Using Semantics for Granularities of Tokenization

Martin Riedl
University of Stuttgart
Institut für maschinelle
Sprachverarbeitung
martin.riedl@ims.uni-stuttgart.de

Chris Biemann
University of Hamburg
Language Technology Group
biemann@informatik.uni-hamburg.de

Depending on downstream applications, it is advisable to extend the notion of tokenization from low-level character-based token boundary detection to identification of meaningful and useful language units. This entails both identifying units composed of several single words that form a multiword expression (MWE), as well as splitting single-word compounds into their meaningful parts. In this article, we introduce unsupervised and knowledge-free methods for these two tasks. The main novelty of our research is based on the fact that methods are primarily based on distributional similarity, of which we use two flavors: a sparse count-based and a dense neural-based distributional semantic model. First, we introduce `DRUID`*, which is a method for detecting MWEs. The evaluation on MWE-annotated data sets in two languages and newly extracted evaluation data sets for 32 languages shows that* `DRUID` *compares favorably over previous methods not utilizing distributional information. Second, we present* `SECOS`*, an algorithm for decompounding close compounds. In an evaluation of four dedicated decompounding data sets across four languages and on data sets extracted from Wiktionary for 14 languages, we demonstrate the superiority of our approach over unsupervised baselines, sometimes even matching the performance of previous language-specific and supervised methods. In a final experiment, we show how both decompounding and MWE information can be used in information retrieval. Here, we obtain the best results when combining word information with MWEs and the compound parts in a bag-of-words retrieval set-up. Overall, our methodology paves the way to automatic detection of lexical units beyond standard tokenization techniques without language-specific preprocessing steps such as POS tagging.*

Submission received: 5 July 2017; revised version received: 10 February 2018; accepted for publication: 15 May 2018.

doi:10.1162/COLI_a_00325

1. Introduction

If we take Ron Kaplan's motivation for tokenization seriously that the "stream of characters in a natural language text must be broken up into distinct meaningful units" (Kaplan 2005) to enable natural language processing beyond the character level, then tokenization is more than the low-level preprocessing task of treating interpunctuation, hyphenation, and enclitics. Rather, tokenization also should aspire to produce *meaningful* units, or, as Webster and Kit (1992) define it, tokens should be linguistically significant and methodologically useful. In practice, however, tokenizers are rather not concerned with meaning or significance—placed right at the beginning of any NLP pipeline and usually implemented in a rule-based fashion, they are merely workhorses to enable higher levels of processing, which includes a reasonable split of the input into word tokens and some normalization to cater to the sensitivity of subsequent processing components. Although it is clear that the methodological utility of a specific tokenization depends on the overall task, it seems much more practical to fix the tokenization in the beginning of the text ingestion process and handle task-specific adjustments later. The work presented in this article operationalizes lexical semantics in order to identify meaningful units. Assuming that low-level processing has already been performed, we devise a method that can identify multiword units, namely, word *n*-grams that have a non-compositional meaning, as well as a method that can split close compound words into their parts. Both methods are primarily based on distributional semantics (Harris 1951): By operationalizing language unit similarity in various ways, we are able to inform the tokenization process with semantic information, enabling us to yield meaningful units, which are shown to be linguistically valid and methodologically useful in a series of suitable evaluations. Both methods do not make use of language-specific processing, thus could be applied directly after low-level tokenization without assuming the existence of, for example, a part of speech tagger.

Depending on the task, the low-level "standard" tokenization can be too fine-grained, as from a semiotic perspective multiword expressions (MWEs) refer to a single concept. On the flipside, tokenization can be too coarse-grained, as close compound words are detected as single words, whereas they are formed by the concatenation of at least two stems and can be considered as MWEs without white spaces. In this article, we will describe two different approaches to represent (nominal) concepts in a similar fashion. This results in an extended tokenization, similar to the work by Hassler and Fliedl (2006). However, they extend their tokenization solely by bracketing phrases and MWEs and do not split text in more fine-grained units. Trim (2013) differentiates between low-level and high-level tokenization. Whereas high-level tokenization concentrates on the identification of MWEs and phrases, low-level tokenization mostly splits words that are connected by apostrophes or hyphens. Our notion of coarse-grained tokenization is similar to high-level tokenization. However, the fine-grained tokenization goes one step beyond low-level tokenization, as we split close compound words.

First, we describe a method for detecting MWEs. For defining MWEs, we follow the definition by Sag et al. (2001, page 2) that claims that MWEs are "idiosyncratic interpretations that cross word boundaries (or spaces)." Furthermore, MWEs are made up of compounds, phrases, or sentences. The detection of named entities (e.g., names, locations, companies, or concepts) is often considered as a task of its own, which aims at identifying a subset of MWEs and is relevant for information extraction (e.g., relation extraction or event extraction), but also for information retrieval or automatic speech recognition systems.

As a second contribution, we present a method for splitting close compounds. Examples for such close compounds include, for example, dishcloth (*English*), pancake (*English*), Hefeweizen (*German for wheat beer*), bijenzwerm (*Dutch for swarm of bees*) or hiilikuitu (*Finnish for carbon fibre*). Similar to MWEs, compounds are created by combining existing words, although in close compounds the stems are not separated by white space. Detecting the single stems, called **decompounding**, showed impact in several natural language processing (NLP) applications like automatic speech recognitions (Adda-Decker and Adda 2000), machine translation (Koehn and Knight 2003), or information retrieval (IR) (Monz and de Rijke 2001) and is perceived as a crucial component for the processing of languages that are productive with respect to this phenomenon.

For both the detection of MWEs and the decompounding of words, most existing approaches rely either on supervised methods or use language-dependent part-of-speech (POS) information. In this work, we present two knowledge-free and unsupervised (and therefore language-independent) methods that rely on information gained by distributional semantic models that are computed using large unannotated corpora, namely, word2vec (Mikolov et al. 2013) and JoBimText (Biemann and Riedl 2013). First, we describe these methods and highlight how their information can help for both tokenization tasks. Then, we present results for the identification of MWEs and afterwards show the performance of the method for decompounding. For both tasks, we first show the performance using manually annotated gold data before we present evaluations for multiple languages using automatically extracted data sets from Wikipedia and Wiktionary. Lastly, we demonstrate how both flavors of such an extended tokenization can be used in an IR setting. The article is partly based on previous work (Riedl and Biemann 2015, 2016) that has been substantially extended by adding experiments for several languages and showing the advantage of combining the methods in an information retrieval evaluation.

The article is organized as follows. Section 2 describes the distributional semantic models that are used to compute similarities between lexical units, which are the main source of information for both fine- and coarse-grained tokenization. Then, we describe how multiword expressions can be detected and evaluate our methodology. In Section 4, we describe the workings and the evaluation for compound splitting. How to use both methods for information retrieval is shown in Section 5. In Section 6, we present the related work. Afterwards, we highlight the main findings in the conclusion in Section 7 and give an overview of future work in Section 8.

2. Using Distributional Semantics for Fine- and Coarse-Grained Tokenization

Both methods described in this article have in common that they rely on distributional semantics, which is based on the distributional hypothesis that was conceived by Harris (1951). This hypothesis states that words that occur in a similar context tend to have similar meaning. Many methods have implemented that assumption in order to compute word similarities using various contexts (e.g., neighboring words, words with syntactic dependencies) (Hindle 1990; Grefenstette 1994; Lin 1998). Usually, words are not only similar to synonyms but also to hypernyms, antonyms, or related terms. For the task of splitting words, the similarity to hypernyms is interesting, as compounds are often similar to more general terms, which are stems of the compound. For example, the word *Hefeweizenbier* [yeast wheat beer] is most similar to the term *Bier* [beer] or *Weizenbier* [wheat beer], which are words that are nested in the more specific word.

Such information is beneficial when it comes to the task of splitting compounds, as we shall see subsequently. When computing similarities not only for words but considering word *n*-grams, we observe that concepts that are composed of several word units are often similar to single-word terms. For example, the word *hot dog* is most similar to food-related terms like *hamburger* or *sandwich*. As shown in the remainder of this article, the information of distributional semantics is beneficial for the tasks of identifying of MWEs but also for the task of compound splitting.

In this work, we compute semantic similarities using the dense vector-based CBOW model from word2vec (Mikolov et al. 2013) and a symbolic graph-based approach called JoBimText (Biemann and Riedl 2013). In order to use both models within the word splitting and the word merging task, we transform them to a so-called distributional thesaurus (DT) as defined by Lin (1997). A DT can be considered as a dictionary where for each word the top *n* most similar words are listed, ordered by their similarity score.

The CBOW model is learned during the task of predicting a word by its context words. For this, the input layer is defined by the contexts of a word. As output layer we use the center word. The prediction is performed using a single hidden layer that represents the semantic model with the specified dimensions. For the computation of word2vec models, we use 500 dimensions, 5 negative samples, and a word window of 5. Because the implementation by Mikolov et al. (2013)[1] does not support the computation of similarities between all *n*-grams within a corpus, we use the word2vecf implementation by Levy and Goldberg (2014).[2] This implementation allows specifying terms and contexts directly and features the functionality to retrieve the most relevant contexts for a word. In order to extract a DT from models computed with word2vec and word2vecf, we compute the cosine similarity between all terms and extract, for each term, the 200 most similar terms.

As opposed to the mainstream of using dense vector representations, the approach by Biemann and Riedl (2013), called JoBimText, uses a sparse count-based context representation that nevertheless scales to arbitrary amounts of data (Riedl and Biemann 2013). Furthermore, this approach has achieved competitive results to dense vector space models like CBOW and SKIP-gram (Mikolov et al. 2013) in word similarity evaluations (Riedl 2016; Riedl and Biemann 2017). To keep the preprocessing language independent, we keep only words in a context window for both approaches, as opposed to, for example, dependency-parsing-based contexts. For the task of MWE identification we do not only represent single words but also *n*-grams using single-word contexts. For the task of decompounding, only unigrams are considered.

Based on the frequencies of words/*n*-grams and contexts, we calculate the lexicographer's mutual information (LMI) significance score (Evert 2005) between terms and features and remove all context features that co-occur with more than 1,000 terms, as these features tend to be too general. In the next step we reduce the number of context features per term by keeping for each term only 1,000 context features with the highest LMI score. The similarity score is defined as the number of shared features of two terms. Such an overlap-based similarity measure is proportional to the Jaccard similarity measure, although we do not conduct any normalization. After computing the feature overlap between all pairs of terms, we retain the 200 most similar terms for each word *n*-gram. In line with Lin (1997) we refer to such a resource as DT.

1 https://code.google.com/archive/p/word2vec/.
2 https://bitbucket.org/yoavgo/word2vecf.

3. Merging Words: Multiword Identification

The detection of multiword units is one of the extensions needed for coarse-grained tokenization. As summarized concisely by Blanc, Constant, and Watrin (2007, page 1), "language is full of multiword units." By inspecting dictionaries, we highlight the importance of MWEs. For example, in WordNet, 41.41% of all words are MWEs, as shown in Table 1. Whereas more than 50% of all nouns are MWEs, only about 26% of all verbs are MWEs. As the majority of all MWEs found in WordNet are nouns (93.73%), developing the method we focus first on the detection of terms belonging to this word class in Section 3.6 and show the performance on all word classes in subsequent sections.

Although it seems intuitive to treat certain sequences of tokens as (single) terms, there is still considerable controversy about the definition of what exactly constitutes a MWE. Sag et al. (2001) pinpoint the need for an appropriate definition of MWEs. For this, they classify a range of syntactic formations that could form MWEs and define MWEs as being non-compositional with respect to the meaning of their parts. Although the exact requirements of MWEs is bound to specific tasks (such as parsing, keyword extraction, etc.), we operationalize the notion of non-compositionality by using distributional semantics and introduce a measure that works well for a range of task-based MWE definitions.

Reviewing previously introduced MWE ranking approaches (cf. Section 6.1), most methods use the following mechanisms to determine multiwordness: POS tags, word/multiword frequency, and significance of co-occurrence of the parts. In contrast, our method uses an additional mechanism, which performs a ranking based on distributional semantics.

Distributional semantics has already been used for MWE identification, but mainly to discriminate between compositional and non-compositional MWEs (Schone and Jurafsky 2001; Hermann and Blunsom 2014; Salehi, Cook, and Baldwin 2014). Here we introduce a concept to describe the multiwordness of a term by its *uniqueness*. This score measures the likeliness that a term in context can be replaced with a single word. This measure is motivated by the semiotic consideration that due to parsimony, concepts are often expressed as single words. Furthermore, we deploy a context-aware punishment term, called **incompleteness**, which degrades the score of candidates that seem incomplete regarding their contexts. For example, the term *red blood* can be called *incomplete* as the following word is most likely the word *cell*. Both concepts are combined into a single score we call `DRUID` (DistRibutional Uniqueness and Incompleteness Degree), which is calculated based on a DT. In the following, we show the performance of this method for French and English and examine the effect of corpus size on MWE extraction. This section extends work presented in Riedl and Biemann (2015). In addition, we

Table 1
Amounts and percentages of MWEs contained in WordNet 3.1 for different POS.

	noun	**adjective**	**adverb**	**verb**	**all**
MWE (count)	60,337	505	695	2,838	64,375
MWE (percentage)	51.51	2.35	15.53	24.59	41.41
all words	117,953	21,499	4,475	11,540	155,467

demonstrate the language independence of the method by evaluating it on 32 languages and give a more detailed data analysis.

