Comparison of Subword Segmentation Methods for Open-vocabulary ASR using a Difficulty Metric

Abbas Khosravani[1], *Claudiu Musat*[2], *Philip N. Garner*[1], *Alexandros Lazaridis*[2]

[1]Idiap Research Institute, Switzerland
[2]Data, Analytics & AI Group — Swisscom AG, Switzerland
name.lastname@{idiap.ch,swisscom.com}

Abstract

We experiment with subword segmentation approaches that are widely used to address the open vocabulary problem in the context of end-to-end automatic speech recognition (ASR). For morphologically rich languages such as German which has many rare words mainly due to compound words, there is an increasing interest in subword-level word representation based on, e.g., byte-pair encoding and unigram language model. However, we are not aware of any systematic comparative analysis of different approaches. To this end, we propose a framework which estimates a difficulty score of a test utterance for the ASR model based on an out-of-vocabulary metric. Using this framework we run experiments on several subword segmentation approaches, which provides us with comparative analysis on the strengths and weaknesses of them. For the ASR model, we employ a fully convolutional sequence-to-sequence encoder architecture using time-depth separable convolution blocks and a lexicon-free beam search decoding with n-grams subword language model. Additionally, we leverage multiple models with different word representations to investigate their impact on ASR performance.

Index Terms: speech recognition, end-to-end, open-vocabulary, subword segmentation, German language

1. Introduction

We are interested in general in automatic speech recognition (ASR) for German, and ultimately for Swiss German. German is characteristically highly inflected with a large vocabulary. Compound words play a significant role. In traditional ASR, these characteristics typically lead to large pronunciation lexicons and high out of vocabulary (OOV) rates. In Swiss German, these challenges are perhaps, on the one hand, eased slightly by the simpler grammar, but on the other hand, made worse by dialectical variation, lack of standard orthography, and prevalence of code switching. In such environments, lexicon free approaches are clearly attractive.

Although classical ASR models still dominate end-to-end systems on common benchmarks, the latter have increasingly seen competitive results, approaching state-of-the-art performance when using more training data, and regularization through data augmentation. The requirement for a handcrafted pronunciation dictionary, designed using linguistic knowledge to map words to phoneme sequences, on the other hand, has always been a problem for conventional ASR systems, especially for languages without such resources. However, end-to-end methods directly model the posterior distribution, $p(W|X)$, of a word sequence W given a speech feature sequence X. To be able to handle the out-of-vocabulary problem, it has become increasingly common to use a subword-level word representation for the language output sequence. Examples include character [1] or word-pieces which are most often implemented using the byte-pair encoding (BPE) [2] or unigram language model [3] techniques, originally developed for machine translation. Although character representation does not lead to any OOV problem, there are still advantages to using a larger vocabulary of subword units as opposed to characters [4]. Finding the best subword-informed word representation remains an open research problem.

Recently, it has been shown that subword regularization techniques, which generate multiple subword segmentations based on either a unigram language model [3] or stochastic BPE [5], produce large gains over BPE as a deterministic subword segmentation approach for machine translation baselines. This idea has been used in the context of speech recognition and implemented in recent speech recognition frameworks [6]. In [4] it has been shown that subword regularization produces significant gains over the unregularized segmentation using an attention-based ASR model. More recent works also use this regularization technique to improve the generalization of the ASR model [7, 8]. However, we are not aware of any comparative analysis on different subword segmentation approaches.

In a lexicon-based ASR system, the OOV rate of the test set can be considered as a drawback of the system. There are two reasons for this: It gives a lower limit to the error rate that can be achieved, and defines tokens that should be considered as missing information from the system. Ultimately, if the OOV words are important for covering the domain of the ASR system, they should be added to the system in an adaptation scenario. On the other hand, in a lexicon-free system, the OOV rate is important as it is the metric that we seek to reduce; however, it is not obvious how to define it. Of course, it could be done on the ground truth, but would penalise phrases that only differ in, say, conjugation or compounded form, that subwords could easily handle. We propose to measure the *difficulty* of a test set in terms of the ratio of BPE tokens to ground-truth words, where the BPE tokens are those from the training set. Intuitively, if this figure is unity (or less), then the test utterance is completely represented at word level (or phrase level) by the training data and the task has low difficulty. If, however, the ratio is greater than unity, the utterance is not well represented by the training set and the task is difficult.

