Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 4410–4417
Marseille, 11–16 May 2020

A Joint Approach to Compound Splitting and Idiomatic Compound Detection

Irina Krotova[1], **Sergey Aksenov**[2], **Ekaterina Artemova**[3]
[1] MobileTeleSystems
[2] Sberbank of Russia
[3] National Research University Higher School of Economics
Moscow, Russia
ivkrotov@mts.ru, aksenov.s.an@sberbank.ru, eartemova@hse.ru

Abstract

Applications such as machine translation, speech recognition, and information retrieval require efficient handling of noun compounds as they are one of the possible sources for out-of-vocabulary (OOV) words. In-depth processing of noun compounds requires not only splitting them into smaller components (or even roots) but also the identification of instances that should remain unsplitted as they are of idiomatic nature. We develop a two-fold deep learning-based approach of noun compound splitting and idiomatic compound detection for the German language that we train using a newly collected corpus of annotated German compounds. Our neural noun compound splitter operates on a sub-word level and outperforms the current state of the art by about 5%.

Keywords: compound splitting, idiomatic compounds detection, word embeddings, sequence models, sub-word models

1. Introduction

Compounding is a common word-formation process in Germanic languages (e.g. German, Dutch, Swedish) that poses challenges for many natural language processing applications, such as machine translation (Daiber et al., 2015; Fritzinger and Fraser, 2010; Popović et al., 2006), speech recognition (Larson et al., 2000), information retrieval (Alfonseca et al., 2008; Monz and De Rijke, 2001) and coreference resolution (Tuggener, 2016).

Difficulties are primarily caused by high productivity and low corpus frequency of compounds, which increases the vocabulary size and leads to sparse data problems. According to (Baroni et al., 2002), almost half (47%) of the word types in a 28-million German newswire corpus are compounds. At the same time, 83% of them are not frequent words or productively formed hapax legomena and have a corpus frequency of 5 or lower.

German compounds are not orthographically separated by hyphen or whitespaces and are mostly written as a single word. For example, the equivalent of the German word *Arbeitstag* is written in English as two-word compound "working day".

This leads to more out-of-vocabulary (OOV) words, which can not be listed in a lexicon and translated, but at the same time, may be split and represented as at least two components or roots. The right-most component is a noun, the head of the compound. The leftmost component is the modifier and can be a noun, verb, adjective, number, or a preposition.

The decomposition of a complex compound or compound splitting is a well defined, but yet not a simple task. Two parts of the compound are not always concatenated as in *Tischtennis* ("table tennis") or *Wolkenkratzer* ("skyscraper"). Compound parts can undergo morphological modifications from the normal form, such as addition (*Arbeitszeit* ("working time")) or truncation of letters (*Kirchhof* ("church garden")), umlaut (*Bücherregal* ("bookshelf")) or a combination of modifications. All these morphological changes need to be considered to correctly split a compound into two lemmas.

The most common way to preprocess German compounds is to split them into components before training and translation (Stymne, 2008). In the majority of cases, noun+noun compound nouns are realized by a determiner or adpositional phrase following the head of the compound (*Haustür - Tür des Hauses* ("house door", "front door"), *Gartenschlauch - Schlauch für den Garten* ("garden hose", "hosepipe").

From a linguistic point of view, this refers to "Frege's principle", the idea of formal semantics. According to this principle, the meaning of a sentence can be deducted from the meaning of its constituents (Kiefer, 2000). This principle can be extended to lower syntax levels, such as a phrase or a word. A compound can be tackled from this perspective because it consists of two independent nouns.

It is generally recognized that certain language phenomena, such as idioms, figures of speech (metaphors), expressions that are subjects to pragmatic interpretation, can not be interpreted in a strictly compositional way (Downing, 1977). The illustration of this difference is a pair of German expressions *Altmaterial* and *altes Material*: the compound means "recovered material" whereas *altes Material* describes the material as being old, where the certain meaning of the word "old" depends on the context.