We want to emphasize that our method works in an unsupervised fashion and is not restricted to certain POS classes. However, most of the competitive methods require POS filtering as a pre-processing step in order to do their statistics. Hence, these methods are mostly evaluated based on noun compounds. Because of comparison reasons, the first evaluation that uses POS filtering (see Section 3.6) is restricted to noun compounds. However, the remaining experiments in Sections 3.7 and 3.8 are not restricted to any particular POS.

First, we describe the new method and show its performance on different data sets; we briefly describe the baseline and previous approaches in the next section.

3.1 Baselines and Previous Approaches

In the first setting, we evaluate our method by comparing the MWE rankings to multiword lists that have been annotated in corpora. In order to show the performance of the method, we introduce an upper bound and two baseline methods and give a brief description of the competitors. Most of these methods rely on lists of pre-filtered MWE candidate terms T. Usually these are extracted by patterns defined on POS sequences.

3.1.1 Upper Bound. As an upper bound, we consider a perfect ranking, where we rank all positive candidates before all negative ones. Within the data set, we only have binary labels for true and false MWEs. Thus, any ordering of the MWEs within the block of MWEs labeled as true, respectively, false, does not change the upper bound.

3.1.2 Lower Baseline and Frequency Baseline. The ratio between true candidates and all candidates serves as a lower baseline, which is also called **baseline precision** (Evert 2008). The second baseline is the frequency baseline, which ranks candidate terms $t \in T$ according to their frequency $freq(t)$. Here, we hypothesize that words with high frequency are multiword expressions.

3.1.3 C-value/NC-value. Frantzi, Ananiadou, and Tsujii (1998) developed the commonly used C-value (see Equation (1)). This value is composed of two factors. As first factor, they use the logarithm of the term length in words in order to favor longer MWEs. The second factor is the frequency of the term reduced by the average frequency of all candidate terms T, which nest the term t (i.e., t is a substring of the terms we denote as T_t).

$$cv(t) = \log_2(|t|) \cdot \left(freq(t) - \frac{1}{|T_t|} \sum_{b \in T_t} freq(b) \right) \tag{1}$$

An extension of the C-value was proposed by Frantzi, Ananiadou, and Tsujii (1998) and is called the NC-value. It takes advantage of context words C_t, which are neighboring words of t, by assigning weights to them. As context words only *nouns*, *adjectives*, and *verbs* are considered.[3] Context words are weighted with Equation (2), where k denotes

3 Frantzi, Ananiadou, and Tsujii (1998) do not specify the context window size.

the number of times the context word $c \in C_t$ occurs with any of the candidate terms. This number is normalized by the number of candidate terms.

$$w(c) = \frac{k}{|T|} \tag{2}$$

The NC-value is a weighted sum of the C-value and the product of the term t occurring with each context c, which form the term t_c.

$$nc(t) = 0.8 \cdot cv(t) + 0.2 \sum_{c \in C_t} freq(t_c) w(c) \tag{3}$$

3.1.4 t-test. The t-test (see, e.g., Manning and Schütze 1999, page 163) is a statistical test for the significance of co-occurrence of two words. It relies on the probabilities of the term and its single words. The probability of a word $p(w)$ is defined as the frequency of the term divided by the total number of terms of the same length. The t-test statistic is computed using Equation (4) with $freq(.)$ being the total frequency of all unigrams.

$$t(w_1 \dots w_n) \approx \frac{p(w_1 \dots w_n) - \prod_{i=1}^{n} p(w_i)}{\sqrt{p(w_1 \dots w_n)/freq(.)}} \tag{4}$$

We then use this score to rank the candidate terms.

3.1.5 Marginal Frequency-Based Geometric Mean (FGM) Score. Nakagawa and Mori (2002, 2003) presented another method that is inspired by the C/NC-value and outperformed a modified C-value measure.[4] It is composed of two scoring mechanisms for the candidate term t, as shown in Equation (5).

$$FGM(t) = GM(t) \cdot MF(t) \tag{5}$$

The first term in the equation is the geometric mean $GM(.)$ of the number of distinct direct left $l(.)$ and right $r(.)$ neighboring words for each single word t_i within t.

$$GM(t) = \left(\sum_{t_i \in t} (|l(t_i)| + 1)(|r(t_i)| + 1) \right)^{\frac{1}{2|t|}} \tag{6}$$

These neighboring words are extracted directly from the corpus; the method relies on neither candidate lists nor POS tags. In contrast, the marginal frequency $MF(t)$ relies on the candidate list and the underlying corpus. This frequency counts how often the candidate term occurs within the corpus and is not a subset of a candidate. Korkontzelos (2010) showed that although scoring according to Equation (5) leads to comparatively good results, it is consistently outperformed by the performance of $MF(t)$.

4 They adjust the logarithmic length in order to be able to use the C-value to detect single-word terms.

3.2 DistRibutional Uniqueness and Incompleteness Degree (DRUID)

Here, we describe the DRUID method for ranking terms regarding their multiwordness, which consists of two mechanisms relying on semantic word similarities: A score for the *uniqueness* of a term and a score that punishes its *incompleteness*.[5] The importance and influence of the results for the combination of both mechanisms is demonstrated in Section 3.9. The DT is computed as described in Section 2, using n-grams ($n = 1, 2, 3, 4$). When using JoBimText to compute such a DT, we use the left and right neighboring words as context. In order to compute the DRUID score using the CBOW model, we compute dense vector representations using word2vecf (Levy and Goldberg 2014) and convert it to a DT by extracting the 200 most similar words for each n-gram. An example using JoBimText for the most similar n-grams to the terms *red blood cell* and *red blood* including their feature overlap is shown in Table 2.

3.2.1 Uniqueness Computation. The first mechanism of our MWE ranking method is based on the following hypothesis: n-grams that are MWEs could be substituted by single words, thus they have many single words among their most similar terms. When a semantically non-compositional word combination is added to the vocabulary, it expresses a concept that is necessarily similar to other concepts. Hence, if a candidate multiword is similar to many single word terms, this indicates multiwordness.

To compute the *uniqueness* score (uq) of an n-gram t, we first extract the n-grams it is similar to using the DT as described in Section 2. The function $similarities(t)$ returns the 200 most similar n-grams to the given n-gram t. We then compute the ratio between unigrams and all similar n-grams considered using the formula, where the function $unigram(.)$ tests whether a word is a unigram:

$$uq(t) = \frac{|\{\forall w \in similarities(t) \mid unigram(w)\}|}{|similarities(t)|}. \quad (7)$$

We illustrate the computation of our measure based on two example terms: the MWE *red blood cell* and the non-MWE *red blood*. When considering only the ten most similar entries for both n-grams as illustrated in Table 2, we observe a uniqueness score of $7/10 = 0.7$ for both n-grams. If considering the top 200 similar n-grams, which are also used in our experiments, we obtain 135 unigrams for the candidate *red blood cell* and 100 unigrams for the n-gram *red blood*. We use these counts for exemplifying the workings of the method in the remainder.

3.2.2 Incompleteness Computation. In order to avoid ranking nested terms at high positions, we introduce a measure that punishes such "incomplete terms". This mechanism is called *incompleteness (ic)* and, similarly to the C/NC-value method (see Section 3.1.3), consists of a context weighting function that punishes incomplete terms. We show the pseudocode for the computation in Algorithm 1. First, we use the function $context(t)$ to extract the 1,000 most significant context features. This function returns a list of tuples of left and right contexts.

5 The DRUID implementation is available open source and pre-computed models can be found here: http://www.jobimtext.org/druid.

Table 2
The ten most similar entries for the term *red blood cell* (left) and *red blood* (right). Here, seven of ten terms are single words in both lists.

red blood cell		*red blood*	
Similar term	Score	Similar term	Score
erythrocyte	133	**red**	148
red cell	129	white blood	111
RBC	95	**Sertoli**	93
platelet	70	**Leydig**	92
red-cell	37	**NK**	86
reticulocyte	34	**mast**	85
white blood	33	**granulosa**	81
leukocyte	29	**endothelial**	81
granulocyte	28	hematopoietic stem	79
the erythrocyte	28	peripheral blood monon	78

Algorithm 1 Computation of the incompleteness score

```
1: function ic(t)
2:     contexts ← context(t)
3:     C ← map()
4:     for all (c_left, c_right) in contexts do
5:         C[c_left, left] ← C[c_left, left] + 1
6:         C[c_right, right] ← C[c_right, right] + 1
7:     end for
8:     return max_value(C)/|contexts|
9: end function
```

For JobimText, these context features are the same that are used for the similarity computation in Section 2 and have been ranked according to the LMI measure. In the case of word2vecf, context features are extracted per word. To be compatible with the JoBimText contexts, we extract the 1,000 contexts with the highest cosine similarity between word and context.

For the example term *red blood*, some of the contexts are ⟨*extravasated, cells*⟩, ⟨*uninfected, cells*⟩, ⟨*nucleated, corpuscles*⟩. In the next step we iterate over all contexts. Using the first context feature results in the tuple *(extravasated, cells)*. Then, we separately count the occurrence of both the left and the right context, including its relative position (left/right) as illustrated in Table 3 for the two example terms.

We subsequently return the maximal count and normalize it by the counts of features $|context(t)|$ considered, which is at most 1,000. This results in the incompleteness measure $ic(t)$. For our example terms we achieve the values $ic(\textit{red blood}) = 557/1,000$ and $ic(\textit{red blood cell}) = 48/1,000$. Whereas the uniqueness scores for the most similar entries are close together (100 vs. 135), we now have a measure that indicates the incompleteness of an *n*-gram, assigning higher scores to more incomplete terms.

3.2.3 Combining Both Measures. As shown in the previous two sections, a high uniqueness score indicates the multiwordness and a high incompleteness score should decrease

Table 3
Top three most frequent context words for the term *red blood cell* and *red blood* in the Medline corpus.

Context term	Position	Count
	red blood cell	
transfusions	right	48
(	right	42
transfusion	right	33
	red blood	
cells	right	557
cell	right	344
corpuscles	right	13

the overall score. In our experiments (see Section 3.9) we reveal that using solely the *uniqueness* score results in good scores. However, often expressions ending with stopwords and incomplete MWEs are detected. In experiments, we found the best combination when we subtract the incompleteness score from the uniqueness score.[6] This mechanism is inspired by the NC-value and motivated as terms that are often preceded/followed by the same word do not cover the full multiword expression and need to be downranked. This leads to Equation (8), which we call **D**ist**R**ibutional **U**niqueness and **I**ncompleteness **D**egree:

$$\texttt{DRUID}(t) = uq(t) - ic(t) \tag{8}$$

Applying the DRUID score to our example terms (considering the 200 most similar terms) we achieve the scores $\texttt{DRUID}(\textit{red blood cell}) = 135/200 - 48/1,000 = 0.627$ and $\texttt{DRUID}(\textit{red blood}) = 100/200 - 557/1,000 = -0.057$. As a higher DRUID score indicates the multiwordness of an *n*-gram, we conclude that the *n*-gram *red blood cell* is a better MWE than the *n*-gram *red blood*.

3.3 Experimental Setting

To evaluate the method, we examine two experimental settings: first, we compute all measures on a small corpus that has been annotated for MWEs, which serves as the gold standard. In the second setting, we compute the measures on a larger in-domain corpus. The evaluation is again performed for the same candidate terms as given by the gold standard. Results for the top k ranked entries are reported using the precision at k:

$$P@k = \frac{1}{k}\sum_{i=1}^{k} x_i \tag{9}$$

6 Our experiments revealed that multiplicative combinations consistently performed worse.

with x_i equal to 1 if the ith ranked candidate is annotated as MWE and 0 otherwise. For an overall performance we use the average precision (AP) as defined by Thater, Dinu, and Pinkal (2009):

$$AP = \frac{1}{|T_{mwe}|} \sum_{k=1}^{|T|} x_k P@k \tag{10}$$

with T_{mwe} being the set of positive MWEs. When facing tied scores, we mix false and true candidates randomly following Cabanac et al. (2010).

3.4 Corpora

For the first experiments, we consider two annotated (small) corpora and two unannotated (large) corpora for the evaluation and computation of MWEs. The language independence of DRUID is demonstrated on various Wikipedia text corpora.

GENIA Corpus and SPMRL 2013: French Treebank. In the first experiments, we use two small annotated corpora that serve as the gold standard MWEs. We use the medical GENIA corpus (Kim et al. 2003), which consists of 1,999 abstracts from Medline[7] and encompasses 0.4 million words. This corpus has annotations regarding important and biomedical terms.[8] In addition, single terms are annotated in this data set, which we ignore.

The second small corpus is based on the French Treebank (Abeillé and Barrier 2004), which was extended for the SPMRL task (Seddah et al. 2013). This version of the corpus also contains compounds annotated as MWEs. In our experiments, we use the training data, which cover 0.4 million words.

Whereas the GENIA MWEs target term matching and medical information retrieval, the SPMRL MWEs mainly focus on improving parsing through compound recognition.

Medline Corpus and Est Républicain Corpus (ERC). In a second experiment, the scalability to larger corpora is tested. For this, we make use of the entire set of Medline abstracts, which consists of about 1.1 billion words. The Est Républicain Corpus (Seddah et al. 2012) is our large French corpus.[9] It is made up from local French newspapers from the eastern part of France and comprises 150 million words.

Wikipedia. Applying the methods to texts extracted from 32 Wikipedias validates their language independence. For this, we use the following languages: Arabic, Basque, Bulgarian, Catalan, Croatia, Czech, Danish, Dutch, English, Estonian, Finnish, French, Galician, German, Greek, Hebrew, Hungarian, Italian, Kazakh, Latin, Latvian, Norwegian, Persian, Polish, Portuguese, Romanian, Russian, Slovene, Spanish, Swedish, Turkish, and Ukrainian.