Our experiments on different evaluation datasets show that this framework provides a good measure of difficulty for test utterances, and therefore provides us with a good tool to analyze different subword segmentation approaches in terms of their effectiveness on various evaluation scenarios. In this study, we investigate three hypotheses:

H1. The proposed framework can measure the difficulty level of an evaluation dataset.

H2. Using this framework, we can study the strengths and weaknesses of different subword segmentation techniques so as to choose a proper one for a particular evaluation scenario.

H3. Using this framework, it is possible to combine different segmentation techniques to improve ASR performance.

The experiments conducted in this study suggest that the proposed framework provides a good measure to compare and analyze different subword segmentation strategies. The remainder of this paper is organized as follows. Section 2 describes the proposed framework and different subword segmentation strategies. Our experimental setup including data and models is presented in Section 3. The test of the hypotheses and analysis is given in Section 4. Finally, Section 5 concludes the paper and provides insight into future work.

2. Methodology

We first present the popular techniques for subword-informed word representations and then introduce the proposed framework for measuring the difficulty of evaluation utterances for ASR system.

2.1. Subword Segmentation

Byte Pair Encoding (BPE) segmentation [2] which is based on a data compression principle, generates a unique subword sequence for each word. It is an iterative procedure which starts with a sequence of characters as tokens and at each step it merges the most frequent pair into a new token. The frequency is computed using a training text dataset, usually the acoustic data transcription, and the merge operations are added to the merge table in order. This is done until the desired vocabulary size is reached. To provide a segmentation for a new word, the same merge table is used to perform merge operations in order on its character sequence.

A recent technique which is more flexible than BPE, is based on a probabilistic language model, and can generate multiple segmentations with associated probabilities for each word; this is essential for subword regularization [3]. This segmentation technique, based on the *unigram language model* (ULM) has been shown to make both Neural Machine Translation (NMT) and ASR models more robust [3, 4].

To overcome the deterministic nature of BPE and generate the multiple segmentations required for subword regularization, in [5] the authors proposed to randomly drop the merge operations in BPE procedure which leads to producing multiple segmentations within the same fixed BPE framework. The authors showed that this *BPE-dropout* outperforms BPE on a wide range of translation tasks.

2.2. Framework

The proposed framework provides a difficulty score for each evaluation utterance based on the transcription information. The transcription is not available for real evaluation scenario, however, we will use it to compare and analyze different subword segmentation techniques. We will later show that, using the transcription provided by the ASR system we can estimate this difficulty score.

In order to compute this score, we use the same data compression technique as in byte-pair encoding [2], but unlike BPE which usually works at word level, we cross word boundaries and split the whole transcription into individual characters as initial tokens. To keep the notion of words, we add a special word separator symbol (e.g., underscore) to the beginning of each word in the training as well as evaluation transcriptions. We start by iteratively merging the most frequent pair of tokens. The frequency of pairs are computed using the transcription of data used to train the ASR system. A merge operation is performed only if the frequency is more than a specified threshold. For simplicity, we can set this threshold to zero which means a pair of tokens is merged only when at least one combination of the tokens has been observed during training. We repeat this process until no more merge operations are possible. We then divide the final number of tokens by the number of words in the transcription to get the difficulty score. This procedure is described in Algorithm 1.