One of the most significant and detailed works on the relationship between non-literal meaning and compositionality was written by Jan G. Kooij, see (Kooij, 1968). He distinguishes between idiomatic and non-idiomatic compounds. The meaning of the idiomatic compound cannot be explained from the constituents and the structure (consider, for example, "egghead" and "egg-shell"). He claims also that some non-idiomatic compounds have meaning specialization: for example, the Dutch word *huisdeur* ("house-door') consists of two words, *huis* ("house") and *deur* ("door"), which are two independent words with the same meaning. However the word *huisdeur* does mean not any door in the house, and rather it refers only to the front door. He also makes an important observation, that the boundary

between idiomatic and non-idiomatic compounds is not a yes-no question, but a matter of degree. A similar point of view is supported by (Goatly, 1997), who also emphasizes the controversy of the strict separation of literal and non-literal language usage. According to this work, they are only more or less tied to conventional meaning.

In this paper, we explore computational approaches to idiomatic meaning modeling and identification. We explore only German compounds and suggest a two-fold approach to idiomatic and literal compounds identification. The first step is to split a compound into constituents. The second step is to evaluate how likely it is that the compound is of idiomatic nature. The first step is treated as a sequence labeling task performed on the sub-word level. For each sub-word, a label, which indicates whether a split should be introduced after this very sub-word, is assigned. The second step targets at detection of idiomatic compounds and operates on the embeddings of the compound and its components. The compound is considered idiomatic, if the lexical meaning of the compound cannot be composed of the lexical meaning of its components, i.e., it is impossible to derive the embedding of the compound from the embeddings of its components. For these means, we adopted a simple yet efficient approach for compositionality detection from (Jana et al., 2019).

The contributions of this paper are as follows: we propose a new German compound splitting method, based on neural sequence models. We introduce a new dataset for the task of non-literal meaning identification and establish a baseline for this task.

2. Related work

German Compound Splitting

Methods for automatic splitting of word compounds has been studied by several research groups. Early approaches used dictionary-based methods as a source for full morphological analysis. (Koehn, 2003) use corpora statistics and present a frequency-based approach to German compound splitting. Compound parts are identified by word frequencies and different possible splits are ranked according to the geometric mean of subword frequencies. Word modifications, such as the deletion of characters and linking elements *"-s"* and *"-'es"* are allowed. (Stymne, 2008) extends this algorithm by adding the 20 most frequent morphological transformations. (Tuggener, 2016) relies on character n-grams and their distribution. (Weller-Di Marco, 2017) combines this approach with linguistic heuristics and focuses on alignment. Other researchers use unsupervised approaches to compound splitting. (Macherey et al., 2011) presents a method that does not rely on any handcrafted rules for transitional elements or morphological operations. This algorithm uses a bilingual corpus and learns morphological operations from it. (Ziering and van der Plas, 2016) do not use parallel corpora, but rather learn "morphological operation patterns" from lemmatized monolingual corpora. (Riedl and Biemann, 2016) explore distributional semantics; the method is based on the assumption that the constituents of a compound are semantically similar and identify the valid splitting point. They utilize a distributional thesaurus and a set of "atomic word units" extracted from corpus data. (Schulte im Walde et al., 2016) detect the semantic relation between the constituents of the compounds.

2.1. Word Segmentation in other languages

The problem of word segmentation has received much attention in Chinese. Since (Xue and Shen, 2003) Chinese word segmentation is addressed as a character labeling task: each character of the input sequence is labeled with one of the four labels $\mathcal{L} = \{B, M, E, S\}$, which stand for character in Begin, Middle or End of the word or Single character word. (Xue and Shen, 2003) uses a maximum entropy tagger to tag each character independently. This approach was extended in (Peng et al., 2004) to the sequence modeling task, and linear conditional random fields were used to attempt it and receive state of the art results. A neural approach to Chinese segmentation mainly uses various architectures of character level recurrent neural networks (Cai and Zhao, 2016; Zhang et al., 2018; Cai et al., 2017) and very deep convolutional networks (Sun et al., 2017). Same architectures are used for dialectal Arabic segmentation (Samih et al., 2017).

The English word formations leads to lesser importance of the word segmentation problem. However a similar problem rises when processing social media data, hashtags in particular. As it was shown by (Berardi et al., 2011) hashtag segmentation for TREC microblog track 2011 (Soboroff et al., 2012) improves the quality of information retrieval, while (Bansal et al., 2015) shows that hashtag segmentation improves linking of entities extracted from tweets to a knowledge base. Both (Berardi et al., 2011; Bansal et al., 2015) use Viterbi-like algorithm for hashtag segmentation. Following the idea of scoring segmentation candidates, (Reuter et al., 2016) introduces other scoring functions, which include a bigram model (2GM) and a Maximum Unknown Matching (MUM), which is adjustable to unseen words.

