Improving Speech Recognition of Compound-rich Languages

Prabhat Pandey[1], *Volker Leutnant*[1], *Simon Wiesler*[1], *Jahn Heymann*[1], *Daniel Willett*[1]

[1]Amazon Development Center, Aachen, Germany

{panprabh, leutnant, wiesler, jahheyma, dawillet}@amazon.de

Abstract

Traditional hybrid speech recognition systems use a fixed vocabulary for recognition, which is a challenge for agglutinative and compounding languages due to the presence of large number of rare words. This causes high out-of-vocabulary rate and leads to poor probability estimates for rare words. It is also important to keep the vocabulary size in check for a low-latency WFST-based speech recognition system. Previous works have addressed this problem by utilizing subword units in the language model training and merging them back to reconstruct words in the post-processing step. In this paper, we extend such open vocabulary approaches by focusing on compounding aspect. We present a data-driven unsupervised method to identify compound words in the vocabulary and learn rules to segment them. We show that compound modeling can achieve 3% to 8% relative reduction in word error rate and up to 9% reduction in the vocabulary size compared to word-based models. We also show the importance of consistency between the lexicon employed during decoding and acoustic model training for subword-based systems.

Index Terms: Speech Recognition, Subword Modeling, Compounding, German

1. Introduction

Large-vocabulary continuous speech recognition (LVCSR) of languages with productive word formation poses a challenge because of high number of word forms. For example, the vocabulary size shoots up enormously in German due to word compounding. Typically, these words are entity names or content words. For a low-latency and low-compute automatic speech recognition (ASR) system, it is important to control the size of vocabulary. One approach to address this problem is to use subword units as the modeling unit [1, 2, 3]. Recently, end-to-end ASR systems [4, 5] have been proposed which also typically use subword units. However, they also struggle to recognize proper nouns due to the lack of enough text data seen by the model [6, 7]. The subword-based systems segment words into smaller units, called subwords, and add some kind of markers to these subwords for the identification of word boundaries. The language model is trained on a mix of word and subword tokens. After the recognition is done, the markers are removed in the post-processing step to reconstruct the original words. Previous works have experimented with different data-driven segmentation approaches like Morfessor [8, 9], Greedy Unigram Segmentation [10] or Byte-Pair-Encoding [11]. One challenge with the subword-based system is to get the correct pronunciations of subword units. In [2], experiments were done with Finnish and Estonian languages for which there is a one-to-one mapping between graphemes and phonemes. Other approaches make use of grapheme-to-phoneme (G2P) models. However, as these data-driven segmentation approaches can potentially produce unpronounceable tokens, G2P is likely to produce incorrect pronunciations. Further, the G2P systems are typically trained on manually curated pronunciations of regular words. There exists few syllable-based segmentation approaches [12] which can avoid this problem, however, they need to be trained separately for each language. In [1, 3] subword units were derived by aligning the graphemes and phonemes of pronunciations of regular words. In this work, we focus on the compounding aspect. We propose a data-driven unsupervised method to identify and segment compound words. As we split compounds such that the segmented tokens are part of the vocabulary, we can use the manually curated lexicon for pronunciation of segmented tokens. We show that by splitting compounds prior to ASR training and joining them back after recognition helps in achieving 3 to 8% relative word error rate reduction (WERR) and results in 9% reduction of vocabulary size for German language. We also show that having consistency between lexicon employed during acoustic model training and ASR decoding can have large impact on the accuracy of subword-based ASR systems.

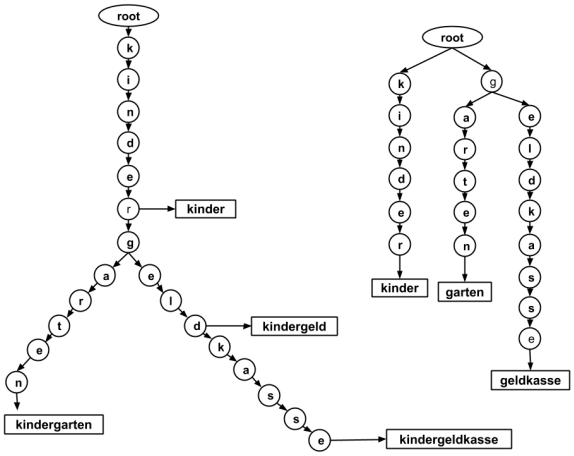

Figure 1: *The figure on the left is a prefix-tree constructed from full_vocab comprising of* {*kinder, kindergeld, kindergarten, kindergeldkasse*}. *The figure on the right is a prefix-tree constructed from segments_vocab consisting of* {*kinder, garten, geldkasse*}.