Algorithm 1: Framework for measuring difficulty

```
Input: Train and test transcriptions, a threshold
Output: Difficulty score
Add a special symbol to each word in train and test
  transcriptions;
$nwords \leftarrow$ Number of words in a test transcription;
$tokens \leftarrow$ Split a test transcription into characters;
while True do
    if $size(tokens) = 1$ then
        break;
    end
    $pairs \leftarrow$ Compute frequency for each pair of
      tokens using the training transcription;
    $pair, freq \leftarrow max(pairs)$;
    if $freq > threshold$ then
        $tokens \leftarrow$ Apply merge for $pair$;
    else
        break;
    end
end
$score \leftarrow size(tokens)/nwords$;
```

If the number of words in the transcription is the same as the number of tokens generated, the difficulty score will be one. In this case the tokens are usually the same as the words in the transcription. However, if the number of tokens is more than the number of words, it is likely that there are words in the transcription that were never seen during training. Therefore, the higher the number of tokens, the more difficult it would be for the ASR system to transcribe it as the probability of OOV words is higher. On the other hand, if the number of tokens is less than the number of words, it is more likely that there are phrases or even sentences being observed during training. Finally, if the number of tokens equals one, then an utterance with the same transcription has already been used during training. In Section 3, we show that the lower the difficulty score is for an evaluation utterance, the easier it would be for ASR system to transcribe it.

3. Experimental Setup

3.1. Speech Data

Compared to, say, English, there are relatively few speech corpora available for German. Fortunately, some efforts have been made recently to collect and contribute such resources for sustainable research [9, 10, 11, 12]. Our experiments are conducted on a $\sim$737 hour training set consisting of 0.5 million German utterances. The training and evaluation utterances come from different open-source German corpora. Since the focus of this study is on analysis of subword segmentation approaches rather than domain mismatch, we design an evaluation plan to reflect this goal. We uniformly select ∼100 hours of speech data from three different German corpora to include a diverse range of topics, speakers and difficulty. Table 1 gives an overview of the data used in our experiments. In the following, we also briefly describe each data resources.

Table 1: *The amount of training and evaluation data used in our experiment.*

	Training		**Evaluation**	
Corpus	Speech	Speakers	Speech	Speakers
SWC-de	111h	221	32h	72
M-AILABS-de	195h	–	34h	–
CV-de	430h	7422	36h	154
	737h		102h	

The *Spoken Wikipedia Corpora* (SWC) [13, 9] is a large collection of speech read by volunteers covering a broad variety of Wikipedia topics under a free license. It is constantly expanding and evolving and is of considerable size for several languages. The German corpus or SWC-de includes 1010 articles with 249h of aligned speech from 288 readers. Due to the encyclopedic nature of the articles and diverse range of topics and large vocabulary size, this corpus is attractive for our study. Moreover, the articles are read completely by volunteers and sound more natural than those collected in controlled conditions. Recent work found this corpus to be helpful for improving ASR performance [14].

The *M-AILABS* resource was distributed by Munich Artificial Intelligence Laboratories[1] under a non-restrictive license and comprises hundreds of hours of speech audio in nine different languages taken from non-professional audio-books of the LibriVox project [11]. Although it contains a wide range of prosodic variation, it lacks speaker variability as the majority of audio-books were spoken by only a few speakers, making it not a good resource for speech-to-text applications. The German set includes ∼237h of audio clips varying in length from 1 to 20 seconds.

The *Common Voice* (CV) corpus [10] is a multilingual collection of transcribed speech data which was collected and validated using crowdsourcing; it intends to provide a free resource for speech technology research and development. It is an on going project and so far it includes 2,500 hours of collected speech data from 50,000 individuals in 38 different languages. The German set (CV-de), includes ∼370,000 validated audio files or a total ∼470 hours of data from 7600 individuals.

3.2. ASR Model

We use wav2letter++, an ASR framework designed from the outset to support end-to-end paradigms [6]. It supports several end-to-end sequence models including sequence-to-sequence models with attention (S2S) [15].