A similar problem may arise outside of natural language processing scope. (Markovtsev et al., 2018) subjected source code identifiers to analysis and use LSTM-derived splitters to extract distinct identifiers from the large chunks of code.

Compositionalty Evaluation

(Hätty and im Walde, 2018) proposes a combined approach for automatic term identification and investigating the understandability of terms by defining fine-grained classes of termhood and framing a classification task. They selected 400 German compounds to annotate for termhood in the domain of cooking. Next they predicted the compound classes in three steps: compound splitting, representation of compound and its components in the feauture space and applying a neural network classifier. To split compounds CharSplit (Tuggener, 2016), CompoST (Cap, 2014) and the Simple Compound Splitter (Weller-Di Marco, 2017) were combined. The feature description includes word embeddings, frequency and productivity of the components. The best classifier model achieved an 80% improvement on F1-score in comparison to the best baseline model.

(Horbach et al., 2016) presented an annotation study on a representative dataset of literal and idiomatic uses of infinitive-verb compounds in German newspaper and jour-

nal texts. They have collected a corpus of 6,000 instances of 6 representative infinitive-verb compounds in German, that was annotated for idiomaticity by expert lexicographers. A Naive Bayes classifier uses context features to classify instances of the verb compounds as either idiomatic or literal with an accuracy of 85%.

3. Dataset

We use the dataset discussed in (Henrich and Hinrichs, 2011), GermaNet v.14.0 (2019). This is a list of 82 309 split nominal compounds extracted from a German wordnet GermaNet (Henrich and Hinrichs, 2010).

The format of the compound splits is one compound per line, where the compound itself, its modifier, and the head are listed. Compound splitting is supported by automatic algorithms, combined from several compound splitters. Then all automatically split compounds are manually post-corrected and enriched with relevant properties. All modifiers in the dataset are lemmatized and in the case the modifier is ambiguous, both possibilities are specified (*Laufschuhe* ("running shoes"): *lauf-* ("to run") (en) [verb] and *(der) Lauf* ("run") [noun]).

Compounds in the dataset include compounds with different properties of head and/or modifier. The dataset includes such specific compound parts like abbreviations (*SIM-Karte* ("SIM card")), affixoids (*Grundfrage* ("basic question") - *grund* ("reason, cause")(affixoid) *Frage* ("question")), foreign words (*Energydrink* ("energy drink"), confixes (*Milligramm* ("milligram") - *milli* ("milli") (confix) *Gramm* ("gram")), opaque morphemes, whose meaning is not transparent without considering its etymology (*Himbeere*("raspberry"), *Lebkuchen*("gingerbread")), proper names (*Hubbleteleskop* ("Hubble telescope")), virtual word forms, which do not exist in the isolation (*Einflussnahme* ("influence"), Fragesteller ("questioner")) and word groups (*Nacht-und-Nebel-Aktion* ("cloak-and-dagger operation"), *Pro-Kopf-Einkommen* ("per capita income")). As a result of the variety of compound components, the task is as close as possible to real-world challenges in machine translation of compound nouns.

The amount of the unique modifiers and head in the original dataset is much lower than the number of compounds. There are 12724 unique modifiers and 9249 unique compound heads in the dataset. Almost half of modifiers (6118) and a large part of compound heads (3752) are hapax legomenon and occur only once in the dataset.

3.1. Data Preprocessing and Annotation

For the task of idiomatic compounds detection, we present the dataset of idiomatic and literal uses of German compound nouns components, based on GermaNet data. Our method includes computing word embeddings for compound nouns and their components. As the performance of word embedding degrades at low-frequent words, we limited the original dataset, namely GermaNet v.14.0 (2019), by word frequencies based on data from the DWDS corpus, constructed at the Berlin-Brandenburg Academy of Sciences (BBAW) (Klein and Geyken, 2010). We produced the compound frequencies list for the Reference and Newspaper Corpora 1990 through 2019 and selected the first 5000 entries for annotation.

After that, we added the definitions from Duden dictionary (Duden Universalwörterbuch, 2006) to the list of compound nouns. Since we selected the most frequent words from the GermaNet compound list, most of them had definitions in Duden dictionary (Duden Universalwörterbuch, 2006).