2. Compound Modeling

2.1. Segmentation

We use a data-driven approach to identify and segment compounds in the vocabulary. We consider all possible segmentation of a compound such that the segmented tokens (also referred as decompounds) are part of the vocabulary. However, the vocabulary of LVCSR task tends to contain spelling errors or multi-lingual words in its tail which may result into incorrect splitting of a compound. For example, it is not uncommon

Algorithm 1 merge(leftTree, rightTree)

Initialize merged $\leftarrow PrefixTree()$
for leftSubtree in leftTree.children() **do**
  label $\leftarrow$ leftSubtree.label
  if rightTree $contains$ label **then**
    rightSubtree $\leftarrow$ rightTree[label]
  else
    rightSubtree $\leftarrow PrefixTree()$
  end if
  merged[label] $\leftarrow merge$(leftSubtree, rightSubtree)
end for
merged.word = [leftTree.word, rightTree.word]
Return merged

to have the English word *no* in the German vocabulary which would result into incorrect segmentation of German compound *nomaden* $\rightarrow$ *no maden*. To address this, we restrict the segmented tokens to a smaller vocabulary consisting of frequently occurring words. We refer to this vocabulary as *segments_vocab* and the full vocabulary considered for identifying compounds as *full_vocab*. We first construct a prefix tree for words in *full_vocab* and *segments_vocab* independently as shown in Figure 1. We then merge the two trees as per the procedure described in Algorithm 1. The merged tree consists of all the paths in the *full_vocab* tree but word labels from both the trees, as shown in Figure 2. Eventually, *segment* procedure from Algorithm 2 is called. To address over-generation, some restrictions can be placed. For example, in order to avoid splitting like *erleben* $\rightarrow$ *er leben*, we only keep those words in *segments_vocab* which consists of more characters than a certain value. One can also chose to limit the maximum number of segments of a compound. In Section 2.3, we describe a method to further filter these segmentation rules based on lexicon.

2.2. Deterministic Splitting

The approach described in Section 2.1 can produce multiple segmentations of a compound. For example, two possible segmentation of *schlafzimmerlicht* is possible with a *segments_vocab* consisting of {*schlaf, zimmer, schlafzimmer, licht*}:

schlafzimmerlicht $\rightarrow$ schlaf zimmer licht
schlafzimmerlicht $\rightarrow$ schlafzimmer licht

One approach could be to split compounds with all possible segmentations [13]. On the other hand, for a n-gram based language model, this would mean distributing the counts among different segmentation forms. In order to have a deterministic splitting, we select the splitting form with the least number of segments. If there are still multiple options, we select the splitting form with the maximum sum of counts of its segments where the counts are calculated over all the learned segmentation rules.

2.3. Filtering Segmentation Rules based on Pronunciations

We earlier discussed in Section 1 how previously proposed subword modeling approaches fail to get correct pronunciations for subword units. When working with lexicon-based ASR system, this can have a significant impact on the accuracy. When we segment a compound, the concatenation of pronunciations of decompounds must result into the pronunciation of the compound. This can also be used to identify incorrect segmentation

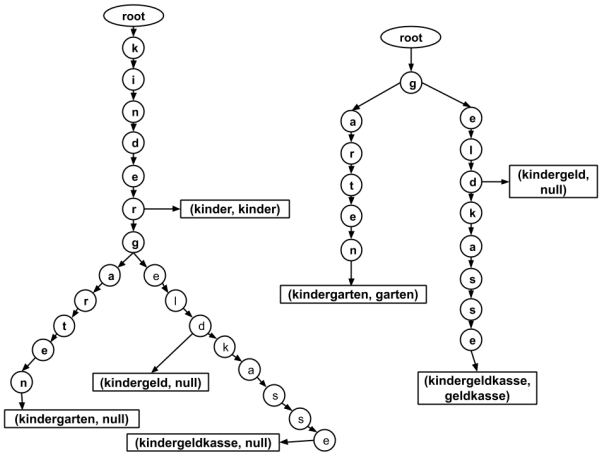

Figure 2: *The tree on the left is the output of merging the two trees of Figure 1. The tree on the right is the output of merging the subtree of node with the word label* {*kinder, kinder*} *in the left tree with the segments_vocab tree of Figure 1.*

Table 1: *Different marking styles used in subword modeling.*

Marking Style	Example
Left-marked (+m)	schlaf +zimmer +licht
Right-marked (m+)	schlaf+ zimmer+ licht
Both-marked (+m+)	schlaf+ +zimmer+ +licht
Word Boundary (<w>)	<w> schlaf zimmer licht <w>

rules. We discard all such segmentation rules where not all the pronunciations of a compound can be constructed by concatenating pronunciations of its decompounds.