7 The Medline corpus is available at: http://www.nlm.nih.gov/bsd/licensee/access/medline_pubmed.html.
8 The GENIA corpus is freely available at: http://www.nactem.ac.uk/genia/genia-corpus/pos-annotation.
9 The ERC is available at: http://www.cnrtl.fr/corpus/estrepublicain.

3.5 Candidate Selection

In the first two experiments, we use POS filters to select candidates. We concentrate on filters that extract noun MWEs, as they constitute the largest number of MWEs (see Table 1) and avoid further preprocessing like lemmatization. We use the filter introduced by Justeson and Katz (1995) for the English medical data sets (see Table 4).

Considering only terms that appear more than ten times yields 1,340 candidates for the GENIA data set and 29,790 candidates for the Medline data set. According to Table 5, we observe that most candidates are bigrams. Whereas about 20% of MWEs are trigrams in both corpora, only a marginal number of longer MWEs have been marked.

For the French data sets, we apply the POS filter proposed by Daille, Gaussier, and Langé (1994), which is suited to match nominal MWEs (see Table 4). Applying the same filtering as for the medical corpora leads to 330 candidate terms for the SPMRL and 7,365 candidate terms for the ERC. Here the ratio between bigrams and trigrams is more balanced but again the number of 4-grams constitutes the smallest class.

In comparison with the Medline data set, the ratio of multiwords extracted by the POS filter on the French corpus is much lower. We attribute this to the fact that in the French data, many adverbial, prepositional MWEs are annotated, which are not covered by the POS filter.

The third experiment shows the performance of the method in absence of language-specific preprocessing. Thus, we only filter the candidates by frequency and do not make use of POS filtering. As most previous methods rely on POS-filtered data, we cannot compare with them in this language-independent setting.

For the evaluation, we compute the scores of the competitive methods in two ways: First, we compute the scores based on the full candidate list without any frequency filter and prune low-frequent candidates only for the evaluation (post-prune). In the second setting, we filter candidates according to their frequency before the computation

Table 4
POS sequences for filtering noun MWEs for English and French. Each letter is a truncated POS tag of length one where `J` is an adjective, `N` a noun, `P` a preposition, and `D` a determiner.

Language	POS filter
English (Korkontzelos 2010)	`(([JN]+[JN]?[NP]?[JN]?)N)`
French (Daille, Gaussier, and Langé 1994)	`N[J]?|NN|NPDN`

Table 5
Number of MWE candidates after filtering for the expected POS tag. Additionally, the table shows the distribution over n-grams with $n \in \{1, 2, 3, 4\}$.

Corpus	Total Number of Candidates	2-gram	3-gram	4-gram
GENIA	1,340	1,056	243	41
Medline	29,790	22,236	6,400	1,154
SPMRL	330	197	116	17
ERC	7,365	3,639	2,889	837

of scores (pre-prune). This leads to differences for context-aware measures, because in the pre-pruned case a lower number of less noisy contexts is used.

The evaluation on Wikipedia is slightly different, as we do not have any gold data. Thus, we compute the ranking regarding the multiwordness for all words in the corpus. Based on this list, we determine the multiwordness of an *n*-gram by testing its existence in the respective language's Wiktionary.

3.6 Results Using POS

First, we present the results based on the GENIA corpus (see Table 6). Almost all competitive methods beat the lower baseline. The C/NC-value performs best when the pruning is done after the frequency filtering. In line with the findings of Korkontzelos (2010) and in contrast to Frantzi, Ananiadou, and Tsujii (1998), the AP of the C-value is slightly higher than for the NC-value. All the FGM-based methods except the GM measure alone outperform the C-value. The results in Table 6 indicate that the best competitive system is the post-pruned FGM system, as it has much higher average precision scores and misses only 50 MWEs in the first 500 entries. A slightly different picture is presented in Figure 1, where we plot the *P@k* scores against the number of candidates. Here `DRUID` computed on the JoBimText similarities performs well for the top-*k* list for small *k*, that is, it finds many valid MWEs with high confidence, thus combines well with MF, which extends to larger *k*, but places too much importance of frequency when used alone. Common errors occur for frequent prepositional phrases, such as

Table 6
Results for P@100, P@500, and the average precision (AP) for various ranking measures. The gold standard is extracted using the GENIA corpus. This corpus is also used for computing the measures.

Method	*P*@100	*P*@500	AP
upper baseline	1.000	1.000	1.0000
lower baseline	0.713	0.713	0.7134
frequency	0.790	0.750	0.7468
t-test	0.790	0.750	0.7573
C-value (pre-pruned)	0.880	0.846	0.8447
NC-value (pre-pruned)	0.880	0.840	0.8405
GM	0.590	0.662	0.6740
MF (pre-pruned)	0.920	0.872	0.8761
FGM (pre-pruned)	0.910	0.840	0.8545
MF (post-pruned)	0.900	0.876	0.8866
FGM (post-pruned)	0.900	0.900	0.8948
`DRUID`	0.930	0.852	0.8663
`DRUID` (using word2vec)	0.800	0.740	0.7352
Uniqueness (using word2vec)	0.680	0.752	0.7283
Incompleteness (using word2vec)	0.760	0.724	0.7375
log(freq)·`DRUID`	**0.970**	0.860	0.8661
MF(post-pruned)·`DRUID`	0.950	0.926	**0.9241**
FGM(post-pruned)·`DRUID`	**0.960**	**0.940**	**0.9262**

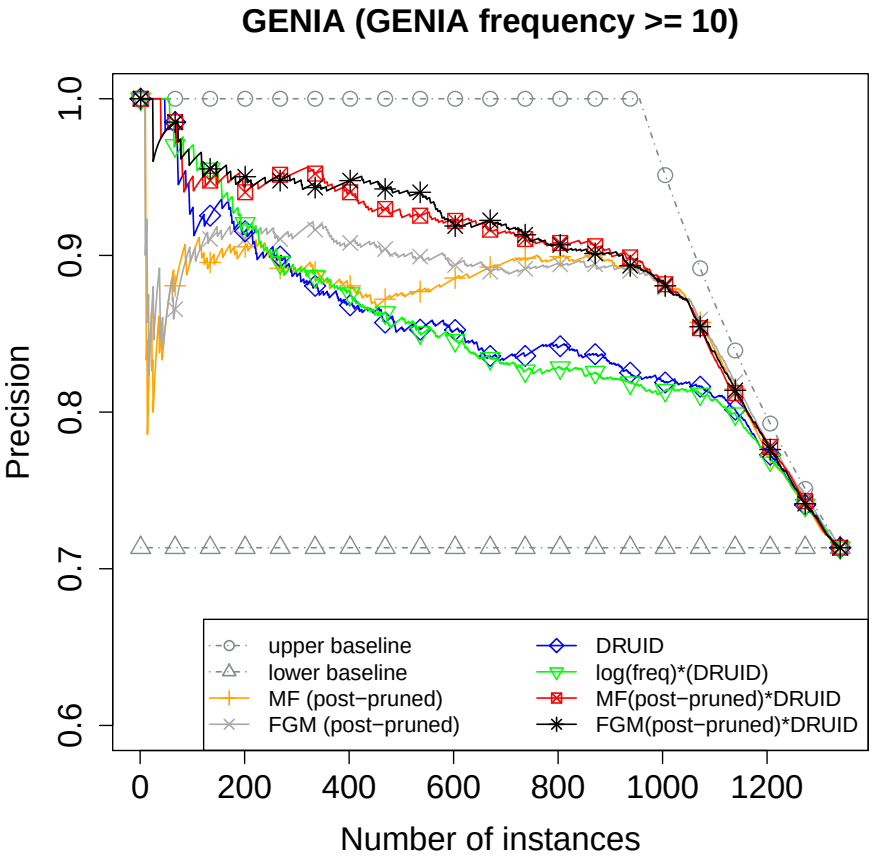

Figure 1
This graph shows the P@k for some measures, plotting the precision against *k*. Using DRUID in combination with the MF and FGM measures yields the highest precision scores.

"in patience" (we give more details on errors in Section 3.10). Using similarities from the word2vec model does not work well for the DRUID method. This is mainly attributed to the fact that multiwords are mostly similar to words of the same frequency (Schnabel et al. 2015; Riedl and Biemann 2017) and often these words are multiwords themselves. Observing, for example, the most similar terms for the term *red blood cells*, we retrieve the words *peripheral blood mononuclear cell*, *show that the*, *U937 cells*, *basal*, *potent*, which are much noisier than the ones we obtain with the JoBimText model (see Table 2 in Section 3.2.1) and within the top 10 most similar terms, we only find four single-worded terms. This is already an indicator that the concept of uniqueness does not apply to similarities computed with word2vec. In contrast, the JoBimText similarities are most similar to more frequent words (we detected 7 out of 10 terms to be unigrams), and we detect more synonyms and hypernyms that are single-word terms. Only for the P@100, can the word2vec-based method beat the t-test and frequency baselines. However, for all other measures, the performance is similar to these baselines or even inferior, and significantly worse than using DRUID with JoBimText. Thus, we will not report results for the other MWE extraction experiments.

When looking at effects between post-pruning and pre-pruning, we observe that FGM scores higher than MF when post-pruning, but the inverse is observed when pre-pruning. Our JoBimText-based DRUID method can outperform FGM only on the top-ranked 300 terms (see Figure 1 and Table 6).

Multiplying the logarithmic frequency with DRUID, the results improve slightly and the best *P*@100 of 0.97 is achieved. All FGM results are outperformed when combining the post-pruned FGM scores with our measure. According to Figure 1, this combination achieves high precision for the first ranked candidates and still exploits the good performance of the post-pruned FGM based method for the middle-ranked candidates.

Different results are achieved for the SPMRL data set, as can be seen in Table 7. Whereas the pre-pruned C-value again receives better results than frequency, it scores below the lower baseline. In addition, the post-pruned FGM and MF method do not exceed the lower baseline. Data analysis revealed that for the French data set only ten out of the 330 candidate terms are nested within any of the candidates. This is much lower than the 637 terms nested in the 1340 candidate terms for the GENIA data set. As both the FGM-based methods and the C/NC-value heavily rely on nested candidates, they cannot profit from the candidates of this data set and achieve similar scores as ordering candidates according to their frequency. Comparing the baselines to our scoring method, this time we obtain the best result for DRUID without additional factors. However, multiplying DRUID with MF or log(frequency) still outperforms the other methods and the baselines.

Most MWE evaluations have been performed on rather small corpora. Here, we examine the performance of the measures for large corpora, to realistically simulate a situation where the MWEs should be found automatically for an entire domain or language.

Using the Medline corpus, all methods except the GM score outperform the lower baseline and the frequency baseline (see Table 8). Regarding the AP the best results are obtained when combining our DRUID method with the MF, whereas for *P*@100 and *P*@500 the log-frequency-weighted DRUID scores best. As we can observe from

Table 7
Results for MWE detection on the French SPMRL corpus. Both the generation of the gold standard and the computations of the measures have been performed on this corpus.

Method	***P*@100**	***P*@200**	**AP**
upper baseline	1.000	0.860	1.0000
lower baseline	0.521	0.521	0.5212
frequency	0.500	0.480	0.4876
t-test	0.500	0.485	0.4934
C-value (pre-pruned)	0.490	0.540	0.5107
MF (post-pruned)	0.510	0.495	0.5017
FGM (post-pruned)	0.460	0.480	0.4703
DRUID	**0.790**	**0.690**	**0.7794**
log(freq)·DRUID	0.770	0.675	0.7631
MF(post-pruned)·DRUID	0.700	0.630	0.6850
FGM(post-pruned)·DRUID	0.600	0.570	0.5948

Table 8
Results of *n*-gram ranking on the medical data. Whereas the gold standard is extracted from the GENIA data set, the ranking measures as well as the frequency threshold for selecting the gold candidates are computed using the Medline corpus.

Method	*P*@100	*P*@500	AP
upper baseline	1.000	1.000	1.0000
lower baseline	0.416	0.416	0.4161
frequency	0.720	0.534	0.4331
C-value (pre-pruned)	0.750	0.564	0.4519
t-test	0.720	0.542	0.4483
GM	0.210	0.272	0.3502
MF (pre-pruned)	0.550	0.542	0.4578
FGM (pre-pruned)	0.580	0.478	0.4200
MF (post-pruned)	0.530	0.500	0.4676
FGM (post-pruned)	0.490	0.446	0.4336
DRUID	0.770	0.686	0.4608
log(freq)·DRUID	**0.860**	**0.720**	0.4693
GM · DRUID	0.770	0.634	0.4497
MF(pre-pruned)·DRUID	0.730	0.634	**0.4824**
MF(post-pruned)·DRUID	0.730	0.626	**0.4889**

Figure 2, using solely the DRUID method or the combined variation with the log-frequency lead to the best ranking for the first 1,000 ranked candidates. However, both methods are outperformed beyond the first 1,000 ranked candidates by the MF-informed DRUID variations. Using the combination with GM results in the lowest scores.

In this experiment, the C-value achieves the best performance from the competitive methods for the *P*@100 and *P*@500, followed by the t-test. But the highest AP is reached with the post-pruned MF method, which also outperforms the sole DRUID slightly. Contrary to the GENIA results, the MF scores are consistently higher than the FGM scores.

In the French ERC, no nested terms are found within the candidates. Thus, the post-pruned and pre-pruned settings are equivalent and thus MF equals frequency. We show the results for the evaluation using the ERC in Table 9.