For many classification tasks, such as word sense disambiguation or named-entity recognition, there is general agreement on a standard set of categories. For the compound-related tasks, on the other hand, although numerous annotation schemes have been proposed, yet there is still little agreement and no standard categories.

Our annotation scheme was designed based on the principle of compositionality, described above. From this perspective, it is possible to give a compound definition using its constituents only if a compound is non-idiomatic. If a compound is not idiomatic and can not be literally translated using its constituents after splitting, the definition does not contain compound parts. See Table 1 for examples of compounds and their definitions. According to the proposed annotation scheme the compound *Arbeitstag* is compositional, as its definition contains both constituents and the compound *Schildkröte* is not compositional.

Arbeitstag	**Tag**, an dem [berufliche] **Arbeit** geleistet wird oder zu leisten ist.	Working day: the day, on which the [professional] work is done or needs to be done
Schildkröte	(besonders in Tropen und Subtropen) auf dem Land und im Wasser lebendes, sich an Land sehr schwerfällig bewegendes Tier mit Bauch- und Rückenpanzer, in den Kopf, Beine und Schwanz eingezogen werden können.	Turtle: (particularly in the tropics and subtropics) animal, which lives on land or in the water, moves slowly on land and has a shell, where its head, legs and tail can be retracted into.

Table 1: Examples of German compounds and definitions. The constituents of compounds are bolded.

On the one hand, the proposed principle allows us to automatically annotate data according to the definitions from Duden dictionary (Duden Universalwörterbuch, 2006). On the other hand, it makes the manual annotation task less challenging, because it is easier to distinguish between idiomatic and literal meaning of each constituent, than of the whole compound.

All compounds[1] were automatically annotated and manually post-corrected according to following schema:

[1]The dataset is available: `https://github.com/PragmaticsLab/kompositionsfreudigkeit`

0 : both of the components can be used in the compound definition, non-idiomatic compound.

1 : the first component is idiomatic; the second is non-idiomatic.

2 : the first component is non-idiomatic; the second is idiomatic.

3 : both components are idiomatic.

Each compound is annotated with a value ranging from 0 to 3, which stands for the category of compound so that 0 means that compound is non-idiomatic and 3 means that the compound is idiomatic. Categories 1 and 2 can be considered borderline and partially idiomatic. The sample of the annotated dataset can be found in Table 2.

Freq	Compound	Modifier	Head	Category
65883	*Jahrhundert* ("century")	*Jahr*	*Hundert*	0
171137	*Freitag* ("friday")	*frei*	*Tag*	1
33681	*Zeitpunkt* ("time moment")	*Zeit*	*Punkt*	2
13519	*Lebensmittel* ("foods")	*Leben*	*Mittel*	3

Table 2: Examples of annotated compounds

4. Problem Formulation and Models

4.1. Compound Splitting Baselines

As a baseline splitters we adopted CharSplit (Tuggener, 2016) and `SECOS` (Riedl and Biemann, 2016) along with open source reference implementation of both splitters. CharSplit calculates the probabilities of n-grams to occur at the word's beginning, and middle and calculates a splitting score at each position in a compound word. `SECOS` leverages information from the distributional thesaurus to rank possible candidate splits.

4.2. Compound Splitting Models

Compound splitting is treated as a sequence labeling task. We develop a set of RNN-derived models, which leverage different types of input representations and hidden units. For an architecture overview, see Figure 1.

Each model is a binary classifier based on a sub-word level bidirectional recurrent neural network. The classifier determines for each sub-word, whether it is in the split position or not. Each subword is assigned with either 0 (there is no split after this sub-word), or 1 (there is a split after this sub-word). The concatenation of the hidden states of the forward and backward RNN forms a feature vector for each character that is then fed to a fully connected layer. The fully connected layer has a *softmax* activation function that computes the probability of a split for each sub-word.

We consider the following design choices:

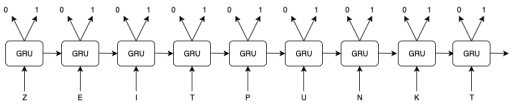

Figure 1: Our architecture

1. **sub-word definition**: a sub-word can be either a character, or a BPE sub-word unit (Heinzerling and Strube, 2018).

