Letter Sequence Labeling for Compound Splitting

Jianqiang Ma **Verena Henrich** **Erhard Hinrichs**
SFB 833 and Department of Linguistics
University of Tübingen, Germany
{jma,vhenrich,eh}@sfs.uni-tuebingen.de

Abstract

For languages such as German where compounds occur frequently and are written as single tokens, a wide variety of NLP applications benefits from recognizing and splitting compounds. As the traditional word frequency-based approach to compound splitting has several drawbacks, this paper introduces a letter sequence labeling approach, which can utilize rich word form features to build discriminative learning models that are optimized for splitting. Experiments show that the proposed method significantly outperforms state-of-the-art compound splitters.

1 Introduction

In many languages including German, compounds are written as single word-tokens without word delimiters separating their constituent words. For example, the German term for 'place name' is *Ortsname*, which is formed by *Ort* 'place' and *Name* 'name' together with the linking element 's' between constituents. Given the productive nature of compounding, treating each compound as a unique word would dramatically increase the vocabulary size. Information about the existence of compounds and about their constituent parts is thus helpful to many NLP applications such as machine translation (Koehn and Knight, 2003) and term extraction (Weller and Heid, 2012).

Compound splitting is the NLP task that automatically breaks compounds into their constituent words. As the inputs to compound splitters often include unknown words, which are not necessarily compounds, splitters usually also need to distinguish between compounds and non-compounds.

Many state-of-the-art splitters for German (Popović et al., 2006; Weller and Heid, 2012) mainly implement variants of the following two-step frequency approach first proposed in Koehn and Knight (2003):

1. Matching the input word with known words, generating splitting hypotheses, including the non-splitting hypothesis that predicts the input word to be a non-compound.

2. Choosing the hypothesis with the highest geometric mean of frequencies of constituents as the best splitting. If the frequency of the input word is higher than the geometric mean of all possible splittings, non-splitting is chosen.

The frequency approach is simple and efficient. However, frequency criteria are *not* necessarily optimal for identifying the best splitting decisions. In practice, this often leads to splitting compounds at wrong positions, erroneously splitting non-compounds, and incorrectly predicting frequent compounds to be non-compounds. Parallel corpora (Koehn and Knight, 2003; Popović et al., 2006) and linguistic analysis (Fritzinger and Fraser, 2010) etc. were used to improve the frequency approach, but the above-mentioned issues remain. Moreover, frequencies encode no information about word forms, which hinders knowledge transfer between words with similar forms. In an extreme yet common case, when one or more compound constituents are unknown words, the correct splitting is not even generated in Step 1 of the frequency approach.

To address the above-mentioned problems, this paper proposes a *letter sequence labeling* (LSL) approach (Section 2) to compound splitting. We cast the compound splitting problem as a sequence labeling problem. To predict labels, we train conditional random fields (CRF; Lafferty et al., 2001), which are directly optimized for splitting. Our

Proceedings of the 14th Annual SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 76–81,
Berlin, Germany, August 11, 2016.

CRF models can leverage rich features of letter n-grams (Section 2.3), such as *ung* (a German nominalization suffix), which are shared among words and applicable to many unknown compounds and constituents. Our method is language independent, although this paper focuses on German.

Evaluated with the compound data from GermaNet (Hamp and Feldweg, 1997; Henrich and Hinrichs, 2010) and Parra Escartín (2014), experiments in Section 3 show that our approach significantly outperforms previously developed splitters. The contributions of this paper are two-fold:

- A novel letter sequence labeling approach to compound splitting
- Empirical evaluations of the proposed approach and developed feature on a large compound list

2 Letter Sequence Labeling (LSL) for Compound Splitting

2.1 Compound splitting as LSL

Before detailing the sequence labeling approach, we first describe the representation of splitting output used for this paper.

Splitting output. The splitting output for the above example *Ortsname* would be "*Orts name*". In general, we consider the output as string sequences obtained by adding whitespaces between constituent words in the original compound. Linking elements between constituents are attached to the ones before them. Moreover, no lemmatization or morphological analysis is performed. Compound splitting also considers the recognition of non-compounds, the output of which is the word itself. The choice for such representation is to avoid bias to any morphological theory or language-specific property. If needed, however, such output can be mapped to lexemes/lemmas.

