






Article

Dynamic Acoustic Unit Augmentation with BPE-Dropout for Low-Resource End-to-End Speech Recognition

Aleksandr Laptev ^{1,*}, Andrei Andrusenko ^{1,†}, Ivan Podluzhny ¹, Anton Mitrofanov ^{1,2} and Ivan Medennikov ^{1,2} and Yuri Matveev ^{1,2}

- ¹ Corporate Laboratory of Human-Machine Interaction Technologies, Information Technologies and Programming Faculty, School of Translational Information Technologies, ITMO University, 196084 Saint-Petersburg, Russia; andrusenkoa@itmo.ru (A.A.); iapodluzhnyi@itmo.ru (I.P.); mitrofanov-aa@itmo.ru (A.M.); medennikov@speechpro.com (I.M.); matveev@speechpro.com (Y.M.)
- ² STC-Innovations Ltd., 194044 Saint-Petersburg, Russia
- * Correspondence: aalaptev@itmo.ru
- † These authors contributed equally to this work and share the first authorship.

Abstract: With the rapid development of speech assistants, adapting server-intended automatic speech recognition (ASR) solutions to a direct device has become crucial. For on-device speech recognition tasks, researchers and industry prefer end-to-end ASR systems as they can be made resource-efficient while maintaining a higher quality compared to hybrid systems. However, building end-to-end models requires a significant amount of speech data. Personalization, which is mainly handling out-of-vocabulary (OOV) words, is another challenging task associated with speech assistants. In this work, we consider building an effective end-to-end ASR system in low-resource setups with a high OOV rate, embodied in Babel Turkish and Babel Georgian tasks. We propose a method of dynamic acoustic unit augmentation based on the Byte Pair Encoding with dropout (BPE-dropout) technique. The method non-deterministically tokenizes utterances to extend the token's contexts and to regularize their distribution for the model's recognition of unseen words. It also reduces the need for optimal subword vocabulary size search. The technique provides a steady improvement in regular and personalized (OOV-oriented) speech recognition tasks (at least 6% relative word error rate (WER) and 25% relative F-score) at no additional computational cost. Owing to the BPE-dropout use, our monolingual Turkish Conformer has achieved a competitive result with 22.2% character error rate (CER) and 38.9% WER, which is close to the best published multilingual system.

Keywords: end-to-end speech recognition; low-resource; BPE-dropout; augmentation; out-of-vocabulary; transformer; BABEL Turkish; BABEL Georgian



Citation: Laptev, A.; Andrusenko, A.; Podluzhny, I.; Mitrofanov, A.; Medennikov, I.; Matveev, Y. Dynamic Acoustic Unit Augmentation with BPE-Dropout for Low-Resource End-to-End Speech Recognition. *Sensors* **2021**, *21*, 3063. <https://doi.org/10.3390/s21093063>

Academic Editors: Amos Azaria and Ariella Richardson

Received: 6 April 2021
Accepted: 25 April 2021
Published: 28 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Digital speech assistants have become ubiquitous in everyday life. According to the survey from Microsoft's latest voice report [1], 75% of English-speaking households are expected to have at least one smart speaker by 2020. Among the key functions of an ordinary speech assistant is voice search. It allows users to search the web by saying queries rather than typing them. In addition, voice search is expected to be as personalized as any modern search. However, the personalization itself is more complicated for this task than for typing-based search since it starts before ranking the results at the speech recognition stage. The part of voice search, which is responsible for transducing speech to words and passing them to the search field, can be thought of as the large vocabulary continuous speech recognition (LVCSR) task of automatic speech recognition (ASR). One of the main challenges in this task is the recognition of words that the ASR system has not encountered before; such words are called out-of-vocabulary (OOV). Recognition errors for such words occur more often than those that the system is aware of. Thus, the presence of OOV words in voice queries may negatively affect the performance of voice search. In turn,

an incorrect voice search may decrease the user-perceived quality of the whole system. Moreover, speech assistants' low personalization ability generally leads to deterioration of user experience [2].

An ASR system is one of the main components of a smart voice assistant. This system recognizes speech information from the user to transform it and pass it on for processing as a command or query. Thus, recognition errors can lead to incorrect interpretation of commands or incorrect formation of search queries. However, to operate effectively, it is not enough for the ASR model to possess a high recognition quality. The system also has to be fast and compact to be able to run on edge devices [3,4] or to have a combined server–device structure with a lightweight model for commands and a high-quality server-grade LVCSR-intended model [5]. To our knowledge, both hybrid [6] and end-to-end [7,8] ASR systems are used for speech assistants (e.g., [9,10]). Regardless of the technology applied, building an ASR system for smart assistants faces the data availability problem. Due to speech data privacy concerns and the existence of underrepresented languages, there exist challenges to gather enough data to build an effective recognition system. Thus, for many languages, excluding English, one has to consider low-resource data availability conditions (the total amount of annotated speech data suitable for training a model is less than a hundred hours).

The aforementioned problems of OOV words handling and low-resource data conditions need to be addressed when building an ASR system. If the system is a conventional hybrid (HMM-DNN-based acoustic model and word-based n-gram language model), the OOV problem is often solved by dynamically expanding the system's vocabulary and/or adapting the language model (e.g., [11–13]). A less common approach is to use a subword-based n-gram language model [14]. The vocabulary of character- or subword-based end-to-end systems is not restricted compared with the hybrid ones. However, it is difficult to build a model using extra unpaired data (viz. large external text corpora), and doing this can lead to poor performance on rare and unseen words. One of the recent approaches to tackle the OOV problem for such systems is biasing towards a given context at decoding time [15]. However, even without such improvements, subword-based end-to-end systems are generally better in handling OOV than conventional hybrid ones. The downside is that the negative impact of low-resource conditions affects end-to-end systems more since the acoustic units (output tokens) of such systems are more high-level than the Hidden Markov Model states of hybrid ones. In other words, there are more data required to saturate the model (without noticeable overfitting) that emits end-to-end acoustic units. Concerning the choice of acoustic units for an end-to-end ASR system, there is a trade-off between better saturation, obtained through the use of less specific tokens, and higher token precision by using more specific and curated tokens that are expected to contain non-trivial lexical information. Without considering logogram-based languages (e.g., Chinese), characters are the least specific tokens, and various word pieces (subwords) are more specific ones.

There are many ways to divide words into subwords. The two most popular methods of subword segmentation are Byte Pair Encoding (BPE) [16] and a unigram language model (ULM) [17]. BPE is agglomerative merging of subwords, starting with characters, according to the frequency of their joint occurrence in a training set. The ULM subword segmentation is an approach for inferring subword units by training a unigram language model on a set of characters and words suffix arrays and iteratively filtering out subwords using the Expectation–Maximization algorithm to maximize the data likelihood. Notably, this approach to make the ULM subword segmentation is not the only one. Another method worth mentioning is Morfessor [18], which finds morphological segmentation of words using greedy local search. Regardless of the subword segmentation method, there is a problem to find the optimal (in terms of the final system performance) subword number. Another problem related to subword usage is the variability of their segmentation. A text segmented with the smallest number of highly specific subwords may not always be optimal. We propose using dynamic acoustic unit augmentation to address these problems. The approach consists of diversifying the subword segmentation during model training

by sampling different segmentations for the same words. In ULM, sampling is supported by a simple varying of its temperature, which is called the subword regularization [17]. A recent Morfessor modification, named Morfessor EM+Prune [19], is also able to perform the subword regularization. Eventually, BPE-dropout [20] was proposed to regularize segmentation by randomly omitting merges.

There are few previous works on ASR related to the investigation of subword augmentation by non-deterministic segmentation. The vanilla subword regularization was studied in [21,22]. In the first work, the method was applied for the WSJ dataset (English, 50 h). In addition, the authors proposed a novel prefix search algorithm that utilizes subword length in the calculation of prefix probability. The second work investigated the improvement of applying the subword regularization to different amounts of data and analyzed its effect on OOV word recognition and hypothesis diversity. Presently, BPE-dropout and Morfessor EM+Prune were applied only to machine translation (MT). BPE-dropout was beneficially used for low-resource MT tasks as a standalone improvement [23–25] or combined with a neural sequence-to-sequence segmentation model [26]. The Morfessor EM+Prune’s subword regularization, along with other improvements, was used for the asymmetric-resource one-to-many MT task [27].

In this work, we have provided extensive research on how BPE-dropout and the ULM subword regularization acoustic unit augmentations contribute to the performance of strong end-to-end ASR system baselines in low-resource conditions. We studied the sensitivity of a model to the total number of target subwords and the regularization rate. We also analyzed how effective the aforementioned subword augmentation techniques are for alleviating the OOV problem. Finally, we built systems that achieved competitive results for IARPA Babel Turkish [28] and Georgian [29] low-resource tasks.

Our main contribution is as follows: We propose and evaluate a dynamic acoustic unit augmentation method for ASR system training, improving speech assistants’ user experience and perceived quality by increasing the OOV word recognition quality. The method is based on a non-deterministic BPE subword segmentation algorithm, BPE-dropout.

2. ASR Modeling

This section provides an overview of two main ASR approaches: hybrid and end-to-end.

2.1. Hybrid Approach

Conventional hybrid ASR systems can be divided into acoustic and language models. The acoustic model is responsible for converting an input feature sequence to output acoustic units (e.g., phonemes). The language model contains the language knowledge and helps the decoder convert acoustic units into the final word sequence. Apart from a few service parts, the model includes pronunciation lexicon and linguistic information, applied as a statistical n-gram model. The pronunciation lexicon defines the rules for mapping graphemes (characters) to phonemes.

