Automatic Compound Segmentation for Leichte Sprache

Jesús Calvillo, Umesh Patil, Johann Seltmann, Anne-Kathrin Schumann

t2k GmbH, Dresden, Germany

j.calvillo@text2knowledge.de umesh.patil@text2knowledge.de

johann.seltmann@text2knowledge.de ak.schumann@text2knowledge.de

Abstract

In German "Easy Language" (Leichte Sprache), complex compound words are often orthographically segmented to facilitate perception and processing by marking their internal structure. This practice has been shown to facilitate reading comprehension, especially for readers with cognitive or reading impairments. We present a lightweight model that combines Compound Segmentation with Complex Word Identification (CWI) to automatically detect and split difficult compounds in text. We evaluate our system both on general compound segmentation and in the specific context of Leichte Sprache. Our results show that our model achieves high segmentation accuracy, outperforming both rule-based and much larger neural systems, identifying which compounds should be segmented. We also release a new evaluation dataset of Leichte Sprache sentences with segmented compounds.

1 Introduction

German is a language that allows for word formation through compounding, i.e., multiple words can join to form a single one without any orthographic separation (e.g. *Armut* "poverty" + *Bekämpfung* "combat" + *Programm* "program" → *Armutsbekämpfungsprogramm*). Compounding can be applied recursively, yielding a potentially infinite amount of linguistically acceptable compounds. In practice, this word formation process is used very productively; in fact, Baroni et al. (2002) report that 47% of all word types in the APA German news corpus are compounds. This creates challenges for NLP as compounds are often low-frequency: 83% of the compounds in the APA corpus had a frequency of less than 5 occurrences (Baroni et al., 2002). This causes data sparsity, which complicates tasks like information retrieval (e.g., Alfonseca et al., 2008; Monz and De Rijke, 2001), speech recognition (e.g., Adda-Decker and Adda, 2000), and machine translation (e.g., Daiber et al., 2015; Koehn and Knight, 2003; Neumannová and Bojar, 2023).

Compound Splitting (also called Compound Decomposition or Decompounding) refers to splitting such words into their constituents, mitigating some of the problems mentioned above. This can be further divided into *compound segmentation* and *compound normalization* (Ziering and van der Plas, 2016). The former refers to segmenting a compound into its constituents while preserving its surface form (e.g. Armutsbekämpfungsprogramm → Armuts + bekämpfungs + programm). The latter corresponds to task of recovering the base form of each constituent (e.g. Armutsbekämpfungsprogramm → Armut + Bekämpfung + Programm).

This task is not always simple: some compounds involve simple concatenation (e.g., *Sport* + *Schuhe* → *Sportschuhe*); however, others require morphological adjustments, including linking elements (Fugenelemente),[1] truncation, or pluralization, among others:

Bekämpfung + *s* + *Programm* → *Bekämpfungsprogramm* (linking element)
Kirsche -e + *Baum* → *Kirschbaum* (truncation)
Buch + *ü_er* + *Regal* → *Bücherregal* (pluralization with umlaut)

Compound splitting can also help readability, as some compounds can be difficult to process for humans, especially for people with reading impairments. In Leichte Sprache (German "Easy Language"), long and difficult compounds are typically visually segmented with hyphens or the mediopunkt (·) to support lexical decoding (Bredel and

[1] Fugenelemente are a limited set of interfixes (e.g., *-s-*, *-es-*) inserted between compound parts, typically for phonetic reasons.

Proceedings of the 21st Conference on Natural Language Processing (KONVENS), Volume 2: Workshops, pages 191–201, 2025

Maaß, 2016) and enhance comprehension for readers with cognitive or reading impairments. This has been stipulated in guidelines for Leichte Sprache (e.g., Deutsches Institut für Normung e.V., 2025; Maaß, 2015; Bundesministerium für Arbeit und Soziales, 2019). Empirical studies confirm to some extent that this helps readers: Deilen (2021) and Wellmann (2020) used eye-tracking to show that visually segmented compounds reduce reading time and cognitive load for both cognitively impaired and unimpaired readers. Similarly, Pappert and Bock (2020) use a lexical decision task to show that visual compound segmentation results in shorter reaction times both for readers with cognitive impairments and functional illiteracy.

Despite the importance of compound segmentation in Leichte Sprache, only a few computational models address this explicitly. Although tools such as Gertwol (Haapalainen and Majorin, 1995; Steiner, 2018) have been integrated into language simplification pipelines (e.g., Suter et al. 2016), to our knowledge, no previous model jointly addresses automatic identification and segmentation of difficult compounds for Leichte Sprache.

Compound segmentation for Leichte Sprache implies two steps: 1) identifying the *difficult* words to be segmented; and 2) segmenting those words. We present a neural model of compound segmentation that in combination with a model of complex word identification (CWI) performs these steps, thus providing an automatic system to fulfill this requirement. Our work is the first to combine a compound segmentation model with a model for complex word identification model to systematically handle compound splitting for Leichte Sprache.