We incorporate a sequence-to-sequence model which has an encoder-decoder architecture using time-depth separable (TDS) convolution blocks [7]. In [16], it was shown that this TDS convolution block generalizes much better than other deep convolutional architectures and requires fewer parameters to train. This generalization is mainly due to some form of regularization including, dropout [17], label smoothing [18] and subword regularization [3]. We fix the network architecture for all experiments and use 12 TDS blocks with dropout and kernel size of 21×1 in three groups and set the number of channels in each group to (10, 14, 18) resulting in 39M parameters. We use a key-value attention [7] mechanism and an encoder of dimension 512. The model is trained using the seq2seq criterion for 75 epochs using SGD and the learning rate initialized to 0.05. We also use 80-dimensional log-mel features, computed with a 25ms window and 10ms frame shift.

3.3. Decoding and Language Modeling

The wav2letter++ decoder support both lexicon-based and lexicon-free decoding. The lexicon-free beam-search decoder uses a word separator which is predicted as a normal token and can also be part of a token to split the sequence of tokens into words. Therefore during training there is no notion of words. The decoder also supports different types of language models to provide LM score (log-probability) accumulated together with AM scores for a one-pass beam decoding. In our experiments we use 6-gram word-piece LM which is trained with KenLM [19] on 8M sentences of German text. They include texts from German Wikipedia (63%), European Parliament transcriptions (22.4%), and crawled German sentences (14.6%) from the Internet. The perplexity of our LM varies for each subword segmentation approach but is around 100 on average. All text was normalized the same way as the training transcription. We use a beam size of 40, beam threshold of 18, tokens beamsize of 15 and tune the LM weight on a development set.

4. Results and Analysis

4.1. Measure of evaluation difficulty

The first experiment is designed to support the first hypothesis, that is, we show that the proposed framework can provide a measure of difficulty for the evaluation utterances. To achieve this, we train three ASR models with different word representations. The first system is a word-based ASR model with 16K words comprising ∼90% of all the words in the training transcription and a 4-gram word-based language model. To handle OOV, we use a character representation for all other words. For the second system we use only characters as output units along with a 20-gram character-based LM. Finally, the third system uses subwords as output units. We train a BPE model with 8K subword units on the training transcription and generate a lexicon with subword representation for each word in the training transcription. As explained in Section 2.2, we compute a difficulty score for each evaluation utterance and then classify them into different level of difficulties, in a way that each level includes significant number of utterances. Table 2 presents the results.

It is clear from the results that subword segmentation provides superior performance over either character or word level models. We can also observe that as the difficulty level increases, the performances of all models drop significantly. The difficulty measure correlates well with the actual ASR performance, which supports the hypothesis that the proposed measure is suitable to the difficulty level of an evaluation dataset in the absence of a lexicon. We can see that CV and SWC have the easiest and more difficult evaluation utterances, respectively.

[1] https://www.caito.de/2019/01/the-m-ailabs-speech-dataset

Table 2: *ASR performance results in terms of WER (%) for different word representation and evaluation difficulty level.*

			Word Representation		
Difficulty	#Words	Top Corpus	Char	BPE	Word
$0.0 - 0.2$	170k	CV (99%)	5.10	**2.07**	3.67
$0.2 - 0.4$	27k	CV (97%)	5.35	**2.42**	3.10
$0.4 - 0.6$	42k	CV (35%)	11.1	**6.77**	7.98
$0.6 - 0.8$	230k	M-AILABS (54%)	19.5	**9.66**	12.9
$0.8 - 1.0$	243k	SWC (53%)	27.3	**13.8**	20.3
$1.0 - 1.2$	33k	SWC (62%)	34.4	**21.7**	29.7
$1.2 - 1.5$	13k	SWC (79%)	34.9	**27.4**	34.5
$1.5 - 2.0$	3.1k	SWC (91%)	51.6	**42.4**	48.4
$2.0 - \infty$	1.2k	SWC (98%)	90.0	**69.0**	71.3
All	762k		18.9	**9.93**	13.9