In recent years, hybrid systems have been well studied and proven to solve many ASR-related problems. However, this approach to training ASR systems has inherent drawbacks:

- Acoustic and language models are built separately from each other and have their different objective functions. This significantly complicates the process of optimizing the ASR system.
- To train the final DNN-based acoustic model, a hard alignment (mapping of each input feature frame to a target acoustic unit) is required. It is generated and refined through several iterations of GMM-HMM-based training, in which the condition-independent assumption is in effect. However, this hard alignment also limits the acoustic context that the model can process before emitting the target token’s spike.
- Decoding with WFST graph is highly memory intensive, which makes it difficult to use the approach in ASR tasks for smart devices where the memory is severely limited.

2.2. End-to-End Approaches

CTC. Connectionist Temporal Classification (CTC) [7] was the first significant step towards addressing hybrid models' problems mentioned earlier. A new loss function was proposed to map input features to final speech recognition labels without using hard alignment and pronunciation lexicon. Any acoustic units (graphemes, phonemes, subwords) can be used as output labels. An auxiliary "blank" symbol controls label repetitions and their absence. However, the CTC-trained end-to-end ASR system does not have its own context- or language model (the system is an encoder only, and it is trained in a context-independent manner), which leads to degradation of the recognition quality. Nevertheless, using pure CTC-trained systems can still be advantageous since they are often the most efficient and deliver competitive quality [30].

Neural transducer. Later, a neural transducer [31] was introduced, which can solve the context-independent problem of the CTC approach. The proposed prediction network is designed to utilize contextual information and thus works similarly to the language model. The encoder and prediction network results are then sent to the joint network, which emits the final result based on acoustic and context information. The entire system is jointly optimized with the single Transducer objective function, which is a modified CTC loss. Recently, the transducer approach has proven its effectiveness both in large-resource (e.g., [32]) and low-resource (e.g., [33]) tasks.

Attention-based. Another approach to building an end-to-end ASR system is using the attention-based sequence-to-sequence architecture [8] that consists of an encoder and a decoder with the attention mechanism. The attention mechanism allows the decoder to use a weighted representation of an encoded input context. Along with an autoregressive decoder, this provides context-depending label modeling. However, this approach is prone to overfitting, which manifests in the output of a highly probable sequence of tokens regardless of acoustic information. It was proven effective to combine attention decoding with CTC to alleviate their shortcomings and improve recognition quality [34]. At the same time, the Transformer [35] attention-based architecture was proposed as more effective than various RNN architectures. A multi-head self-attention (MHA) mechanism significantly improved the quality of models over the recurrent models. A transformer model, trained with CTC-Attention, can outperform neural transducer systems (e.g., [36]) and benefit from various augmentation techniques [37]. Recently, the Conformer [38] was introduced, which is a modification of the transformer layer. Convolution blocks and advanced activation functions were added to each layer of the model encoder. The latest reports (e.g., [39]) demonstrate that the Conformer outperforms the Transformer in almost all tasks.

3. Subword Modeling

This section describes the subword augmentation techniques that were the subject of our investigation.

3.1. ULM Subword Regularization

The subword segmentation algorithm [17] is based on a simple unigram language model. It allows getting multiple subword segmentation variants with the corresponding probabilities. The probability of a subword sequence $\mathbf{x} = (x_1, x_2, \dots, x_M)$ is the product of unigram probabilities of these subwords. To obtain the most probable subword segmentation \mathbf{x}^* for the input word sequence \mathbf{W} , the Viterbi algorithm is used.

For subword regularization, one first has to get l -best segmentations according to a probability distribution $P(\mathbf{x}|\mathbf{W})$ over subword segmentation variants corresponding to a source word sequence. Next, one can sample a new segmentation \mathbf{x}_i from the multinomial distribution:

$$P(\mathbf{x}_i|\mathbf{W}) = \frac{P(\mathbf{x}_i)^\alpha}{\sum_{i=1}^l P(\mathbf{x}_i)^\alpha} \quad (1)$$

where α is a temperature parameter, which controls the smoothness of the distribution. If $\alpha = 0$, then the segmentation is sampled from uniform distribution (segmentation is

uniformly sampled from the n -best (if $l = n$) or lattice (if $l = \infty$). A larger α allows the selection of the most probable Viterbi segmentation. The parameter l is restricted by the Forward-Filtering and Backward-Sampling algorithm [40] because the number of all possible subword segmentation variants increases exponentially with respect to the sentence length.

3.2. BPE-Dropout Augmentation

The Byte Pair Encoding (BPE) [16] segmentation defines a simple deterministic mapping of words to subword tokens. The algorithm starts by creating an initial token vocabulary consisting of characters of the input text's words. The end-of-word mark is also added to disambiguate word boundaries. Next, the tokens are agglomeratively merged according to their co-occurrence frequency. The merge operations are written in the merge table. The algorithm iterates until the maximum number of merges is exceeded, or the desired vocabulary size is reached. The resulting merge table contains all allowed rules and the order of merging subwords.

During the segmentation process, the word is split into characters with the addition of the end-of-word mark. Then, the tokens are assembled according to the merge table until the merge rules are exhausted. The training and inference procedures are deterministic, thus the result of the algorithm is always unambiguous. Such formulation does not imply any regularization or augmentation.

Recently, BPE was reformulated [20], which made applying augmentation possible. The method, named BPE-Dropout, is based on random discarding of a certain number of merges with some probability p . If $p = 0$, then it operates like standard BPE segmentation. When $p = 1$, all merge operations are omitted, and words are split into single characters.

4. Method Description

In this section, we present the method of our dynamic acoustic unit augmentation. An evaluation criterion for the recognition performance of OOV words is also provided here.

4.1. Dynamic Acoustic Unit Augmentation

A typical ASR pipeline involves static preparation of acoustic units prior to model training. Grapheme-based segmentation breaks down words into characters. In subword segmentation, a pre-built subword tokenization system transforms word transcripts into subword sequences used as targets in the ASR model training. The transcripts are segmented deterministically by design (even if the segmentation itself is non-deterministic) as the model processes the whole training text in a single shot. For each training data batch in each epoch, there will be identical target subword sequences.

Using subword augmentation techniques allows for getting different subwords for the same word. Using a non-deterministic segmentation during training, rather than before it, enables obtaining various target subword sequences each time a word sequence occurs in a batch. This leads to a diversification of the targets by epochs, while acoustic data remain the same. This augmentation method enriches acoustic units and regularizes the training process.

4.2. Recognition of OOV Words

It is assumed that an end-to-end ASR model trained on subword-segmented utterances is capable of recognizing any new word in the target language. However, if a word was not sufficiently represented in the training data, then the model can assign a low probability to the subword sequence representing the word during the decoding process. Therefore, instead of an OOV word, the decoder is likely to emit the most similar seen word. We assumed that non-deterministic subword tokenization should improve the recognition of unseen words, as this technique allows for enhancing the diversity of subword sequences during the ASR model's training process.

To analyze OOV word recognition performance, we used an F-score metric similar to [22]. The method based on counting after decoding how many times the model emitted (true positive, tp) or did not emit (false negative, fn) the OOV words from the evaluation set. Words that were neither in training nor in evaluation transcripts (false positive, fp) were also counted and used for calculating $precision = tp / (tp + fp)$, $recall = tp / (tp + fn)$, and $F\text{-score} = 2 \cdot precision \cdot recall / (precision + recall)$. This allows for estimating the quality of OOV word recognition.

5. Experiments

This section describes the experiments performed and provides the results obtained.

5.1. Data

For our experiments, we used two telephone conversations datasets for the IARPA Babel Turkish [28] and Georgian [29] languages. We formed training sets from utterances with duration from 10 to 2000 frames and not exceeding 300 characters to avoid GPU memory overflow and stabilize the training process. We also extracted one hour of data from each training set for validation purposes. Final data training sizes were 73.40 h for the Turkish set and 50.52 h for the Georgian one. All results were obtained for the official development sets, which consist of 9.82 h (5.40% OOV words) and 12.36 h (8.95% OOV words) of Turkish and Georgian, respectively.

5.2. End-to-End Setup

The main end-to-end model architecture for our experiments was the Transformer. The encoder consisted of a 2-layer CONV2D subsampling block (to reduce input feature sequence by four times) (In object detection, Convolutional Neural Networks (CNNs) are used as main architecture blocks (e.g., [41]). Some inference-efficient ASR systems also use purely-convolutional solutions (e.g., [30]). However, CNNs seem to be the most effective only for the initial time compression to our knowledge.) followed by 12 Transformer layers with 1024 units feed-forward dimension. The decoder was a 6-layer Transformer with 1024 feed-forward units. We used 8-headed self-attention with 360 dimensions for both model parts. The model was trained with joint CTC and attention-based loss for 100 epochs. We used Adam optimizer [42] with OneCycleLr training scheduler [43] as this combination showed the best model convergence during preliminary architecture search. The input feature sequence for both hybrid and end-to-end setups were cepstral mean- and variance-normalized 40-dimensional log-Mel filterbank coefficients with three-dimensional pitch features. In our end-to-end training setup, we additionally used SpecAugment [44] data augmentation.

To the extent of our knowledge, the Sentencepiece tokenizer (Available at <https://github.com/google/sentencepiece> accessed on 23 February 2021) [45] is the only tool that currently supports both the ULM and BPE subword segmentation algorithms and their non-deterministic segmentation techniques. We used it to dynamically tokenize utterances when training our models in the ESPnet speech recognition toolkit (Available at <https://github.com/espnet/espnet> accessed on 23 February 2021) [46,47].