Our model has a similar architecture to Tuggener (2018) with minimal modifications and a more fine-grained training procedure. The model of complex word identification that we used is an XGBoost classifier presented by Patil et al. (2025). We evaluated whether this CWI model can identify the words that need segmentation. Then we evaluated our compound segmentation model showing that it can in general segment German compounds and even has better results than much larger state-of-the-art models. Then we evaluated our combined model of compound segmentation and CWI on the task of compound segmentation for Leichte Sprache with a new dataset of Leichte Sprache with segmented compounds that we make available.[2]

Our contributions are:

- A parallel dataset of sentences with segmented compounds, with their unsegmented pairs.
- A training procedure to train *lightweight* compound segmentation models, leveraging unsupervised and supervised training.
- A pipeline for compound segmentation designed for the domain of Leichte Sprache, with its evaluation.

2 Related Work: German Compound Splitting

Many models for German Compound Splitting have been proposed, ranging from rule-based systems to state-of-the-art neural models. Early systems such as those by Koehn and Knight (2003), Weller and Heid (2012) and Weller-Di Marco (2017) relied on dictionaries, corpus frequencies and manually designed rules. While precise, such systems are limited in coverage and often fail on out-of-vocabulary words.

To overcome this, unsupervised methods were introduced. Macherey et al. (2011) use a bilingual corpus to learn morphological operations, avoiding handcrafted rules. SECOS (Riedl and Biemann, 2016) uses a distributional thesaurus to find semantically coherent compound constituents. Another important advance was the MOP Compound Splitter (Ziering and van der Plas, 2016), which learns morphological operations.

Among the neural approaches, the one that is the most relevant to our work is that of Tuggener (2018), who evaluated several supervised and unsupervised neural models, showing that they were able to generalize better on out-of-domain data, compared to earlier approaches. Among other architectures, the study proposes a bi-directional character LSTM (Hochreiter and Schmidhuber, 1997) trained auto-regressively with a language model objective (at a given time step, the model is expected to predict the next/previous character), whose forward and backward hidden states are then fed to a multilayer perceptron that classifies whether after each position of a character sequence, a separator needs to be inserted (Figure 1 shows our very

[2] https://github.com/text2knowledge/ls_comp_segmentation_hurraki, https://huggingface.co/datasets/text2knowledge/ls_comp_seg_hurraki

similar architecture). This model was compared against dictionary/ngram-based models, an LSTM-based unsupervised approach, and a sequence to sequence with attention model. The Bi-LSTM with the multilayer perceptron was the best among the compared models. This architecture was later used also for compound splitting in Icelandic (Dađason et al., 2020), and later by Krotova et al. (2020), who also performed idiomatic compound detection.

More recently, Minixhofer et al. (2023) built a wide-scale multilingual dataset for compound splitting formed by 255K words across 56 languages, that was used to train and evaluate several variations of sequence to sequence transformer models, including large language models (LLMs). Similar to Tuggener (2018), they apply a two-staged approach with a self-supervised stage and a supervised one. Their results showed that character-based models outperform all subword-based models, **including LLMs**, especially on compounds where subword boundaries do not coincide with compound constituent boundaries. They report their best models outperformed all the baselines they used, however, they did not compare against the neural approaches mentioned above (i.e., Tuggener, 2018; Krotova et al., 2020; Dađason et al., 2020), whose performance in German is actually very similar ($\sim 95\%$ average accuracy) to that of Minixhofer et al. (2023).

3 Compound Segmentation for Leichte Sprache

While compound segmentation has been shown to facilitate reading comprehension of long and difficult compounds, to apply this in practice we also need to identify the words that need segmentation. We propose a two-staged approach for this task:

1. **Complex Word Identification**: Words or sentences are analyzed to identify difficult terms that may require compound segmentation.

2. **Compound Segmentation**: Difficult words are segmented into their constituent parts.

These steps can be used in a language simplification pipeline, and are further explained below.

3.1 Complex Word Identification Model

To identify which words should be split in the context of Leichte Sprache, we employ a Complex Word Identification (CWI) model. CWI is a subtask of lexical simplification where systems detect words that are likely to be difficult for a target audience. We used the CWI model reported in Patil et al. (2025). The model performs the CWI task for German. It was trained as an XGBoost classifier using a combination of word-level linguistic features, corpus-level distributional patterns, frequency data from a Leichte Sprache corpus, and human-annotated complexity ratings. The model achieves a relatively high F1-score of 0.85, evaluated on the dataset reported in Yimam et al. (2018).