4.2. Subword segmentation analysis

To test the second hypothesis, we train two new models using subword regularization based on ULM [3] and stochastic BPE [5]. In [4], it was shown that regularization helps the generalization of ASR model. Using our framework, we want to analyze the effect of regularization on the performance of the ASR system. Similarly, we set the number of subword units to 8K but generate a lexicon with 10-best subword segmentation for each word in the training transcription. During training, for each word the best representation is used but we uniformly sample over other alternatives with a small probability. Based on our experiments, we set this probability to 0.05. We also train an unregularized subword model using the best segmentation of the ULM approach. Table 3 shows the results.

Table 3: *ASR performance results in terms of WER (%) with 95% confidence interval for regularized and unregularized subword models.*

	Unregularized		**Regularized**	
Difficulty	BPE	ULM	BPE	ULM
$0.0 - 0.2$	**2.07** ±0.07	2.41±0.07	2.87±0.08	3.03±0.08
$0.2 - 0.4$	**2.42**±0.18	2.76±0.19	3.08±0.21	3.34±0.21
$0.4 - 0.6$	**6.77**±0.24	7.23±0.25	7.05±0.25	6.76±0.24
$0.6 - 0.8$	9.66±0.12	9.42±0.12	8.98±0.12	**8.63**±0.11
$0.8 - 1.0$	13.8±0.14	13.3±0.14	12.0±0.13	**11.9**±0.13
$1.0 - 1.2$	21.7±0.45	21.0±0.44	18.9±0.42	**18.7**±0.42
$1.2 - 1.5$	27.6±0.77	28.2±0.78	24.1±0.74	**23.8**±0.74
$1.5 - 2.0$	42.4±1.74	44.2±1.75	44.0±1.75	**37.6**±1.7
$2.0 - \infty$	69.0±2.57	67.9±2.60	66.6±2.62	**62.4**±2.7
OOV	64.7±0.0	64.2±0.49	60.4±0.51	**57.8**±0.52
All	9.93±0.07	9.80±0.07	9.19±0.06	**9.03**±0.06

Due to the fact that the evaluation utterances range from easy to difficult for the ASR system, we may not always see the effect of regularization relying only on the overall performance. However, using the proposed framework, the results indicate that for difficulty level above 0.6, UML with regularization provides superior performance over BPE without regularization. This indicates that, regularization helps generalization of the model to unseen words, at a cost of some degradation in performance for less difficult utterances. The benefit gained by regularization can best be observed from the result for regularized and unregularized ULM as well as BPE. Moreover, you can infer that for CV evaluation utterances with lower difficulty level, BPE is a better choice than ULM. This experiment supports our second hypothesis that the proposed framework provides a systematic comparative tool and helps us to choose an appropriate subword segmentation for a specific evaluation scenario.

4.3. Model combination

Finally, to test the third hypothesis, we conduct an experiment to see whether we can use the proposed framework to combine multiple representations and improve ASR performance. Due to the fact that we do not know in advance the transcription for an evaluation utterance, we may not be able to compute the difficulty score. However, we can estimate the score using the transcription generated by the ASR system provided that the the system is accurate enough. We use regularized ULM, which performs the best for higher difficulty levels, and compute the difficulty score for each evaluation utterance. If the score is higher than a specified threshold, e.g., 0.5 as implied from Table 3, we keep the transcription, but if the score is lower than this threshold, we use the BPE model to generate the transcription. This simple fusion technique provides some notable performance improvement. The results are shown in Table 4.