5.3. The Augmentation Impact

Figure 1 shows how Word Error Rate (WER) depends on the usage of augmentation techniques. In the first series of experiments, for the Turkish language, we trained the Transformer model described above using two different subword tokenization methods: BPE and ULM. The vocabulary size was set to 1000 units. For each method, a line graph plots the dependence of α value (ULM sampling smoothing parameter and BPE dropout probability of a subword segmentation model in the Sentencepiece tokenizer) on WER (green and red colors respectively). The scale mark $\alpha = 0$ denotes deterministic tokenization. It can be observed that using both augmentation methods is beneficial for the models.

The best result was obtained using the BPE-trained model with $\alpha = 0.1$, which provided an absolute WER improvement of 2.5%.

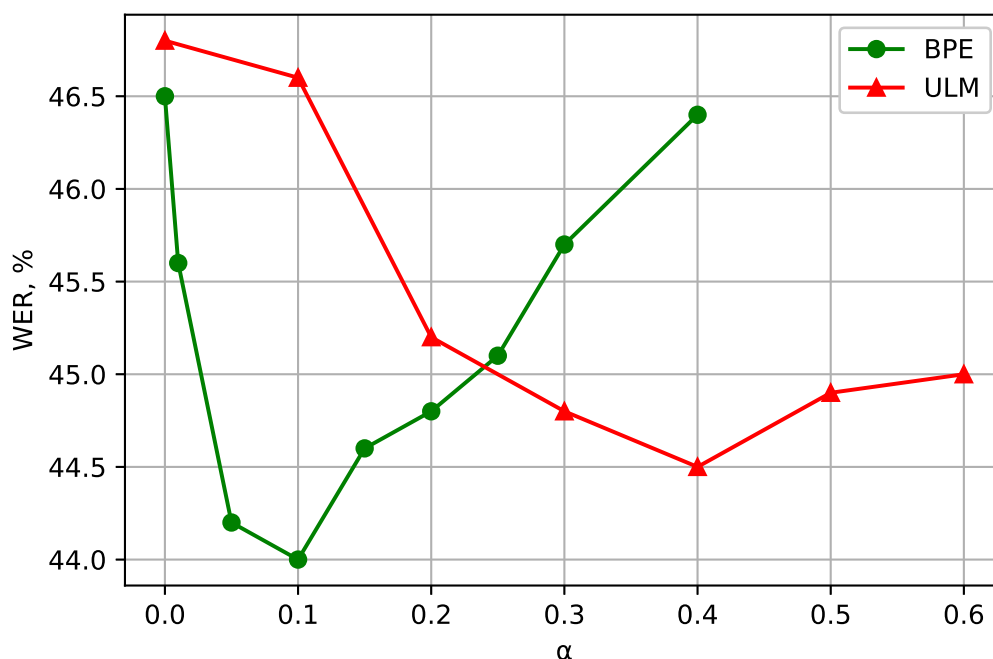


Figure 1. (Turkish) WER for different values of α for the BPE- and ULM-trained subword models. $\alpha = 0.0$ means that deterministic segmentation is used. The vocabulary size is 1000 units.

Having settled on the BPE-dropout with $\alpha = 0.1$ dropout probability, we investigated how this augmentation technique performs for different vocabulary sizes. The results for the Turkish and Georgian languages are presented in Figures 2 and 3, respectively (more detailed representations are available in Tables 1 and 2). The models with character-based acoustic units performed worse (with 53.0 and 51.2 WER% for Turkish and Georgian) than the ones with subword-based optimal vocabulary sizes. Despite this, character-based models had a high recall and thus a competitive OOV recognition F-score. Using the chosen unit augmentation technique was beneficial both in WER and F-score. With the vocabulary size of 3000, the Turkish recognition quality was improved by 2.9 WER% and 0.034 F-score compared to the best non-augmented models and by 6.0 WER% and 0.062 F-score for the models of the same vocabulary size. Similarly, the improvements of 3.2 WER% and 0.032 F-score were obtained for the Georgian language with 500 acoustic units. Overall, using BPE-dropout lessened the need for optimal subword vocabulary size choice to build a more effective model.

Another study was to check the BPE-dropout augmentation when applied with a more advanced Conformer architecture and other augmentation approaches. We chose a Conformer with the depth-wise convolution kernel of size 15 and the 3-fold speed perturbation (SP) [48]. The rest of the model hyperparameters and the training environment were the same as in our Transformer setup. The tokenization setup was as follows: 3000 BPE vocabulary units and the dropout probability $\alpha = 0.1$. The results for the Babel Turkish are presented in Table 3. The BPE-dropout augmentation improvement remained for the Conformer setup with 2.4 WER% and 0.035 F-score compared to 6.0 WER% and 0.064 F-score for the Transformer setup. It was also productively combined with the SP augmentation, resulting in 38.9 WER% and 0.224 F-score of the final system. The training of the Conformer model training did not converge for the Babel Georgian in our setup. It can be assumed that 50 h of data may not be enough to train an advanced end-to-end model from scratch.

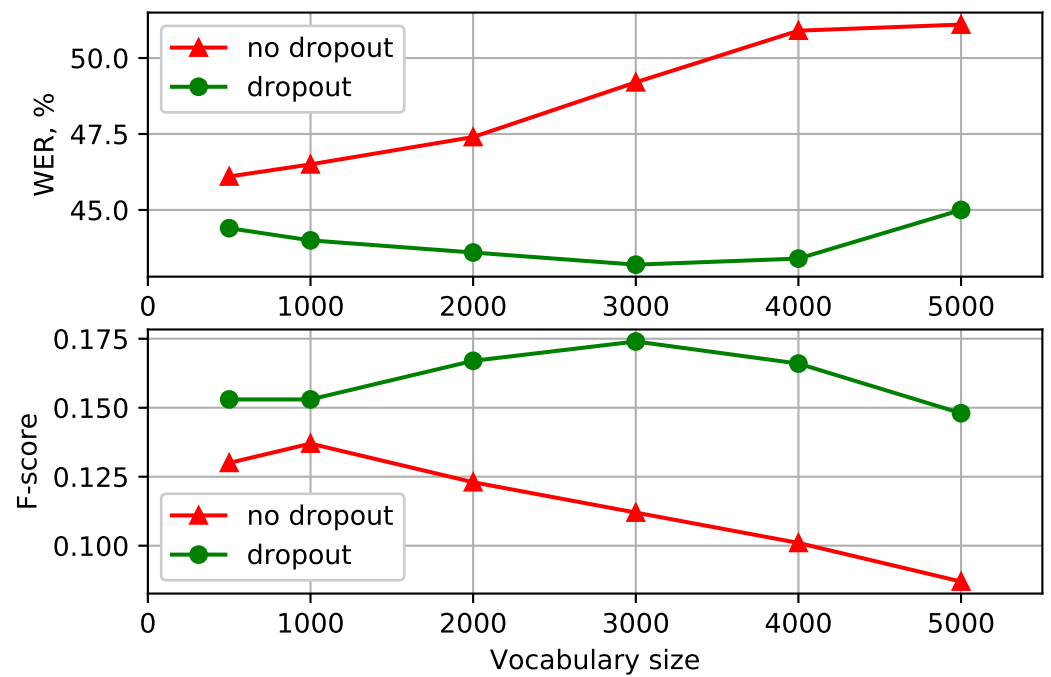


Figure 2. (Turkish): WER and F-score for different vocabulary size for BPE segmentation with the dropout (**top**) and without (**bottom**).

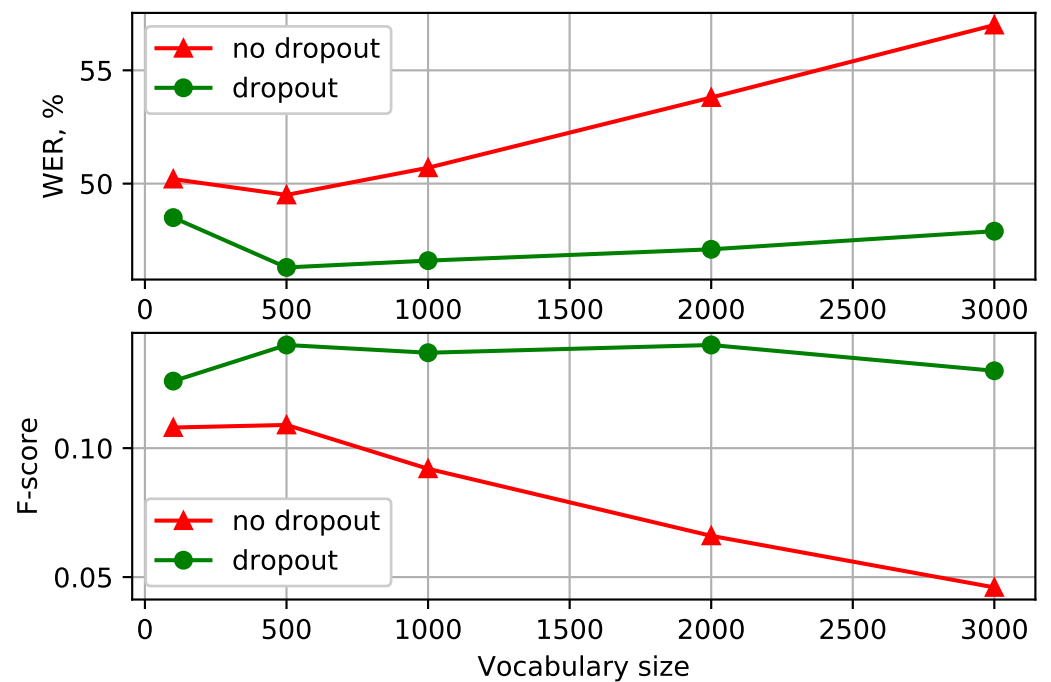


Figure 3. (Georgian): WER and F-score for different vocabulary size for BPE segmentation with the dropout (**top**) and without (**bottom**).