Sequence labeling. With the above-mentioned representation, compound splitting can be viewed as a sequence of predictions of what positional role each letter plays in a word/string. Specifically, we label each letter with the **BMES** tag-set. For multi-letter strings, label **B** indicates "the first letter of a string", label **E** indicates "the last letter of a string", and label **M** indicates "a letter in the middle of a string". The rare cases of single-letter strings are labeled as **S**. The label sequence for the example *Ortsname* would be: **B-M-M-E-B-M-M-E**. The splitting output strings can be constructed by extracting either single letters that are labeled as **S** or the consecutive letters such that (1) the first letter is labeled as **B**; (2) the last letter is labeled as **E**; (3) all the others in between are labeled as **M**.

We call the above formulation of compound splitting *letter sequence labeling*. It falls into the broader category of sequence labeling, which is widely used in various NLP tasks, such as POS tagging (Hovy et al., 2014) and Chinese word segmentation (Ma and Hinrichs, 2015). As many state-of-the-art NLP systems, we build conditional random fields models to conduct sequence labeling, which are detailed in the next subsections.

2.2 Conditional random fields (CRFs)

Conditional random fields (Lafferty et al., 2001) are a discriminative learning framework, which is capable of utilizing a vast amount of arbitrary, interactive features to achieve high accuracy. The probability assigned to a label sequence for a particular letter sequence of length T by a CRF is given by the following equation:

$$p_\theta(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z_\theta(\mathbf{X})} \exp\left\{ \sum_{t=1}^{T} \sum_{k=1}^{K} \theta_k f_k(y_{t-1}, y_t, x_t) \right\} \quad (1)$$

In the above formula, X is the sequence of letters in the input compound (or non-compound), Y is the label sequence for the letters in the input and $Z(X)$ is a normalization term. Inside the formula, θ_k is the corresponding weight for the *feature function* f_k, where K is the total number of features and k is the index. The letters in the word being labeled is indexed by t: each individual x_t and y_t represent the current letter and label, while y_{t-1} represents the label of the previous letter.

For the experiments in this paper, we use the open-sourced CRF implementation *Wapiti*, as described in Lavergne et al. (2010).

2.3 Feature templates

A feature function $f_k(y_{i-1}, y_i, x_i)$ for the letters x_i under consideration is an indicator function that can describe previous and current labels, as well as a complete letter sequence in the input word. For example, one feature function can have value **1** only when the previous label is **E**, the current label is **B** and the previous three letters are *rts*. Its value is **0** otherwise. This function describes a possible feature for labeling the letter *n* in *Ortsname*.

In our models, we mainly consider functions of *context features*, which include *n*-grams that ap-

pear in the local window of h characters that centers at letter x_i. In this paper, we use $1 \leq n \leq 5$ for *n*-grams and $h = 7$ for window size, as we found that smaller windows or only low order n-grams lead to inferior results. The contexts are automatically generated from the input words using feature templates by enumerating the corresponding n-grams, the index of which is relative to the current letter (i.e. x_i). Table 1 shows the templates for the context features used in this work.

Type	Context features
unigram	$x_{i-3}, x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}, x_{i+3}$
bigram	$x_{i-3}x_{i-2}, x_{i-2}x_{i-1}, ..., x_{i+2}x_{i+3}$
trigram	$x_{i-3}x_{i-2}x_{i-1}, ..., x_{i+1}x_{i+2}x_{i+3}$
4-gram	$x_{i-3}x_{i-2}x_{i-1}x_i, ..., x_ix_{i+1}x_{i+2}x_{i+3}$
5-gram	$x_{i-3}x_{i-2}x_{i-1}x_ix_{i+1}, ...$

Table 1: Context feature templates.

Besides context features, we also consider *transition features* for current letter x_i, each of which describes the current letter itself in conjunction with a possible transition between the previous and the current labels, i.e. (x_i, y_{i-1}, y_i) tuples.