2. **RNN architecture**: we compare vanilla RNN to GRU and LSTM (Hochreiter and Schmidhuber, 1997) architectures (we used keras[2] implementation of each architecture)

3. **whether the embeddings are trainable**[3]: the character embeddings are initialized randomly and thus are always learned as model parametres. The adopted pretrained BPE embeddings can be either learned as model parametrs or can be kept non-trainable and thus remain unchanged.

BPE tokenization has become a de-facto standard way for processing sub-words in the era of BERT (Devlin et al., 2019) and BERT-like models. Thus we decided to draw a comparison between BPE tokenization and simpler character-level models, frequently used for segmentation in Chinese (Xue and Shen, 2003) or Arabic (Samih et al., 2017). These models process input words in a character by character way so that each character is treated as a single sub-word.

The size of BPE vocabulary is one of the architecture choices. We choose from vocabulary size equal to 10^3 and 10^4. We choose RNN, GRU and LSTM units to be 256-dimensional. All models were trained for 30 epochs with Adam optimizer with the default learning rate equal to 10^{-3}.

4.3. Idiomatic Compounds Detection

We establish new baselines for the task idiomatic compound detection by adopting methods of compositionality detection (Jana et al., 2019).

Idiomatic compounds detection is considered as a binary classification task, where one class stands for non-idiomatic compounds (labeled with 0) and the other – for borderline idiomatic compounds (labels 1, 2, and 3). We do not distinguish between different degrees of idiomaticity and consider a compound to be either idiomatic or not. We simply train various supervised machine learning methods on vector representations of a compound and its components. We use the following classification algorithms: logistic regression (LogReg) and gradient boosting (XGBoost). For feature representation, we use a concatenation of a compound embedding with embeddings of compound components acquired from various distributional semantics models (DSMs), such as `word2vec` (Mikolov et al., 2013) or `fastText` (Joulin et al., 2017). To obtain the components

[2]http://keras.io

[3]this option corresponds to the `trainable` argument of the `Embedding` layer

of a compound, we use either the source gold standard split or our own splitter, based on Char-GRU, as it significantly outperforms other splitters.
We used two pre-trained DSMs:

- `word2vec` model pre-trained on Wikipedia. We use the Word2Vec Skip-gram model with a window size of 5 and a minimum word frequency of 10 to generate a 300-dimensional vector for each word.
- `fastText` model pre-trained on Wikipedia. Similarly to `word2vec` model, the vector for each word has 300 dimensions.

The feature vector for each compound has 900 dimensions. The core difference between pre-trained DSMs is based on the way OOV words are treated. While `word2vec` suggests using a special embedding for unknown words (`[unk]`), `fastText` us capable to infer an embedding for any word based on n-grams.
To detect whether the compound word is idiomatic or not, we used two classification algorithms:

- Logistic Regression from `scikit-learn`[4] with regularization strength parameter `C`=1.
- Gradient Boosting (XGBoost[5]) over decision trees with 200 estimators. Minimum size of a leaf in each tree is 25. Weights for classes `0` and `1` are 1 and 10 respectively.

Model	**Embedding layer**	**Accuracy**
CharSplit (Baseline)		0.879
`SECOS` (Baseline)		0.914
Char-level models		
vanilla RNN	trainable	0.915 ± 0.002
GRU	trainable	$\mathbf{0.956} \pm 0.002$
biLSTM	trainable	0.944 ± 0.003
BPE-level models		
BPE vocab size = 10^3		
vanilla RNN	non-trainable	0.726 ± 0.003
GRU	non-trainable	0.746 ± 0.003
biLSTM	non-trainable	0.734 ± 0.005
vanilla RNN	trainable	0.731 ± 0.004
GRU	trainable	0.759 ± 0.003
biLSTM	trainable	0.752 ± 0.004
BPE vocab size = 10^4		
vanilla RNN	non-trainable	0.788 ± 0.004
GRU	non-trainable	0.802 ± 0.002
biLSTM	non-trainable	0.810 ± 0.004
vanilla RNN	trainable	0.779 ± 0.004
GRU	trainable	0.823 ± 0.003
biLSTM	trainable	0.825 ± 0.005

Table 3: Compound splitters performance

5. Results and discussion

5.1. Compound Splitting

The results of the compound splitting experiment are presented in Table 3. Mean accuracy values with standard deviation for 30 runs for each model are reported. It can be seen that all character-level models perform better than any of the BPE-level models. Character-level models learn orthographic patterns only, as they are not provided with any semantic input. Hence they are better aimed for constituent boundary detection. There is no significant difference, whether the BPE-embeddings are trainable or not. However, the size of BPE vocabulary matters: when trained with a larger vocabulary, the model performs better though it does not make sense to use even larger BPE vocabulary since it would include whole compounds as a single token. Vanilla RNN architectures are always outperformed by LSTM and GRU.