Table 4: *ASR performance results in terms of WER (%) for BPE and ULM as well as their fusion.*

Difficulty	BPE	ULM	Fusion
$0.0 - 0.2$	**2.07** ±0.07	3.03±0.08	2.37±0.07
$0.2 - 0.4$	**2.42**±0.18	3.34±0.21	3.31±0.21
$0.4 - 0.6$	**6.77**±0.24	6.76±0.24	6.83±0.24
$0.6 - 0.8$	9.66±0.12	8.63±0.11	**8.56**±0.11
$0.8 - 1.0$	13.8±0.14	**11.9**±0.13	**11.9**±0.13
$1.0 - 1.2$	21.7±0.45	18.7±0.42	**18.6**±0.42
$1.2 - 1.5$	27.6±0.77	**23.8**±0.74	**23.8**±0.74
$1.5 - 2.0$	42.4±1.74	**37.6**±1.7	**37.6**±1.7
$2.0 - \infty$	69.0±2.57	**62.4**±2.7	**62.4**±2.7
All	9.93±0.07	9.03±0.06	**8.86**±0.06

5. Conclusion and Future work

A framework based on BPE, to associate a difficulty with test utterances in the absence of a lexicon, correlates well with actual ASR accuracy. The framework reveals that different subword approaches vary in performance with difficulty.

Our results show that the more stochastic approaches are more suited to more difficult (out of domain) test sets. The combination of different subword approaches can also lead to improvement in ASR results.

In future work, we will use the framework to inform the training process given adaptation data appropriate for a new domain.

6. Acknowledgements

This work is part of Domain Adaptation via Hierarchical Lexicons (DAHL) project funded by Swisscom company to start from November 2019.

7. References

[1] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4945–4949.

[2] 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.

[3] 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.

[4] J. Drexler and J. Glass, "Subword regularization and beam search decoding for end-to-end automatic speech recognition," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6266–6270.

[5] I. Provilkov, D. Emelianenko, and E. Voita, "Bpe-dropout: Simple and effective subword regularization," *arXiv preprint arXiv:1910.13267*, 2019.

[6] V. Pratap, A. Hannun, Q. Xu, J. Cai, J. Kahn, G. Synnaeve, V. Liptchinsky, and R. Collobert, "Wav2letter++: A fast open-source speech recognition system," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6460–6464.

[7] A. Hannun, A. Lee, Q. Xu, and R. Collobert, "Sequence-to-sequence speech recognition with time-depth separable convolutions," *Proc. Interspeech 2019*, pp. 3785–3789, 2019.

[8] G. Synnaeve, Q. Xu, J. Kahn, E. Grave, T. Likhomanenko, V. Pratap, A. Sriram, V. Liptchinsky, and R. Collobert, "End-to-end asr: from supervised to semi-supervised learning with modern architectures," *arXiv preprint arXiv:1911.08460*, 2019.

[9] T. Baumann, A. Köhn, and F. Hennig, "The spoken wikipedia corpus collection: Harvesting, alignment and an application to hyperlistening," *Language Resources and Evaluation*, vol. 53, no. 2, pp. 303–329, 2019.

[10] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.

[11] "LibriVox: Free public domain audiobooks," pp. 7–8, Jan. 2014. [Online]. Available: https://librivox.org

[12] S. Radeck-Arneth, B. Milde, A. Lange, E. Gouvêa, S. Radomski, M. Mühlhäuser, and C. Biemann, "Open source german distant speech recognition: Corpus and acoustic model," in *International Conference on Text, Speech, and Dialogue*. Springer, 2015, pp. 480–488.

[13] A. Köhn, F. Stegen, and T. Baumann, "Mining the spoken wikipedia for speech data and beyond," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2016, pp. 4644–4647.

[14] B. Milde and A. Köhn, "Open source automatic speech recognition for german," in *Speech Communication; 13th ITG-Symposium*. VDE, 2018, pp. 1–5.

[15] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, 2015, pp. 577–585.

[16] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1243–1252.

[17] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[19] K. Heafield, "KenLM: Faster and smaller language model queries," in *Proceedings of the sixth workshop on statistical machine translation*. Association for Computational Linguistics, 2011, pp. 187–197.