines in Table 4). This analysis supports the previous finding: most errors of our SECOS method consist of missed splits.

scores of each candidate assuming $p < 0.01$.

dataset	700	c't	GermaNet	Dutch
	number of compounds			
# incorrect	85	35177	12532	7258
% incorrect	12.17	22.17	23.26	32.60
under-split	47	23773	7972	5849
over-split	33	7843	3578	806
wrongly-split	5	3561	982	603
	number of splits			
missed	55	29213	8968	6612
wrong	43	12703	4669	1520
correct	43	20381	3777	1743

Table 4: Number of compounds that have been split incorrectly with respect to the gold data. We report numbers of how many of these compounds are split fewer (*under-split*), more often (*over-split*) or equally (*wrongly-split*) in comparison to the gold standard. In addition, we show the total number of missed, wrong and correct splits for these compounds.

9 Conclusion

In this paper we have introduced an unsupervised method for decompounding words that is based on distributional semantics. We show the impact of its components and tune its parameters on a small German dataset. On two large German datasets, we demonstrate a performance of our method that is competitive to supervised and rule-based tools and outperforms two unsupervised baselines by a large margin. Further, we demonstrated its language-independence by achieving a good performance on a Dutch dataset. In the future, we would like to assess the impact of SECOS in task-based settings as well as apply it to other compounding languages.

Acknowledgments

This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) within the SeMSch project. Additionally, we want to thank the anonymous reviewers for their helpful comments.

References

Martine Adda-Decker and Gilles Adda. 2000. Morphological Decomposition for ASR in German. In *In Pro-*

ceedings of the Workshop on Phonetics and Phonology in ASR, PHONUS 5, pages 129–143, Saarbrücken, Germany.

Enrique Alfonseca, Slaven Bilac, and Stefan Pharies. 2008. Decompounding Query Keywords from Compounding Languages. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, ACL-HLT '08, pages 253–256, Columbus, OH, USA.

Chris Biemann and Martin Riedl. 2013. Text: Now in 2D! A Framework for Lexical Expansion with Contextual Similarity. *Journal of Language Modelling*, 1(1):55–95.

Chris Biemann, Uwe Quasthoff, Gerhard Heyer, and Florian Holz. 2008. ASV Toolbox: a Modular Collection of Language Exploration Tools. In *Proceedings of the International Conference on Language Resources and Evaluation*, LREC '08, pages 1760–1767, Marrakech, Morocco.

Joachim Daiber, Lautaro Quiroz, Roger Wechsler, and Stella Frank. 2015. Splitting Compounds by Semantic Analogy. In *Proceedings of the 1st Deep Machine Translation Workshop*, pages 20–28, Prague, Czech Republic.

Zellig S. Harris. 1951. *Methods in Structural Linguistics*. University of Chicago Press, Chicago.

Verena Henrich and Erhard Hinrichs. 2011. Determining Immediate Constituents of Compounds in GermaNet. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 420–426, Hissar, Bulgaria.

Florian Holz and Chris Biemann. 2008. Unsupervised and Knowledge-Free Learning of Compound Splits and Periphrases. In *Proceedings of the 9th International Conference on Computational Linguistics and Intelligent Text Processing (CICLING)*, pages 117–127, Haifa, Israel.

Philipp Koehn and Kevin Knight. 2003. Empirical Methods for Compound Splitting. In *In Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '03, pages 187–193, Budapest, Hungary.

Klaus Macherey, Andrew M. Dai, David Talbot, Ashok C. Popat, and Franz Och. 2011. Language-independent Compound Splitting with Morphological Operations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, HLT '11, pages 1395–1404, Portland, OR, USA.

Torsten Marek. 2006. Analysis of German Compounds Using Weighted Finite State Transducers. Bachelor thesis, Universität Tübingen.

Christof Monz and Maarten de Rijke. 2001. Shallow Morphological Analysis in Monolingual Information Retrieval for Dutch, German, and Italian. In *Evaluation of Cross-Language Information Retrieval Systems, Second Workshop of the Cross-Language Evaluation Forum*, CLEF 2001, pages 262–277, Darmstadt, Germany.

Roeland Ordelman, Arjan van Hessen, and Franciska de Jong. 2003. Compound decomposition in Dutch large vocabulary speech recognition. In *Proceedings of the European Conference on Speech Communication and Technology*, EUROSPEECH '03, pages 225–228, Geneva, Switzerland.

Matthias Richter, Uwe Quasthoff, Erla Hallsteinsdóttir, and Chris Biemann. 2006. Exploiting the Leipzig Corpora Collection. In *Proceedings of the Fifth Slovenian and First International Language Technologies Conference*, IS-LTC '06, pages 68–73, Ljubljana, Slovenia.

Roland Schäfer and Felix Bildhauer. 2013. *Web Corpus Construction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool.

Anne Schiller. 2005. German Compound Analysis with wfsc. In *Proceedings of the 5th International Workshop on Finite-State Methods and Natural Language Processing*, FSMNLP 2005, pages 239–246, Helsinki, Finnland.

Menno van Zaanen, Gerhard van Huyssteen, Suzanne Aussems, Chris Emmery, and Roald Eiselen. 2014. The Development of Dutch and Afrikaans Language Resources for Compound Boundary Analysis. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, LREC '14, pages 1056–1062, Reykjavík, Iceland.