5.2. Idiomatic Compounds Detection

The results of the idiomatic compound detection experiment are presented in Table 4. A simple model that always predicts that the word is idiomatic is referred to as *Dummy model*. It can be seen that classification models with pre-trained word embeddings perform significantly better than the *Dummy model*.
We used two compound splitters for the task. First, we used the gold standard split from GermaNet. Second, we use our own splitter, which, according to previous experiments, happens to outperform other well-known splitters.
Among two DSMs under consideration, `fastText`, seems to be a better source for word embeddings. As `fastText` model is capable of inferring a word embedding for out of vocabulary words, it is less sensitive to splitter errors.
XGBoost and logistic regression perform almost the same, with XGBoost producing slightly higher scores. Due to the high complexity of the task, the results of both classifiers are moderate. Though when compared to the *Dummy model*, we can stay that the classifiers are capable of detecting idiomatic compounds, which means that the task itself is cab be approached by means of machine learning and distributional semantics.

Model	**F_1-score**
Dummy model	0.21
Gold Split + `word2vec` + XGBoost	0.567
Gold Split + `word2vec` + LogReg	0.579
Gold Split + `fastText` + XGBoost	**0.584**
Gold Split + `fastText` + LogReg	0.577
Char-GRU Split + `word2vec` + XGBoost	0.545
Char-GRU Split + `word2vec` + LogReg	0.521
Char-GRU Split + `fastText` + XGBoost	**0.554**
Char-GRU Split + `fastText` + LogReg	0.541

Table 4: Performance of idiomatic compounds detection

[4] `https://scikit-learn.org/stable/`
[5] `https://xgboost.readthedocs.io`

6. Error Analysis

6.1. Compound splitting

In order to understand the errors of methods we compared, we analyzed the compounds that have been split incorrectly. CharSplit often fails when encountering the linking element like the *Fugen-s* or plural marker *-(e)n-* (*Gruppe-nerste* instead of *Gruppen-erste* ("top of the group"), *Namensgebung* instead of *Namens-gebung* ("naming"). They are often attached to the head component of the compound noun. The second problem is splitting compounds with frequent suffixes: suffixes like *'-ung'* or *'-schaft'* are often recognized as a head noun (*Grenzverschieb-ung* instead of *Grenz-verschiebung* ("shifting of boundaries")).

SECOS works in a different way and returns all the possible splitting boundaries of the compound (like *Bundesfinanz-ministerium* ("Federal Ministry of Finance")). The most frequent errors (55% of all errors) are wrong splits of the compounds, where the modifier or both parts are compounds too (e.g. *Todeszeit-punkt* ("time of death") instead of *Todes-zeitpunkt*, *Arbeitszeit-raum* ("working period") instead of *Arbeits-zeitraum*, *Süßwasserzier-fisch* ("freshwater") instead of *Süßwasser-zierfisch*).

The second most frequent type of SECOS errors are those compounds, where the modifier starts with a character or pair of characters ("er", "s", "en"), which are often used as a transitional element by compounds building (e.g. *Trags-chrauber* ("autogyro") instead of *Trag-schrauber*, *Gasten-gagement* ("guest engagement) instead of *Gastengagement*, *Norden-gland* ("nothern England") instead of *Nord-england*).

More than one third (36%) of the remaining errors are such compounds where at least one of the subword roots has Latin, Greek or English origin (e.g. *Nitrogly-zerin* ("nitroglycerin") instead of *Nitro-glyzerin*, *Lymp-hödem* ("lymphedema") instead of *Lymph-ödem*). Most of these words are scientific terms or English loan words. These roots are not frequent compared to compound parts of German origin and are not widely represented in the GermaNet data. See Table 5 for examples of some compound parts of different origins and their absolute frequencies.

