IDIAP RESEARCH REPORT

COMPARISON OF SUBWORD SEGMENTATION METHODS FOR OPEN-VOCABULARYEND-TO-END SPEECH RECOGNITION

Abbas Khosravani Claudiu Musat Philip N. Garner
Alexandros Lazaridis[a]

Idiap-RR-34-2020

DECEMBER 2020

[a]Idiap Research Institute

Centre du Parc, Centre du Parc, Rue Marconi 19, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

COMPARISON OF SUBWORD SEGMENTATION METHODS FOR OPEN-VOCABULARY END-TO-END SPEECH RECOGNITION

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

ABSTRACT

To address the open vocabulary problem in the context of end-to-end automatic speech recognition (ASR), we experiment with subword segmentation approaches, specifically byte-pair encoding and unigram language model. Such approaches are attractive in general for morphologically rich languages, and in particular for German. We propose a technique which computes the tokenization rate of an utterance transcription in the spirit of the out-of-vocabulary (OOV) metric that would be used for closed vocabularies. We show that this tokenization rate can then be used to rank evaluation utterances in terms of recognition difficulty. Using this technique we show that the optimal choice of subword segmentation technique depends on the expected tokenization rate of the domain. We further show that a hybrid solution exists and can lead to improved performance. For the ASR model, we employ wav2letter, a fully convolutional sequence-to-sequence encoder architecture using time-depth separable convolution blocks and a lexicon-free beam search decoding with n-gram subword language model.

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-informed 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 compound form, that subwords could easily handle. We propose a technique to rank the evaluation utterances in terms of recognition difficulty. Specifically, we measure the tokenization rate of an utterance transcription in terms of the ratio of the segmentation length to ground-truth number of words. Intuitively, the lower this figure is for a test utterance, the better it is represented by the training data in the spirit of out-of-vocabulary and the task has low difficulty. If, however, the tokenization rate is greater than unity, the utterance is not well represented by the training set in terms of OOV rate and the task is difficult.

Our experiments on different evaluation datasets show that the proposed technique provides a measure to rank the evaluation utterances in terms of recognition difficulty which enables us to compare and analyze different subword segmentation approaches, e.g. in terms of generalization, as well as their effectiveness on various evaluation scenarios. In short, we investigate the following hypotheses in this study:

H1. The proposed technique can measure the difficulty level of an evaluation dataset in the spirit of out-of-vocabulary metric.

H2. Using this technique, it is possible to combine different segmentation approaches to improve ASR performance.

The remainder of this paper is organized as follows. Section 2 describes the proposed technique and different subword segmentation strategies. Our experimental setup including data and recogniton systems 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

In this section, we first present the popular techniques for subword-informed word representations and then introduce the proposed technique to rank the evaluation utterances in terms of recognition difficulty 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 for each word (usually all words in the training transcriptions) as tokens and at each step it merges the most frequent pair into a new token. The merge operations are added to a merge table in order. This is done until the desired vocabulary size is reached. To provide segmentation for a new word, the same merge table is used to perform merge operations in order on the word 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. Tokenization Rate

The proposed technique provides a measure, which we call *tokenization rate*, to assess the recognition difficulty of an evaluation utterance in respect to OOV rate using the transcription information for that utterance as well as all the training utterances. Although the transcription is not available in real evaluation scenarios, the aim of this study is to compare and analyze different subword segmentation techniques. We later show that, using the hypothesised transcript provided by the ASR system we can estimate this tokenization rate.

In order to compute the tokenization rate for an evaluation utterance, we use the same data compression technique as in byte-pair encoding [2] to encode the transcript into a unique sequence of tokens. Initially, we split the whole transcript into individual characters and consider them 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. We start by iteratively merging the most frequent pair of tokens in respect to the training transcriptions. A merge operation is performed only when the frequency of a pair is more than a specified threshold. In this way, a pair of tokens is merged only when at least a specified number of them are available in the training transcriptions. We repeat this process until no further merge operation is possible. The tokenization rate is then defined as the ratio of the segmentation length to the number of words in the transcript. This procedure is described in Algorithm 1.

