Investigating Word Segmentation Techniques for German Using Finite-State Transducers

Gábor Pintér(✉), Mira Schielke, and Rico Petrick

Linguwerk GmbH, 01069 Dresden, Germany
{gabor.pinter,mira.schielke,rico.petrick}@linguwerk.com

Abstract. Word segmentation plays an important role in speech recognition as a text pre-processing step that helps decrease out-of-vocabulary items and lowers language model perplexity. Segmentation is applied mainly for agglutinative languages, but other morphologically rich languages, such as German, can also benefit from this technique. Using a relatively small, manually collected broadcast corpus of 134k tokens, the current study investigates how Finite-State Transducers (FSTs) can be applied to perform word segmentation in German. It is shown that FSTs incorporating word-formation rules can reach high segmentation performance with 0.97 precision and 0.93 recall rate. It is also shown that FSTs incorporating n-gram models of manually segmented data can reach even higher performance with accuracy and recall rates of 0.97. This result is remarkable considering the fact that the bottom-up approach performs on par with the expert system without requiring explicit knowledge about morphological categories or word formation rules.

Keywords: Word segmentation · Text processing · Morphology

1 Introduction

Agglutinative languages, such as Turkish or Japanese, are commonly reported to be challenging for automatic speech recognition (ASR), partly because the vocabulary of these languages cannot be effectively accounted for by a simple enumeration of words. Listing is impractical due to the highly productive derivational and inflectional morphology. This morphological characteristic may pose problems for speech recognition as the large number of word types can lead to high out-of-vocabulary rates in the pronunciation lexicon and high perplexities in language models.

German is not an agglutinative language, but its relatively complex inflectional system and its productive compounding characteristics raise problems similar to that of agglutinative languages [4–6,13,16]. For example, German adjective modifiers can have different endings according to the gender, number and case of the nouns they modify (e.g., *das kalt-e Bier* 'cold beer[NOM]' *dem kalt-en Bier* 'cold beer[DAT]' *kalt-es Wasser* 'cold water[ACC]'). Calculating

A. Karpov et al. (Eds.): SPECOM 2018, LNAI 11096, pp. 511–521, 2018.
https://doi.org/10.1007/978-3-319-99579-3_53

only with eleven forms per adjective (*null*, *-e -en -em -er -es -ste -sten -stem -ster -stes*) would require adding each adjective eleven times to the lexicon. Compounding can also considerably increase the vocabulary size in German as spelling conventions require most compounds to be written as single words, without any hints of morpheme boundaries. This practice results in such super-long word formations as the infamous *Donaudampfschifffahrtsgesellschaft* 'Danube Steamboat Shipping Company'. An apparent solution to these problems is to split up word forms, that is, to introduce some kind of word segmentation step that provides input for lexicon and language model related tasks.

There are several word segmentation techniques and tools available ranging from morphological analyzers to completely data-driven, unsupervised segmentation techniques. General purpose morphological analyzers, such as TAGH [7] for German, or CHASEN [11] for Japanese provide full-fledged, linguistically accurate morphological analysis, in which morpheme boundaries can be used as splitting points. Although it is not uncommon for studies to implement custom, morphology-based segmentation tools [2,14], the costs associated with the development and maintenance of general morphological analyzers is prohibitive. Languages without appropriate morphological analyzers can be processed by self- or semi-supervised, data-driven algorithms that identify sub-word units automatically, without relying on morphological information. Besides some sporadic, heuristically formulated attempts [10,17], MORFESSOR [3] has to be highlighted as an established data-driven segmentation tool, frequently occurring in studies concerned with sub-word models of speech recognition [18–20]. While data-driven tools are extremely convenient and their performance tend to improve with more data, they can produce unexpected errors, and their behavior is difficult to control.

The current study aims to briefly overview how Finite-State Transducers (FSTs) can be used for word segmentation, and provide a simple performance measure for the techniques introduced—using German data. FSTs can function as a convenient mechanism to segment words, and are often used in morphological analyzers. But FSTs can also operate using bottom-up information, for example in the form of n-gram models. This study introduces and compares two top-down and a range of bottom-up FST models for word segmentation. As preliminary experiments show, more morphological knowledge leads to better segmentation performance, but self-supervised approaches—with no morphological knowledge—can perform on par with expert systems.