3 Experiments

3.1 Gold-standard data

The training of CRFs requires gold-standard labels that are generated from the gold-standard splittings (non-splittings) of compounds (non-compounds). We use GermaNet (GN) for this purpose, as it has a large amount of available, high-quality annotated compounds (Henrich and Hinrichs, 2011). We have extracted a total of 51,667 unique compounds from GermaNet 9.0. The compounds have 2 to 5 constituents, with an average of 2.1 constituents per compound. The remaining words are, nevertheless, not necessarily non-compounds, as not all the compounds are annotated in GN. So we extract 31,076 words as non-compounds, by choosing words that have less than 10 letters and are not known compounds. The heuristic here is that compounds tend to be longer than simplex words. The resulting non-compound list still contains some compounds, which adversly affects modeling.

We also employ the Parra Escartín (2014; henceforth PE) data to allow a fair comparison of our approach with existing compound splitters. The PE dataset has altogether 342 compound tokens and 3,009 non-compounds. PE's compounds have 2 to 5 constituents, with an average amount of 2.3 constituents.

3.2 Evaluation metrics

In our experiments, we use evaluation metrics proposed in Koehn and Knight (2003), which are widely used in the compound splitting literature. Each compound splitting result falls into one of the following categories: **correct split**: words that should be split (i.e. compounds) and are correctly split; **correct non-split**: words that should not be split (i.e. non-compounds) and are not split; **wrong non-split**: words that should be split but are not split; **wrong faulty split**: words that should be split and are split, but at wrong position(s); **wrong split**: words that should not be split but are split. As in Koehn and Knight (2003), the following scores are calculated from the above counts to summarize the results that only concern compounds:

- **precision**: (correct split) / (correct split + wrong faulty split + wrong split)
- **recall**: (correct split) / (correct split + wrong faulty split + wrong non-split)
- **accuracy**: (correct) / (correct + wrong)

3.3 Experiments on GermaNet data

In the experiments of this subsection, a random set of 70% of the GN data is used for training the LSL model and another 10% is used as a development set for choosing hyper parameters of the model. The remaining 20% is the test set, which is put aside during training and only used for evaluation.

Model	Precision	Recall	Accurracy
uni- & bigrams	0.873	0.833	0.857
+ trigrams	0.937	0.920	0.925
+ 4-grams	0.952	0.940	0.942
+ 5-grams	**0.955**	**0.941**	**0.943**

Table 2: Results of models with different context features on GermaNet. Best results in **bold face**.

Since our models predict splittings solely based on the information about the input word, different tokens of the same word type appearing in various sentences would result in exactly the same prediction. Therefore the learning and evaluation with the GN data is based on *types* rather than tokens. As shown in Table 2, the model performance improves steadily by adding higher-order letter n-gram features.

Models	Correct		Wrong			Scores		
	split	non	non	faulty	split	precision	recall	accuracy
Popović et al. (2006)	248	**3009**	84	10	**0**	0.961	0.725	0.972
Weller and Heid (2012)	259	3008	82	**1**	1	**0.992**	0.757	0.975
Letter sequence labeling (this work)	**319**	2964	**14**	10	44	0.855	**0.930**	**0.980**

Table 3: Comparison with the state-of-the-art. Best results are marked in **bold face**.

The best overall accuracy of 0.943 is achieved by the model that uses features of n-grams up to order 5. No further improvement is gained by even higher order n-grams in our experiments, as the model would overfit to the training data. The high accuracy on the GN data is a reliable indicator for performance in real-life scenario, due to its rigid non-overlapping division of large training and test sets.

3.4 Experiments on Parra Escartín's data

When comparing our method with frequency-based ones, it would be ideal if each method was trained and tested (on disjoint partitions of) the same benchmark data, which provides *both* gold-standard splitting and frequency information. Unfortunately, GermaNet provides no frequency information and most large-scale word frequency lists have no gold-standard splits, which makes neither suitable benchmarks. Another practical difficulty is that many splitters are not publicly available. We plan to complement the GN data with frequency information extracted from large corpora to construct such benchmark data in the future. For the present work, we evaluate our model on the test data that other methods have been evaluated on. For this purpose, we use the PE data, as two state-of-the-art splitters, namely Popović et al. (2006) and Weller and Heid (2012)[1], have been evaluated on it.

We train the best model from the last subsection using modified GN data, which has longer non-compounds up to 15 letters in length and excludes words that also appear in the PE data. The model is evaluated on the PE data using the same metrics as described in Section 3.2, except that the evaluation is by *token* rather than by type, to be compatible with the original PE results. Table 3 shows the results, which are analyzed in the remainder of this section.