If the segmentation of a transcript leads to an equal number of tokens compared to words, tokenization rate is unity. In this case the segmentation usually breaks up the transcript into the same sequence of words with no out-of-vocabulary words. However, if the segmentation breaks up a word into a sequence of subwords, this is an indication of an OOV word and this leads to a higher tokenization rate and ultimately increases the recognition difficulty. On the other hand, if the tokenization rate is less than unity, it is likely that there is

Algorithm 1: Computing the tokenization rate

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

at least a sequence of words which are observed more than a specified number of times in the training transcriptions. Therefore, the ASR is more capable in recognizing such utterances and the difficulty decreases. Finally, when only one token is generated, the whole transcript has been observed during training more than a specified number of times and it has the lowest recognition difficulty. In Section 3, we will show that the ASR performance is highly correlated with the tokenization rate of the evaluation utterances.

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 training set consisting of half a million German utterances with ∼737 hours of speech data. The training and evaluation utterances come from different open-source German corpora. We designed an evaluation plan to enable the study of different segmentation approached with less focus on domain mismatch. We uniformly select ∼100 hours of speech data from three different German corpora which 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	

3.1.1. Spoken Wikipedia Corpora

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].

3.1.2. M-AILABS

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 an attractive resource for speech-to-text applications. The German set includes ∼237h of audio clips varying in length from 1 to 20 seconds.

3.1.3. Common Voice

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.

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

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 approaches 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 supports 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 for subword-unit approaches and 4-gram word LM for word-based model 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 the 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 want to show that tokenization rate provides a measure to rank the evaluation utterances in terms of recognition difficulty. To achieve this, we build three different ASR systems with different word representations. The first system is a word-level ASR system with a vocabulary size of 100K and a 4-gram word-based LM. We use a lexicon-based decoding which restricts the search to a fixed vocabulary and therefore, it is not capable of recognizing OOV words. The second system is a character-level ASR system with a 20-gram character-based LM. Finally, the third system is a subword-informed ASR system in which subword units are discovered using BPE technique. We experiment with different vocabulary sizes to find the best performing one on our evaluation dataset. Table 2 presents the results.

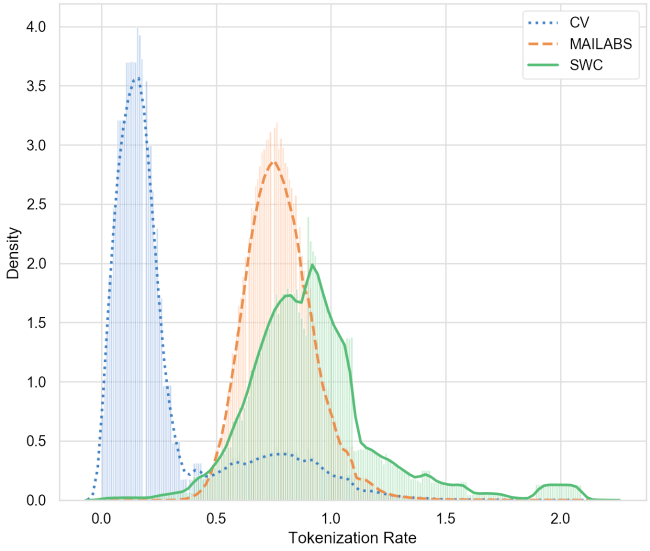

Fig. 1. The probability distribution of the tokenization rate of utterances in each evaluation dataset.

Table 2. ASR performance on various evaluation datasets in terms of WER(%) for character, word and subword level representations. BPE is used to generate subword units with different vocabulary sizes. The number of words and percentage of OOV words are also reported for each evaluation dataset.

			Segmentation				
Evaluation	#Words	%OOV	Char	BPE$_{4K}$	BPE$_{8K}$	BPE$_{10K}$	Word
CV	252k	0.8	5.10	5.11	**4.30**	4.31	5.39
MAILABS	255k	2.6	5.35	11.5	**10.9**	11.3	13.6
SWC	254k	6.6	11.1	16.9	**14.6**	15.6	23.2
All	762k	3.3	18.9	11.2	**9.93**	10.4	14.1

It is clear from the results that subword-level ASR system provides superior performance over either character or word-level ASR systems. Moreover, a vocabulary size of 8K for subword units results in the best performance in all evaluation scenarios.