2 Word Segmentation with Transducers

2.1 Morphological Analysis as Segmentation

The simplest word segmentation transducer can be constructed similarly to a two-level morphological parser [8], except that instead of underlying morphemes and features the output contains only the split input. Input and output labels share the same set of characters with extra segment boundary symbols (e.g. `'+'`) on the output side. The segmentation transducer is defined as a closure over all

acceptable segments. Figure 1 demonstrates a sample transducer that splits up the input compound *zeitraum* 'time period' into its components: *zeit+raum*.[1]

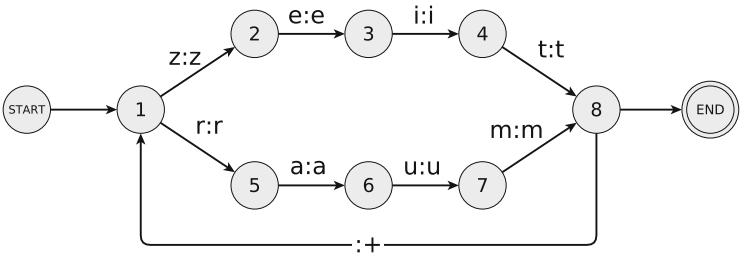

Fig. 1. A sample word segmenter FST.

A transducer with a simple closure over all lexical items, however, is not an effective segmenter, because it accepts any sequence of segments in any order. For example, the transducer above also accepts nonsense words like *zeitzeit* or *raum-raumraum*. This problem of over-generation can be addressed by incorporating word-formation constraints into the transducer. A widely utilized technique is to formulate constraints over morphological categories, such as prefixes and suffixes. Figure 2 displays a transducer that accepts prefixes only at the beginning of words and suffixes only at the end. For example, prefix *ab-* attaches only to left side of verbs, suffix *-ung* only to their right side (e.g. *ab+schaff+ung*).

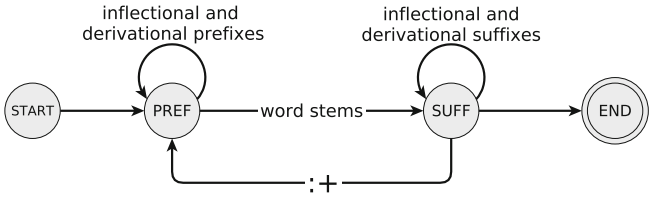

Fig. 2. Segmenter FST with morphological knowledge about prefixes and suffixes.

This naive prefix/suffix only approach also leaves plenty of room for over-generation. The system can be greatly improved by incorporating more fine-grained word formation rules through taking affix types, part of speech categories and other subcategorization features into consideration. Figure 3 represents a more sophisticated attempt for a morphological approach using various derivational and inflectional suffix types. Although discovering and implementing word formation rules is a tedious task, it can lead to remarkable segmentation performance as demonstrated by the German morphological analyzers such as TAGH [7].

[1] Words are lowercased for clarity. In German nouns start with capital letters, so the segmentation would be more correctly: *Zeitraum* → *Zeit+Raum*.

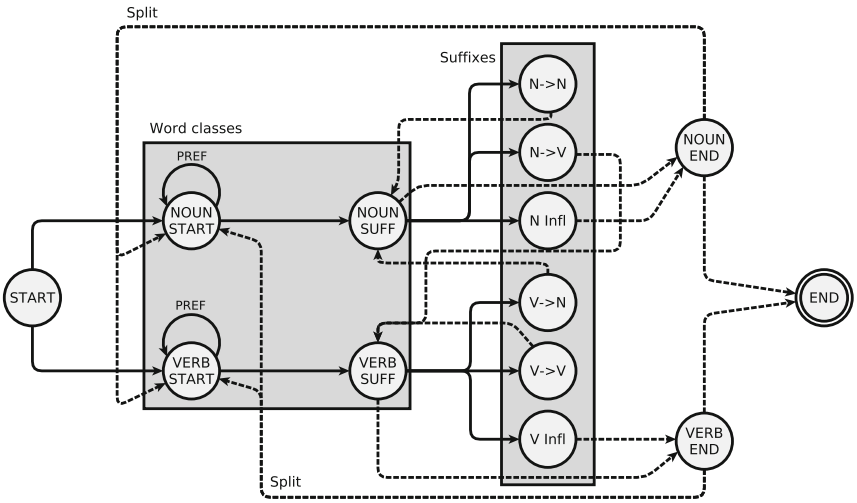

Fig. 3. Excerpt of a segmenter FST with expert morphological knowledge.

