German Compound Splitting Using the Compound Productivity of Morphemes

Sugisaki, Kyoko ; Tuggener, Don

German Compound Splitting Using the Compound Productivity of Morphemes

Kyoko Sugisaki
German Department
University of Zurich
Schönberggasse 9, 8001 Zurich
Switzerland
sugisaki@ds.uzh.ch

Don Tuggener
School of Engineering
Zurich University of Applied Sciences
Steinberggasse 13, 8400 Winterthur
Switzerland
don.tuggener@zhaw.ch

Abstract

In this work, we present a novel compound splitting method for German by capturing the compound productivity of morphemes. We use a giga web corpus to create a lexicon and decompose noun compounds by computing the probabilities of compound elements as bound and free morphemes. Furthermore, we provide a uniformed evaluation of several unsupervised approaches and morphological analysers for the task. Our method achieved a high F1 score of 0.92, which was a comparable result to state-of-the-art methods.

1 Introduction

Compounds are pervasive in German. According to Baroni (2002), 7% of the tokens and 47% of the types in a 28-million token German newswire corpus are compounds. German compounds are very productive. Two or more words can be combined to build a compound if a semantic relation is present between them. For example, a compound, such as *Fischfrau* ('fish woman'), can be produced ad-hoc. Indeed, *Fischfrau* can be interpreted as a woman who sells fish or a woman who has a face like a fish (Heringer (1984) cited in Eisenberg (2013)).

In natural language processing, German compound splitting has been investigated intensively because the task is crucial for several applications, such as machine translation (Daiber et al., 2015; Cap et al., 2014; Fritzinger and Fraser, 2010; Popović et al., 2006), information retrieval (Alfonseca et al., 2008; Monz and de Rijke, 2002), speech recognition (Larson et al., 2000), coreference resolution (Tuggener, 2016), and lexicography (Roth, 2014).

Formally, a compound is defined as consisting of at least two roots. The right-most component of a noun compound is a noun, while the left roots can be a noun, verb, adjective, or a preposition (Eisenberg, 2013). However, not every combination of a noun with nouns, verbs, adjectives and prepositions is a compound. In particular, some of the noun compounds with prepositions are often lexicalized so that we cannot decompose them semantically. For example, *Anflug* (lit. 'to-flight') means approaching flight, but Ausflug (lit. 'out-flight') is not a departing flight, but means an excursion. The departure is *Abflug* (lit. 'from-flight'). The second problem, which is also related to the first problem, is that there are many ambiguities between bound and free morphemes in German. For example, the verb *ist* ('is') can be a bound morpheme for persons (e.g., *Dadaist, Feminist, Marxist*) or the noun *Bar* can be a bound morpheme for capability and possibility (e.g., *machbar* 'executable', *trinkbar* 'potable').

In this paper, we aim to solve the bound and free morpheme ambiguity in an unsupervised approach. In German, there are noun, verb, adjective and adverb compounds. As the majority of compounds are noun compounds (62% of the compounds in (Baroni et al., 2002)), we focus on noun compounds in this work.

The paper is structured as follows. After providing an overview of previous unsupervised German compound splitting approaches (Section 2), we present our novel compound productivity based method for the task (Section 3). In our approach, we create a lexicon from a web giga corpus and disambiguate compound split points by measuring the probabilities of compound elements as bound and free morphemes. In Section 4, we test several unsupervised compound splitting methods and morphological analysers in a uniform design. We conclude our paper with an error analysis and a discussion.

2 Related Work

So far, three main methods have been used to split German compounds: 1) rule-based, that is, with finite state machines (Marek, 2006; Schiller, 2006), and morphological analysers (Schmid et al., 2004); 2) unsupervised, corpus-driven methods (Koehn and Knight, 2003; Larson et al., 2000); 3) supervised approaches (Tuggener, 2018; Alfonseca et al., 2008). These methods are also often combined (e.g., Weller-Di Marco(2017); Roth (2014); Henrich and Hinrichs (2011); Fritzinger and Freaser (2010)). Recently, semantic approaches have been proposed to analyse the semantic relationships between the constituents of compounds (Ziering et al., 2016; Daiber et al., 2015). In this section, we provide a brief overview of previous unsupervised approaches to the decomposition of German compounds. In Section 4, we provide an evaluation of these methods.