As claimed in Section 2.2, tokenization rate provides a measure to rank the evaluation utterances according to the

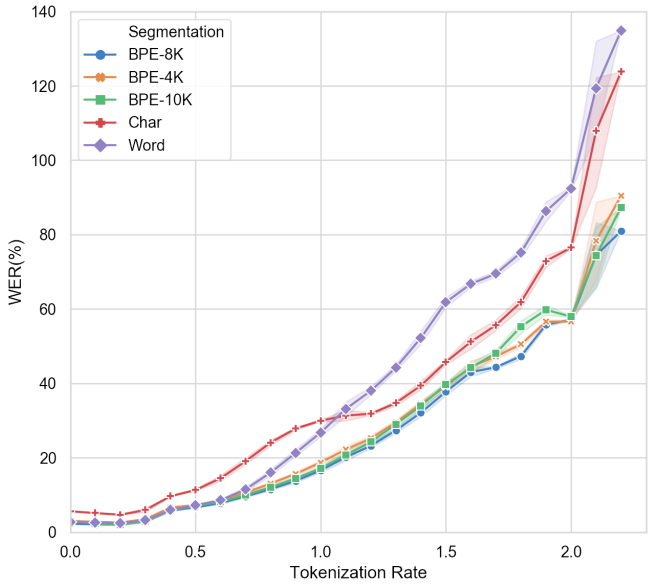

Fig. 2. ASR performance in terms of WER(%) as a function of tokenization rate for character, word and subword-level representations. BPE is used to generate subword units with different vocabulary sizes.

recognition difficulty which we will show this shortly. Figure 1 shows the probability distribution of the tokenization rate for utterances in each evaluation datasets. We can see that CV and SWC have the lowest and the highest overall tokenization rates, respectively. As explained in Section 3.1, due to the diverse range of topics and large vocabulary size of SWC, a significant number of OOV words was expected and this is well illustrated in Figure 1.

In Figure 2, we plot the performance of ASR system as a function of tokenization rate for various word representation techniques. It is clear that the tokenization rate correlates well with the actual ASR performance, which supports the first hypothesis that the proposed measure is suitable to discern the difficulty level of an evaluation dataset. Due to the fact that the word-level ASR system is bound to a fixed lexicon, it provides the worst performance on higher tokenization rates, but very good performance on lower ones. This is mainly attributed to the existence of enough training speech data for words and even sequences of words. The character-level ASR system is able to reconstruct OOV words and therefore at some point in the curve it beats the word-level ASR system. Subword-level ASR systems outperforms both the character and word-level ones in the absence of a lexicon, with 8K vocabulary size constantly achieving the best performance.

4.2. Subword segmentation analysis

In order to compare different subword segmentation techniques, we train two new models using subword regularization based on ULM [3] and stochastic BPE [5]. In [4], it was shown that regularization helps generalization of ASR system. Using the proposed technique, we aim at analyzing the effect of regularization on the performance of ASR systems. We set the number of subword units to 8K and generate a lexicon with 10-best subword segmentations for each word in the training transcription. During training, the best representation for each word is used, however, with a small probability other alternatives are selected randomly. We set this probability to 0.05 which provides the best overall performance according to our experiments. In addition to the BPE-based ASR system which does not benefit from regularization, we train an unregularized ULM-based ASR model using the best segmentation of each word. Table 3 shows the results.

Table 3. ASR performance in terms of WER(%) for regularized and unregularized subword-level ASR systems. The error rate for OOV words which contribute to almost 3.3% of all the words in the evaluation datasets is also reported.

	Unregularized		**Regularized**	
Evaluation	BPE	ULM	BPE	ULM
CV	**4.30**	**4.30**	4.79	4.89
MAILABS	10.9	11.2	10.8	**10.6**
SWC	14.6	13.9	11.9	**11.5**
All	9.93	9.80	9.19	**9.03**
OOV	64.7	64.2	60.4	**57.8**

From the results, it is obvious that regularization leads to significant performance improvement for SWC dataset with the highest overall tokenization rate but hurts the performance on CV dataset with few OOV words. The OOV word error rate is an indication of the ability of subword-informed approaches in recognizing unseen words. As a result, the ASR system can benefit from regularization to improve generalization and better recognition of OOV words.

Figure 3 plots the ASR performance as a function of tokenization rate for different subword segmentation techniques. We use logarithmic scale for WER to better illustrate the point in the curve where a regularized subword unit approach outperforms the unregularized version. For tokenization rate lower than this point ($\sim$ 0.6), unregularized techniques significantly outperforms regularized ones and the other way around. This figure is in alignment with the previous finding that, regularization helps generalization of the ASR system, at the cost of some degradation in performance for utterances with low tokenization rate. As evidenced by the experiments, the proposed technique provides a systematic comparative tool and helps us to choose an appropriate subword segmentation for a specific evaluation scenario.