The best results are again obtained with our method with and without the logarithmic frequency weighting. Again, the AP of the C-value and most of the FGM-based methods are inferior to the frequency scoring. Only the t-test and the MF score slightly higher than the frequency.[10] In contrast to the results based on the smaller SPMRL data set, the MF, FGM, and C-value can outperform the lower baseline.

In comparison to the smaller corpora, the performance for the larger corpora is much lower. Especially low-frequent terms in the small corpora that have high frequencies in the larger corpora have not been annotated as MWEs.

10 This is achieved by chance for the MF, as it is equal to the frequency. The different scores are due to the randomly sorted tied scores used during our evaluation and reflect the variance of randomness.

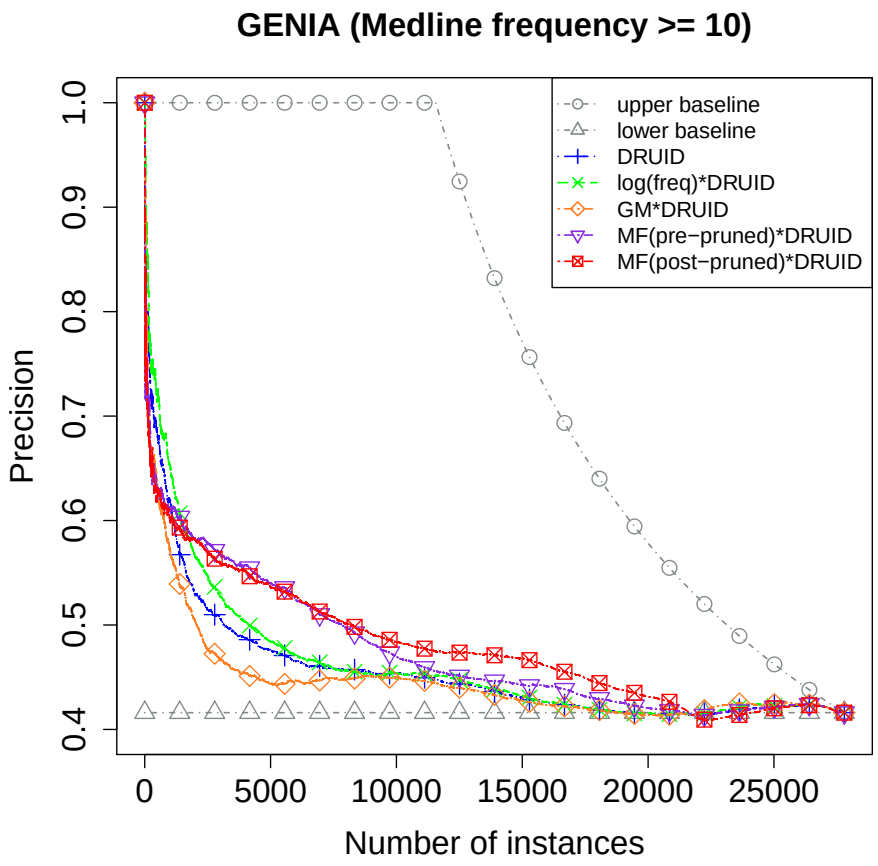

Figure 2
Precision scores when considering different number of highest ranked words for DRUID and combined DRUID variations. Here, the gold standard is extracted from the GENIA data set, whereas the scores for the methods are computed using the Medline corpus.

3.7 Results Without POS Filtering

Next, we apply our method to candidates without any POS filtering and report results for candidates surpassing a frequency threshold of 10. Thus, we do not only restrict the evaluation on noun MWEs but use all MWEs of all POS classes that have been annotated in both corpora. As most competitive methods from the previous section rely on POS tags, we only use the t-test for comparison.

Analysis revealed that the top-scored candidates according to the t-test begin with stopwords. As an additional heuristic for the t-test, we shift those MWEs to the last ranks that start or end with one of the most frequent ten words in the corpus. For the smaller data set the best results are achieved with the sole DRUID (see Table 10) and frequency weighting does not seem to be beneficial, as highly frequent n-grams ending with stopwords are ranked higher in absence of POS filtering. This, however, is not observed for larger corpora. Here, the best results for Medline are achieved with the frequency-weighted DRUID. Whereas for French, the sole DRUID method performs best, the difference between the DRUID and the log-frequency-weighted DRUID is rather

Table 9
Results for ranking *n*-grams, according to their multiwordness, based on the French ERC. The candidates are extracted based on the smaller SPMRL corpus.

Method	*P*@100	*P*@500	AP
upper baseline	1.000	1.000	1.0000
lower baseline	0.220	0.220	0.2201
frequency	0.370	0.354	0.3105
C-value	0.420	0.366	0.3059
t-test	0.390	0.360	0.3134
GM	0.010	0.052	0.1694
MF	0.370	0.356	0.3148
FGM	0.280	0.260	0.2405
DRUID	0.700	0.568	0.3962
log(freq)·DRUID	**0.760**	**0.582**	**0.4075**
MF · DRUID	0.570	0.516	0.3776
FGM · DRUID	0.510	0.418	0.3234

Table 10
MWE ranking results based on different methods without using any linguistic preprocessing.

Corpora	Method	Medical		French	
		P@100	AP	*P*@100	AP
small corpora	upper baseline	1.000	1.0000	1.000	1.0000
	lower baseline	0.107	0.1071	0.083	0.0832
	frequency	0.150	0.1135	0.060	0.0906
	t-test	0.160	0.1261	0.080	0.1097
	t-test + sw	0.530	0.3643	0.180	0.1481
	DRUID	**0.700**	**0.4048**	**0.670**	**0.2986**
	log(freq)·DRUID	0.690	0.3644	0.460	0.2527
large corpora	upper baseline	1.000	1.0000	1.000	1.0000
	lower baseline	0.036	0.0361	0.019	0.0191
	frequency	0.010	0.0361	0.060	0.0366
	t-test	0.020	0.0412	0.080	0.0440
	t-test + sw	0.000	0.0989	0.200	0.0485
	DRUID	0.610	0.1378	**0.660**	**0.1009**
	log(freq)·DRUID	**0.760**	**0.1649**	0.600	0.0988

small. The low APs can be explained by the large number of considered candidates. The second best scores are achieved with the stopword-filtered t-test (t-test + sw). As in this setting the C-value cannot make use of candidate filtering based on POS tags, we do not list its performance, as it performs on par with frequency.

3.8 Multilingual Evaluation

In order to demonstrate the performance of DRUID for several languages, we perform an evaluation on 32 languages. For this experiment, we compute similarities on their respective Wikipedias.[11] The evaluation is performed by extracting the 1,000 highest ranked words using DRUID. In order to determine whether a word sequence is a MWE, we use Wiktionary as "gold" standard and test whether it occurs as word entry.[12] Using this information, we compute the AP for these 1,000 ranked words.

We present the results for this experiment in columns 2 to 5 in Table 11. The t-test with stopword filtering mostly performs similar to the frequency baseline and improves from an average score of 0.07 to 0.08. We observe that in comparison to two baselines, frequency (freq.) and the t-test with stopword filtering, the DRUID method yields the best scores for 6 out of the 32 languages. However, if we multiply the logarithmic frequency by the DRUID measure, we gain the best performance for 30 languages. In general, numerical scores are low—for example, for Arabic, Slovene, or Italian, we obtain APs below 0.10. The highest scores are achieved for Swedish (0.33), German (0.36), Turkish (0.36), French (0.44), and English (0.70). Analyzing the results, we observe that many "false" MWEs are multiword units that are in fact multiword units, which are just not covered in the respective language's Wiktionary. Furthermore, we detect that these word sequences often are titles of Wikipedia articles. The absence of word lemmatization causes further decline, as words in Wiktionary are recorded in lemmatized form. To alleviate this influence, we extend our evaluation and check the occurrence of word sequences both in Wiktionary and Wikipedia. Using the Wikipedia API also normalizes query terms and, thus, we obtain a better word sequence coverage. This is confirmed by much higher results, as shown in columns 6 through 9 in Table 11. Using the frequency combination with DRUID, we even gain higher APs for languages, which attained worse scores in the previous setting (e.g., Arabic [0.62], Slovene [0.17], and Italian [0.44]). Except for Estonian and Polish, using the logarithmic frequency weighting performs best for all languages. For these two languages, using the sole DRUID measure performs best. The best performance is obtained for English (0.87), Turkish (0.66), French (0.66), German (0.62), and Portuguese (0.64). Based on these multilingual experiments, we have demonstrated that DRUID not only performs well for English and French, but also for other languages, showing that its elements, uniqueness and incompleteness, are language-independent principles for multiword characterization.

3.9 Components of DRUID

Here, we show different parameters for DRUID, relying on the English GENIA data set without POS filtering of MWE candidates and by considering only terms with a frequency of 10 or more. Inspecting the two different components of the DRUID measure (see Figure 3 top), we observe that the uniqueness measure contributes most to the DRUID score. The main effect of the incompleteness component is the downranking of a rather small number of terms with high uniqueness scores, which improves the overall

11 We use Wikipedia dumps from late 2016.

12 For querying terms, we use the Wiktionary API: https://en.wiktionary.org/w/api.php, February 2017.

Table 11
AP for the frequency baseline, t-test, and DRUID evaluated against Wiktionary and a combination of Wiktionary and Wikipedia, including word normalization.

Language	Wiktionary				Wiktionary & Wikipedia			
	freq	t-tests +sw	DRUID	log(freq)· DRUID	freq	t-test +sw	DRUID	log(freq)· DRUID
Arabic	**0.01**	**0.01**	0.00	**0.01**	0.27	0.30	0.32	**0.62**
Basque	0.01	0.01	0.01	**0.03**	0.05	0.06	**0.33**	0.23
Bulgarian	0.01	0.01	0.00	**0.03**	0.28	0.35	0.23	**0.54**
Catalan	0.02	0.02	0.06	**0.07**	0.13	0.18	0.29	**0.39**
Croatia	0.04	0.05	0.01	**0.06**	0.14	0.15	0.11	**0.21**
Czech	0.07	0.07	0.01	**0.08**	0.17	0.20	0.14	**0.28**
Danish	0.01	0.02	0.01	**0.25**	0.19	0.21	0.19	**0.32**
Dutch	0.09	0.11	0.05	**0.18**	0.20	0.25	0.27	**0.53**
English	0.10	0.49	0.21	**0.70**	0.19	0.54	0.56	**0.87**
Estonian	0.03	0.03	0.03	**0.05**	0.12	0.13	**0.17**	0.14
Finnish	**0.14**	0.12	0.02	0.11	0.11	0.11	0.16	**0.19**
French	0.17	0.18	0.21	**0.44**	0.30	0.32	0.38	**0.66**
Galician	**0.12**	0.10	0.03	**0.12**	0.29	0.29	0.19	**0.42**
German	0.25	0.23	0.07	**0.36**	0.28	0.27	0.40	**0.65**
Greek	0.07	0.08	0.04	**0.08**	0.14	0.17	0.15	**0.26**
Hebrew	0.05	0.06	0.01	**0.12**	0.27	0.31	0.05	**0.34**
Hungarian	0.09	0.10	0.03	**0.20**	0.14	0.16	0.09	**0.29**
Italian	**0.10**	**0.10**	0.01	**0.10**	0.28	0.30	0.06	**0.44**
Kazakh	0.01	0.01	0.01	**0.05**	0.07	0.08	0.27	**0.32**
Latin	0.01	0.01	0.04	**0.09**	0.09	0.11	0.13	**0.28**
Latvian	0.00	0.00	0.00	**0.01**	0.10	0.10	0.07	**0.13**
Norwegian	0.02	0.02	**0.39**	0.21	0.19	0.20	0.28	**0.40**
Persian	0.08	0.11	0.04	**0.19**	0.29	0.37	0.41	**0.55**
Polish	0.07	0.08	0.02	**0.19**	0.12	0.14	**0.36**	0.32
Portuguese	0.14	0.14	0.05	**0.20**	0.31	0.34	0.32	**0.64**
Romanian	0.05	0.06	0.05	**0.16**	0.20	0.25	0.19	**0.47**
Russian	0.07	0.07	0.02	**0.15**	0.16	0.17	0.16	**0.27**
Slovene	0.01	0.01	0.01	**0.05**	0.09	0.11	0.09	**0.17**
Spanish	**0.12**	0.14	0.02	**0.12**	0.34	0.42	0.26	**0.63**
Swedish	0.07	0.10	0.03	**0.33**	0.19	0.26	0.42	**0.58**
Turkish	0.08	0.10	0.20	**0.36**	0.20	0.22	0.50	**0.66**
Ukrainian	0.01	0.01	0.02	**0.04**	0.09	0.11	0.12	**0.14**
Average	0.07	0.08	0.05	0.16	0.19	0.22	0.24	0.40

ranking. We can also see that for the top-ranked terms, the negative incompleteness score does not improve over the frequency baseline but merely outperforms frequency for candidates in the middle range. Used in DRUID, we observe a slight improvement for the complete ranking.

We achieve a P@500 of 0.474 for the uniqueness scoring and 0.498 for the DRUID score.

When filtering similar entries, used for the *uq* scoring, by their similarity score (see Figure 3 bottom), we observe that the amount of similar *n*-grams considered seems to be more important than the quality of the similar entries: With the increasing filtering, the quality of extracted candidate MWEs diminishes.