2.1 Word Frequency Based Decomposition

In this paradigm, a compound is split into two or more known words from a lexicon that are disambiguated based on the frequency of the words in a corpus.

Frequency Based Score Koehn and Knight (2003) selected the best split point S with the highest score of the averaged multiplication of word frequency of n component elements e:

$$argmax\, S = (\prod_{e_i \in S} count(e_i))^{\frac{1}{n}}$$

In this method, it is assumed that an element is likely to be a part of a compound if it occurs frequently as an independent word in a corpus.

Probability Based Score Marek (2006) and Alfonseca (2008) proposed a probability based score for compound splitting. The split score is calculated by the (sum of the negative logarithms of the) probabilities of the component elements, which is calculated by their frequency count(e) divided by the corpus size N.

$$argmin\, S = \sum_{e_i \in S} -log(\frac{count(e)}{N})$$

The assumption is that the high probability of occurrence of the component elements in a corpus has a positive effect on the prediction of compound elements.

2.2 Character N-Gram Based Decomposition

An approach to compound splitting that does not rely on a lexicon is presented in Tuggener (2016). The approach is based on character n-grams and their distributions in words. The basic idea is that character n-grams have a different probability at the beginning, middle and end of a word. These probabilities are calculated for n-grams with size 3 to N using an unlabelled corpus, where N was empirically fixed to 20. For each word in the corpus, all n-grams are collected and counted at each character position. After counting, the conditional probability of being at one of the three word positions are calculated for each observed ngram. For example, for the n-gram *"ung"*, the conditional probability to be at the end of a word is calculated:

$$P(word\ ending|\text{“}ung\text{”}) \approx \frac{count(\text{“}ung\text{”} \wedge word\ ending)}{count(\text{“}ung\text{”})}$$

Each character position in a compound is scored based on the three probabilities. For example, in the case of the split point *Tür|Schloss* 'door key', we consider the probabilities of all 3 to N n-grams ending at the split position (here, only *tür*) to emit a word ending, all 3 to N n-grams starting at the split position (*sch, schl, schlos, schloss*) to emit a word beginning or a word middle. Position i is then scored by taking the highest probability to emit a word start (here $P(word\ start|\text{“}schlo\text{”}) = 0.68$), the highest probability to emit a word ending ($P(word\ end|\text{“}tür\text{”}) = 0.45$) and the lowest probability for being in the middle of a word $P(word,\ middle|\text{“}schloss\text{”}) = 0.03$):

$$\begin{aligned} Score(i) = {} & max(P(word\ start)) + \\ & max(P(word\ end)) - min(P(word\ middle)) \end{aligned}$$

After scoring all positions, the position with the highest score is selected as the split position. In addition, a minimum threshold can be set so that simplexes are discarded. For multiple compound splitting, compounds are recursively split into binary components.

3 Compound Productivity Based Disambiguation

In this section, we propose our disambiguation method based on the compound productivity of morphemes.

3.1 Model

In our approach, we address the following two main challenges in compound decomposition: First, free morphemes can be bound morphemes (affixes) (e.g., *ist, er, schaft, bar,* and *haft*) or affix-like (i.e., prepositions/verb-particles; e.g., *auf, ab,* and *an*). Some ambiguous morphemes such as *ist* ('is') and *er* ('he') are highly problematic in frequency-based decomposition, as they are some of the most frequent words in a corpus. Second, the frequency of free morphemes does not necessarily indicate the semantic compatibility of compound elements. In the decomposition of *Hundesteuer*, there are two candidate splits: *Hunde|steuer* ('dog tax') and *Hundes|teuer* ('dog pricy'). *Teuer* is much more frequent than *Steuer*. However, *Hunde|steuer* is the correct compound decomposition.