Splitting compounds. *Accuracy* and *precision* consider both non-compounds and compounds and are influenced by the ratio of the two, which is 8.8:1 for the PE data. It means that both metrics are mostly influenced by how well the systems distinguish compounds from non-compounds. By contrast, *recall* depends solely on compounds and is thus the best indicator for splitting performance. The recall of our model is significantly higher than that of previous methods, which shows that it generalizes well to splitting unknown compounds.

Recognizing non-compounds. The relatively low *precision* of our model is mainly caused by the high *wrong split* count. We found that almost half of these "non-compounds" that our model "wrongly" splits *are* compounds, as the PE annotation skips all adjectival and verbal compounds and also ignores certain nominal compounds. The remaining of wrong split errors can be reduced by using higher quality training cases of non-compounds, as the current gold-standard non-compounds were chosen by the word length heuristic, which introduced noise in learning.

4 Discussion and Related Work

Work on compound splitting emerged in the context of machine translation (Alfonseca et al., 2008b; Stymne, 2008; El-Kahlout and Yvon, 2010) and speech recognition (Larson et al., 2000) for German, Turkish (Bisazza and Federico, 2009), Finnish (Virpioja et al., 2007) and other languages (Alfonseca et al., 2008a; Stymne and Holmqvist, 2008). Most works, including discriminative learning methods (Alfonseca et al., 2008a; Dyer, 2009), follow the frequency approach. A few exceptions include, for example, Macherey et al. (2011) and Geyken and Hanneforth (2005), the latter of which builds finite-state morphological analyzer for German, where compound splitting is also covered. In contrast to most previous work, this paper models compound splitting on the lower level of letters, which can better generalize to unknown compounds and constituents. Moreover, it is possible to integrate word-level knowledge into

[1] Parra Escartín (2014) evaluated Weller and Heid (2012) 'as is', using a model pre-trained on unknown data, which might have overlaps with the test data.

the proposed sequence labeling model, by adding features such as "the current letter starts a letter sequence that matches a known word in the lexicon".

The basic idea of letter or phoneme sequence-based analysis goes back to early structural linguistics work. Harris (1955) studies the distribution of distinct phoneme unigrams and bigrams before or after a particular phoneme, i.e. *predecessor/successor variety*. The change of these variety scores in an utterance is used to determine the word boundaries. That idea has been adopted and further developed in the context of word segmentation of child-directed speech (Çöltekin and Nerbonne, 2014), where all the intra-utterance word boundaries are absent. Another instance of such sentence-wise word segmentation is Chinese word segmentation (Peng et al., 2004), where it is a standard solution to conduct CRF-based sequence labeling, using ngrams of orthographic units as features. To some extent, compound splitting can be seen as a special case of the above two word segmentation tasks. In particular, our method is clearly inspired by that of Chinese word segmentation, such as Peng et al. (2004). Although it might seem obvious to model compound splitting as letter sequence labeling in hindsight, it is not really so in foresight. Both the dominance of word frequency-based approach and the extra challenges in morphology makes is less natural to think in terms of letter operation and labeling.

5 Conclusion and Future Work

Conclusion. This paper has introduced a novel, effective way of utilizing manually split compounds, which are now available for many languages, to boost the performance of automatic compound splitting. The proposed approach is language independent, as it only uses letter n-gram features that are automatically generated from word forms. Such features capture morphological and orthographic regularities without explicitly encoding linguistic knowledge. Moreover, our approach requires no external NLP modules such as lemmatizers, morphological analyzers or POS taggers, which prevents error propagation and makes it easy to be used in other NLP systems. The proposed approach significantly outperforms existing methods.

Future work. We would like to conduct extrinsic evaluations on tasks such as machine translation to investigate how compound splitting impacts the performance of NLP applications. It is interesting to study how new features and alternative sets of labels for letters would influence the results and to test our approach on other languages such as Dutch and Swedish.

Acknowledgments

The authors would like to thank Daniël de Kok and the anonymous reviewers for their helpful comments and suggestions. The financial support for the research reported in this paper was provided by the German Research Foundation (DFG) as part of the Collaborative Research Center "The Construction of Meaning" (SFB 833), project A3.