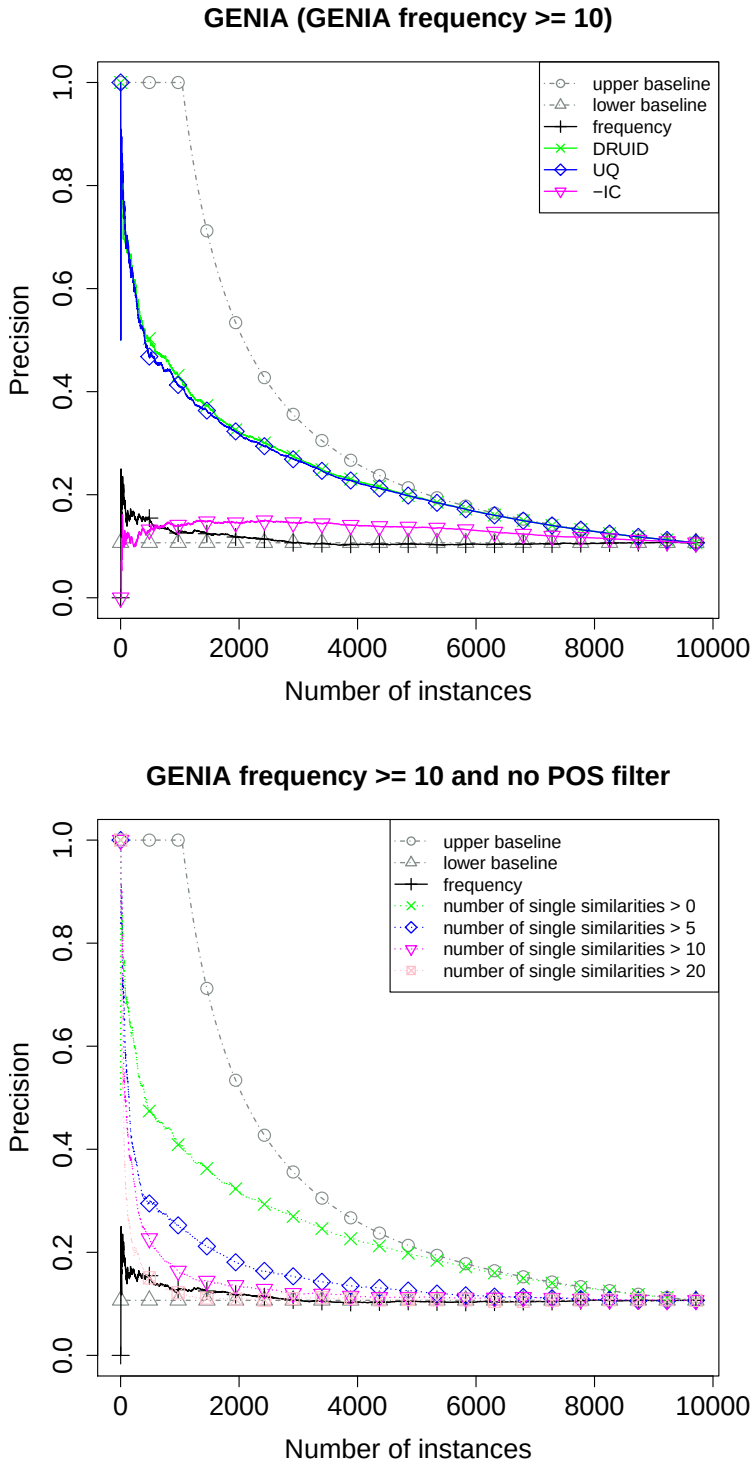

Figure 3
Results for the components of the DRUID measure (top) and for different filtering thresholds (bottom) of the similar entries considered for the uniqueness scoring.

3.10 Discussion and Data Analysis

The experiments confirm that our DRUID measure, either weighted with the MF or alone, works best across two languages and across different corpus sizes. It also achieves the best results in absence of POS filtering for candidate term extraction. The optimal weighting of DRUID depends on the nestedness of the MWEs: Using DRUID with the MF should be applied when there are more than 20% of nested candidates. If there are no nested candidates, we recommend using the log-frequency or no frequency weighting.

We present the best-ranked candidates obtained with our method and with the best competitive method in terms of *P*@100 for the two smaller corpora. Using the GENIA data set, our log-frequency based DRUID (see left column in Table 12) ranks only true MWE within the 15 top-scored candidates.

The right-hand side shows results extracted with the pre-pruned MF method that yields three non-MWE terms. Whereas these terms seem to be introduced as candidates due to a POS error, the MF, and the C-value are not capable of removing terms starting with stopwords. The DRUID score alleviates this problem with the uniqueness factor. For the French data set, only one false candidate is ranked in the top 15 candidates.

In comparison, eight non-annotated candidates are ranked in the top 15 candidates by the MF (post-pruned) method as shown in Table 13.

Whereas the unweighted DRUID method scores better than its competitors on the large corpora, the best numerical results are achieved when using DRUID with frequency-based weights on smaller corpora. For a direct comparison, we evaluated the small and large corpora using an equal candidate set. We observed that all methods computed on the large corpora achieve slightly inferior results than when computing them using the small corpora.

Data analysis revealed that we personally would consider many of these high ranked "false" candidates as MWEs.

For examining the effect, we extracted the top ten ranked terms, which are not annotated as MWE from the methods with the best *P*@100 performance, resulting in

Table 12
Top ranked candidates from the GENIA data set using our ranking method (left) and the competitive method (right). Each term is marked if it is an MWE (1) or not (0).

log(freq)·DRUID		**MF (pre-pruned)**	
NF-kappa B	1	T cells	1
transcription factors	1	NF-kappa B	1
transcription factor	1	transcription factors	1
I kappa B alpha	1	activated T cells	1
activated T cells	1	T lymphocytes	1
nuclear factor	1	human monocytes	1
human monocytes	1	I kappa B alpha	1
gene expression	1	nuclear factor	1
T lymphocytes	1	gene expression	1
NF-kappa B activation	1	NF-kappa B activation	1
binding sites	1	in patients	0
MHC class II	1	important role	0
tyrosine phosphorylation	1	binding sites	1
transcriptional activation	1	in B cells	0
nuclear extracts	1	transcriptional activation	1

Table 13
Top ranked candidates from the SPMRL data set for the best DRUID method (left) and the best competitive method (right). Each term is marked if it is an MWE (1) or not (0).

DRUID		MF (post-pruned)	
hausse des prix	1	milliards de francs	0
mise en oeuvre	1	millions de francs	0
prise de participation	1	Etats - Unis	1
chiffre d' affaires	1	chiffre d' affaires	1
formation professionnelle	1	taux d' intérêt	1
population active	1	milliards de dollars	0
taux d' intérêt	1	millions de dollars	0
politique monétaire	1	Air France	1
Etats - Unis	1	% du capital	0
Réserve fédérale	1	milliard de francse	0
comité d' établissement	1	directeur général	1
projet de loi	1	M. Jean	0
système européen	0	an dernier	1
conseil des ministres	1	années	1
Europe centrale	1	% par rapport	0

Table 14
Top ranked terms for the Medline corpus, which are not marked as MWEs. The rank is denoted to the left of each term and all terms, which can be found within a lexicon, are marked in **bold**.

	log(freq)DRUID		C-value (pre-pr.)
26	**carboxylic acid**	1	present study
28	**connective tissue**	7	important role
40	**cathepsin B**	11	degrees C
41	**soft tissue**	13	**risk factors**
42	**transferrin receptor**	15	significant differences
53	DNA damaging	18	other hand
61	**foreign body**	22	significant difference
62	radical scavenging	33	**magnetic resonance**
71	spatial distribution	39	first time
74	**myosin heavy chain**	48	significant increase

the log(freq) DRUID and the pre-pruned C-value methods. We show the terms including their ranking position based on the GENIA data set in Table 14.

First, we observe that the first "false" candidate for our method appears at rank 26 and at rank 1 for the C-value. Additionally, only 10 out of the top 74 candidates are not annotated as MWEs for our method, whereas the same number of 10 non-MWEs is found in the first 48 candidates for the competitor. When searching the terms within the MeSH dictionary, we find seven terms ranked from our method and two for the competitive method, showing that most such errors are at least questionable, given that these terms are contained in a domain-specific lexicon.[13] This leads us to the conclusion that our method scales to larger corpora.

13 The MeSH dictionary is available at: http://www.nlm.nih.gov/mesh/.

Table 15
The highest ranked single-worded terms for Medline and ERC without any POS filtering, based on the DRUID score.

Medline		ERC	
GATA-1	antiatherosclerotic	mesure	Bergé
function	Smad6	activités	carnets
Sp1	Evi-1	politique	Bouvet
used	ETS1	prix	promesse
increased	3q26	réduction	préoccupe
shown	Tcf	analyse	composants
IFN-gamma	LEF-1	crise	aspirations
decreased	hypolipidaemic	stratégie	hostilité
IL-10	down-regulatory	tête	dettes
IL-5	Xq13	campagne	Brunet

In contrast to the competitive measures introduced in this section, our method is also able to rank single-worded terms. We show the 20 highest ranked single-worded terms in Table 15 for the Medline and the ERC corpus. In both lists we did not filter by POS and removed numbers, which often have a high DRUID score. Both for French and for the medical data, we observe some verbs, but mostly common and proper nouns. These are well suited as keyword lists that are required for document indexing used, e.g., for search engines or automatic speech recognition, as we have demonstrated in Milde et al. (2016).

3.11 Summary on MWE Identification

In this section, we have demonstrated the capabilities of our method for the identification of MWEs in order to treat them as single tokens. Using similarities from word2vec does not work well for DRUID and applying the symbolic JoBimText approach works better. This is mainly attributed to the fact that JoBimText prefers to extract similarities to more frequent words, which are often single-worded terms (e.g., hypernyms), whereas word2vec mostly predicts multiword expressions and words of the same frequency for similarity queries. This is possible as JoBimText does not embed terms in a metric space that is subject to the triangle inequality and is in line with the research by Schnabel et al. (2015) and Riedl and Biemann (2017). These findings show that using similarities from a symbolic semantic method brings added value when it comes to the identification of MWEs. Uniqueness is a well-working mechanism in MWE modeling. Whereas frequency and co-occurrence have been captured in many previous approaches (see Manning and Schütze [1999], Ramisch, De Araujo, and Villavicencio [2012], and Korkontzelos [2010] for a survey), we boost multiword candidates t by their grade of distributional similarity with single word terms. We implement such contextual substitutability with a model where the term t can consist of multiword tokens and similarity is measured based on the right and neighboring word between all (single and multiword) terms. Because it is the default to express concepts with single words, a high uniqueness score is assigned to multiwords that belong to the same category just as single words would. For example, using an English open-domain corpus, *hot dog* is most similar to the terms: *food*, *burger*, *hamburger*, *sausage*, and *roadside*. Candidates with a low number of single-word similarities also serve the same function, but more

frequently we observe single *n*-grams with function words or modifying adjectives concatenated with content words—for example, *small dog* is most similar to "*various cat*", "*large amount of*", "*large dog*", "*certain dog*", and "*dog*". To be able to kick in, the measure requires a certain minimum frequency for candidates in order to find enough contextual overlap with other terms. Additionally, we demonstrate effective performance on larger corpora and show its applicability when used in a completely unsupervised evaluation setting. Furthermore, we have demonstrated the language independence of the measure by evaluating it on 32 languages using Wiktionary and Wikipedia for the evaluation.

4. Splitting Words: Decompounding

In order to enable a tokenization for sub-word units, we introduce SECOS (SEmantic COmpound Splitter), which is based on the hypothesis that compounds are similar to their constituting word units.[14] Again, our method is based on a DT. In addition, it does not require any language-specific rules and can be applied in a knowledge-free way. 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*). This section extends the work presented in Riedl and Biemann (2016) by adding results on an Afrikaans and a Finnish data set. Additionally, we introduce an evaluation based on automatically extracted compounds from Wiktionary and present results for 14 languages.

4.1 SEmantic COmpound Splitter (SECOS)

Our method consists of three stages: First, we extract a candidate word set that defines the possible sub-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 different candidate sets, we obtain different compound splits. Finally, we define a mechanism that ranks these splits and returns the top-ranked one.

Candidate Extraction. For the extraction of all candidates in C, we use a 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 do 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 in 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

14 An implementation of SECOS is available at: https://github.com/riedlma/SECOS. Furthermore, we provide models for all the languages that have been processed in this article.

atomic word units. For this, we iterate over the entire vocabulary of the background corpus, applying the compound splitter (see Section 4.1) 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**.

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" that are in fact short abbreviations.

We show candidates, extracted from the similar candidate unit, with $ml = 3$ for the example term in Table 16. 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 overgenerates split points, as can be observed for the example word, which is split into six 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—for example, 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 add a smoothing factor ϵ to each frequency in order to assign non-zero

Table 16
Examples of the output of our algorithms for the example term *Bundesfinanzministerium*.

word w	Bundesfinanzministerium
candidates C	Finanzministerium, Ministerium,
with ml=3	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

values to unknown units.[15] This yields the following formula for a compound w, which is decomposed into the units $w_1, \ldots, w_N$:

$$p(w) = \left(\prod_{i}^{N} \frac{wordcount(w_i) + \epsilon}{total\_wordcount + \epsilon \cdot \#words} \right)^{\frac{1}{N}} \tag{11}$$

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.[16] In our example, this is the prefix-suffix-merged candidate *Bundes-finanz-ministerium*.

Split Ranking. We have examined schemes of priority ordering for integrating information from different candidate sets—for example, using the similar candidate units first and only applying 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.