To address these two problems, we propose using the binding strength of morphemes (i.e., the compound elements). Specifically, we compute the ratio of dependence and independence. How much more likely is it that a morpheme is independent rather than dependent? Thus, for each morpheme, we capture its probability of occurrence as a bound morpheme (dependent) and as a free morpheme (independent). Then we split a compound using the following formula to score the potential decomposition:

$$Score = \frac{P(e_1...e_k, E)}{P(e_1)...P(e_k)P(E)}$$

where a word consists of k compound elements e and E signifies the end of a word. For example, we decompose the word *Hundesteuer* as follows.

$$\frac{P(\text{“}hunde\text{”}, \text{“}steuer\text{”}, E)}{P(\text{“}hunde\text{”})P(\text{“}steuer\text{”})P(E)}$$

The joint probability is decomposed according to the chain rule and the Markov assumption:

$$\frac{P(\text{“}steuer\text{”}|\text{“}hunde\text{”})P(E|\text{“}hundesteuer\text{”})}{P(\text{“}hunde\text{”})P(\text{“}steuer\text{”})P(E)}$$

$P(E)$ remains constant over all tokens. Therefore, it is removed:

$$\frac{P(\text{“}steuer\text{”}|\text{“}hunde\text{”})P(E|\text{“}hundesteuer\text{”})}{P(\text{“}hunde\text{”})P(\text{“}steuer\text{”})}$$

Because we want to capture the binding strength of morphemes, we decompose them into a 'back-off model', in which a star means any strings that are followed by the conditional term:

$$\frac{P(*|\text{“}hunde\text{”})P(E|\text{“}steuer\text{”})}{P(\text{“}hunde\text{”})P(\text{“}steuer\text{”})}$$

In the corpus, we calculate by c(ounting):

$$\frac{\frac{c(\text{“}hunde-\text{”})}{c(\text{“}hunde\text{”})}\frac{c(\text{“}-steuer\text{”},E)}{c(\text{“}steuer\text{”})}}{\frac{c(\text{“}hunde\text{”})}{N}\frac{c(\text{“}steuer\text{”})}{N}}$$

The score is calculated for each compound decomposition candidate. In our example *Hundesteuer*, we compute a score for two candidates, *Hunde|steuer* and *Hundes|teuer*. The underlying idea is that the score captures the compound productivity of morphemes. Here, we expect that there are much more compounds with *-steuer* than with *-teuer*. We choose the candidate with the lowest score, which is the best one with an acceptable balance of the frequency of bound morphemes as compound elements and free morphemes as words.

3.2 Implementation

Compound productivity based decomposition consists of two steps: 1) creating a set of candidate split points using string matching with words extracted from a monolingual corpus; 2) disambiguating (or ranking) them according to the model (Sec. 3.1).

Step 1 We split words into multiple split points. To generate candidate split points, we follow the method in Koehn and Knight (2003). We extract the words from a corpus and create a lexicon. We then carry out recursive binary string matching of the right component with the lexicon entries. The minimal number of characters to be segmented is two. In this step, we are recall-oriented, and we try to split all possible components because the second processing component disambiguates and boosts precision.

Step 2 We disambiguate the candidate sets created by the splitter in Step 1 by using the compound productivity based model. To count frequency, we use a training corpus. Then we store the lexicon with the frequency in a MySQL database. We carry out the string matching of morphemes on the fly by using the database.

For frequency count, we handle compound formation suffixes by delating and adding strings in the middle of words. In particular, we operate a delation operation of *=s=* and an addition operation of *=e=/=en=/=eln=/=ern=*. In compounding, some nouns, especially feminine nouns are formed by adding a linking morpheme *=s=* (e.g., *Verständigung=s=problem* 'communication problem'), and some nouns are truncated at the end (e.g., *Schulhaus* 'school building' and

Südwest 'south west'). Verb-noun compounds consist of verb stems and nouns, which requires the concatenation of *=en=/=eln=/=ern=* after verb stems (e.g., *schauen/wandern/sammeln* instead of *schau/wander/sammel*). To gain a reliable frequency count of compound words, each word undergoes these string manipulation operations, and the highest frequency count is selected.

4 Evaluation

We evaluate our compound productivity based method against three approaches to compound splitting: 1) a word frequency based approach with two disambiguation methods (Sec. 2.1); 2) a character n-gram based approach (Sec. 2.2); 3) two morphological analysers, Gertwol (Haapalainen and Majorin, 1994) and SMOR (Schmid et al., 2004). To test these methods in a uniform design, we define a compound decomposition task with multiple splits, i.e., where each compound contains one or more split points.