Table 1. (Turkish): WER and F-score for different vocabulary sizes with ($\alpha = 0.1$) and without BPE-dropout augmentation.

Vocab Size	BPE-Dropout	WER (%)	Precision	Recall	F-Score
char	-	53.0	0.067	0.165	0.095
500	-	46.1	0.114	0.152	0.130
	+	44.4	0.120	0.209	0.153
1000	-	46.5	0.130	0.144	0.137
	+	44.0	0.126	0.194	0.153
2000	-	47.4	0.126	0.118	0.123
	+	43.6	0.144	0.198	0.167
3000	-	49.2	0.129	0.099	0.112
	+	43.2	0.156	0.197	0.174
4000	-	50.9	0.124	0.085	0.101
	+	43.4	0.151	0.183	0.166
5000	-	51.1	0.115	0.070	0.087
	+	45.0	0.137	0.160	0.148

Table 2. (Georgian): WER and F-score for different vocabulary sizes with ($\alpha = 0.1$) and without BPE-dropout augmentation.

Vocab Size	BPE-Dropout	WER (%)	Precision	Recall	F-Score
char	-	51.2	0.090	0.162	0.116
100	-	50.2	0.087	0.143	0.108
	+	48.5	0.101	0.167	0.126
500	-	49.5	0.095	0.126	0.108
	+	46.3	0.117	0.172	0.140
1000	-	50.7	0.088	0.096	0.092
	+	46.6	0.118	0.161	0.137
2000	-	53.8	0.070	0.061	0.066
	+	47.1	0.124	0.160	0.140
3000	-	57.0	0.054	0.039	0.046
	+	47.9	0.116	0.147	0.130

Table 3. (Turkish): Our end-to-end models' performance depending on the BPE-dropout regularization use ($\alpha = 0.1$).

model	BPE-Dropout	WER (%)	Precision	Recall	F-Score
Transformer	-	49.2	0.129	0.099	0.112
	+	43.2	0.156	0.197	0.174
Conformer	-	42.9	0.188	0.142	0.162
	+	40.5	0.194	0.201	0.197
Conformer+SP	+	38.9	0.199	0.255	0.224

5.4. Final Comparison

Apart from our best end-to-end systems, we established baselines with a conventional hybrid architecture consisting of an LF-MMI trained TDNN-F acoustic model and a 3-gram word language model. The acoustic features were the same that we used for our end-to-end models. The models setup and training process (except for acoustic features) were performed according to the *librispeech/s5* recipe of the Kaldi [49] toolkit.

Our baselines and best models for both languages were compared to other published results in Table 4. There is a specific type of recognition result scoring named sclite (sclite is a part of the SCTK toolkit. Available at <https://github.com/usnistgov/SCTK> accessed on 23 February 2021) [50]. It was used in all NIST OpenKWS evaluations and provided for all BABEL languages. Thus, the considered Babel Turkish and Georgian development sets are expected to be scored with it. However, the exact comparison is not formally possible since all the works except for [51] do not mention the use or non-use of the sclite scoring tool. The results that are known to have been sclite-scored are marked with an asterisk.

Table 4. The final comparison. * indicates that sclite is used for scoring.

Language	Model	CER (%)	WER (%)
Turkish	Our LF-MMI TDNN-F	(* 21.4)	43.9 (* 38.6)
	Our Conformer	22.2 (* 17.3)	38.9 (* 34.7)
	CTC-BLSTM [51]	-	50.7 (* 45.8)
	BLSTMP+VGG-Multilingual [52]	28.7	-
	XLSR-Monolingual [53]	26.1	-
	XLSR-53-Multilingual [53]	18.8	-
Georgian	Our LF-MMI TDNN-F	(* 25.4)	51.6 (* 43.3)
	Our Transformer	24.6 (* 21.0)	46.3 (* 41.7)
	BLSTMP+VGG-Multilingual [52]	36.0	-
	Multilingual hybrid fusion [54]	-	32.2
	XLSR-Monolingual [53]	30.5	-
	XLSR-53-Multilingual [53]	17.2	31.1

Our Turkish end-to-end model performed well compared to all the systems. It delivered 22.2% CER and 38.9% WER (17.3% CER and 34.7% WER with sclite scoring). These results are better than those of the previous monolingual systems. The model may even have outperformed the best Babel multilingual system (assuming sclite was used in [53]) in CER. This might indicate that applying advanced data augmentation techniques can compete with out-of-language-domain data addition in terms of the quality improvement. However, there are currently few works covering the Turkish speech recognition; therefore, the topic has yet to be fully explored. As for Babel Georgian, our model with 24.6% CER and 46.3% WER (21.0% CER and 41.7% WER with sclite scoring) was competitive among the monolingual systems, but their quality is considerably low compared to the previous multilingual results. Apart from out-of-language-domain data usage, this gap can be explained by additional text data usage in building a language model for decoding [54] and advanced multilingual pre-training approaches [53].

6. Discussion

This section attempts to explain the results provided in Experiments (Section 5).

With a subword text segmentation, tokens can be unevenly represented in training data, and a model can be biased towards recognizing frequent tokens. Nevertheless, even frequent tokens can have a small limited number of words, which they are a part of (context words), and this can lead to overfitting. In addition, short (in terms of character number) tokens in such conditions may have poor saturation, especially in low-resource cases. BPE-dropout can address both of these problems: it increases the frequency of short tokens and the number of context words for all tokens (except for subwords representing a full word) in the training process.

The increase in the frequency of short tokens occurs due to “forgetting” to apply some merging rules when assembling short tokens into more complex ones. Without “forgetting,” these short non-terminal tokens become a part of other tokens, which causes the appearance of non-terminal tokens in words that are otherwise occupied by more advanced terminal subwords. Thus, the model receives more diverse contexts for these tokens during the training process. It can be seen in Figures 4 and 5 that, with BPE-dropout, short tokens

appear even more often during the training (left line charts) and in a broader set of unique words (right scatter charts). The latter is also true for longer tokens (3–4 characters long).

We argue that short (1–2 characters long) tokens play an essential role in the recognition of OOV words. It was observed that their amount in the OOV recognition results ranges from 60–70% to 80%. Consequently, extensive statistics for short tokens and the variability of their contexts may help the model better produce unseen words based on the short token utterances already encountered in the training.

BPE-dropout can be studied in terms of augmentation and regularization properties. For tokens 1–2 characters long, the method has strong augmentation properties. The use of BPE-dropout increased the amount of single-character tokens in the training process by 2–3 times: from 14% to 29% for the BPE vocabulary size of 1000 and from 7% to 22% for the vocabulary size of 3000. In other cases, BPE-dropout performed more like a regularization technique: the number of tokens with more than two symbols did not increase or even slightly decreased for the tokens longer than four characters. For such subwords, diversification of token sequences is one of the regularization properties, as it reduces overfitting of the attention decoder.

Another important regularization property of BPE-dropout is reducing the influence of the vocabulary size on the model quality (according to Figures 2 and 3). A small vocabulary allows for better saturation of tokens when training, but the recognition may become non-robust to unconventional and alternative pronunciations, as modeling long-term language dependencies becomes difficult and acoustic information dominates decoding. Alternatively, increasing the BPE vocabulary allows more words to be recognized “directly” in one piece, which benefits the quality of recognition. At the same time, an increased BPE vocabulary substantially shifts the balance of tokens in training towards long tokens, thereby obstructing the OOV recognition ability. The BPE-dropout technique facilitates the trade-off between these options. As can be seen in Figure 6, BPE-dropout compensates for the decrease in the number of short token appearances at the cost of a slight decrease in the percentage of long ones.