Component	Absolute frequency
Tag ("day")	311
Land ("country")	552
Sport ("sport")	297
Lymph ("lymphe")	8
Ödem ("edema")	4
Nitro ("nitro")	4
Glyzerin ("glycerin")	1

Table 5: Examples of compound parts and their absolute frequencies

6.2. Idiomatic Compounds Detection

The majority of the errors (413 of 477) are the non-idiomatic compounds labeled as idiomatic. The most frequent compound components of the wrong classified words can be found in Table 6.

Bund	modifier	15
Regierung	modifier	9
Staat	modifier	8
Wirtschaft	modifier	7
groß	modifier	6
Wahl	modifier	6
Chef	head	5
Verband	head	5
Rat	head	5

Table 6: Example of erroneous idiomatic compounds Detection

These components are inactive metaphors, which idiomatic meaning is difficult to distinguish because of its frequency. Most of them are from the domains of politics, economics, and law on a daily basis. Most likely these compounds are challenging even for human annotators. For example, compounds with modifiers *Bund-* ("national", "federal"), *Regierung* ("government") and *Staat* ("state", "country") were labeled like idiomatic.

7. Conclusion

In this paper, we present a two-stage approach to compound splitting and idiomatic compound detection in German. Our neural compound splitter is based on character-level recurrent neural networks. We outperform two well-known methods, CharSplit, and SECOS. To detect compounds, which should not be split, as they are of idiomatic nature, we exploit a common technique for compositionality detection.

The suggested approach to idiomatic compound detection in its present state presents more of a proof of concept nature. It clearnly benefits from the semantic information encoded in the word embeddings though there is enough space for improvement. One of the possible directions of the future work is to use other word embedding models, that encode not only distributional but also structural features, such as Poincare embeddings, or contextual embedding models, such as ELMo or BERT.

8. Acknowledgements

Ekaterina Artemova was supported by the framework of the HSE University Basic Research Program and Russian Academic Excellence Project "5-100".

9. Bibliographical References

Alfonseca, E., Bilac, S., and Pharies, S. (2008). Decompounding query keywords from compounding languages. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, HLT-Short '08, pages 253–256, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bansal, P., Jain, S., and Varma, V. (2015). Towards semantic retrieval of hashtags in microblogs. In *Proceedings of the 24th International Conference on World Wide Web*, pages 7–8.

Baroni, M., Matiasek, J., and Trost, H. (2002). Predicting the components of german nominal compounds. In *ECAI*, pages 470–474.

Berardi, G., Esuli, A., Marcheggiani, D., and Sebastiani, F. (2011). Isti@ trec microblog track 2011: Exploring the use of hashtag segmentation and text quality ranking. In *TREC*.

Cai, D. and Zhao, H. (2016). Neural word segmentation learning for chinese. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 409–420.

Cai, D., Zhao, H., Zhang, Z., Xin, Y., Wu, Y., and Huang, F. (2017). Fast and accurate neural word segmentation for chinese. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 608–615.

Cap, F. (2014). Morphological processing of compounds for statistical machine translation.

Daiber, J., Quiroz, L., Wechsler, R., and Frank, S. (2015). Splitting compounds by semantic analogy. *CoRR*, abs/1509.04473.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Downing, P. (1977). On the creation and use of english compound nouns. *Language*, pages 810–842.

Duden Universalwörterbuch, D. (2006). Duden. *Deutsches Universalwörterbuch. CD-ROM. Hrsg. vd Dudenredaktion. Mannheim/Leipzig/Wien/Zürich: Dudenverlag*.

Fritzinger, F. and Fraser, A. (2010). How to avoid burning ducks: Combining linguistic analysis and corpus statistics for german compound processing. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 224–234, Stroudsburg, PA, USA. Association for Computational Linguistics.

Goatly, A. (1997). *The language of metaphors*. Routledge.

Hätty, A. and im Walde, S. S. (2018). Fine-grained termhood prediction for german compound terms using neural networks. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 62–73.

Heinzerling, B. and Strube, M. (2018). Bpemb: Tokenization-free pre-trained subword embeddings in 275 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Henrich, V. and Hinrichs, E. (2010). GernEdiT: A graphical tool for GermaNet development. In *Proceedings of the ACL 2010 System Demonstrations*, pages 19–24, Uppsala, Sweden, July. Association for Computational Linguistics.