Data As test data, we use the list of split noun compounds provided by Germanet (Henrich and Hinrichs, 2011). Germanet contains 74,983 noun compounds. In the Germanet list, the compounds are split into two lemmas. We extend the binary segmentation into multiple segmentation by tracing binary branching points upwards. We then convert the lemma-based binary and multiple splits into word-based ones. For example, *Heimtierhaltung* (lit. 'domestic animal keeping') is separated into *Heim* 'domestic', *Tier* 'animal' and *Haltung* 'keeping'. Thus, our test set contains 64,627 noun compounds with binary split and 10,356 with multiple split.

Measurement To measure the accuracy of the compound split methods, we extend the word-level precision and recall score proposed by Koehn and Knight (2003) to the split point level. We evaluate the split point-level precision and recall to measure how many returned split points are those that should be split (precision) and how many of the correct split points are actually returned (recall). To harmonise these two scores, we use the F1 score. In addition, we use a simple measure to determine the number of words that are correctly split (i.e., word-level accuracy) and how many words are not split at all (i.e., word-level miss rate). It is important to note that our test data does not contain words that are not compounds.

Unsupervised Approaches For the unsupervised approaches, we use two types of German corpora: the 0.88-billion-word web corpus SdeWaC (Faaß and Eckart, 2013) and the German newspaper corpus TüBa D/Z (Telljohann et al., 2015). We create four training sets to build a model or a lexicon with word frequency counts for each method: (1) all data from SdeWaC(SdeWaCALL, 754,462,344 tokens, 8,357,974 token types); (2) only lexical words from SdeWac (SdeWaCFunc 412,617,920 tokens, 5,392,723 token types); (3) only nouns from SdeWaC (SdeWaCNoun, 211,384,683 tokens, 6,840,728 token types); and (4) all data from TüBa (TüBaALL, 1,524,822 tokens, 143,990 token types). For SdeWaCNoun, we extract capitalised words excluding first words in sentences. For SdeWaCFunc, we extract lexical words by discarding function words (i.e., high frequency words in the corpus (i.e., $<$ frequency class 5 in Perkuhn et. al (2012)), and abbreviations and noises (i.e., low frequency words ($>$ frequency class 19 in ibd.)), vowelless words, camel cases, and foreign diacritics. For the test, we set the minimal score of the numerator to 0.0001 in the compound productivity based method and that of the score and to 0.5 in the character n-gram based method.

Morphological Tools For comparison, we test Gertwol and SMOR. The evaluation of Gertwol is straightforward because it returns compound boundaries. We convert the lemma-based representation to word-based representation by using a string matching method. Regarding SMOR, we convert the lemma-based and fine-grained representations returned by SMOR by matching the word form and the lemma form. We separately handle verbal nominalisation forms such as *-ung* and *-er*. We convert all part-of-speech and compound tags into compound boundaries if they were not marked with a suffix.[1]

Results Table 1 shows the results. The unsupervised approaches and morphological tools achieved a similar accuracy with a F1 score of split points between 0.91 and 0.93. As expected, the morphological tools failed to split compounds more often than the data-driven unsupervised approaches (cf. w-miss), but the lexicon-based approaches suffered if the size of a training corpus was small

[1] Part-of-speech and compound tags are NN, ADJ, ADV, PREF, NPROP, CARD, ORD, PRED, V, ANS, ATTR, COMPAR, DEF, INDEF, INDEF, NEG, PERS, PRFL, PRO, REC, REFL, SUB, SUBST, KSF, TRUNC, F (capitalisation normalised) in SMOR. `http://www.cis.uni-muenchen.de/~schmid/tools/SMOR/dspin/index.html`

	train	spl-F1	spl-prec	spl-rec	w-acc	w-miss
wFreq	sdeWaCALL	.68	.53	.94	.31	**.00**
	sdeWaCFunc	.82	.71	.96	.58	**.00**
	sdeWaCNoun	.75	.62	**.97**	.44	**.00**
	TüBaALL	.87	.85	.89	.76	.03
wProb	sdeWaCALL	.92	.98	.86	.85	**.00**
	sdeWaCFunc	.91	.98	.85	.84	**.00**
	sdeWaCNoun	.92	**.99**	.86	.85	**.00**
	TüBaALL	.86	.93	.79	.78	.03
Prod	sdeWaCALL	.92	.97	.87	.84	**.00**
	sdeWaCNoun	.92	.98	.87	.85	**.00**
	sdeWaCFunc	.91	.97	.85	.84	**.00**
	TüBaALL	.85	.92	.79	.77	.03
charN	sdeWaCALL	.89	.91	.86	.78	**.00**
	sdeWaCFunc	.88	.94	.84	.78	**.00**
	sdeWaCNN	**.93**	.95	.90	.85	**.00**
	TüBaALL	.69	.96	.55	.52	**.00**
SMOR	_	.91	.95	.87	.85	.04
Gertwol	_	**.93**	.93	.93	**.87**	.05