BPE 1000

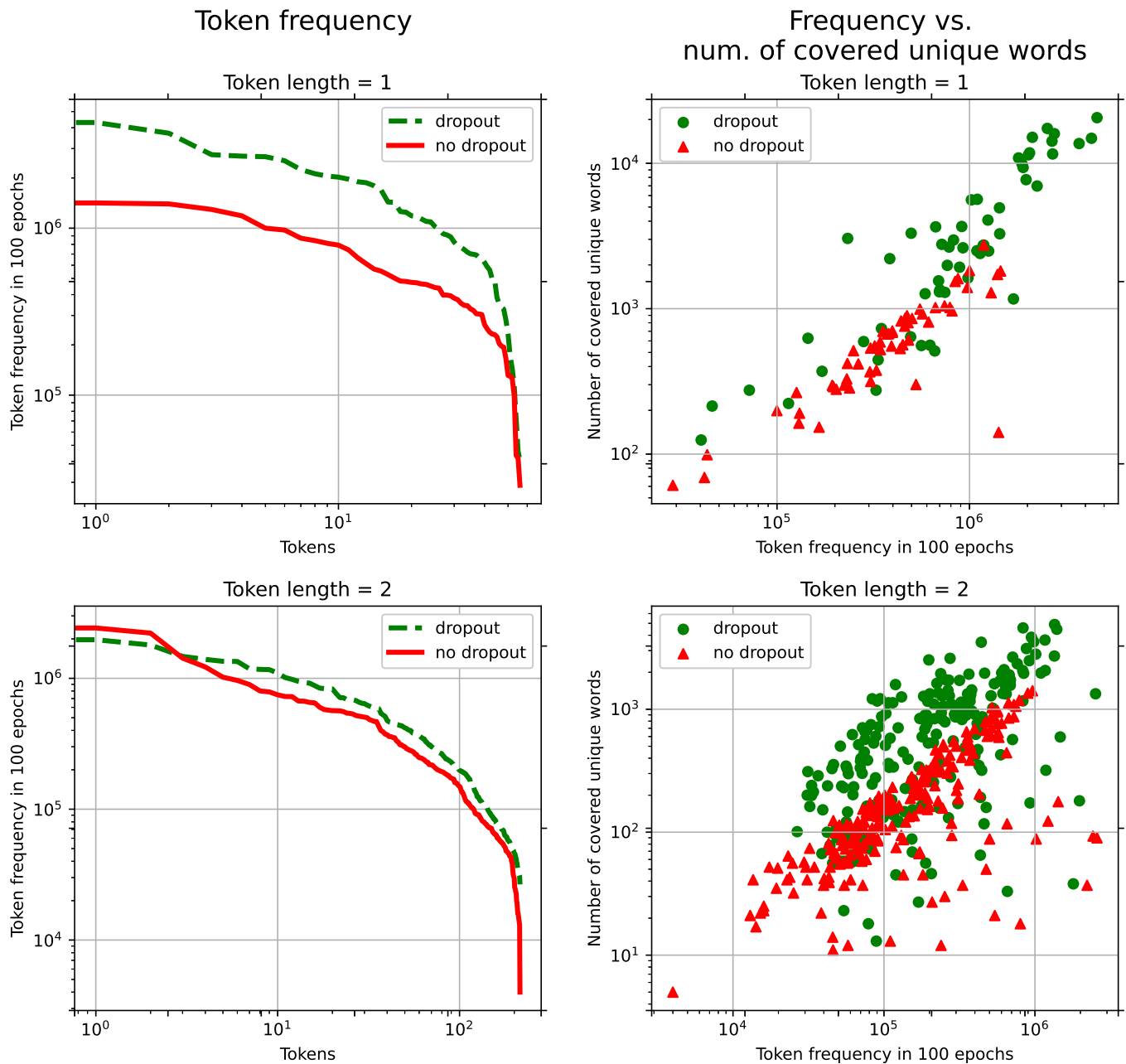


Figure 4. (Turkish): **Left:** Token frequency distribution in 100 epochs. The horizontal axis represents tokens sorted by their frequencies in the descending order. The vertical axis shows frequencies of tokens. **Right:** token frequency vs. number of unique words in which these tokens are present. Points represent individual tokens. Both statistics were computed on the training set for token lengths 1 and 2 with the dropout and without. The BPE vocabulary size was set to 1000.

BPE 3000

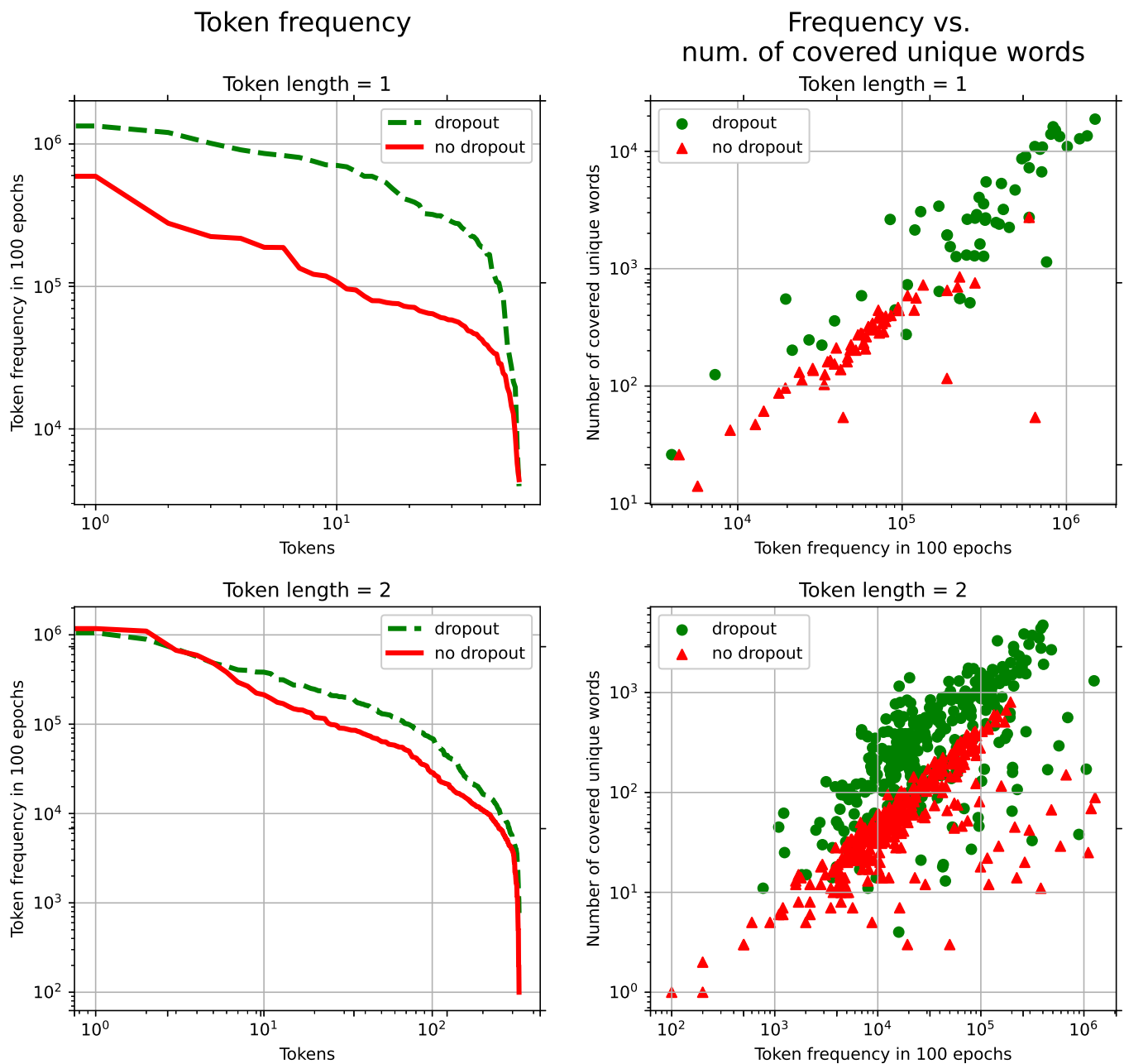
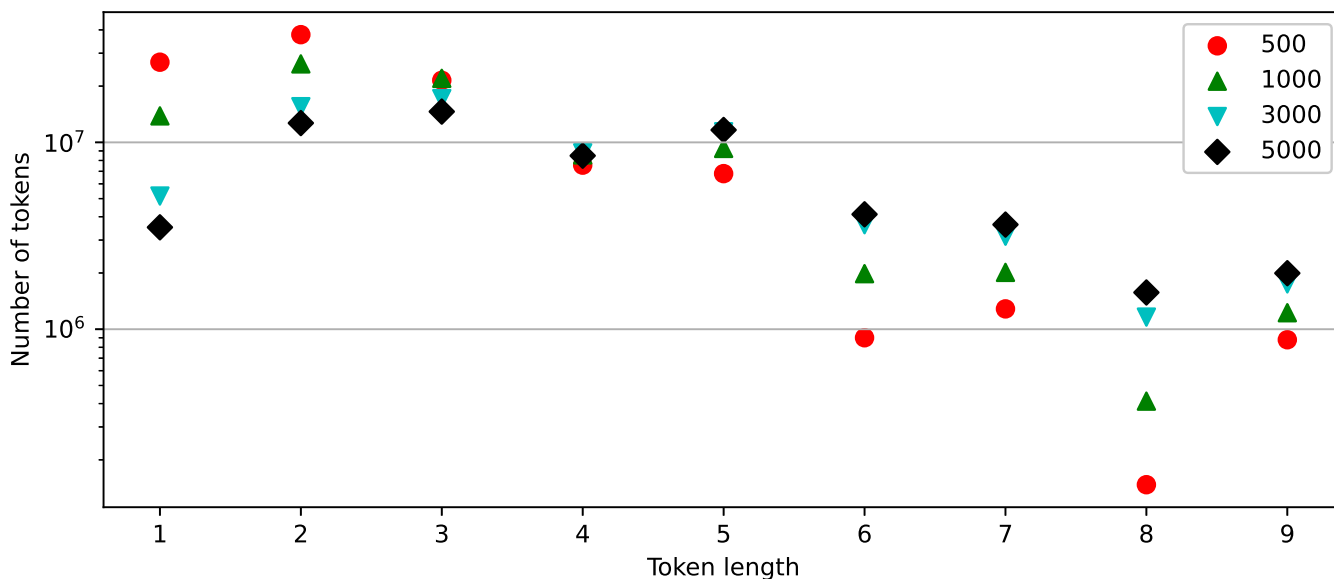


Figure 5. (Turkish): **Left:** Token frequency distribution in 100 epochs. The horizontal axis represents tokens sorted by their frequencies in the descending order. The vertical axis shows frequencies of tokens. **Right:** token frequency vs. number of unique words in which these tokens are present. Points represent individual tokens. Both statistics were computed on the training set for token lengths 1 and 2 with the dropout and without. The BPE vocabulary size was set to 3000.

By revisiting Figures 2 and 3, it can be seen that the larger the vocabulary size, the more noticeable the improvement from using BPE-dropout augmentation. To explain this, we compared actual token distributions in the recognition results obtained. As demonstrated in Figure 7, BPE-dropout increases the number of relatively short (1–3 characters long) tokens in the OOV words from 61% to 75% for the BPE vocabulary size 3000. However, in the case of the vocabulary size 1000, token length distributions in OOV are almost identical. This may mean that the greater the improvement from the use of BPE-dropout, the more it

reshapes and shifts the model token distribution towards the shorter ones, assuming that the dropout parameter remains the same.