2.4. Marking Styles

In order to be able to combine subword units in the post-processing step quickly and deterministically, usually some kind of markers are used. Table 1 shows four different kinds of marking styles typically used. Three of them add markers to the subword tokens whereas word boundary marking style (<w>) introduces an additional token per word. As it introduces an extra token per word, it requires a higher n-gram order for language modeling. m+ and +m marking styles are different than +m+ style as they split a word into marked as well as unmarked tokens. As all the rest of the vocabulary is also unmarked, these marking styles can theoretically construct more new words than +m+ marking style. There is no conclusive evidence in previous works about which marking style performs better. [2] reports that +m+ marking style attains the best performance on ASR task on Finnish and Estonian languages. In [14], left-marked style (+m) was compared against word boundary style (<w>) and it was shown that +m style outperforms <w> style for Hungarian ASR. In [15], experiments were done with all the four marking style and was shown that m+ style performs best for ASR with large vocabularies while <w> works best on smaller vocabularies.

Algorithm 2 segment(mergedTree, segmentsTree, segmentNum=0)

Initialize segmentationList ← []
Initialize subtrees ← {$subtree$ | $subtree$ ∈ mergedTree and $subtree$.root.word[1] != null}
for subtree in subtrees **do**
  if segmentNum > 0 and subtree.word[0] != null **then**
    segmentationList.Add([subtree.word[1]])
  end if
  if $isNotLeaf$(subtree) **then**
    mergedTree ← $merge$(subtree, segmentsTree)
    for segments in $segment$(mergedTree, segmentsTree, segmentNum+1) **do**
      segmentationList.Add([[subtree.root.word[1]] + segments])
    end for
  end if
end for
Return segmentationList

Table 2: *Handling of position-dependent phonemes for compound modeling. For illustration purpose, graphemes of the words are used as its pronunciation.*

Marking Style	Token	Pronunciation
-	schlafzimmerlicht	s_B c h l a f z i m m e r \ l i c h t_E
+m+	schlaf+	s_B c h l a f
	+zimmer+	z i m m e r
	+licht	l i c h t_E
m+	schlaf+	s_B c h l a f
		s c h l a f
	zimmer+	z i m m e r
		z_B i m m e r
	licht	l_B i c h t_E

2.5. Lexicon

In a hybrid WFST-based speech recognition system [16], the search graph is constructed by composing four FSTs: H which maps states of the Hidden Markov Model to context-dependent phonemes, C which transduces context-dependent phonemes to context-independent phonemes, L which transduces phoneme sequence to words, and G which is constructed from the language model (LM). The output symbols of L, which is the set of words in the lexicon, should match input symbols of G, which is vocabulary of the language model. Apart from decoding lexicon, there is another lexicon which is used during acoustic model (AM) training to generate forced-alignments.

Modeling pronunciations for subword-based systems which use position-dependent phonemes is not straight-forward. For example, in Kaldi system [17], there are four variants of a phoneme depending on whether it occurs in the beginning, middle, end of the pronunciation or is a single-phoneme pronunciation. Suffixes *_B* and *_E* are added to the first and last phoneme of a pronunciation, respectively. If we treat the subwords as regular words, we would end up modifying the position-dependent phonemes in the pronunciation of the original word. This also makes pronunciation of marked and unmarked tokens with the same root indistinguishable. In [2], this problem was addressed by modifying the lexicon FST to have separate paths for subword and regular word tokens. As mentioned in Section 2.4, +m and m+ marking styles are more capable of recognizing out-of-vocabulary (OOV) words but the restrictive lexicon FST of [2] can limit this potential greatly. In this work, we keep the original structure of the L FST and address the problem by handling position-dependent phonemes as per Table 2. For +m+ style, suffixes are added to the boundary phonemes of pronunciation of marked tokens depending on whether the marker is on the left or right or both side of the token. For m+ and +m styles, we add two pronunciations for the marked tokens as they can appear in the start or middle of a word in case of m+ style and in the middle or end of a word in case of +m. The pronunciations for unmarked tokens are handled similar to any regular word.