In addition to binary classification as complex or non-complex, the model can also estimate a word's complexity level on a continuous scale ranging from 0 to 1. The model's predictions of word complexity were validated using psycholinguistic data of online word processing. Using a dataset of German nouns from the Developmental Lexicon Project (DeveL, Schröter and Schroeder, 2017), the model was shown to account for human word recognition times beyond what traditional word-level predictors can explain. Furthermore, the model was shown to produce promising results in identifying the CEFR levels of words.

3.2 Compound Segmentation Model

Since prior work has shown that character-based models outperform those using *subword tokenizers* (e.g., in LLMs), we adopted a very similar architecture to the best one proposed by Tuggener (2018). It consists of a character-level bidirectional LSTM pretrained with a language modeling objective, followed by a binary classification layer trained to predict whether a split (label 1) or no split (label 0) should occur at each character boundary (see Figure 1).

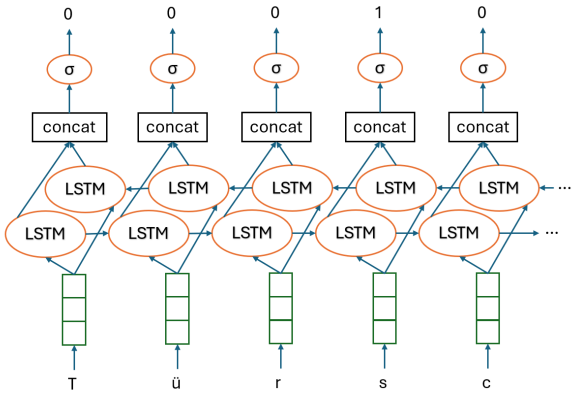

Figure 1: Model Architecture: A Bi-LSTM model with a sigmoid classification layer.

More formally, each character c_t in the input sequence is embedded as a vector $\mathbf{e}_{c_t} \in \mathbb{R}^{d_{\text{emb}}}$,

where $d_{\text{emb}} = 128$.

These embeddings are then passed to forward and backward LSTM layers with *unshared* parameters, allowing each direction to be trained auto-regressively. The forward and backward hidden states are computed as:

$$\mathbf{h}_t^{\rightarrow} = \text{LSTM}_{\rightarrow}(\mathbf{e}_{c_t}, \mathbf{h}_{t-1}^{\rightarrow}), \quad (1)$$

$$\mathbf{h}_t^{\leftarrow} = \text{LSTM}_{\leftarrow}(\mathbf{e}_{c_t}, \mathbf{h}_{t+1}^{\leftarrow}), \quad (2)$$

where $\mathbf{h}_t^{\rightarrow}, \mathbf{h}_t^{\leftarrow} \in \mathbb{R}^{128}$ denote the hidden states in the respective directions. Each LSTM is trained with a language modeling objective to predict the next (or previous) character using a softmax layer, with the following cross-entropy losses:

$$\mathcal{L}_{\text{LM}}^{\rightarrow} = \text{CE}\big(\mathbf{W}^{\rightarrow}\mathbf{h}_t^{\rightarrow},\, c_{t+1}\big), \quad (3)$$

$$\mathcal{L}_{\text{LM}}^{\leftarrow} = \text{CE}\big(\mathbf{W}^{\leftarrow}\mathbf{h}_t^{\leftarrow},\, c_{t-1}\big), \quad (4)$$

where $\mathbf{W}^{\rightarrow}, \mathbf{W}^{\leftarrow} \in \mathbb{R}^{|\mathcal{V}|\times 128}$ are learned output projection matrices and $\mathcal{V}$ is the character vocabulary.

For the compound splitting task, the forward and backward hidden states are concatenated and passed through a sigmoid-activated linear layer to predict the probability of a split at each character position:

$$p_t = \sigma\big(\mathbf{w}^{\top}[\mathbf{h}_t^{\rightarrow};\, \mathbf{h}_t^{\leftarrow}] + b\big), \quad (5)$$

where $\mathbf{w} \in \mathbb{R}^{256}$, $b \in \mathbb{R}$, and $[\cdot\,;\cdot]$ denotes vector concatenation.

The binary cross-entropy loss for split prediction is defined as:

$$\mathcal{L}_{\text{split}} = \text{BCE}(p_t,\, y_t), \quad (6)$$

where $y_t = 1$ if a split precedes character t, and $y_t = 0$ otherwise. To address the class imbalance (i.e., the relative sparsity of split positions), positive labels are up-weighted by a factor of 10 during training.

A key advantage of this architecture is its efficiency: the resulting model is lightweight, easy to train, and suitable for deployment in resource-constrained environments.

3.2.1 Training Datasets

We used three datasets to train and evaluate our compound segmentation model:

German Sentences: to train auto-regressively, we used 140K German sentences from the Leipzig Corpora Collection (2021): half of them were extracted from Wikipedia and the other half from mixed sources.