Without dropout



With dropout

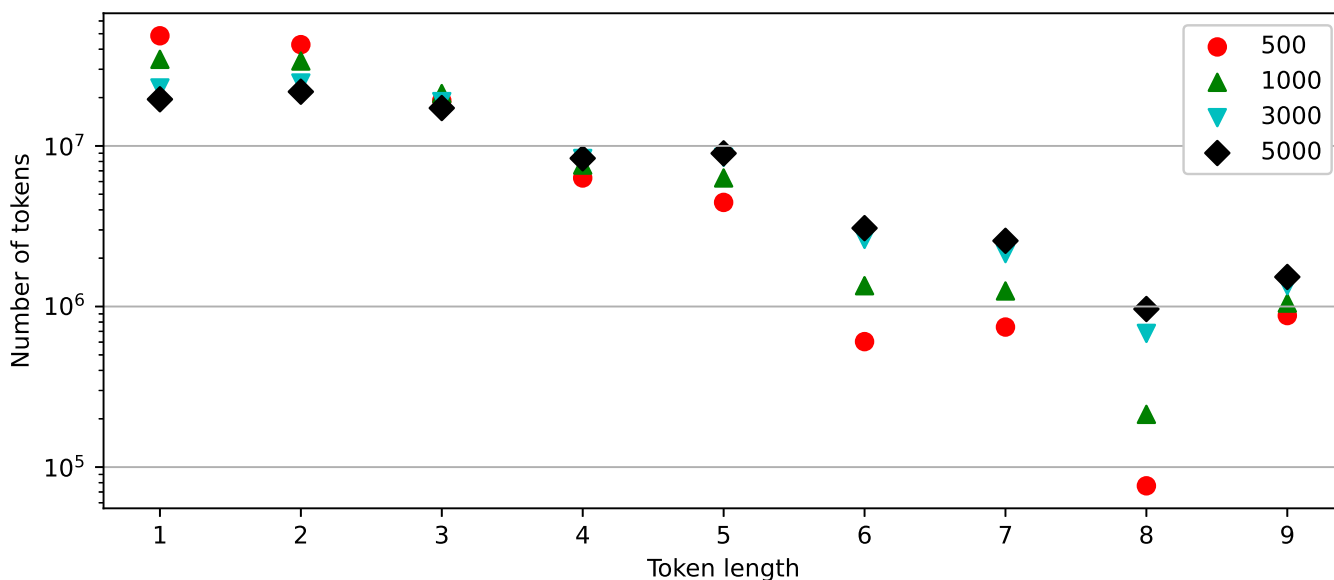
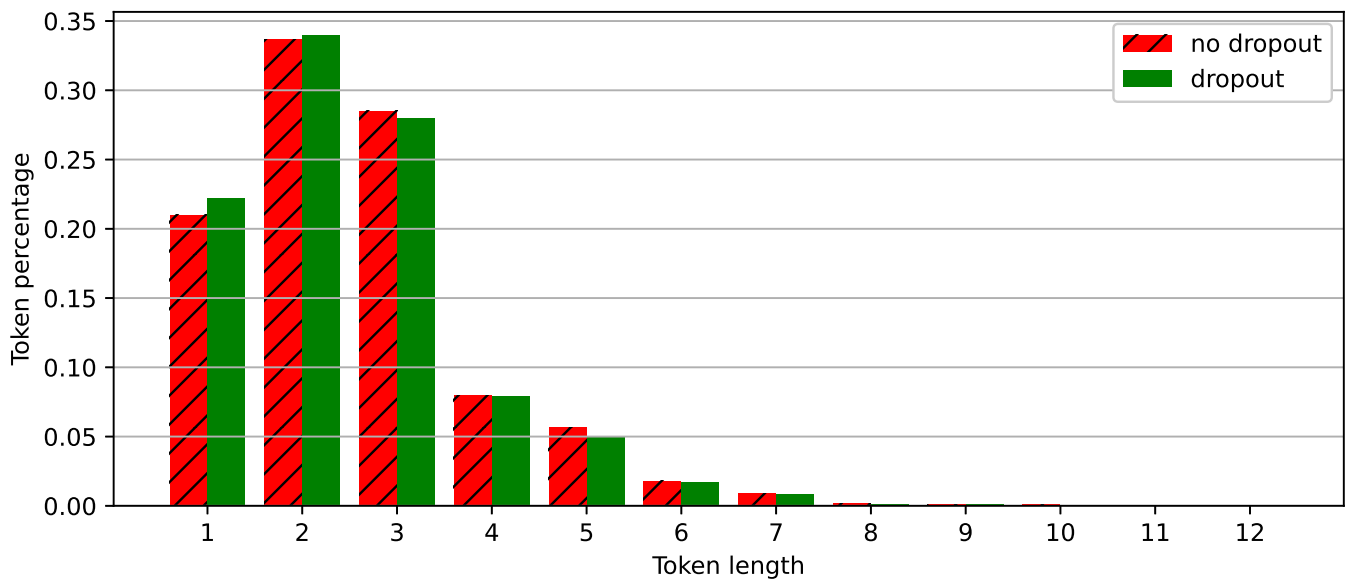


Figure 6. (Turkish): Number of tokens in 100 epochs vs. token length for different BPE subword vocabulary sizes with the dropout (**bottom**) and without (**top**).

Overall, the BPE-dropout-based augmentation provides the model with more complete and diverse statistics for tokens during the training, especially for the short ones. In addition, training with BPE-dropout allows the model to utilize a character-based model's properties to recognize OOV words while maintaining the subword-based model quality for regular speech recognition tasks.

BPE 1000



BPE 3000

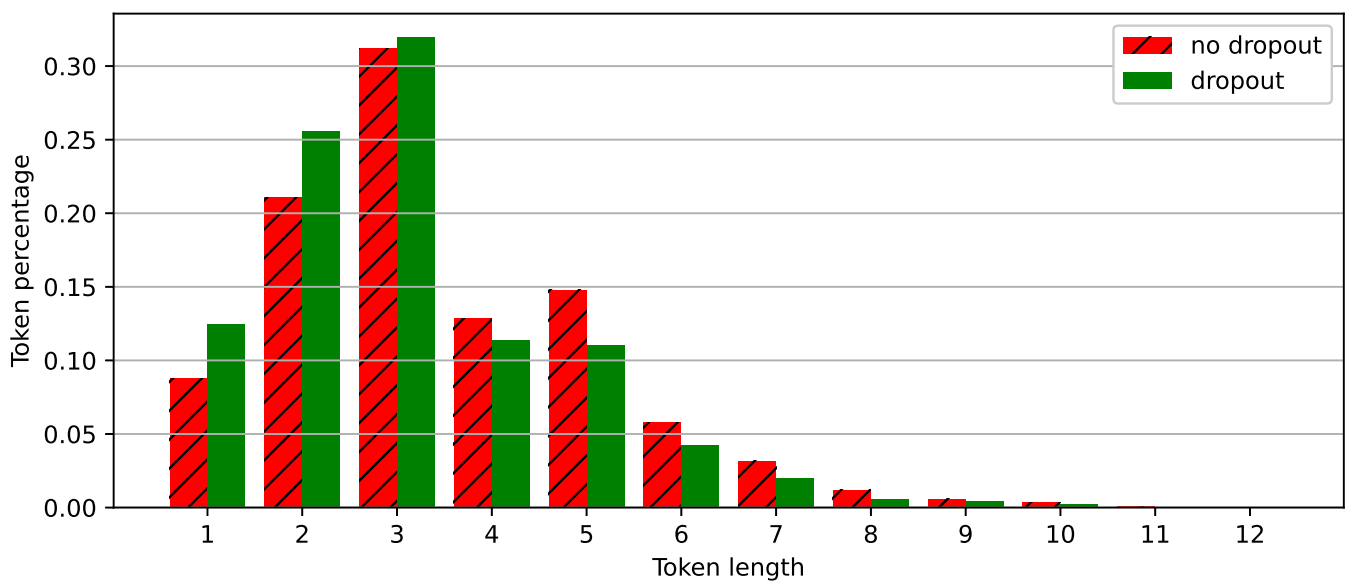


Figure 7. (Turkish): BPE token length distribution in OOV words emitted at decoding. **Top:** Vocabulary size 1000. **Bottom:** Vocabulary size 3000.

7. Limitations

Below are the main limitations of the study:

- While the proposed acoustic unit augmentation approach significantly improves the OOV recognition rate, it is still cannot compete with or replace explicit personalization techniques for those personalized ASR tasks where the quality is more important than the speed.
- Since the use of BPE-dropout shifts the distribution of acoustic units towards shorter ones, the expected quality improvement might diminish if the method is applied to a system with a higher frame subsampling factor (e.g., 8 or 16).
- The end-to-end systems used in this study may not be suitable for the use in smart assistants “as is”, as the research focus was on the quality improvement. Additional enhancements may be required to make the systems more efficient (e.g., model compression, decoding optimization, and streaming training mode). Our best Conformer model has almost 40 million parameters. Its decoding (inference on GPU Nvidia GTX 1080TI and beam search on CPU) has nine real-time (RT) (calculated as the total test set duration divided by the decoding time). After moving to an edge device, the speed will drop significantly, which can make our model impractical for the real-time ASR. In particular, 1080TI has 11.34 tera floating-point operations per second (TFLOPs), while, for example, Nvidia Jetson TX2 Series devices have 0.67 TFLOPs.
- The data used in this study may not be sufficient to build an effective ASR system for smart assistants. It may require augmenting telephone waveforms with synthetic room impulse responses and extending them with target microphone data.