Henrich, V. and Hinrichs, E. (2011). Determining immediate constituents of compounds in germanet. In *Proceedings of the international conference recent advances in natural language processing 2011*, pages 420–426.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Horbach, A., Hensler, A., Krome, S., Prange, J., Scholze-Stubenrecht, W., Steffen, D., Thater, S., Wellner, C., and Pinkal, M. (2016). A corpus of literal and idiomatic uses of german infinitive-verb compounds. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 836–841.

Jana, A., Puzyrev, D., Panchenko, A., Goyal, P., Biemann, C., and Mukherjee, A. (2019). On the compositionality prediction of noun phrases using poincaré embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3263–3274, Florence, Italy, July. Association for Computational Linguistics.

Joulin, A., Grave, E., and Mikolov, P. B. T. (2017). Bag of tricks for efficient text classification. *EACL 2017*, page 427.

Kiefer, F. (2000). Jelentéselmélet, corvina.

Klein, W. and Geyken, A. (2010). Das digitale wörterbuch der deutschen sprache (dwds). In *Lexicographica: International annual for lexicography*, pages 79–96. De Gruyter.

Koehn, P. (2003). Empirical methods for compound splitting. In *Proceedings of EACL, 2003*, pages 187–193.

Kooij, J. G. (1968). Compounds and idioms. *Lingua*, 21:250–268.

Larson, M., Willett, D., Köhler, J., and Rigoll, G. (2000). Compound splitting and lexical unit recombination for improved performance of a speech recognition system for german parliamentary speeches. In *Sixth International Conference on Spoken Language Processing*.

Macherey, K., Dai, A. M., Talbot, D., Popat, A. C., and Och, F. (2011). Language-independent compound splitting with morphological operations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1395–1404, Stroudsburg, PA, USA. Association for Computational Linguistics.

Markovtsev, V., Long, W., Bulychev, E., Keramitas, R., Slavnov, K., and Markowski, G. (2018). Splitting source code identifiers using bidirectional lstm recurrent neural network. *arXiv preprint arXiv:1805.11651*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Monz, C. and De Rijke, M. (2001). Shallow morphological analysis in monolingual information retrieval for dutch, german, and italian. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 262–277. Springer.

Peng, F., Feng, F., and McCallum, A. (2004). Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th international con-*

ference on Computational Linguistics, page 562. Association for Computational Linguistics.

Popović, M., de Gispert, A., Gupta, D., Lambert, P., Ney, H., Mariño, J. B., Federico, M., and Banchs, R. (2006). Morpho-syntactic information for automatic error analysis of statistical machine translation output. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 1–6, New York City, June. Association for Computational Linguistics.

Reuter, J., Pereira-Martins, J., and Kalita, J. (2016). Segmenting twitter hashtags. *Intl. J. on Natural Lang. Computing*, 5(4).

Riedl, M. and Biemann, C. (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, California, June. Association for Computational Linguistics.

Samih, Y., Attia, M., Eldesouki, M., Abdelali, A., Mubarak, H., Kallmeyer, L., and Darwish, K. (2017). A neural architecture for dialectal arabic segmentation. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 46–54.

Schulte im Walde, S. S., Hätty, A., Bott, S., and Khvtisavrishvili, N. (2016). Ghost-nn: A representative gold standard of german noun-noun compounds. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2285–2292.

Soboroff, I., Ounis, I., Macdonald, C., and Lin, J. J. (2012). Overview of the trec-2012 microblog track. In *TREC*, volume 2012, page 20.

Stymne, S. (2008). German compounds in factored statistical machine translation. In *International Conference on Natural Language Processing*, pages 464–475. Springer.

Sun, Z., Shen, G., and Deng, Z. (2017). A gap-based framework for chinese word segmentation via very deep convolutional networks. *arXiv preprint arXiv:1712.09509*.

Tuggener, D. (2016). *Incremental coreference resolution for German*. Ph.D. thesis, Universität Zürich.

Weller-Di Marco, M. (2017). Simple compound splitting for German. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 161–166, Valencia, Spain, April. Association for Computational Linguistics.

Xue, N. and Shen, L. (2003). Chinese word segmentation as lmr tagging. In *Proceedings of the second SIGHAN workshop on Chinese language processing-Volume 17*, pages 176–179. Association for Computational Linguistics.

Zhang, Q., Liu, X., and Fu, J. (2018). Neural networks incorporating dictionaries for chinese word segmentation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Ziering, P. and van der Plas, L. (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, California, June. Association for Computational Linguistics.