Compound Piece (Minixhofer et al., 2023): This resource contains 255K compound words across 56 languages. It also includes a set of hyphenated words automatically harvested from the mC4 corpus (Xue et al., 2021). Though the segmentations of the latter are noisy and not perfectly aligned with true compound boundaries, they provide substantial weak supervision. Minixhofer et al. (2023) leverage this dataset both to pretrain a decompounding model and to refine subword tokenization. Similarly, we use the German part of the hyphenated words for the early training phases of our compound segmentation model.

GermaNet: For fine-tuning, we used GermaNet (Henrich and Hinrichs, 2011), a curated resource of 126,733 segmented German compounds. These include rich linguistic annotations such as linking elements, truncations, affixoids, confixes,etc. We used the most recent version (v19.0, University of Tübingen 2024) for final training.

3.2.2 Training regime

Our training regime combines some of the techniques used by Tuggener (2018) and Minixhofer et al. (2023), aiming at leveraging auto-regressive language model pretraining with semi-supervised training and further supervised fine-tunning. This is done in four phases:

Phase 1: auto-regressive pre-training. We first train the LSTMs to predict their next and previous characters in normal German sentences. This ensures that the model is familiar with the patterns of German character sequences, which allows the model to build the appropriate representations that are necessary for the classification task.

We train for up to 20 epochs (Adam optimizer, batch size of 16, $lr = 0.01$, early-stopping patience of 3); the loss typically plateaus after 10 epochs. Three random seeds are tried and the checkpoint with best validation loss is retained. Note that we could have used many more sentences, but because of its small size, the model cannot memorize the training data, so the 140K were

Text with Unsegmented Compounds	Text with Segmented Compounds
Der Weihnachtsbaum hat sogar eine eigene Internetseite.	Der Weihnachts•Baum hat sogar eine eigene Internet•Seite.
Sie tragen Plastikhandschuhe.	Sie tragen Plastik•Handschuhe.
Dann können andere Facebookbenutzer sehen, was sie mögen.	Dann können andere Facebook•Benutzer sehen, was sie mögen.

Table 1: Example sentences from the evaluation dataset.

enough. Preliminary experiments with more sentences yielded similar results. We used 130K sentences for training and 10K for validation.

Phase 2: classification warm-up with frozen weights. The best auto-regressive weights are loaded, *embedding weights and both LSTMs are frozen*, and only the classifier head is trained on the Compound Piece data. We keep an aggressive learning rate ($lr = 0.01$); training converges within 3–5 epochs. This phase is focused on training the classification layer while keeping the previously trained LSTM weights safe from abrupt modifications due to the initially-high loss gradients of the classification layer.

Phase 3: full classification training. All layers are unfrozen and training resumes on the same Compound Piece data with a reduced rate ($lr = 0.0015$); approximately 10 epochs suffice for convergence.

Phase 4: GermaNet fine-tuning. Finally, the model is fine-tuned on the GermaNet corpus. ($lr = 0.0015$, 10 epochs). The resulting checkpoint is the model used for the experiments in §4.

After training, as a final post-processing step, we remove segmentations concerning prepositions (e.g., *bei, mit*) at the beginning of words (e.g. *Beispiel* "example", *Mitarbeiter* "employee") and some bound morphemes[3] (e.g., *her* in *Herkunft* "origin").

4 Evaluation

4.1 Evaluation Dataset

To evaluate our compound segmentation system, we compiled a list of sentences in Leichte Sprache from two sources:

- **Hurraki**:[4] a Leichte Sprache encyclopedia.
- **Einfach Stars**:[5] a Leichte Sprache news site covering famous people and pop culture.

We selected these two from several different possible sources, as they both consistently segment difficult words and do not segment simpler words.

In addition to segmented compound nouns, these sources also sometimes segment verbs (e.g. "heraus-finden" and "aus-gesprochen"). Since we (and our model) do not consider these constructions as compounds, we used a heuristic based on regular expressions and part-of-speech tagging to identify them and remove the separator characters ("-" and "•"), thereby avoiding evaluating this particular type of words. We leave for future discussion whether these words should be segmented or not.

We then created a parallel dataset of sentence pairs by applying the same heuristic to the segmented nouns in each sentence, so that each example consists of one sentence with segmented nouns and one with unsegmented nouns. Table 1 contains examples of these pairs. This results in a dataset with 38114 sentence pairs with at least one segmented compound. To this, we added 500 examples that do not contain any segmented compound and 500 examples that originally contained a segmented verb (e.g., "heraus-finden"), but we removed this segmentation.

The vast majority of the pairs come from Einfach Stars (34187). Most sentences only contain one segmented compound (see Table 2). There are 42006 segmented compounds in total, with 15933 unique ones. Table 3 shows the most common segmented compounds and their counts. Within these examples, we also identified 1858 compounds (587 unique) that were not segmented in their sources, perhaps because they were deemed simple enough. Table 4 shows the most common ones.