2.2 Supervised Word Segmentation with N-Grams

While morphological analyzers are obvious choices for segmenting words, the analysis they provide is not necessarily optimal for further processing. For instance, word stems combined with morphological features, instead of the written forms, do not provide optimal input for grapheme-to-phoneme algorithms (e.g. *wirfst* → *werfen* $< V >< 2 >< Sg >$). Also, too short morphemes can be sub-optimal for speech recognition tasks. These along with similar constraints can easily result in disagreements with the morphological analysis. Data-driven segmentation techniques can remedy this problem by providing means to learning arbitrary segmentation patterns. One way of doing this is by training n-gram models on segmented data. The idea of using n-gram-based segmentation as a text pre-processing step is an established method for Asian languages [9] but it has also been applied to German. Incorporation of n-gram models into segmentation FSTs is not a complicated task: FST-based language models are commonly used in various speech and language processing tasks [12,15]. A notable problem concerning the combination of segmentation and n-gram FSTs is that FST-based segmenters typically operate on characters, while n-gram models are defined over words or morphemes. This mismatch can be easily remedied by rewriting character sequences to morpheme labels in segmenters as demonstrated in Fig. 4.

As a first approximation, n-gram information can be integrated into the segmentation process in two steps. First a lattice of possible segmentations is created; second, this lattice is re-scored with an n-gram FST. The two-step approach, however, is slow and cumbersome. A more elegant approach is to merge the segmenter and the n-gram transducers into one FST. The merged—or composed—FST preserves the overall structure and weights of the n-gram

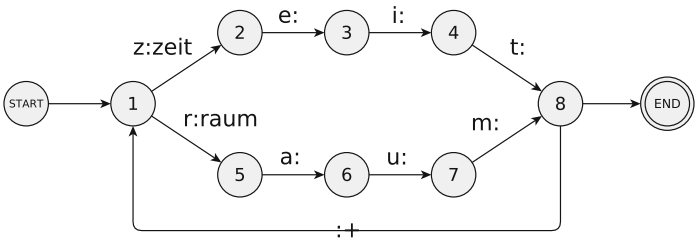

Fig. 4. Word segmenter from Fig. 1 with morpheme-level output labels.

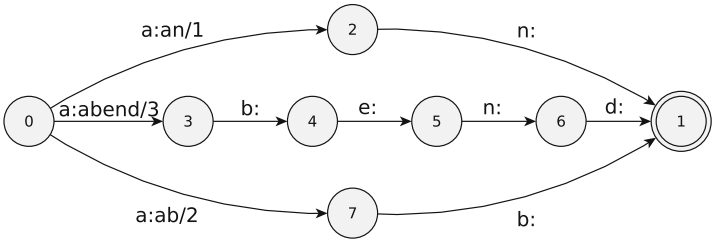

Fig. 5. Fragment of a transducer n-gram model with arcs for *an*, *ab* and *abend*. Epsilon output labels are omitted for clarity.

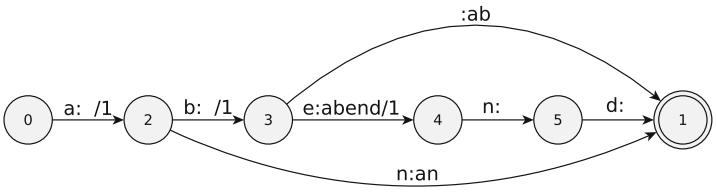

Fig. 6. Determinized and weight-pushed version of transducer in Fig. 5.

transducer. Figure 5 displays a fragment of an n-gram FST whose input morpheme arcs were replaced by characters.

Making the resulting transducer deterministic and sorting it by input label are useful optimization steps as they help reduce model size and enable faster search of arcs. Figure 6 represents an optimized version of the FST of Fig. 5. A disadvantage of these models is that they require custom search and composition algorithms, as their treatment of back-off and epsilon arcs is different from standard FST-based n-gram models.

3 Experiment

A series of experiments was conducted to compare the performance of top-down with bottom-up approaches to FST-based segmentation. The top-down approach was represented by two FST models that implemented different amounts—*Naive*

and *Expert* levels—of morphological knowledge. The bottom-up approach was associated with transducers that were based on n-gram models. A relatively small (134k) broadcast news corpus was used in a 10-fold cross-validation setup to evaluate segmentation performance. The folds were analyzed for perplexity and OOV rates as well as precision, recall and f-measure. In preparation of those calculations n-gram models with Katz smoothing were trained using 9 folds out of 10. The quality measures were calculated against the retained folds.