Table 1: Evaluation: Multiple Compound Decomposition: Lexicon-based approaches: wFreq = frequency based score (Section 2.1), wProb = probability based score (Section 2.1), Prod = our compound productivity based score (Section 3.1); CharN=charater-ngram based approach (Section 2.2)

(TüBaALL). Comparing the training sets of the unsupervised approaches, the performance tends to increase if the size of the corpus increases. However, the benefit of a very large corpus was rather limited. The unsupervised methods achieved a comparative accuracy with much smaller but more linguistically motivated training sets (i.e., SdeWaCNoun and SdeWaCFunc) than the whole corpus.

Error Analysis We analysed the compound split points returned by the best model for each approach. The compounds that are successfully decomposed by all the systems are those that consist of at least one frequent word (e.g., *Sommerschule* 'summer school', *Botenstoff* 'semiochemical', *Exilverlag* 'exile publishing house', *Biosensor* 'biosensor', *Augustinerkloster* 'Augustinian monastery'). A hard problem shared by all the systems was to identify split points between two infrequent components such as (1) technical terms (e.g., *Doldensurre* 'jagged chickweed', *Röhrenholothurie* 'cotton spinner'), (2) confixes (e.g., *Bioei* 'organic egg', *Erotomanie* 'erotomania'), and (3) foreign words (e.g. *Backstage, Newsticker*). Some of technical terms were difficult to split correctly because of lexicalized compounds in compounds (e.g., *Venus|fliegentau*, *Eisen|nickelkies*). In addition, all systems have a difficulty in preposition-noun compounds (*Umweg* 'detour', *Antransport* 'carry-in').

Our system performed comparably well for compounds containing foreign words (e.g., *Millionen|seller* or *Pneumo|thorax*). However, the major proprietary error was due to possible linking morpheme *-s* placed in the middle of words. Of which, there are two error types. First, all components are common nouns (e.g., *Barocks|aal* instead of *Barocks|aal*). Secondly, one of the components is some proper name (e.g., *Kletters|eil* instead of *Kletter|seil*) or some abbreviation (e.g., *Fensters|ims* instead of *Fenster|sims*). We consider that this major problem could be resolved by using information about the discourse context to discard implausible compound elements.

5 Conclusion

In this paper, we demonstrated that our compound productivity based disambiguation achieved a comparative result to other lexicon-based and character n-gram based unsupervised approaches and morphological tools. We observed that the benefit of large training data held to some extent, but we demonstrated that linguistically motivated training data provide more benefits than the size of a corpus. In future work, we will test the methods on non-compounds and on binary decomposition.

Acknowledgments

This work has been funded under SNSF grant 160238 for the first author. We thank Nicolas Diener for various discussion on the task and two anonymous reviewers for their helpful comments.

References

Enrique Alfonseca, Slaven Bilac, and Stefan Pharies. 2008. German decompounding in a difficult corpus. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing: 9th International Conference (CICLing)*, pages 128–139, Berlin, Heidelberg. Springer.

Marco Baroni, Johannes Matiasek, and Harald Trost. 2002. Predicting the components of German nominal compounds. In *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI'02)*, pages 470–474.

Fabienne Cap, Alexander Fraser, Marion Weller, and Aoife Cahill. 2014. How to produce unseen teddy bears: Improved morphological processing of compounds in SMT. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 579–587.

Joachim Daiber, Lautaro Quiroz, Roger Wechsler, and Stella Frank. 2015. Splitting compounds by semantic analogy. In *Proceedings of the 1st Deep Machine Translation Workshop*.