While we used both sources for our evaluation, because of copyrights, we can make available only the Hurraki part.

[3] Namely: *ent-, ge-, her-, hin-, un-, ur-, ver-, -bar, -los*

[4] https://hurraki.de

[5] https://einfachstars.info

# Segmented Compounds	# Sentences
0	998
1	33027
2	3728
3	372
4+	89

Table 2: Count of sentences with different numbers of compounds in the evaluation dataset.

4.2 Evaluation Tasks

We evaluate our compound segmentation pipeline in three settings:

1. **Compound Segmentation**: We evaluate compound segmentation by comparing the output of different models of compound segmentation against the original segmentation, considering only the words that were originally segmented. We also evaluate the case of unsegmented compounds, where we assume that they were not segmented in the original sources because they are not difficult, therefore, we assume they should be labeled as simple.

2. **Complex Word Identification**: We evaluate the performance of the CWI model to identify the compounds that may need to be segmented. Here we assume that the segmented compounds in the dataset were selected because of their complexity. Then, the CWI model is expected to identify the segmented compounds as difficult and the unsegmented compounds as simple.

3. **CWI + Compound Segmentation**: In our final setting, we first apply the CWI model to select words that may need segmentation and then we apply different segmenters on those words. We assess the differences in the full sentences.

4.3 Baselines

We compared our segmentation system against two baselines:

- **MOP Compound Splitter** (MCS, Ziering and van der Plas, 2016), which is a splitter based on morphological operations. We used this as an example of classical rule-based models for compound splitting.

Segmented Compound	Count
Internet•Seite	386
Kranken•Haus	293
Fernseh•Sendung	290
Groß•Britannien	226
Welt•Tag	218
Königs•Familie	202
Königs•Haus	146
Fernseh•Sendungen	145
National•Mannschaft	136
Ehe•Frau	133

Table 3: Most common segmented compounds in the evaluation dataset.

Unsegmented Compound	Count
Mitglied	130
Hochzeit	87
Geburtstag	77
Tatort	70
Flugzeug	52
Zeitschrift	51
Kennenlernen	51
Sonntag	42
Feuerwehr	39
Wochenende	32

Table 4: Most common unsegmented compounds in the evaluation dataset.

- **Compound Piece**, which is a sequence to sequence compound normalization model whose output is then used to perform compound segmentation. We used this model as an example of large-scale neural compound segmentation models.

We did not find an instance of the models by Tuggener (2018) and similar variants (i.e. Daðason et al., 2020; Krotova et al., 2020), but we expect our model to have a similar performance, potentially benefiting from the extra training phases.

5 Results

From the Leichte Sprache sentence pairs described in subsection 4.1, we used a random sample of 10K sentences for our evaluation experiments (8946 from Einfach Stars, 1054 from Hurraki), out of which 260 contained no segmented compounds. We used the CWI model to tokenize the sentences and infer word complexity values for each token. Then, we applied the compound segmentation

Model	Accuracy(%)
MCS	37.28
Compound Piece	85.29
Our model	**87.66**

Table 5: Accuracy comparison of different compound segmentation models.

models on these tokenized sentences. The CWI model uses spaCy (Honnibal et al., 2020) for tokenization, which is mostly driven by white spaces and punctuations, the resulting tokens are words or punctuations.

5.1 Compound Segmentation

We took the segmented compounds in the test dataset as ground truth and compared it against our model and the baselines' output. Note that this ground truth often does not segment a compound into all of its constituents (e.g., *Weihnachtsfest•Post*), while the compared models aim at fully segmenting the input compounds (i.e., *Weihnachts•Fest•Post*).

The model accuracies can be found in Table 5. By looking at the mistakes of the MCS model, we can see that its low accuracy is mainly because it works as a compound normalizer rather than a segmenter, so the compound constituents are often changed to its lemma form (e.g., *Menschenaffen* → *Mensch•Affe*). It also relies on its lemma dictionary, so unknown words are not handled properly (e.g., *Spidermanfilme* → *Sid•Erman•Film*).

Our model had the best segmentation performance, while the Compound Piece model had a very comparable accuracy. Since they exclusively perform segmentation, no mistakes regarding modifications of the original constituents are made. Some of the main mistakes are concerning words that have more than 2 constituents and the models segment more than the ground truth (e.g., *Fußballmannschaft* → *Fuß•Ball•Mannschaft* instead of *Fußball•Mannschaft*). Another common difference is that compounds that start or end with a preposition are usually not segmented by the models while sometimes they are segmented in the ground truth (e.g., *Mit•Arbeiter*,*Vor•Name*).