4.2 Evaluation Setting

For the computation of our method, we use similarities computed on various languages. First, we compute the DTs using JoBimText using the left and the right neighboring word as context representation. In addition, we extract a DT from the CBOW method from word2vec (Mikolov et al. 2013) using 500 dimensions, as described in Section 2. We compute the similarities for German based on 70M sentences and for Finnish on 4M sentences that are provided via the Leipzig Corpora Collection corpus (Richter et al. 2006). For the generation of the Dutch similarities, we use the Dutch web corpus (Schäfer and Bildhauer 2013), which is composed of 259 million sentences.[17] Similarities for Afrikaans are computed using the Taalkommissie corpus (3M sentences) (Taalkommissie 2011) and we use 150GB of texts for Russian.[18] The evaluation for various languages based on the automatically extracted data set is performed on similarities computed on text from the respective Wikipedias.

We evaluate the performance of the algorithms using a splitwise precision and recall measure that is inspired by the measures introduced by Koehn and Knight (2003). Our evaluation is based on the splits of the compounds and is defined as shown:

$$\begin{aligned} precision &= \frac{correct\ split}{correct\ split + wrong\ splits} \\ recall &= \frac{correct\ split}{correct\ split + missing\ splits} \\ F1 &= 2 \cdot \frac{precision \cdot recall}{precision + recall} \end{aligned} \tag{12}$$

15 We set $\epsilon = 0.01$. In the range of $\epsilon = [0.0001, 1]$ we observe marginally higher scores using smaller values.
16 Although our method mostly does not assume language knowledge, we uppercase the first letter of each w_i, when we apply our method on German nouns.
17 Available at: `http://webcorpora.org/`.
18 The sentences are extracted from: `http://lib.rus.ec`.

As unsupervised baselines we use the semantic analogy-based splitter (SAS) from (Daiber et al. 2015)[19] and the split ranking by Koehn and Knight (2003), called KK.

4.3 Data Sets

For the intrinsic evaluation, we chose data sets of various languages. We use one small German data set for tuning the parameters of the methods. This data set consists of 700 manually labeled German nouns from different frequency bands created by Holz and Biemann (2008). For the evaluation, we consider two larger German data sets. The first data set comprises 158,653 nouns from the German newspaper magazine *c't* and was created by Marek (2006).[20] As second data set we use a noun compound data set of 54,571 nouns from GermaNet,[21] which has been constructed by Henrich and Hinrichs (2011).[22] While converting these data sets for the task of compound splitting, we do not separate words in the gold standard, which is made up of prepositions (e.g., the word *Abgang [outflow]* is not split into *Ab-gang [off walk]*).

In addition, we apply our method to a Dutch data set of 21,997 compound nouns and an Afrikaans data set that consists of 77,651 compound nouns. Both data sets have been proposed by van Zaanen et al. (2014). Furthermore, we perform an evaluation on a recent Finnish data set proposed by Shapiro et al. (2017) that comprises 20,001 words. In contrast to the other data set it does not only contain compound words but also 16,968 words with a single stem that must not be split. To show the language independence of our method, we further report results data sets for 14 languages that we collected from Wiktionary. [23]

4.4 Tuning the Method

In order to show the influence of the various candidate sets and to find the best performing parameters of our method, we use the small German data set with 700 noun compounds. We obtain the highest F1 scores (see Table 17) considering only candidates with a frequency above 50 ($wc = 50$) and that have more than four characters ($ml = 5$). Furthermore, we append only prefixes and suffixes equal or shorter than three characters ($ms = 3$ and $mp = 3$).

As observed in Table 17, the highest precision using the JoBimText similarities is achieved with the similar candidate units. However, the recall is lowest because for many words no information is available. Using the extended similarities, the precision decreases and the recall increases. Interestingly, we observe an opposite trend for word2vec. However, the best overall performance is achieved with the generated dictionary, which yields an F1 measure of 0.9583 using JoBimText and 0.9627 using word2vec. Using geometric mean scoring to select the best compound candidate lifts the F1 measure up to 0.9658 using JoBimText and 0.9675 using the word2vec similarities on this data set.

19 https://github.com/jodaiber/semantic_compound_splitting.
20 Available at: http://heise.de/ct.
21 Available at: http://www.sfs.uni-tuebingen.de/lsd/documents/compounds/split_compounds_from_GermaNet10.0.txt.
22 We follow Schiller (2005) and remove all words including dashes. This only affects the GermaNet data set and reduces the effective test set to 53,118 nouns.
23 The data set was collected in February 2017 and is available here: http://ltdata1.informatik.uni-hamburg.de/SECOS/datasets/wiktionary_compounds.tar.gz.

Table 17
Precision (P), Recall (R), and F1 Measure (F1) on split positions for the 700 compound nouns using different split candidates.

	JoBimText			word2vec		
	P	**R**	**F1**	**P**	**R**	**F1**
similar candidates	**0.9880**	0.6855	0.8094	0.9554	0.9548	0.9551
extended similar candidates	0.9617	0.7523	0.8442	**0.9859**	0.6813	0.8058
generated dictionary	0.9576	**0.9589**	**0.9583**	0.9644	**0.9610**	**0.9627**
geometric mean scoring	0.9698	0.9617	0.9658	0.9726	0.9624	0.9675

4.5 Decompounding Evaluation

In this section, we first show results for manually extracted data sets and then demonstrate the multilingual capabilities of our method using a data set that was automatically extracted from Wiktionary. We compare our results to previously available methods, which will be discussed in Section 6.2.

4.5.1 Results for Manually Annotated Data Sets. Now, we compare the performance of our method against unsupervised baselines and knowledge-based systems (see Table 18).

For the 700 nouns we achieve the highest precision, recall, and F1 measure using our method with similarities from word2vec. Because we have tuned our parameters on this comparably small data set, which might be prone to overfitting, we do not discuss these results in depth but provide them again for completeness.

On the c't data set, the best results are observed by using (supervised) JWordSplitter (JWS) followed by supervised Automatische Sprachverarbeitungs Toolbox (ASV), and our method. Here, JWS achieves significant improvements against all other methods in terms of F1 score.[24] Nevertheless, our method yields the highest precision value; SAS and KK score lowest.

Evaluating on the GermaNet data set, our method with similarities both from JoBimText is only outperformed by the supervised ASV method. Similar to the results for 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 (KK and SAS) and yields the overall highest precision.

On the Afrikaans data set we observe higher precision using the baseline method (KK) than using SECOS. By approach, more words get split than using the KK method. Whereas the KK approach identifies most compounds correctly, many compounds are not detected at all. Here, our method performs best using JoBimText.

For Dutch, no trained models for JWS and ASV are available. Thus, we did not use these tools but compare to the 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 significantly beats the KK baseline.

Furthermore, we show results based on a Finnish data set proposed by Shapiro (2016). Whereas her method performs better in terms of recall in comparison to SECOS,

24 We perform a Wilcoxon signed-rank test between the F1 scores of each candidate assuming $p < 0.01$. However, we only obtain a p-value below 0.5.

Table 18
Results based on manually created data sets for German, Dutch, Afrikaans, and Finnish. We mark the best results in **bold** font and use an asterisk (*) to show if a method performs significantly better than the baseline methods and use two asterisks (**) when a single method outperforms all others significantly.

Data Set	Method	Precision	Recall	F1 Measure
700	JWS	0.9328	**0.9666**	0.9494
	SAS	0.8723	0.6848	0.7673
	ASV	0.9584	0.9624	0.9604
	KK	0.9532	0.7801	0.8580
	SECOS (JoBimText)	0.9698	0.9617	0.9658
	SECOS (word2vec)	**0.9726**	0.9624	**0.9675**
c't	JWS	0.9557	**0.9441**	**0.9499****
	SAS	0.9303	0.5658	0.7037
	ASV	0.9571	0.9356	0.9462
	KK	0.9432	0.8531	0.8959
	SECOS (JoBimText)	0.9606	0.9139	0.9367
	SECOS (word2vec)	**0.9624**	0.9143	0.9377
GermaNet	JWS	0.9248	0.9402	0.9324
	SAS	0.8861	0.6723	0.7645
	ASV	0.9346	**0.9453**	**0.9399****
	KK	0.9486	0.7667	0.8480
	SECOS (JoBimText)	0.9543	0.9158	0.9347
	SECOS (word2vec)	**0.9781**	0.8869	0.9303
Afrikaans	KK	**0.9859**	0.6527	0.7854
	SECOS (JoBimText)	0.9224	**0.7524**	**0.8288****
	SECOS (word2vec)	0.9157	0.7512	0.8253
Dutch	NL Splitter	**0.9706**	**0.8929**	**0.9301****
	KK	0.9579	0.8007	0.8723
	SECOS (JoBimText)	0.9624	0.8548	0.9055
	SECOS (word2vec)	0.9718	0.8595	0.9122
Finnish	(Shapiro 2016)	0.8817	0.9266	0.9036
	KK	0.9043	0.9294	0.9167
	SECOS (JoBimText)	0.9574	**0.9472**	**0.9523****
	SECOS (word2vec)	**0.9677**	0.9187	0.9426*

we attain much higher precision and thus also obtain the highest F1 measure. Again, the best results are obtained using similarities from JoBimText, which even significantly outperform the second best results retrieved with our method using similarities from word2vec.

4.5.2 Results for Automatically Extracted Data Sets. In a second evaluation, we report results based on a data set that was automatically extracted from Wiktionary.[25] We extracted all compounds for which we were also able to find their stems in the

25 For the extraction of compounds, we rely on compounds listed on: https://en.wiktionary.org/wiki/Category:Compound_words_by_language.

dictionary. We show only results for languages where we could find at least more than 90 compounds (see Table 19). For the evaluation, we computed similarities using the corresponding Wikipedias.

In contrast to the previous experiments, we do not restrict to noun compounds only but also include words of other POS. As baseline system, we use the unsupervised KK baseline method. Comparing the results achieved with SECOS and KK, the highest precision is obtained by the word2vec-based KK method. However, recall is always much lower than when using SECOS. As the KK baseline does not use any smoothing, it misses many splits, which results in few but precise word splits. In general, the results for German, Dutch, and Finnish are lower than using the manually annotated data sets. This is expected, as the data set is presumably noisier. In comparison to the experiments in Section 4.5.1, the best results using SECOS, except for Latin, are achieved using similarities from JoBimText and not the ones from word2vec. Although the precision scores are mostly comparable, the recall scores are much lower. Also, for German, we observe performance drops when using word2vec. These performance drops can be explained by the different nature of the Wiktionary compounds. First, the Wiktionary data sets mainly have only one split point marked within the compound word, although these words might consist of more than two stems. We tried to resolve this issue by recursively splitting words with nested compounds also contained in the data set. However, recall changed only marginally. The main reason for the inferior performance of word2vec seems to be that most words in the Wiktionary data set are not contained in the processed corpora. For example, whereas on the German c't data set 19% of the words are not contained in the corpus, 60% of the words are unknown in the German Wiktionary data set. It seems that the dictionary-based approach with JoBimText similarities is less sensitive to unknown words than with word2vec similarities.

We observe that SECOS is able to achieve precision values above 95% for most languages. Recall, on the other hand, is consistently lower, resulting in F1 scores around 80%. This again confirms that the method is splitting cautiously.

For Latin, we observe the lowest performance, which we mainly attribute to its small Wikipedia. Using the JoBimText similarities, only 24 of the 98 compounds are contained in the text at all, and only 10 with a frequency above 10. The best results in terms of F1 score are obtained for German (0.8756), Dutch (0.8669), Hungarian (0.8383), and Finnish (0.8544). On average, we obtain an F1 score of 0.8141. The ability to gain good results also on morphologically rich languages such as Hungarian and Finnish demonstrates the language-independence of our method.

4.6 Discussion

In order to understand the errors of our method in the intrinsic evaluation, we analyzed the compounds that were split incorrectly using JoBimText. Whereas previous results were reported per split-point, we now look at the number of wrongly split entire compounds. For the c't compounds data set, our method splits 22.17% of compounds incorrectly, and we observe 32.6% wrongly split compounds in the Dutch data set (see Table 20).

In addition, we analyzed how many compounds have been split in fewer parts (*under-split*) and more parts (*over-split*) than the gold data, or have the same number of splits, that, however, are incorrect (*wrongly-split*). For all data sets we observe a general trend: Our method tends to rather undersplit 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

Table 19
Results of SECOS for compounds that have been automatically extracted from Wiktionary for 14 languages. We show results for the KK baseline and the SECOS method using similarities computed with JoBimText and word2vec.