Table 3: *Comparison of relative reduction in the vocabulary size against the size of segmentation rules relative to the initial vocabulary size.*

Relative Size of Segmentation Rules (%)	Relative Reduction in Vocabulary Size (%)		
	+m	m+	+m+
4e−3	3e−3	2e−4	−5e−4
1.11	0.85	0.72	0.47
2.26	1.71	1.58	1.15
5.69	4.09	4.01	3.23
7.84	5.35	5.28	4.38
14.23	8.57	8.56	7.36

3. Experiments

3.1. Setup

All the experiments were done for German language. The LM training data comprising of various internal and external text data sources, was used to learn compound segmentation rules. All the n-gram language models are 4-gram models and the hypothesis generated by the first-pass ASR system is rescored by a Neural Language Model (NLM). The training data for rescoring NLM is consistent with the first-pass n-gram LM in all the experiments. We experimented with three marking styles: left-marked (+m), right-marked (m+) and both-marked (+m+). We left out word boundary style (<w>) as it requires higher n-gram order compared to the rest of marking styles in order to have a fair comparison. We measure the accuracy of ASR systems in terms of word error rate (WER). We report relative word error rate reduction with respect to baseline word-based model. Two

Table 4: *Comparison of effective OOV rate on the vocabulary of Knowledge test set against the size of segmentation rules relative to the initial vocabulary size. The OOV rate of word-based model is 4.20%.*

Relative Size of Segmentation Rules (%)	Effective OOV rate (%)		
	+m	m+	+m+
2.26	2.62	2.39	3.69
5.69	2.45	2.24	3.52
7.84	2.33	2.12	3.40
14.23	2.10	2.01	3.20

Table 5: *Accuracy results of different compound ASR systems compared to word-based system. The AM remains same in all the systems. As it is trained on token-type of word-based model, the decoding lexicon for compound systems is inconsistent with the AM training lexicon.*

Marking Style	Generic WERR (%)	Knowledge WERR (%)
Left-marked (+m)	1.1	3.2
Right-marked (m+)	1.4	3.7
Both-marked (+m+)	0.9	1.8

internal test sets were used, one is *Generic* speech recognition task and the other is *Knowledge* test set consisting of queries with the intent of seeking information about entities.

3.2. Vocabulary Size

Table 3 shows the behaviour of reduction in vocabulary size compared to the size of learned segmentation rules. As +m+ marking style introduces markers to all segments, the vocabulary size reduction is smaller compared to m+ and +m marking styles. In fact, it may result into vocabulary increase when working with a small vocabulary and only a handful of segmentation rules are learned. +m style results into slightly more reduction in vocabulary size compared to m+ style. This can be attributed to the fact that there is a higher degree of variation in prefixes than suffixes of compound words in German. In Table 4, we compare the OOV rate of the compound models on the vocabulary of *Knowledge* test set. As the compound models can potentially recognize words beyond its vocabulary, we calculate the *effective* OOV (EOOV) rate. We consider only such words as OOV which can't be constructed by the vocabulary of compound models. Consistent with the expectation that m+ and +m marking styles are more capable to construct new words, we see lower EOOV rate compared to +m+ style. m+ style has marginally lower EOOV rate than +m style.

3.3. ASR Results

Table 5 compares the WERR observed when using compound language models with different marking styles over word-based model. The acoustic model remains same for all the systems and is trained on token-type of word-based system. For compound ASR systems, there is a mismatch between the decoding lexicon and the lexicon used in AM training. In the decoding lexicon, the position-dependent phonemes in the pronunciations

Table 6: *Accuracy results of different compound models compared to word-based model. Both AM and LM were trained with consistent token-type and so, there is consistency between the AM training and decoding lexicon.*

Marking Style	Pronunciation Modeling	Generic WERR (%)	Knowledge WERR (%)
+m	Word	2.5	4.3
	Subword	2.2	5.5
m+	Word	3.1	6.2
	Subword	2.7	8.2
+m+	Word	1.8	4.5
	Subword	1.0	4.8