References

Enrique Alfonseca, Slaven Bilac, and Stefan Pharies. 2008a. Decompounding query keywords from compounding languages. In *Proceedings of ACL: Short Papers*, pages 253–256.

Enrique Alfonseca, Slaven Bilac, and Stefan Pharies. 2008b. German decompounding in a difficult corpus. In *Computational Linguistics and Intelligent Text Processing*, pages 128–139. Springer.

Arianna Bisazza and Marcello Federico. 2009. Morphological pre-processing for Turkish to English statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 129–135.

Çağrı Çöltekin and John Nerbonne. 2014. An explicit statistical model of learning lexical segmentation using multiple cues. In *Proceedings of EACL 2014 Workshop on Cognitive Aspects of Computational Language Learning*.

Chris Dyer. 2009. Using a maximum entropy model to build segmentation lattices for MT. In *Proceedings of NAACL*, pages 406–414.

Ilknur Durgar El-Kahlout and François Yvon. 2010. The pay-offs of preprocessing for German-English statistical machine translation. In *Proceedings of International Workshop of Spoken Language Translation*, pages 251–258.

Fabienne Fritzinger and Alexander Fraser. 2010. How to avoid burning ducks: combining linguistic analysis and corpus statistics for German compound processing. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 224–234.

Alexander Geyken and Thomas Hanneforth. 2005. Tagh: A complete morphology for german based on weighted finite state automata. In *Finite-State Methods and Natural Language Processing*, pages 55–66. Springer.

Birgit Hamp and Helmut Feldweg. 1997. GermaNet – a lexical-semantic net for German. In *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, Madrid.

Zellig S Harris. 1955. From phoneme to morpheme. *Language*.

Verena Henrich and Erhard Hinrichs. 2010. GernEdiT – the GermaNet editing tool. In *Proceedings of LREC*, pages 2228–2235, Valletta, Malta, May.

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

Dirk Hovy, Barbara Plank, and Anders Søgaard. 2014. Experiments with crowdsourced re-annotation of a POS tagging data set. In *Proceedings of ACL*, pages 377–382.

Philipp Koehn and Kevin Knight. 2003. Empirical Methods for Compound Splitting. In *EACL*, page 8.

John Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of International Conference on Machine Learning*, pages 282–289.

Martha Larson, Daniel Willett, Joachim Köhler, and Gerhard Rigoll. 2000. Compound splitting and lexical unit recombination for improved performance of a speech recognition system for german parliamentary speeches. In *Proceedings of INTERSPEECH*, pages 945–948.

Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. Practical very large scale CRFs. In *Proceedings of ACL*, pages 504–513.

Jianqiang Ma and Erhard Hinrichs. 2015. Accurate linear-time Chinese word segmentation via embedding matching. In *Proceedings of ACL-IJCNLP (Volume 1: Long Papers)*, pages 1733–1743, Beijing, China, July.

Klaus Macherey, Andrew M Dai, David Talbot, Ashok C Popat, and Franz Och. 2011. Language-independent compound splitting with morphological operations. In *Proceedings of ACL*, pages 1395–1404.

Carla Parra Escartín. 2014. Chasing the perfect splitter: a comparison of different compound splitting tools. In *LREC*, pages 3340–3347.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of COLING*, pages 562–568.

Maja Popović, Daniel Stein, and Hermann Ney. 2006. Statistical machine translation of German compound words. In *Advances in Natural Language Processing*, pages 616–624. Springer.

Sara Stymne and Maria Holmqvist. 2008. Processing of Swedish compounds for phrase-based statistical machine translation. In *Proceedings of the 12th Annual Conference of the European Association for Machine Translation*, pages 180–189.

Sara Stymne. 2008. German compounds in factored statistical machine translation. In *Advances in Natural Language Processing*, pages 464–475. Springer.

Sami Virpioja, Jaakko J Väyrynen, Mathias Creutz, and Markus Sadeniemi. 2007. Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner. *Machine Translation Summit XI*, 2007:491–498.

Marion Weller and Ulrich Heid. 2012. Analyzing and Aligning German Compound Nouns. In *Proceedings of LREC*, pages 2–7.