5.2 Complex Word Identification

Our dataset contains segmented compounds that were identified with a heuristic that uses the separation character (a hyphen or middle point). However, among the sentences in the dataset, there are also compounds that were not segmented by the original text authors, which we expect to be simpler words. To identify these and given the relatively high quality of our segmentation model (as shown above), we applied our segmentation model on the unsegmented sentences; then, we considered as an unsegmented compound any word that was segmented by the model and that was not originally segmented. We speculate that they were not segmented because they were simple enough for Leichte Sprache. With this process, we obtained 1522 instances of unsegmented compounds.

The CWI model returns a continuous value between 0 and 1 that corresponds to how difficult the word is (0 being not difficult, 1 being difficult). Using a threshold of 0.5 to label the words in the test set (i.e., any word whose CWI value is greater than 0.5 is labeled as difficult, or simple otherwise), we counted how many times those compounds were labeled as difficult: out of the 10982 segmented compounds in the test set, 9229 (84%) were identified as difficult. Regarding the unsegmented compounds, which are expected to be labeled as simple, out of the 1522 unsegmented compounds that we identified, 983 (63%) were labeled as simple.

Nevertheless, a complexity threshold of 0.5 may not properly fit the properties of Leichte Sprache. To try to calibrate it, we applied a grid search with threshold $\tau = 0.0$ to $\tau = 0.95$ with an increment of 0.5. The results showed that since there are many more segmented compounds than unsegmented compounds, the threshold that maximizes a naive F1-score is 0.0 (i.e. to segment all compounds). A more balanced approach is computing accuracies separately between segmented and unsegmented compounds and then averaging them. In this case, the threshold with maximum average accuracy is 0.65 (i.e. to avoid segmenting most compounds), but its value (74.6) is not much different from more balanced thresholds (e.g., when $\tau = 0.5$, the average accuracy is 74.3). Because of these results, we continued the rest of the analysis with our initial threshold of 0.5. However, we expect that a more carefully curated and balanced dataset may yield more fine-grained results.

5.3 CWI + Compound Segmentation

We evaluated our whole pipeline by applying CWI on the sentences and then applying the different

segmenters on the words that were labeled as difficult. Here, a perfect system would output the sentences exactly as in the ground truth.

Table 6 shows the token-level accuracy (Tok Acc%), the percentage of perfectly segmented sentences (Perfect%), and two key error types: false positives (FP) and false negatives (FN). Here we define a false positive as a token that was not supposed to be modified but was modified by a given model. We define a false negative as a compound (identified via the ground truth annotations) that was erroneously left unchanged by the model, indicating a missed segmentation. We report false negatives both as a percentage of all compound tokens (FN %Comps) and as a percentage of all tokens (FN %Toks).

Without CWI, our model achieves the highest token-level accuracy (97.0%), the highest percentage of perfect output sentences (75.2%), and the lowest false positive rate (1.5%). It also has the lowest false negative rates: 4.3% of compound tokens and 0.48% of all tokens, showing that most compounds were successfully segmented. Compound Piece follows with 94.5% token accuracy and 58.8% perfect output sentences, while MCS trails further behind.

With CWI, the MCS and Compound Piece models improve, and while our model remains in the lead, its token accuracy slightly drops to 96.5%, with 71.6% perfect sentence outputs. Its false positive rate drops to just 0.6%, but its false negative rates increase to 16.4% of compounds and 1.82% of all tokens. While Compound Piece narrows the gap slightly, our model continues to deliver the best results. These results confirm that the integration of a CWI filter reduces the amount of false positives, however, this filter may need to be further tuned to reach an appropriate balance between false positives and negatives. In our case, the reduction of false positives is not as drastic because it was already relatively low (from 1.5% to 0.6%), compared to the reduction of false positives of the other two models (5.4% to 1.5% and 4.0% to 1.1), this can explain why applying CWI actually hurts slightly the overall accuracy of our model: the decrease of false positives does not compensate the increase of false negatives. A more careful false negative/positive balance may yield better results.

6 Discussion

We evaluated our pipeline for compound segmentation in Leichte Sprache, which consists of selecting the difficult words to be segmented, and then applying segmentation on those words.

Selecting what compounds to segment is very important: segmentation helps readers to process difficult compounds, however, segmenting some compounds may actually make them more difficult because the readers are usually more familiar with their unsegmented form. Moreover, segmenting some compounds may lead to wrong interpretations (e.g., *Groß•Mutter* "grandmother" should not be interpreted as "big mother"). Regarding this aspect, our dataset is rather noisy, as there are no quantitative criteria to inform annotators about what compounds should be segmented and what constitutes a *difficult* compound: perceptions of difficulty vary between individuals—what one annotator considers challenging, another may find straightforward. Indeed, some compounds in our corpus are segmented even though they may not be particularly difficult, e.g., *Mit•Arbeiter* ("employee"). It is important to have clear definitions of how difficult a compound should be to be segmented and how we can measure this. Nonetheless, while these concepts require more definition, we can already see that using CWI can help identify these compounds.