	SECOS						KK						
	JoBimText			word2vec			JoBimText			word2vec			
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	#
Finnish	0.9494	**0.7768**	**0.8544**	0.9676	0.7489	0.8443	0.9596	0.6537	0.7776	**0.9892**	0.6185	0.7612	11,052
English	0.9827	**0.7087**	**0.8235**	0.0001	0.0000	0.0001	0.9779	0.6414	0.7747	**0.9962**	0.6125	0.7585	10,865
German	0.9451	**0.8156**	**0.8756**	0.9484	0.7310	0.8257	0.9453	0.7463	0.8341	**0.9903**	0.6298	0.7700	5,688
Dutch	0.9675	**0.7852**	**0.8669**	0.0002	0.0001	0.0002	0.9620	0.7138	0.8195	**0.9888**	0.6658	0.7958	4,058
Norwegian *(Bokmål)*	0.9878	**0.7118**	**0.8274**	0.9851	0.6508	0.7838	0.9862	0.6266	0.7663	**0.9971**	0.5957	0.7458	1,426
Swedish	0.9607	**0.7451**	**0.8393**	0.9770	0.6798	0.8018	0.9558	0.6685	0.7867	**0.9930**	0.6130	0.7580	1,279
Norwegian *(Nynorsk)*	0.9917	**0.6995**	**0.8204**	0.9924	0.6410	0.7789	0.9914	0.6210	0.7636	**0.9975**	0.5878	0.7397	1,025
Hungarian	0.9923	**0.7256**	**0.8383**	0.9917	0.6680	0.7983	0.9813	0.6712	0.7971	**1.0000**	0.6368	0.7781	625
Danish	0.9677	**0.6998**	**0.8122**	0.9745	0.6592	0.7864	0.9807	0.6197	0.7595	**0.9966**	0.5943	0.7446	493
Estonian	0.9937	**0.6414**	**0.7796**	**1.0000**	0.6107	0.7583	0.9857	0.5656	0.7188	**1.0000**	0.5410	0.7021	244
Latvian	0.9901	**0.5952**	**0.7435**	0.9949	0.5774	0.7307	1.0000	0.5000	0.6667	**1.0000**	0.5119	0.6772	168
Persian	0.8290	**0.8080**	**0.8183**	0.9275	0.6957	0.7950	0.8217	0.7681	0.7940	**0.9450**	0.6848	0.7941	138
Spanish	0.9854	**0.6888**	**0.8108**	0.9919	0.6224	0.7649	0.9767	0.6429	0.7754	**1.0000**	0.5816	0.7355	98
Latin	0.9904	**0.5255**	**0.6867**	**1.0000**	0.5204	0.6846	1.0000	0.5204	0.6846	**1.0000**	0.5102	0.6757	98
Average	0.9667	**0.7091**	**0.8141**	0.8394	0.5575	0.6681	0.9660	0.6399	0.7656	**0.9924**	0.5988	0.7455	2661

Table 20
Number of compounds that have been split incorrectly with respect to the gold data. We report numbers of how many of these compounds have fewer split points (under-split), too many split points (over-split), or the correct number but wrong split points (wrongly-split). In addition, we show the total number of missed, wrong, and correct splits for these compounds.

Data Set	c't	GermaNet	Dutch
	number of compounds		
# incorrect	35,177	13,612	7,258
% incorrect	22.17	26.63	32.60
under-split	23,773	7,972	5,849
over-split	7,843	3,578	806
wrongly-split	3,561	982	603
	number of splits		
missed	29,213	12,537	6,612
wrong	12,703	2,348	1,520
correct	20,381	5,216	1,743

how often our method missed a split, performed a wrong split, and split correctly (see bottom three lines in Table 20). This analysis supports the previous finding: Most errors of our SECOS method consist of missed splits. Depending on the application, this might be a less detrimental behavior than splitting wrongly.

4.7 Summary on SECOS

In order to enable a fine-grained tokenization on sub-word units, we have introduced an unsupervised method for decompounding words that is based on distributional semantics. We have shown the impact of its components and have tuned its parameters on a small German data set. On six data sets for four different languages, SECOS has been shown to perform competitively to supervised and rule-based tools and to outperform two unsupervised baselines by a large margin. Further, we demonstrated its language-independence using automatically extracted compound data sets for 14 languages. Comparing two methods for the generation of distributional semantic models within SECOS, we obtain the best results for German, Dutch, and Afrikaans using word2vec. However, for Finnish the best results are achieved with JoBimText. On the automatically extracted data sets, JoBimText yields on average F1 scores of 0.8122, whereas the word2vec-based method achieves solely scores of 0.7705, which we attribute to the larger numbers of out-of-vocabulary words within the Wiktionary data set.

5. Using Coarse- and Fine-Grained Tokenization for Information Retrieval

In order to show the benefits of using both coarse-grained and fine-grained tokenization, we report results on an information retrieval task. In previous research, the incorporation of compound nouns and MWE information was used successfully in IR (Acosta, Villavicencio, and Moreira 2011; da Silva and Rocha Souza 2012). Also splitting compounds turned out to be a useful processing step in order to improve IR systems

```
<top>
<num> Number: 372
<title> Native American casino
<desc> Description:
Identify documents that discuss the growth of Native American
casino gambling.
<narr> Narrative:
Relevant documents include discussions regarding Native American
casino gambling: its social implications, effects on local and
Native American economies, and legal aspects related to Native
American tribal autonomy.
</top>
```

Figure 4
Listing of a topic from the TREC 2004 Robust Track.

(Monz and de Rijke 2001; Koehn and Knight 2003; Witschel and Biemann 2005; Airio 2006).

For this, we selected the TREC 2004 Robust Track (Voorhees 2005), in which an IR system is evaluated based on 250 topics for which we use titles of the topic description as query.[26] For performing the evaluation, we setup an index for 528,155 documents from the TREC Disks 4 and 5 (without the Congressional Record on Disk 4). We use Lucene[27] with Okapi BM25 and build indices based on the words of the entire documents, the decompounded words within the documents, and also add indices for the detected MWEs within the documents that have a `DRUID` score above 0.3, 0.5, and 0.7. In order to compute models for decompounding and MWE detection, we use an English Wikipedia dump. This experiment focuses on demonstrating the impact of using the additional information gained by our methods in an extrinsic evaluation, rather than aiming at state-of-the-art retrieval performance. Furthermore, we want to highlight that we do not apply any language-dependent information. Thus, the results should generalize across languages.

For the query, we use the title of each topic. In Figure 4, we show all content that is available for topic 372. For building the query, we only use the title. Using the description (<*desc*>)or the narrative (<*narr*>) requires further pre-processing and did not yield to better scores than using solely the title. We combine the different fields for building queries considering all fields as optional. Building the query for the example using tokens, decompounded tokens, and MWEs, we will obtain the title itself both querying against the tokens and decompounded tokens. In addition, we will query for the MWE *Native American*. As English does not contain many close compounds, the decompounding does not apply to many words and queries. However, words like *hydroelectric* will be split into *hydro* and *electric*.

We show the mean average precision (MAP) scores for various combinations of the queries in Table 21.

As the queries use only words from the titles, querying solely against the MWE index does not make sense, as not all titles contain MWEs. We observe that using tokens of the content does result in better MAP scores than using decompounded tokens.[28]

26 `http://trec.nist.gov/data/robust/04.guidelines.html`.
27 We use version 6.6.0, which is available at: `https://lucene.apache.org/core/`.
28 As we do not perform any pre-processing like POS-tagging, we split all words, not only nouns. This might additionally introduce some mismatches.

Table 21
Results on the information retrieval TREC 7 task using compound and MWE information.

tokens	decompounded tokens	MWE 0.3	MWE 0.5	MWE 0.7	MAP
x					0.2023
	x				0.1980
x	x				0.2038
x		x			0.1964
x			x		0.2000
x				x	0.2028
x	x	x			0.2018
x	x		x		0.2037
x	x			x	0.2040

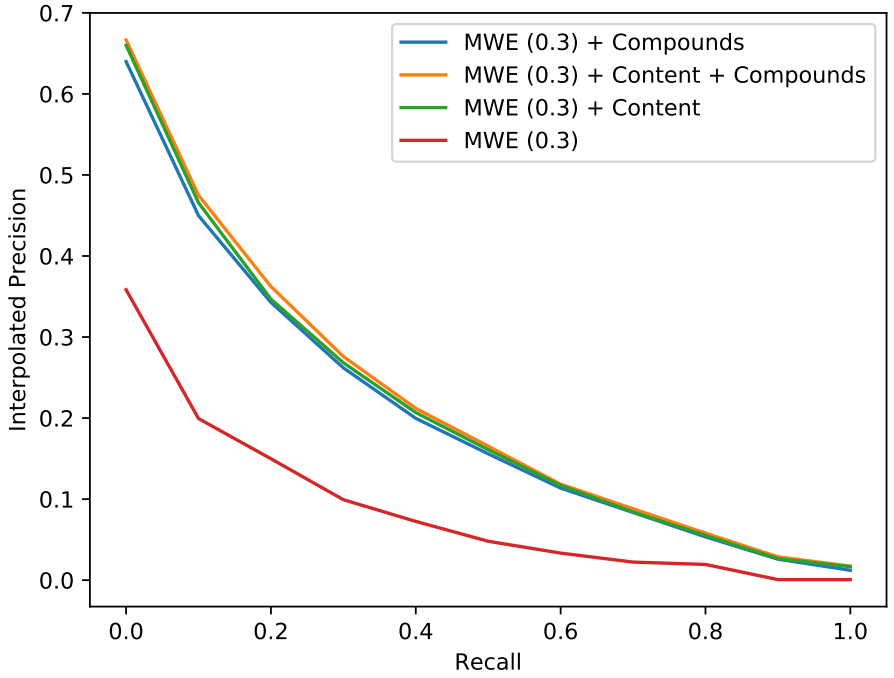

Figure 5
Interpolated precision–recall curve for the TREC 2004 Robust task.

Creating queries with both token and decompounded tokens results in higher MAP scores (0.2038). Combining the original content with MWE information, we obtain inferior results when considering MWEs with a low threshold (0.3 and 0.5) and gain some improvements when using the index with MWEs of high quality. Adding the MWE information to original tokens and decompounded tokens improves only when indexing MWEs with high scores (above 0.7). Using this combination performs best; however, none of the improvements in this experiment are significant with respect to the token-only baseline (using t-test and Wilcoxon rank sum test). Inspecting the interpolated precision–recall curve (see Figure 5), we also observe that the best results

are obtained using MWE information in combination with compounds and content words.

6. Related Work

6.1 Related Work on Merging Words

The generation of MWE dictionaries has drawn much attention in the field of NLP. Early computational approaches (e.g., Justeson and Katz 1995) use POS sequences as MWE extractors.

Other approaches, relying on word frequency, statistically verify the hypothesis whether the parts of the MWE occur more often together than would be expected by chance (Evert 2005; Ramisch 2012). One of the first measures that consider context information (co-occurrences) are the C-value and the NC-value, introduced by Frantzi, Ananiadou, and Tsujii (1998). These methods first extract candidates using POS information and then compute scores based on the frequency of the MWE and the frequency of nested MWE candidates. The method described by Wermter and Hahn (2005) is based on the limited modifiability of MWEs. For this, they introduce a measure that combines frequencies of modifications of the candidate, where modifications are considered as occurrences of the candidate where a single word is replaced with a different one.

A newer method is introduced by Lossio-Ventura et al. (2014), who re-rank scores based on an extension of the C-value, which uses a POS-based probability and an inverse document frequency. Using different measures and learning a classifier that predicts the multiwordness was first proposed by Pecina (2010), who, however, restricts his experiments to two-word MWEs for the Czech language only. Korkontzelos (2010) comparatively evaluates several MWE ranking measures. The best MWE extractor reported in his work is the scorer by Nakagawa and Mori (2002, 2003), who use the un-nested frequency (called marginal frequency) of each candidate and multiply these by the geometric mean of the distinct neighbor of each word within the candidate.

Distributional semantics is mostly used to detect compositionality of MWEs (Katz and Giesbrecht 2006; Salehi, Cook, and Baldwin 2014). For this, most approaches compare the context vector of a MWE with the combined vectors based on the constituent words of the MWE. Then, the similarity between the vectors is used as the degree of compositionality. In machine translation, words are sometimes considered as multiwords if they can be translated as single term (cf. Bouamor, Semmar, and Zweigenbaum 2012; Anastasiou 2010). Although this follows the same intuition as our *uniqueness* measure described in Section 3.2.1, we do not require any bilingual corpora, but rather test if a multiword can likely be substituted for a single word.

Regarding the evaluation, mostly precision at k ($P@k$) and recall at k ($R@k$) are applied (e.g., Frantzi, Ananiadou, and Tsujii 1998; Evert 2005; Lossio-Ventura et al. 2014). Another general approach is using the AP, which is also used in IR (Thater, Dinu, and Pinkal 2009) and has also been applied by Ramisch, De Araujo, and Villavicencio (2012).

6.2 Related Work on Splitting Words

Approaches to automatic decompounding can be classified into corpus-driven approaches and supervised approaches. Corpus-driven approaches are usually informed using frequency lists (Koehn and Knight 2003), probabilistic models (Schiller 2005), parallel corpora (Koehn and Knight 2003; Macherey et al. 2011), or periphrases (i.e.,

reformulations) in large monolingual corpora (Holz and Biemann 2008). As with other NLP tasks, supervised approaches are usually superior to unsupervised approaches if sufficient training material is available. A straightforward yet effective supervised decompounding system is contained in the ASV Toolbox (Biemann et al. 2008), which uses trie-based (Morrison 1968; Witschel and Biemann 2005) datastructures for recursively splitting compounds based on training set splits. Alfonseca, Bilac, and Pharies (2008) combine several signals, including web anchor text, in an SVM-based supervised splitter. More recently, Shapiro (2016) proposes another supervised method that trains a morphology component on compounds and uses a language model and handcrafted constraints in order to split compounds. The method is evaluated on a Finnish data set.

A widely used German decompounder is JWS, which is based on word lists of compound parts as well as manually crafted blacklists and whitelists.[29] The NL Splitter uses similar technology for Dutch compound decomposition.[30] 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. Daiber et al. (2015) propose an unsupervised system using an analogy-based approach that relies on word embeddings. Ziering and van der Plas (2016) introduced an unsupervised method based on morphology that is informed by lemmatization information. Although this approach is unsupervised, it is not knowledge-free, as it is informed by a language-specific morphology component.