8. Conclusions

In this work, we have proposed a method of dynamic acoustic unit augmentation based on the BPE-dropout technique. This method allows for improved ASR system quality at no additional training and decoding computational cost. Its regularization properties eliminate the need for optimal subword vocabulary size search, and its augmentation properties provide a consistent word error rate reduction (at least 6% relative WER improvement compared to the best non-augmented models) in low-resource setups. In addition, BPE-dropout’s ability to significantly improve the recognition of out-of-vocabulary words makes it useful for personalized ASR tasks. Using this approach can make speech assistants’ user experience better and improve the perceived quality. We found that our method is more effective than the previously used ULM subword regularization technique. Applying BPE-dropout unit augmentation to models trained on the Babel Turkish and Georgian low-resource datasets helped our end-to-end monolingual models to be competitive with the previous hybrid and multilingual systems.

Future work may concern adding Morfessor EM+Prune into consideration and comparison with BPE-dropout and the ULM subword regularization. In addition, non-deterministic subword tokenization should be studied in conjunction with the use of high frame subsampling factors. Finally, the dropout probability can be scheduled during the training to make a model behave differently (more like character- or purely subword-based) depending on the training stage.

Author Contributions: A.L. and A.A. contributed equally and share first authorship. I.M. and Y.M. share senior authorship. Conceptualization, A.L.; methodology, A.L., A.A., I.P., and A.M.; software, A.L., A.A., and I.P.; validation, A.L., A.M., and I.M.; formal analysis, I.P.; investigation, A.L., A.A., and I.P.; resources, A.A., I.P., and A.M.; data curation, A.L. and A.A.; writing—original draft preparation, A.L. with significant contributions by I.M. and Y.M.; writing—review and editing, A.L., A.A. and I.P., and I.M.; visualization, A.A. and I.P.; supervision, I.M. and Y.M.; project administration, Y.M.; funding acquisition, A.A. and Y.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially financially supported by ITMO University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this study were obtained from Linguistic Data Consortium, Catalog No. LDC2016S10 (<https://catalog.ldc.upenn.edu/LDC2016S10> accessed on 23 February 2021) and №No. (<https://catalog.ldc.upenn.edu/LDC2016S12> accessed on 23 February 2021). The following restrictions (More in <https://catalog.ldc.upenn.edu/license/ldc-non-members-agreement.pdf> accessed on 23 February 2021) apply: noncommercial linguistic education, research and technology development. Requests to access these datasets should be directed to Linguistic Data Consortium, ldc@ldc.upenn.edu.

Conflicts of Interest: Authors Anton Mitrofanov, Ivan Medennikov, and Yuri Matveev were employed by the company STC-innovations Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

MDPI	Multidisciplinary Digital Publishing Institute
ASR	Automatic Speech Recognition
OOV	Out-Of-Vocabulary
CER	Character Error Rate
WER	Word Error Rate
LVCSR	Large Vocabulary Continuous Speech Recognition
HMM	Hidden Markov Model
DNN	Deep Neural Network
BPE	Byte Pair Encoding
ULM	Unigram Language Model
MT	Machine Translation
WFST	Weighted Finite-State Transducer
CTC	Connectionist Temporal Classification
RNN	Recurrent Neural Network
MHA	Multi-Head (Self) Attention
IARPA	Intelligence Advanced Research Projects Activity
GPU	Graphics Processing Unit
CNN	Convolutional Neural Network
CONV2D	2D CNN
SP	speed perturbation
LF-MMI	Lattice-Free-Maximum Mutual Information (Criterion)
TDNN-F	Time Delay Neural Network-Factorized
BLSTM	Bidirectional Long Short-Term Memory
VGG	Visual Geometry Group
RT	Real Time
TFLOPs	FLoating-point Operations Per second

References

1. Olson, C.; Kemery, K. Voice Report: From Answers to Action: Customer Adoption of Voice Technology and Digital Assistants; Microsoft. Available online: <https://about.ads.microsoft.com/en-us/insights/2019-voice-report> (accessed on 26 December 2020).
2. Pal, D.; Arpnikanondt, C.; Funilkul, S.; Varadarajan, V. User Experience with Smart Voice Assistants: The Accent Perspective. In Proceedings of the IEEE 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019; pp. 1–6. [CrossRef]
3. Sainath, T.; He, Y.; Li, B.; Narayanan, A.; Pang, R.; Bruguier, A.; Chang, S.Y.; Li, W.; Alvarez, R.; Chen, Z.; et al. A Streaming On-Device End-To-End Model Surpassing Server-Side Conventional Model Quality and Latency. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6059–6063. [CrossRef]

4. Huang, W.; Hu, W.; Yeung, Y.T.; Chen, X. Conv-Transformer Transducer: Low Latency, Low Frame Rate, Streamable End-to-End Speech Recognition. In Proceedings of the Interspeech 2020, ISCA, Shanghai, China, 25–29 October 2020; pp. 5001–5005. [\[CrossRef\]](#)
5. Sigtia, S.; Haynes, R.; Richards, H.; Marchi, E.; Bridle, J. Efficient Voice Trigger Detection for Low Resource Hardware. In Proceedings of the Interspeech 2018, ISCA, Hyderabad, India, 2–6 September 2018; pp. 2092–2096. [\[CrossRef\]](#)
6. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.E.; Mohamed, A.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.N.; et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Process. Mag. IEEE* **2012**, *29*, 82–97. [\[CrossRef\]](#)
7. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd international conference on Machine learning—ICML, Pittsburgh, PA, USA, 25–29 June 2006; pp. 369–376. [\[CrossRef\]](#)
8. Chan, W.; Jaitly, N.; Le, Q.V.; Vinyals, O. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 4960–4964. [\[CrossRef\]](#)
9. Aleksic, P.; Allauzen, C.; Elson, D.; Kracun, A.; Casado, D.M.; Moreno, P.J. Improved recognition of contact names in voice commands. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, Australia, 19–24 April 2015; pp. 5172–5175.
10. Tulsiani, H.; Sapru, A.; Arsikere, H.; Punjabi, S.; Garimella, S. Improved Training Strategies for End-to-End Speech Recognition in Digital Voice Assistants. In Proceedings of the Interspeech 2020, ISCA, Shanghai, China, 25–29 October 2020; pp. 2792–2796. [\[CrossRef\]](#)
11. Khokhlov, Y.; Tomashenko, N.; Medennikov, I.; Romanenko, A. Fast and Accurate OOV Decoder on High-Level Features. In Proceedings of the Interspeech 2017, ISCA, Stockholm, Sweden, 20–24 August 2017; pp. 2884–2888. [\[CrossRef\]](#)
12. Gandhe, A.; Rastrow, A.; Hoffmeister, B. Scalable Language Model Adaptation for Spoken Dialogue Systems. In Proceedings of the IEEE 2018 Spoken Language Technology Workshop (SLT), Athens, Greece, 18–21 December 2018; pp. 907–912. [\[CrossRef\]](#)
13. Malkovsky, N.; Bataev, V.; Sviridkin, D.; Kizhaeva, N.; Laptev, A.; Valiev, I.; Petrov, O. Techniques for Vocabulary Expansion in Hybrid Speech Recognition Systems. *arXiv* **2020**, arXiv:2003.09024.
14. Smit, P.; Virpioja, S.; Kurimo, M. Improved Subword Modeling for WFST-Based Speech Recognition. In Proceedings of the Interspeech 2017, ISCA, Stockholm, Sweden, 20–24 August 2017; pp. 2551–2555. [\[CrossRef\]](#)
15. Jain, M.; Keren, G.; Mahadeokar, J.; Zweig, G.; Metze, F.; Saraf, Y. Contextual RNN-T for Open Domain ASR. In Proceedings of the Interspeech 2020, ISCA, Shanghai, China, 25–29 October 2020; pp. 11–15. [\[CrossRef\]](#)
16. Sennrich, R.; Haddow, B.; Birch, A. 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)*; Association for Computational Linguistics: Berlin, Germany, 2016; pp. 1715–1725. [\[CrossRef\]](#)
17. Kudo, T. 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)*; Association for Computational Linguistics: Melbourne, Australia, 2018; pp. 66–75. [\[CrossRef\]](#)
18. Creutz, M.; Lagus, K. Unsupervised Discovery of Morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2002; pp. 21–30. [\[CrossRef\]](#)
19. Grönroos, S.A.; Virpioja, S.; Kurimo, M. Morfessor EM+Prune: Improved Subword Segmentation with Expectation Maximization and Pruning. In *Proceedings of the 12th Language Resources and Evaluation Conference*; ELRA: Marseilles, France, 2020.
20. Provilkov, I.; Emelianenko, D.; Voita, E. BPE-Dropout: Simple and Effective Subword Regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Seattle, WA, USA, 2020; pp. 1882–1892. doi:10.18653/v1/2020.acl-main.170. [\[CrossRef\]](#)
21. Drexler, J.; Glass, J. Subword Regularization and Beam Search Decoding for End-to-end Automatic Speech Recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 6266–6270. [\[CrossRef\]](#)
22. Lakomkin, E.; Heymann, J.; Sklyar, I.; Wiesler, S. Subword Regularization: An Analysis of Scalability and Generalization for End-to-End Automatic Speech Recognition. In Proceedings of the Interspeech 2020, ISCA, Shanghai, China, 25–29 October 2020; pp. 3600–3604. [\[CrossRef\]](#)
23. Tapo, A.A.; Coulibaly, B.; Diarra, S.; Homan, C.; Kreutzer, J.; Luger, S.; Nagashima, A.; Zampieri, M.; Leventhal, M. Neural Machine Translation for Extremely Low-Resource African Languages: A Case Study on Bambara. In *Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages (LoResMT 2020)*; Association for Computational Linguistics: Suzhou, China, 2020; pp.23–32.
24. Knowles, R.; Larkin, S.; Stewart, D.; Littell, P. NRC Systems for Low Resource German-Upper Sorbian Machine Translation 2020: Transfer Learning with Lexical Modifications. In Proceedings of the Fifth Conference on Machine Translation, Seattle, WA, USA, 19–20 November 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 1112–1122.
25. Libovický, J.; Hangya, V.; Schmid, H.; Fraser, A. The LMU Munich System for the WMT20 Very Low Resource Supervised MT Task. In Proceedings of the Fifth Conference on Machine Translation, online, 19–20 November 2020; Association for Computational Linguistics: Seattle, WA, USA, 2020; pp. 1104–1111.