Regarding the correct compound segmentation, it is also not entirely clear whether compounds should be fully segmented, only their main constituents, or only the *difficult* ones; which is reflected in the variety of ways in which compounds are segmented by humans. The models that we evaluated used full segmentation, but other formats could be applied.

In general, clearer guidelines are needed regarding what compounds should be segmented in Leichte Sprache, and to what degree (binary vs. partial vs. full). Such guidelines would improve the quality of the data, which would improve the evaluation and automation of this task.

We also see that comparing the different segmentation models, our smaller model performed better than the larger transformer-based Compound Piece model, which is very useful, as our model can be deployed in more resource-constrained environments. We leave further experiments with LLMs for future work. There is, however, evidence that LLMs that use sub-word tokens have

Model	Tok Acc%	Perfect%	FP %Toks	FN %Comps	FN %Toks
Without CWI					
MCS	86.9	22.8	5.4	6.6	0.73
Compound Piece	94.5	58.8	4.0	4.4	0.49
Our Model	**97.0**	**75.2**	**1.5**	**4.3**	**0.48**
With CWI					
MCS	90.1	28.3	1.5	16.6	1.84
Compound Piece	95.8	65.7	1.1	**16.1**	**1.78**
Our Model	**96.5**	**71.6**	**0.6**	16.4	1.82

Table 6: Token Accuracy (Tok Acc), percentage of perfect output sentences (Perfect), percentage among all tokens of false positive segmentations (FP Toks), percentage among all compounds of false negative segmentations (FN Comps) and percentage among all tokens of false negative segmentations (FN Toks) of different compound segmentation models in Leichte Sprache sentences, with and without Complex Word Identification (CWI).

problems with tasks that involve information about the internal structure of those tokens (Shin and Kaneko, 2024; Cosma et al., 2025). This is in line with what was already reported by Minixhofer et al. (2023). Because of this and the amount of resources that LLMs require, we consider our approach to be more advantageous, especially in cases with restricted resources.

7 Conclusion

We introduced a two-stage system for compound segmentation in Leichte Sprache, combining a lightweight character-level model with a complex word identification component. Our system automatically identifies and segments difficult German compounds, supporting the goals of linguistic accessibility.

The model outperforms both rule-based and larger neural baselines while remaining efficient and deployable in resource-constrained settings, and shows that large transformer-based architectures are not always necessary in NLP. We also contribute a new evaluation dataset of segmented Leichte Sprache sentences to facilitate further research.

Our results show that segmentation accuracy improves through CWI, though further calibration is needed to reduce under-segmentation. We emphasize that clearer guidelines are needed about what compounds should be segmented and to what extent (full vs. partial).

Future work includes refining the complexity threshold, exploring alternative segmentation strategies, and conducting human evaluations.

Limitations

Our CWI component uses a fixed complexity threshold, which may not suit all users/contexts. We also assume that all segmented compounds are difficult and that unsegmented ones are easy, which may not always hold. The model performs full segmentation, while human-annotated texts often use partial. Finally, we have not yet validated the model through human studies, which are essential to assess its usefulness in real-world applications.

Acknowledgments

The work leading to this paper was partially funded by the Federal German Ministry for Labour and Social Affairs through the Civic Innovation Platform[6]. We are also grateful for the feedback that we received from the anonymous reviewers, which helped improve this article.

References

Martine Adda-Decker and Gilles Adda. 2000. Morphological decomposition for asr in german. In *Workshop on Phonetics and Phonology in ASR, Saarbrücken, Germany*, pages 129–143.

Enrique Alfonseca, Slaven Bilac, and Stefan Pharies. 2008. Decompounding query keywords from compounding languages. In *Proceedings of ACL-08: HLT, Short Papers*, pages 253–256, Columbus, Ohio. Association for Computational Linguistics.

Marco Baroni, Johannes Matiasek, and Harald et al. Trost. 2002. Predicting the components of German nominal compounds. In *ECAI*, volume 2002, page 15th.

[6] https://www.knowledgegraph.de/

Ursula Bredel and Christiane Maaß. 2016. *Leichte Sprache: Theoretische Grundlagen. Orientierung für die Praxis*. Duden.

Bundesministerium für Arbeit und Soziales. 2019. Barrierefreie-informationstechnik-verordnung (bitv2.0). Bundesrechtsverordnung auf Grundlage des Behindertengleichstellungsgesetzes (§§12d,16Abs.8 BGG). Inkraftgetreten am 25. Mai 2019; Neufassung vom 12.September2011.

Adrian Cosma, Stefan Ruseti, Emilian Radoi, and Mihai Dascalu. 2025. The strawberry problem: Emergence of character-level understanding in tokenized language models. *arXiv preprint arXiv:2505.14172*.