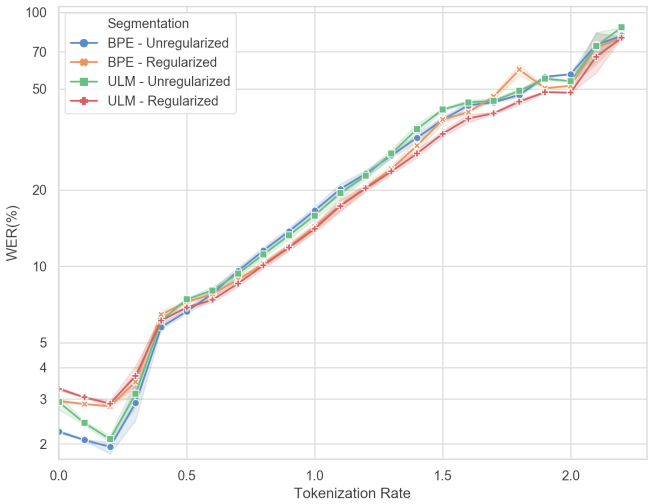

Fig. 3. ASR performance in terms of WER(%) as a function of tokenization rate. The different lines represent different subword segmentation techniques with and without regularization. The WER axis is plotted in logarithmic scale.

4.3. Model combination

Finally, to test the second hypothesis, we conduct an experiment to combine multiple subword segmentation approaches and improve ASR performance. The tokenization rate can not be computed without the ground-truth transcript, however, we can estimate it using a hypothesised transcript generated by the ASR system, provided that the system is fairly accurate. We use regularized ULM-based ASR system which obtained the best overall performance to transcribe an evaluation utterance. We then compute the tokenization rate using this hypothesised transcript. If the tokenization rate is higher than a specified threshold, e.g., 0.6 as implied from Figure 3, we keep the transcription, otherwise we use the BPE-based ASR system. The results are shown in Table 4. Although not significant, this simple fusion technique provides some performance improvement. As you can see, for CV dataset, the performance of the fusion system improves compared to the ULM-based system, whereas, it remains the same for the others. This indicates the effectiveness of tokenization rate to combine multiple subword segmentation approaches.

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

Evalaution	BPE	ULM	Fusion
CV	**4.30**	4.89	4.43
MAILABS	10.9	**10.6**	**10.6**
SWC	14.6	**11.5**	**11.5**
All	9.93	9.03	**8.86**

5. CONCLUSION AND FUTURE WORK

A technique which we call tokenization rate, inspired by the data compression principle, to rank the evaluation utterances in order of ASR recognition difficulty is proposed. This tokenization rate correlates well with the actual ASR accuracy.

Our results show that regularization techniques are more suited for test sets with high number of OOV words, but hurts test sets with low numbers. The combination of different subword approaches can also lead to improvement in ASR results.

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

6. REFERENCES

[1] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua 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] Rico Sennrich, Barry Haddow, and Alexandra 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] Taku 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] Jennifer Drexler and James 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] Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita, "Bpe-dropout: Simple and effective subword regularization," *arXiv preprint arXiv:1910.13267*, 2019.

[6] Vineel Pratap, Awni Hannun, Qiantong Xu, Jeff Cai, Jacob Kahn, Gabriel Synnaeve, Vitaliy Liptchinsky, and Ronan 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] Awni Hannun, Ann Lee, Qiantong Xu, and Ronan Collobert, "Sequence-to-sequence speech recognition with

time-depth separable convolutions," *Proc. Interspeech 2019*, pp. 3785–3789, 2019.

[8] Gabriel Synnaeve, Qiantong Xu, Jacob Kahn, Edouard Grave, Tatiana Likhomanenko, Vineel Pratap, Anuroop Sriram, Vitaliy Liptchinsky, and Ronan Collobert, "End-to-end asr: from supervised to semi-supervised learning with modern architectures," *arXiv preprint arXiv:1911.08460*, 2019.

[9] Timo Baumann, Arne Köhn, and Felix 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] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.

[11] "LibriVox: Free public domain audiobooks," Jan. 2014.

[12] Stephan Radeck-Arneth, Benjamin Milde, Arvid Lange, Evandro Gouvêa, Stefan Radomski, Max Mühlhäuser, and Chris 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] Arne Köhn, Florian Stegen, and Timo 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] Benjamin Milde and Arne Köhn, "Open source automatic speech recognition for german," in *Speech Communication; 13th ITG-Symposium*. VDE, 2018, pp. 1–5.

[15] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua 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] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann 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] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[18] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew 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] Kenneth 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.