26. He, X.; Haffari, G.; Norouzi, M. Dynamic Programming Encoding for Subword Segmentation in Neural Machine Translation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Seattle, WA, USA, July 6–8 2020; pp. 3042–3051. [\[CrossRef\]](#)
27. Grönroos, S.A.; Virpioja, S.; Kurimo, M. Transfer learning and subword sampling for asymmetric-resource one-to-many neural translation. In *Machine Translation*; Springer International Publishing: Berlin/Heidelberg, Germany, 2021; doi:10.1007/s10590-020-09253-x. [\[CrossRef\]](#)
28. Andresen, J.; Bills, A.; Dubinski, E.; Fiscus, J.; Gillies, B.; Mary Harper, T.; Jarrett, A.; Roomi, B.; Ray, J.; Rytting, A.; et al. *IARPA Babel Turkish Language Pack, IARPA-babel105bv0.5 LDC2016S10*; Linguistic Data Consortium: Philadelphia, PA, USA, 2016; doi:10.35111/mb8z-6p26. [\[CrossRef\]](#)
29. Bills, A.; Conners, T.; David, A.; Dubinski, E.; Fiscus, J.; Hammond, S.; Gann, K.; Harper, M.; Hefright, B.; Kazi, M.; et al. *IARPA Babel Georgian Language Pack, IARPA-babel404b-v1.0a LDC2016S12*; Linguistic Data Consortium: Philadelphia, PA, USA, 2016; doi:10.35111/dcr5-ga44. [\[CrossRef\]](#)
30. Kriman, S.; Beliaev, S.; Ginsburg, B.; Huang, J.; Kuchaiev, O.; Lavrukhin, V.; Leary, R.; Li, J.; Zhang, Y. Quartznet: Deep Automatic Speech Recognition with 1D Time-Channel Separable Convolutions. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6124–6128. [\[CrossRef\]](#)
31. Graves, A. Sequence Transduction with Recurrent Neural Networks. In Proceedings of the 29th International Conference on Machine Learning—ICML, Workshop on Representation Learning, Edinburgh, Scotland, 26 June–1 July 2012.
32. Li, J.; Zhao, R.; Meng, Z.; Liu, Y.; Wei, W.; Parthasarathy, S.; Mazalov, V.; Wang, Z.; He, L.; Zhao, S.; et al. Developing RNN-T Models Surpassing High-Performance Hybrid Models with Customization Capability. In Proceedings of the Interspeech 2020, ISCA, Shanghai, China, 25–29 October 2020; pp. 3590–3594. [\[CrossRef\]](#)
33. Andrusenko, A.; Laptev, A.; Medennikov, I. Towards a Competitive End-to-End Speech Recognition for CHiME-6 Dinner Party Transcription. In Proceedings of the Interspeech 2020, ISCA, Shanghai, China, 25–29 October 2020; pp. 319–323. [\[CrossRef\]](#)
34. Kim, S.; Hori, T.; Watanabe, S. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 4835–4839. [\[CrossRef\]](#)
35. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.U.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*; Curran Associates, Inc.: Long Beach, CA, USA 2017; Volume 30, pp. 5998–6008.
36. Andrusenko, A.; Laptev, A.; Medennikov, I. Exploration of End-to-End ASR for OpenSTT – Russian Open Speech-to-Text Dataset. In *Speech and Computer*; Springer International Publishing: Cham, Switzerland, 2020; pp. 35–44. [\[CrossRef\]](#)
37. Laptev, A.; Korostik, R.; Svischev, A.; Andrusenko, A.; Medennikov, I.; Rybin, S. You Do Not Need More Data: Improving End-To-End Speech Recognition by Text-To-Speech Data Augmentation. In Proceedings of the IEEE 2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Chengdu, China, 17–19 October 2020; pp. 439–444. [\[CrossRef\]](#)
38. Gulati, A.; Qin, J.; Chiu, C.C.; Parmar, N.; Zhang, Y.; Yu, J.; Han, W.; Wang, S.; Zhang, Z.; Wu, Y.; et al. Conformer: Convolution-augmented Transformer for Speech Recognition. In Proceedings of the Interspeech 2020, ISCA, Shanghai, China, 25–29 October 2020; pp. 5036–5040. [\[CrossRef\]](#)
39. Guo, P.; Boyer, F.; Chang, X.; Hayashi, T.; Higuchi, Y.; Inaguma, H.; Kamo, N.; Li, C.; Garcia-Romero, D.; Shi, J.; et al. Recent Developments on ESPnet Toolkit Boosted by Conformer. *arXiv* **2020**, arXiv:2010.13956.
40. Scott, S.L. Bayesian Methods for Hidden Markov Models: Recursive Computing in the 21st Century. *Taylor Fr.* **2002**, *97*, 337–351. [\[CrossRef\]](#)
41. Alexeev, A.; Kukharev, G.; Matveev, Y.; Matveev, A. A Highly Efficient Neural Network Solution for Automated Detection of Pointer Meters with Different Analog Scales Operating in Different Conditions. *Mathematics* **2020**, *8*, 1104. [\[CrossRef\]](#)
42. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR, San Diego, CA, USA, 7–9 May 2015.
43. Smith, L.N.; Topin, N. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*; International Society for Optics and Photonics, SPIE: Bellingham, WA, USA 2019; Volume 11006, pp. 369–386. [\[CrossRef\]](#)
44. Park, D.S.; Chan, W.; Zhang, Y.; Chiu, C.C.; Zoph, B.; Cubuk, E.D.; Le, Q.V. SpecAugment: A simple data augmentation method for automatic speech recognition. In Proceedings of the Interspeech 2019, ISCA, Graz, Austria, 15–19 September 2019; doi:10.21437/interspeech.2019-2680. [\[CrossRef\]](#)
45. Kudo, T.; Richardson, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 66–71. [\[CrossRef\]](#)
46. Watanabe, S.; Hori, T.; Karita, S.; Hayashi, T.; Nishitoba, J.; Unno, Y.; Enrique Yalta Soplin, N.; Heymann, J.; Wiesner, M.; Chen, N.; et al. ESPnet: End-to-End Speech Processing Toolkit. In Proceedings of the Interspeech 2018, ISCA, Hyderabad, India, 2–6 September 2018; pp. 2207–2211. [\[CrossRef\]](#)

47. Watanabe, S.; Boyer, F.; Chang, X.; Guo, P.; Hayashi, T.; Higuchi, Y.; Hori, T.; Huang, W.C.; Inaguma, H.; Kamo, N.; et al. The 2020 ESPnet update: New features, broadened applications, performance improvements, and future plans. *arXiv* **2020**, arXiv:2012.13006.
48. Ko, T.; Peddinti, V.; Povey, D.; Khudanpur, S. Audio augmentation for speech recognition. In Proceedings of the Interspeech 2015, ISCA, Dresden, Germany, 6–10 September 2015.
49. Povey, D.; Ghoshal, A.; Boulianne, G.; Burget, L.; Glembek, O.; Goel, N.; Hannemann, M.; Motlicek, P.; Qian, Y.; Schwarz, P.; et al. The kaldı speech recognition toolkit. In Proceedings of the IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, IEEE Signal Processing Society, Waikoloa, HI, USA, 11–15 December 2011.
50. Fisher, W.M.; Fiscus, J.G. Better alignment procedures for speech recognition evaluation. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Minneapolis, MN, USA, 27–30 April 1993; Volume 2, pp. 59–62. [[CrossRef](#)]
51. Bataev, V.; Korenevsky, M.; Medennikov, I.; Zatvornitskiy, A. Exploring end-to-end techniques for low-resource speech recognition. In Proceedings of the International Conference on Speech and Computer, Leipzig, Germany, 18–22 September 2018 ; pp. 32–41. [[CrossRef](#)]
52. Cho, J.; Baskar, M.K.; Li, R.; Wiesner, M.; Mallidi, S.H.; Yalta, N.; Karafiát, M.; Watanabe, S.; Hori, T. Multilingual Sequence-to-Sequence Speech Recognition: Architecture, Transfer Learning, and Language Modeling. In Proceedings of the 2018 IEEE Spoken Language Technology Workshop (SLT), Athens, Greece, 18–21 December 2018; pp. 521–527. [[CrossRef](#)]
53. Conneau, A.; Baevski, A.; Collobert, R.; Mohamed, A.; Auli, M. Unsupervised cross-lingual representation learning for speech recognition. *arXiv* **2020**, arXiv:2006.13979.
54. Alumäe, T.; Karakos, D.; Hartmann, W.; Hsiao, R.; Zhang, L.; Nguyen, L.; Tsakalidis, S.; Schwartz, R. The 2016 BBN Georgian telephone speech keyword spotting system. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 5755–5759. [[CrossRef](#)]