Decompounding is evaluated either intrinsically or in a task that benefits from it, for example, information retrieval (Monz and de Rijke 2001), machine translation (Koehn and Knight 2003; Macherey et al. 2011), or automatic speech recognition (Adda-Decker and Adda 2000; Ordelman, van Hessen, and de Jong 2003).

7. Conclusion

In this article, we have introduced fine-grained and coarse-grained tokenization methods. Whereas normal tokenization considers the separation of words and interpunctuation marks, we have introduced two methods that join multiple words that form a concept and another method for splitting words that are formed by several stems. Both methods are unsupervised and knowledge-free and only rely on distributional semantic models.

As a side note, we have evaluated two models for distributional similarity in this context, showing that the compound splitting method works slightly better with neural word2vec similarities when most of the words are also contained in the corpus used for similarity computations. For the MWE identification we obtain significantly better results when using similarities on the basis of the sparse count-based JoBimText method, which we attribute to the different characteristics of similarity neighborhoods produced by these models.

For the detection of MWE we have evaluated our method using two annotated corpora of French and English medical texts. In addition, we have demonstrated the capability of detecting MWEs for 32 languages using an automatic evaluation on Wiktionary and Wikipedia. Furthermore, in order to split words, we have proposed `SECOS` and shown its performance on five gold standard data sets for German, Dutch,

29 `https://github.com/danielnaber/jwordsplitter`.
30 `http://ilps.science.uva.nl/resources/compound-splitter-nl/`.

Afrikaans, and Finnish. We obtain state-of-the-art performance on two out of five data sets and additionally show the language independence of the method using an automatically extracted data set from Wiktionary for 14 languages. Lastly, we have shown that incorporating both coarse- and fine-grained tokenization results in performance gains for information retrieval.

8. Future Work

In future work, we want to expand the fine-grained tokenization and identify even smaller units within the compounds, which is also one of the major error classes for the compounding. Furthermore, we want to extend the compounding method to detect not only compounds but also morphemes. For the coarse-grained tokenization we want to develop methods that allow labeling the parts of MWEs. Furthermore, we propose to demonstrate the impact of our fine-grained and coarse-grained tokenization for further tasks like machine translation (Koehn and Knight 2003), question answering (Rinaldi et al. 2003; de Marneffe, Padó, and Manning 2009), and to apply it to texts of different languages and domains.

References

Abeillé, Anne and Nicolas Barrier. 2004. Enriching a French Treebank. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, pages 2233–2236, Lisbon.

Acosta, Otavio Costa, Aline Villavicencio, and Viviane Pereira Moreira. 2011. Identification and treatment of multiword expressions applied to information retrieval. In *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*, pages 101–109, Portland, OR.

Adda-Decker, Martine and Gilles Adda. 2000. Morphological decomposition for ASR in German. In *Proceedings of the Workshop on Phonetics and Phonology in ASR*, pages 129–143, Saarbrücken.

Airio, Eija. 2006. Word normalization and decompounding in mono- and bilingual IR. *Information Retrieval*, 9(3):249–271.

Alfonseca, Enrique, 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*, pages 253–256, Columbus, OH.

Anastasiou, Dimitra. 2010. *Idiom Treatment Experiments in Machine Translation*. Ph.D. thesis, Universität des Saarlandes, Saarbrücken, Germany.

Biemann, Chris, 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 2008*, pages 1760–1767, Marrakech.

Biemann, Chris and Martin Riedl. 2013. Text: Now in 2D! A framework for lexical expansion with contextual similarity. *Journal of Language Modelling*, 1(1): 55–95.

Blanc, Olivier, Matthieu Constant, and Patrick Watrin. 2007. A finite-state super-chunker. In *Proceedings of the 12th International Conference on Implementation and Application of Automata*, pages 306–308, Prague.

Bouamor, Dhouha, Nasredine Semmar, and Pierre Zweigenbaum. 2012. Identifying bilingual multi-word expressions for statistical machine translation. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, pages 674–679, Istanbul.

Cabanac, Guillaume, Gilles Hubert, Mohand Boughanem, and Claude Chrisment. 2010. Tie-breaking bias: Effect of an uncontrolled parameter on information retrieval evaluation. In *Conference on Multilingual and Multimodal Information Access Evaluation*, pages 112–123, Padua.

Daiber, Joachim, 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.

Daille, Béatrice, Éric Gaussier, and Jean-Marc Langé. 1994. Towards automatic extraction of monolingual and bilingual terminology. In *Proceedings of the 15th Conference on Computational Linguistics - Volume 1*, pages 515–521, Kyoto.

Evert, Stefan. 2005. *The Statistics of Word Cooccurrences: Word Pairs and Collocations*. Ph.D. thesis, Institut für maschinelle Sprachverarbeitung, University of Stuttgart, Germany.

Evert, Stefan. 2008. A lexicographic evaluation of German adjective-noun collocations. In *Proceedings of the LREC 2008 Workshop Towards a Shared Task for Multiword Expressions*, pages 3–6, Marrakech.

Frantzi, Katerina T., Sophia Ananiadou, and Jun-ichi Tsujii. 1998. The C-value/ NC-value method of automatic recognition for multi-word terms. In *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries*, pages 585–604, Heraklion.

Grefenstette, Gregory. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers.

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

Hassler, Marcus and Günther Fliedl. 2006. Text preparation through extended tokenization. *Data Mining VII: Data, Text and Web Mining and Their Business Applications*, 37:13–21.

Henrich, Verena and Erhard Hinrichs. 2011. Determining immediate constituents of compounds in GermaNet. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing 2011*, pages 420–426, Hissar.

Hermann, Karl Moritz and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 58–68, Baltimore, MA.

Hindle, Donald. 1990. Noun classification from predicate-argument structures. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 268–275, Pittsburgh, PA.

Holz, Florian 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.

Justeson, John S. and Slava M. Katz. 1995. Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1:9–27.

Kaplan, Ronald M. 2005. A method for tokenizing text. In M. Butt, M. Dalrymple, and T. H. King, editors. *Inquiries into Words, Constraints and Contexts*, CSLI Studies in Computational Linguistics Online, pages 55–64.

Katz, Graham and Eugenie Giesbrecht. 2006. Automatic identification of non-compositional multi-word expressions using latent semantic analysis. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, pages 12–19, Sydney.

Kim, Jin-Dong, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. GENIA corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(Suppl 1):i180–i182.

Koehn, Philipp and Kevin Knight. 2003. Empirical methods for compound splitting. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 187–193, Budapest.

Korkontzelos, Ioannis. 2010. *Unsupervised Learning of Multiword Expressions*. Ph.D. thesis, University of York, UK.

Levy, Omer and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, MA.

Lin, Dekang. 1997. Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 64–71, Madrid.

Lin, Dekang. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th International Conference on Computational Linguistics*, pages 768–774, Montreal.

Lossio-Ventura, Juan Antonio, Clement Jonquet, Mathieu Roche, and Maguelonne Teisseire. 2014. Yet another ranking function for automatic multiword term extraction. In *Proceedings of the 9th International Conference on Natural Language Processing*, pages 52–64, Warsaw.

Macherey, Klaus, 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*, pages 1395–1404, Portland, OR.

Manning, Christopher D. and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.
Marek, Torsten. 2006. Analysis of German compounds using weighted finite state transducers. Bachelor thesis, Universität Tübingen, Germany.
de Marneffe, Marie-Catherine, Sebastian Padó, and Christopher D. Manning. 2009. Multi-word expressions in textual inference: Much ado about nothing? In *Proceedings of the 2009 Workshop on Applied Textual Inference*, pages 1–9, Suntec.
Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Machine Learning*, pages 1310–1318, Scottsdale, AZ.
Milde, Benjamin, Jonas Wacker, Stefan Radomski, Max Mühlhäuser, and Chris Biemann. 2016. Ambient search: A document retrieval system for speech streams. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2082–2091, Osaka.
Monz, Christof 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*, pages 262–277, Darmstadt.
Morrison, Donald R. 1968. PATRICIA - Practical Algorithm To Retrieve Information Coded in Alphanumeric. *Journal of the ACM*, 0004-5411 15(4):514–534.
Nakagawa, Hiroshi and Tatsunori Mori. 2002. A simple but powerful automatic term extraction method. In *Proceedings of COLING-02 on COMPUTERM 2002: Second International Workshop on Computational Terminology - Volume 14*, pages 1–7, Taipei.
Nakagawa, Hiroshi and Tatsunori Mori. 2003. Automatic term recognition based on statistics of compound nouns and their components. *Terminology*, 9(2):201–219.
Ordelman, Roeland, 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*, pages 225–228, Geneva.
Pecina, Pavel. 2010. Lexical association measures and collocation extraction. *Language Resources and Evaluation*, 44:137–158.
Ramisch, Carlos. 2012. *A Generic and Open Framework for Multiword Expressions Treatment: From Acquisition to Applications*. Ph.D. thesis, Universidade Federal Do Rio Grande do Sul, Brazil.
Ramisch, Carlos, Vitor De Araujo, and Aline Villavicencio. 2012. A broad evaluation of techniques for automatic acquisition of multiword expressions. In *Proceedings of the Student Research Workshop of the 50th Meeting of the Association for Computational Linguistics*, pages 1–6, Jeju Island.
Richter, Matthias, 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*, pages 68–73, Ljubljana.
Riedl, Martin. 2016. *Unsupervised Methods for Learning and Using Semantics of Natural Language*. Ph.D. thesis, Technische Universität Darmstadt, Germany.
Riedl, Martin and Chris Biemann. 2013. Scaling to large^3 data: An efficient and effective method to compute distributional thesauri. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 884–890, Seattle, WA.
Riedl, Martin and Chris Biemann. 2015. A single word is not enough: Ranking multiword expressions using distributional semantics. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP 2015, pages 2430–2440, Lisbon.
Riedl, Martin and Chris Biemann. 2016. Unsupervised compound splitting with distributional semantics rivals supervised methods. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 617–622, San Diego, CA.
Riedl, Martin and Chris Biemann. 2017. There's no 'count or predict' but task-based selection for distributional models. In *Proceedings of the 12th International Conference on Computational Semantics*, pages 264–272, Montpellier.
Rinaldi, Fabio, James Dowdall, Kaarel Kaljurand, Michael Hess, and Diego Mollá. 2003. Exploiting paraphrases in a question answering system. In *Proceedings of the Second International Workshop on Paraphrasing - Volume 16*, pages 25–32, Sapporo.

Sag, Ivan Andrew, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2001. Multiword expressions: A pain in the neck for NLP. In *Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics*, pages 1–15, Mexico City.

Salehi, Bahar, Paul Cook, and Timothy Baldwin. 2014. Using distributional similarity of multi-way translations to predict multiword expression compositionality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 472–481, Gothenburg.

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

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

Schnabel, Tobias, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, Lisbon.

Schone, Patrick and Daniel Jurafsky. 2001. Is knowledge-free induction of multiword unit dictionary headwords a solved problem? In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 100–108, Pittsburgh, PA.

Seddah, Djamé, Marie Candito, Benoit Crabbé, and Enrique Henestroza Anguiano. 2012. Ubiquitous usage of a broad coverage French corpus: Processing the Est Républicain corpus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*, pages 3249–3254, Istanbul.

Seddah, Djamé, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho D. Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola Galletebeitia, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiórkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villomente de la Clergerie. 2013. Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 146–182, Seattle, WA.

Shapiro, Naomi T. 2016. Splitting compounds with n-grams. In *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, pages 630–640, Osaka.

Shapiro, Naomi T., Joshua Falk, Kati Kiiskinen, and Arto Anttila. 2017. FinnSyll 2.0.0: A Finnish syllabifier. Technical report, Stanford University, https://pypi.python.org/pypi/FinnSyll.

da Silva, Edson and Renato Souza. 2012. Information retrieval system using multiwords expressions (MWE) as descriptors. *Journal of Information Systems and Technology Management*, 9(2):213–234.

Taalkommissie. 2011. *Taalkommissiekorpus 1.1*, Taalkommissie van die Suid-Afrikaanse Akademie vir Wetenskap en Kuns. Centre for Text Technology (CTexT), North-West University, Potchefstroom, South Africa.

Thater, Stefan, Georgiana Dinu, and Manfred Pinkal. 2009. Ranking paraphrases in context. In *Proceedings of the 2009 Workshop on Applied Textual Inference in conjunction with the ACL '09*, pages 44–47, Suntec.

Trim, Craig. 2013. The art of tokenization. Technical Report, IBM Developer Works. `https://www.ibm.com/developerworks/community/blogs/nlp/entry/tokenization?lang=en_us`.

Voorhees, Ellen M. 2005. The TREC robust retrieval track. *SIGIR Forum*, 39(1):11–20.

Webster, Jonathan J. and Chunyu Kit. 1992. Tokenization as the initial phase in NLP. In *Proceedings of the 14th Conference on Computational Linguistics*, pages 1106–1110, Nantes.

Wermter, Joachim and Udo Hahn. 2005. Effective grading of termhood in biomedical literature. In *Annual AMIA Symposium Proceedings*, pages 809–813, Washington, DC.

Witschel, Hans Friedrich and Chris Biemann. 2005. Rigorous dimensionality reduction through linguistically motivated feature selection for text categorization. In *Proceedings of the 15th Nordic Conference of Computational Linguistics*, NODALIDA 2005, pages 210–217, Joensuu.

van Zaanen, Menno, 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, pages 1056–1062, Reykjavík.

Ziering, Patrick and Lonneke van der Plas. 2016. Towards unsupervised and language-independent compound splitting using inflectional morphological transformations. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–653, San Diego, CA.