3.1 Corpus Data and Segmentation

As there is no standardized way to segment German text, there is also no standardized segmented corpus available. For development and testing purposes, German news broadcast text was collected from the Deutsche Welle news portal `www.dw.de` between early 2017 and early 2018. The texts collected, extracted from 207 news reports, was manually normalized and segmented. After normalization each file contained on average 646.7 tokens. Segmentation involved only the splitting up of words, no morphological categories or features were added. Some examples from the corpus are: *Woche-n-arbeit-s-zeit* 'hours worked per week', *Zahl-reich-e Häuser sind zer-stör-t* 'several houses are destroyed'. Admittedly, this manual segmentation diverged from traditional morphological analyses. For example, in order to keep the lexical model simple, words were kept together if segmentation would have produced alternative pronunciation, such as with *Häuser**→*Haus*+*er*.

3.2 Perplexity and OOV Rates

The corpus had a relatively small size of circa 134k tokens after text normalization. The segmentation has increased the token count to 198k, while it almost halved the type count. As expected, the segmented corpus had a lower perplexity of 14.1 compared to 21.4 of the original text (Table 1). Perplexity values were calculated using 3-gram language models with Katz smoothing. As shown in Fig. 7, word segmentation has achieved a considerable decrease in perplexity in unseen data: from 219.98 to 79.69 on average.

OOV type and token ratios were also calculated for the unseen folds. The weighted average of OOV tokens was 7.47%, which dropped to 1.89% after segmentation. A similar decrease was observed with types: from 20.88% to 9.30% on average. Values for each fold are seen in Fig. 8.

Table 1. Text-normalized and segmented news broadcast data.

Unit	Token counts	Type counts	Perplexity
word	133,664 (100.00%)	18,131 (100.00%)	21.377
morpheme	197,536 (147.79%)	9,934 (54.79%)	14.057

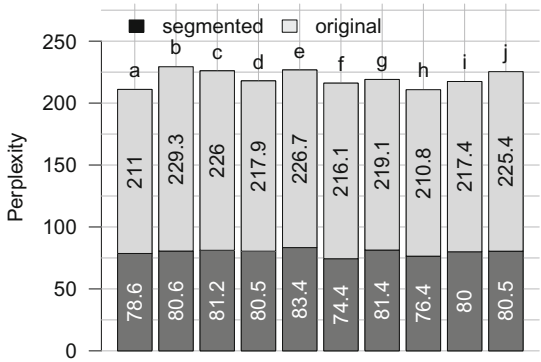

Fig. 7. Perplexity values in unseen folds with segmented and unsegmented text.

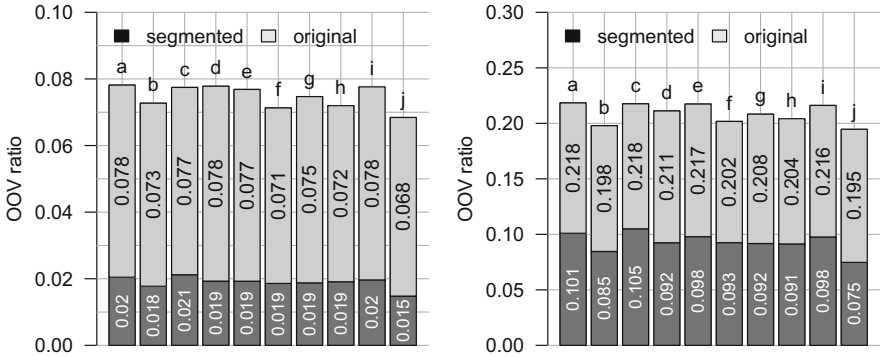

Fig. 8. Out-of-vocabulary ratios for tokens (left) and for types (right) in unseen folds.

3.3 Segmentation Models

A series of FST-based word segmenters was created following the concepts outlined in Sect. 2. A *Naive* model was created with an FST structure relying only on three morpheme categories: prefixes, suffixes and stems (cf. Fig. 2). Weights were set to a constant value for all segments to prefer longer chunks. The *Expert* model implemented a thorough, but non-exhaustive set of morphological rules (see Fig. 3). The weights were defined manually, based on experimentation. Both *Naive* and *Expert* models used around 80% of the corpus as a development set. Other sources, such as affix dictionaries and word lists were also used to define morpheme classes, transducer structure and weights.