Jón Friðrik Daðason, David Erik Mollberg, Hrafn Loftsson, and Kristín Bjarnadóttir. 2020. Kvistur 2.0: a bilstm compound splitter for icelandic. *arXiv preprint arXiv:2004.07776*.

Joachim Daiber, Lautaro Quiroz, Roger Wechsler, and Stella Frank. 2015. Splitting compounds by semantic analogy. *arXiv preprint arXiv:1509.04473*.

Silvana Deilen. 2021. Segmenting compounds in German Easy Language: Does facilitated perception lead to reduced cognitive processing costs? In *3rd Swiss Conference on Barrier-free Communication (BfC 2020)*, page 135.

Deutsches Institut für Normung e.V. 2025. Din spec 33429:2025-03 – empfehlungen für deutsche leichte sprache. Veröffentlicht durch den Normenausschuss Ergonomie (NAErg). Konsortiumsentwicklung mit Förderung durch das Bundesministerium für Arbeit und Soziales (BMAS).

Mariikka Haapalainen and Ari Majorin. 1995. Gertwol und morphologische disambiguierung für das deutsche. In *Proceedings of the 10th Nordic Conference on Computational Linguistics*.

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.

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

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python. https://doi.org/10.5281/zenodo.1212303. Zenodo release.

Philipp Koehn and Kevin Knight. 2003. Empirical methods for compound splitting. *arXiv preprint cs/0302032*.

Irina Krotova, Sergey Aksenov, and Ekaterina Artemova. 2020. A joint approach to compound splitting and idiomatic compound detection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4410–4417, Marseille, France. European Language Resources Association.

Leipzig Corpora Collection. 2021. German corpus based on Wikipedia material from 2021 and mixed material from 2011. https://corpora.uni-leipzig.de?corpusId=deu_news_2023. Leipzig Corpora Collection. Dataset.

Christiane Maaß. 2015. *Leichte Sprache. Das Regelbuch*. LIT Verlag Dr. W. Hopf, Berlin.

Klaus Macherey, Andrew Dai, David Talbot, Ashok Popat, and Franz Josef 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.

Benjamin Minixhofer, Jonas Pfeiffer, and Ivan Vulić. 2023. CompoundPiece: Evaluating and improving decompounding performance of language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 343–359, Singapore. Association for Computational Linguistics.

Christof Monz and Maarten De Rijke. 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.

Kristỳna Neumannová and Ondřej Bojar. 2023. The role of compounds in human vs. machine translation quality. In *Proceedings of Machine Translation Summit XIX, Vol. 1: Research Track*, pages 248–260.

Sandra Pappert and Bettina M. Bock. 2020. Easy-to-read german put to the test: Do adults with intellectual disability or functional illiteracy benefit from compound segmentation? *Reading and Writing*, 33:1105–1131.

Umesh Patil, Jesus Calvillo, Sol Lago, and Anne-Kathrin Schumann. 2025. Quantifying word complexity for Leichte Sprache: A computational metric and its psycholinguistic validation. In *Proceedings of the First Worshop Artificial Intelligence and Easy and Plain Language in Institutional Contexts*, Geneva, Switzerland.

Martin Riedl 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, California. Association for Computational Linguistics.

Pauline Schröter and Sascha Schroeder. 2017. The Developmental Lexicon Project: A behavioral database to investigate visual word recognition across the lifespan. *Behavior Research Methods*, 49(6):2183–2203.

Andrew Shin and Kunitake Kaneko. 2024. Large language models lack understanding of character composition of words. In *ICML 2024 Workshop on LLMs and Cognition*.

Petra Steiner. 2018. Merging the trees: Building a morphological treebank for german from two resources. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories (TLT16)*, pages 146–160. Association for Computational Linguistics.

Julia Suter, Sarah Ebling, and Martin Volk. 2016. Rule-based automatic text simplification for german. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*, pages –. Österreichische Computer Gesellschaft.

Don Tuggener. 2018. Evaluating neural sequence models for splitting (swiss) german compounds. In *SwissText*, pages 42–49.

University of Tübingen. 2024. Germanet compound word splitting resource. `https://uni-tuebingen.de/.../germanet-1/beschreibung/compounds/`. Accessed June 2025.

Marion Weller and Ulrich Heid. 2012. Analyzing and aligning German compound nouns. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2395–2400, Istanbul, Turkey. European Language Resources Association (ELRA).

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

Katharina Wellmann. 2020. Mediopunkt oder Binde-Strich? Eine Eyetracking-Studie. *Leichte Sprache–Empirische und multimodale Perspektiven*, pages 23–42.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online / Virtual. Association for Computational Linguistics.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 66–78, New Orleans, Louisiana. Association for Computational Linguistics.

Patrick Ziering 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, California. Association for Computational Linguistics.