of decompounds are handled like any regular word with both *_B* and *_E* suffixes added to the boundary phonemes. On the other hand, as the tokens for AM training are not segmented, the phonemes at the edge of segments will not have any suffixes in the AM training lexicon. For example, the pronunciation for *schlafzimmerlicht* in the AM training lexicon would be like *s_B c h l a f z i m m e r l i c h t_E*. Whereas in the decoding lexicon, for +m+ marking style, the pronunciation for the token sequence *schlaf+ +zimmer+ +licht* would be like *s_B c h l a **f_E z_B** i m m e **r_E l_B** i c h t_E*. To address this, we retrained the AM for each marking style with a lexicon consistent with the decoding lexicon. We tried two approaches for position-dependent phonemes. In the first approach, we treat marked and unmarked tokens in the same way as a regular word and add suffixes to both boundary phonemes. We call this *word-based* pronunciation modeling. The other approach is based on Section 2.5 and Table 2 which we call as *subword-based* pronunciation modeling. The results are shown in Table 6. It can be seen that the accuracy improves significantly for compound ASR systems when we match the AM training lexicon and the decoding lexicon. m+ marking style, which attains the lowest *effective* OOV rate, achieves the best performance with 3.1% WERR on *Generic* test set and 6.2% WERR on *Knowledge* test set with *word-based* pronunciation modeling. Using *subword-based* pronunciation modeling results in reduced WERR compared to *word-based* counterpart on the *Generic* test set but performs better on the *Knowledge* test set.

4. Conclusions

We proposed a data-driven approach for identifying and segmenting compound words in textual data. We showed that compound modeling can improve ASR performance over word-based models, especially on tail entities and reduces the size of ASR vocabulary. We also showed the importance of having consistency between decoding and AM training lexicon. We compared different marking styles and showed that right-marked (m+) style, which has the lowest OOV rate, achieves the best performance for our setup. Here are some examples of the compounds which were OOV to word-based model and recognized by the compound ASR models: *privatkliniken, burgergurken, lebensmitteltabelle, hauttemperatur*. Compound modeling not only enables recognition of OOV compound words but also improves recognition of rare compounds which have low language model probabilities in the word-based models.

5. References

[1] H. Xu, S. Ding, and S. Watanabe, "Improving end-to-end speech recognition with pronunciation-assisted sub-word modeling," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7110–7114.

[2] P. Smit, S. Virpioja, M. Kurimo *et al.*, "Improved Subword Modeling for WFST-Based Speech Recognition." in *INTERSPEECH*, 2017, pp. 2551–2555.

[3] M. A. B. Shaik, A. E.-D. Mousa, R. Schlüter, and H. Ney, "Hybrid language models using mixed types of sub-lexical units for open vocabulary German LVCSR," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[4] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[5] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[6] A. Kannan, Y. Wu, P. Nguyen, T. N. Sainath, Z. Chen, and R. Prabhavalkar, "An analysis of incorporating an external language model into a sequence-to-sequence model," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 1–5828.

[7] D. Zhao, T. N. Sainath, D. Rybach, P. Rondon, D. Bhatia, B. Li, and R. Pang, "Shallow-Fusion End-to-End Contextual Biasing," *Proc. Interspeech 2019*, pp. 1418–1422, 2019.

[8] M. Creutz and K. Lagus, "Unsupervised discovery of morphemes," in *Proceedings of the ACL-02 workshop on Morphological and phonological learning-Volume 6*. Association for Computational Linguistics, 2002, pp. 21–30.

[9] ——, "Unsupervised models for morpheme segmentation and morphology learning," *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 4, no. 1, pp. 1–34, 2007.

[10] M. Varjokallio, M. Kurimo, and S. Virpioja, "Learning a subword vocabulary based on unigram likelihood," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2013, pp. 7–12.

[11] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1715–1725.

[12] B. Möbius, "Word and syllable models for German text-to-speech synthesis," in *The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*, 1998.

[13] T. Kudo, "Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 66–75.

[14] A. Varga, B. Tarján, Z. Tobler, G. Szaszák, T. Fegyó, C. Bordás, and P. Mihajlik, "Automatic Close Captioning for Live Hungarian Television Broadcast Speech: A Fast and Resource-Efficient Approach," in *Speech and Computer: 17th International Conference, SPECOM 2015, Athens, Greece, September 20-24, 2015, Proceedings*, vol. 9319. Springer, 2015, p. 105.

[15] P. Smit, S. R. Gangireddy, S. Enarvi, S. Virpioja, and M. Kurimo, "Aalto system for the 2017 Arabic multi-genre broadcast challenge," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 338–345.

[16] M. Mohri, F. Pereira, and M. Riley, "Speech recognition with weighted finite-state transducers," in *Springer Handbook of Speech Processing*. Springer, 2008, pp. 559–584.

[17] P. Daniel, G. Arnab, B. Gilles, B. Lukas, G. Ondrej, G. Nagendra, H. Mirko, M. Petr, Q. Yanmin, S. Petr *et al.*, "The Kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584, 2011.