In addition to the two top-down approaches, five data-driven models were created using 1- to 5-gram language models with Katz smoothing. In preparation of these models, first transducer-based n-gram models were trained using normalized and segmented text of the training folds. Next, word and morpheme labels in the n-gram transducer were replaced by character sequences on the input side (cf. Fig. 4). Finally, the transducers were determinized, minimized, and the weights were pushed forward for faster performance (cf. Fig. 6). A special, non-epsilon symbol was used as back-off arc label. All transducers and necessary tools were developed using OpenFst [1] and OpenGrm [15].

3.4 Results

Recall, precision and f-measure values were calculated to evaluate segmentation performance. The unseen data folds from the cross-validation setup were used as test sets for the n-gram models. For *Naive* and the *Expert* models, the separation of seen and unseen data was not consistent, as parts of the corpus were used—besides other sources—to manually discover morphological generalizations. For easier comparison the same "unseen" folds were used for all segmentation models. Table 2 summarizes the means of performance metrics over the test sets. A visual presentation of precision and recall values with medians are presented in Fig. 9.

3.5 Discussion

In terms of f-measures the best segmentation performance was achieved by the 2-gram model. This result, however, is not significantly different from other

Table 2. Segmentation performance: mean values over "unseen" folds.

	Naive	Expert	1-gram	2-gram	3-gram	4-gram	5-gram
Recall	0.9140	0.9324	0.9410	0.9684	**0.9686**	**0.9686**	**0.9686**
Precision	0.8739	0.9656	0.9468	**0.9673**	0.9661	0.9657	0.9657
f-Measure	0.8935	0.9487	0.9438	**0.9678**	0.9673	0.9671	0.9671

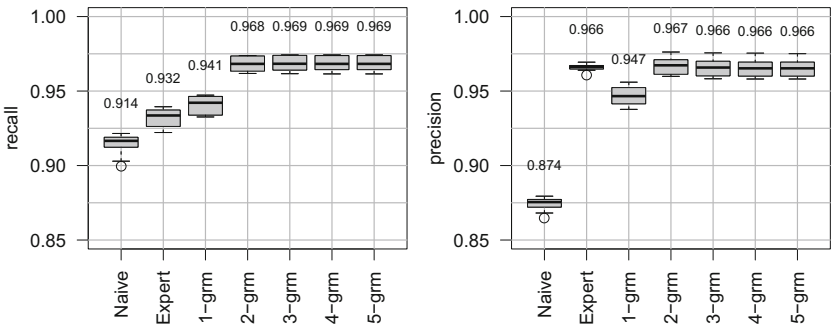

Fig. 9. Segmentation performance: recall (left) and precision (right).

higher order n-gram models. Unquestionably the *Naive* approach had the worst performance among the compared models. This result is not surprising given its over-simplified morphological model. Incorporating more sophisticated morphological knowledge proved to be useful as demonstrated by the performance improvements of the *Expert* model. Of course the question is if such expert systems are worth developing as n-gram models without morphological knowledge can deliver similar performance.

A closer look at the errors may influence the interpretation of the seemingly outstanding results. Almost half of OOV words in the unseen folds were named entities in non-affixed forms. These unsplit OOV items did not contribute to the evaluation as non-parsable input words were treated as single units.[2] Thus neither the reference nor the hypotheses had morpheme boundaries. Provided that words used for training are segmented correctly, the seen data together with non-splittable OOV items can account for the seemingly impressive results. The low error rates are attributable to the low number of multi-segment OOV items.

4 Conclusion

The goal of this article was to present a brief overview and a few examples of how FSTs can be used for word segmentation. The introduced top-down and bottom-up approaches, while performing well in the experiments, provided only a limited insight of what FSTs are capable of. For example, top-down models can easily be augmented with stochastic elements; or inversely, the n-gram approach can integrate morphological classes. It is also possible to detect word-embedded OOV tokens with fall-back arcs in combination with confidence measures. Orthogonal to the direction of these technical improvements, another straightforward extension of this research would involve evaluation of segmentation models in context. The presented low perplexity and OOV rates may imply better ASR performance, but the actual effect on recognition accuracy needs to be verified through experimentation. Although the current literature does not provide a conclusive answer, it seems that segmentation may lead to better ASR performance, but this gain may decrease with the increase of vocabulary size [19]. While we cannot answer questions related to speech recognition performance at present, we believe that our work provides a useful base for further studies concerning word segmentation using finite-state techniques.