Peter Eisenberg. 2013. *Grundriss der deutschen Grammatik*. Metzler, Stuttgart, 4 edition.

Gertrud Faaß and Kerstin Eckart. 2013. SdeWaC – a corpus of parsable sentences from the web. In Iryna Gurevych, Chris Biemann, and Torsten Zesch, editors, *Language Processing and Knowledge in the Web*, pages 61–68, Berlin, Heidelberg. Springer Berlin Heidelberg.

Fabienne Fritzinger and Alexander Fraser. 2010. How to avoid burning ducks: Combining linguistic analysis and corpus statistics for German compound processing. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR (WMT '10)*, pages 224–234.

Mariikka Haapalainen and Ari Majorin. 1994. GERTWOL: ein System zur automatischen Wortformerkennung deutscher Wörter. Technical report, Lingsoft, Inc.

Verena Henrich and Erhard Hinrichs. 2011. Determining immediate constituents of compounds in GermaNet. In *Proceedings of Recent Advances in Natu- ral Language Processing (RANLP)*, pages 420—426.

Hans Jürgen Heringer. 1984. Wortbildung: Sinn und chaos. *Deutsche Sprache*, 12:1–13.

Philipp Koehn and Kevin Knight. 2003. Empirical methods for compound splitting. In *Proceedings of the tenth Conference on European Chapter of the Association for Computational Linguistics (EACL '03)*, pages 187–193.

Martha Larson, Daniel Willett, Joachim Köhler, and Gerhard Rigoll. 2000. Compound splitting and lexical unit recombination for improved performance of a speech recognition system for German parliamentary speeches. In *Sixth International Conference on Spoken Language Processing (ICSLP)*, pages 945–948.

Torsten Marek. 2006. Analysis of German compounds using weighted finite state transducers. Bachelor of Arts Thesis.

Christof Monz and Maarten de Rijke. 2002. Shallow morphological analysis in monolingual information retrieval for Dutch, German, and Italian. In Carol Peters, Martin Braschler, Julio Gonzalo, and Michael Kluck, editors, *Evaluation of Cross-Language Information Retrieval Systems*, pages 262–277, Berlin, Heidelberg. Springer Berlin Heidelberg.

Rainer Perkuhn, Holger Keibel, and Marc Kupietz. 2012. *Korpuslinguistik*. UTB für Wissenschaft : Uni-Taschenbücher. Fink, Paderborn.

Maja Popović, Daniel Stein, and Hermann Ney. 2006. Statistical machine translation of german compound words. In Tapio Salakoski, Filip Ginter, Sampo Pyysalo, and Tapio Pahikkala, editors, *Advances in Natural Language Processing*, pages 616–624, Berlin, Heidelberg. Springer Berlin Heidelberg.

Tobias Roth. 2014. *Wortverbindungen und Verbindungen von Wörtern : lexikografische und distributionelle Aspekte kombinatorischer Begriffsbildung zwischen Syntax und Morphologie*. Basler Studien zur deutschen Sprache und Literatur. Francke, Tübingen.

Anne Schiller. 2006. German compound analysis with wfsc. In Anssi Yli-Jyrä, Lauri Karttunen, and Juhani Karhumäki, editors, *Finite-State Methods and Natural Language Processing*, pages 239–246, Berlin, Heidelberg. Springer Berlin Heidelberg.

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 (LREC 2004)*, pages 1263–1266.

Heike Telljohann, Erhard W. Hinrichs, Kübler Sandra, Zinsmeister Heike, and Beck Kathrin. 2015. Stylebook for the Tübingen treebank of written German (TüBa-D/Z). Technical report, Universität Tübingen.

Don Tuggener. 2016. *Incremental Coreference Resolution for German*. Ph.D. thesis, University of Zurich.

Don Tuggener. 2018. Evaluating neural sequence models for splitting (Swiss) German compounds. In *Proceedings of the 3rd Swiss Text Analytics Conference - SwissText 2018*, CEUR Workshop Proceedings.

Marion Weller-Di Marco. 2017. Simple compound splitting for German. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 161–166.

Patrick Ziering, Stefan Müller, and Lonneke van der Plas. 2016. Top a splitter: Using distributional semantics for improving compound splitting. In *Proceedings of the 12th Workshop on Multiword Expressions*, pages 50–55.