[2] However, a few OOV words were falsely segmented into—typically short—morphemes, leading to errors (e.g. *Tories* → *Tor+ie+s*).

References

1. Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., Mohri, M.: OpenFst: A General and Efficient Weighted Finite-State Transducer Library. In: Holub, J., Žďárek, J. (eds.) CIAA 2007. LNCS, vol. 4783, pp. 11–23. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76336-9_3
2. Arisoy, E., Saraclar, M.: Compositional neural network language models for agglutinative languages. In: Interspeech 2016, pp. 3494–3498 (2016)
3. Creutz, M., Lagus, K.: Unsupervised discovery of morphemes. In: ACL Workshop on Morphological and Phonological Learning, pp. 21–30 (2002)
4. El-Desoky, A., Shaik, A., Schlüter, R., Ney, H.: Sub-lexical language models for German LVCSR. In: Spoken Language Technology Workshop, pp. 159–164 (2010)
5. El-Desoky, A., Shaik, A., Schlüter, R., Ney, H.: Morpheme level feature-based language models for German LVCSR. In: Interspeech 2012, pp. 170–173 (2012)
6. Geutner, P.: Using morphology towards better large-vocabulary speech recognition systems. In: IEEE International Conference on Acoustic, Speech Signal Processing, vol. 1, pp. 445–448 (1995)
7. Geyken, A., Hanneforth, T.: TAGH: a complete morphology for german based on weighted finite state automata. In: Yli-Jyrä, A., Karttunen, L., Karhumäki, J. (eds.) FSMNLP 2005. LNCS (LNAI), vol. 4002, pp. 55–66. Springer, Heidelberg (2006). https://doi.org/10.1007/11780885_7
8. Jurafsky, D., Martin, J.: Speech and Language Processing, 2nd edn. Prentice-Hall Inc., Upper Saddle River (2009)
9. Kang, S.-S., Hwang, K.-B.: A language independent n-gram model for word segmentation. In: Sattar, A., Kang, B. (eds.) AI 2006. LNCS (LNAI), vol. 4304, pp. 557–565. Springer, Heidelberg (2006). https://doi.org/10.1007/11941439_60
10. Larson, M., Willett, D., Köhler, J., Rigoll, G.: Compound splitting and lexical unit recombination for improved performance of a speech recognition system for German parliamentary speeches. In: Interspeech 2000, pp. 945–948 (2000)
11. Matsumoto, Y.: Easy to use practical freeware for natural language processing: morphological analysis system ChaSen. IPSJ Mag. **41**(11), 1208–1214 (2000)
12. Mohri, M.: Finite-state transducers in language and speech processing. Comput. Linguist. **23**(2), 269–311 (1997)
13. Nußbaum-Thom, M., El-Desoky, A., Schlüter, R., Ney, H.: Compound word recombination for German LVCSR. In: Interspeech 2011, pp. 1449–1452 (2011)
14. Renshaw, D., Hall, K.: Long short-term memory language models with additive morphological features for automatic speech recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 5246–5250 (2015)
15. Roark, B., Sproat, R., Allauzen, C., Riley, M., Sorensen, J., Tai, T.: The OpenGrm open-source finite-state grammar software libraries. In: ACL 2012 System Demonstrations, pp. 61–66 (2012)
16. Shaik, A., El-Desoky, A., Schlüter, R., Ney, H.: Feature-rich sub-lexical language models using a maximum entropy approach for German LVCSR. In: Interspeech 2013, pp. 3404–3408 (2013)
17. Shamraev, N., Batalshchikov, A., Zulkarneev, M., Repalov, S., Shirokova, A.: Weighted finite-state transducer approach to german compound words reconstruction for speech recognition. In: AINL-ISMW FRUCT, pp. 96–101 (2015)
18. Smit, P., Virpioja, S., Kurimo, M.: Improved subword modeling for WFST-based speech recognition. In: Interspeech 2017, pp. 2551–2555 (2017)

19. Tachbelie, M., Abate, S., Menzel, W.: Using morphemes in language modeling and automatic speech recognition of Amharic. Nat. Lang. Eng. **20**, 235–259 (2012)
20. Zablotskiy, S., Minker, W.: Sub-word language modeling for Russian LVCSR. In: Ronzhin, A., Potapova, R., Fakotakis, N. (eds.) SPECOM 2015. LNCS (LNAI), vol. 9319, pp. 413–421. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23132-7_51