COMPARATIVE STUDY OF DIFFERENT TOKENIZATION STRATEGIES FOR STREAMING END-TO-END ASR

Sachin Singh, Ashutosh Gupta, Aman Maghan, Dhananjaya Gowda, Shatrughan Singh, Chanwoo Kim

Samsung Research

2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) | 978-1-6654-3739-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/ASRU51503.2021.9687921

ABSTRACT

Most End-to-End Automatic Speech Recognition (ASR) models use character-based vocabularies: characters, sub-words (BPE), or words. While these work well for training a monolingual model, there are certain limitations when applying these to a multilingual model. Rare characters from character-rich languages like Korean can easily result in large vocabulary size, limiting the model's compactness. Representing text at the level of bytes has also been proposed. However, a byte sequence representation of text is often much longer, which increases the decoding time and makes it computationally expensive for on-device use. Byte-based sub-words (BBPE) are proposed in neural machine translation for word representation but are still unexplored in the ASR domain. In this work, we conduct an empirical study comparing the above three tokenization strategies across three metrics: Word Error Rate (WER), model size, and the decoding time, which are critical for an on-device ASR. We did extensive experiments for both monolingual and bilingual, with languages belonging to same (English and Spanish) and different (English and Korean) language families. Our experiments show that BBPE and BPE models yield a similar WER for English and Spanish. While for a character-rich language like Korean, we get 26% and 14% relative WER improvement with BBPE monolingual and bilingual models, respectively. In contrast, the byte models trade-off small model size and a fixed vocabulary at the cost of high xRT. Among all three, we found the BBPE strategy to be the most flexible and optimal for most cases.

Index Terms— end-to-end speech recognition, multilingual, RNN-Transducer

1. INTRODUCTION

End-to-End Automatic Speech Recognition (E2E ASR) solutions are rapidly gaining popularity [1, 2, 3, 4, 5, 6, 7] due to their unified structure, smaller memory footprint and simplistic training pipeline. With recent advancements like spec-augmentation [8], semi-supervised training [9] they outperform the large conventional ASR systems in terms of accuracy. Despite this great success, E2E ASR systems are not easily scalable for most world languages. Because they need a large amount of annotated data for training which is both expensive and challenging to collect. To this end, several new studies [10, 11, 12] have successfully demonstrated that a multilingual ASR could be a possible solution to this. A Multilingual ASR model leverages the fact that even syntactically different looking languages share some low-level representations analogous to CNNs in visual systems. Furthermore, multilingual systems greatly simplify the training and deployment pipeline by eliminating the need to build multiple models for multiple languages. However, extending a monolingual ASR system to a multilingual system is not straightforward. One major challenge is the enlarged vocabulary size and hence an extended softmax layer. Even if the vocabulary is naively constructed with just the characters of each language, the vocabulary size of a moderately sized multilingual system can easily shoot up to 100k tokens. In comparison, a typical vocabulary size of a monolingual model is in the order of thousands.

Table 1: Tokenization of a Korean sequence with Char, Byte, BPE, and BBPE vocabulary. Here Bytes are represented in hexadecimal format

Chars	멋			있			는		
Bytes	EB	A9	8B	EC	9E	88	EB	8A	94
BPE	멋있						는		
BBPE	멋EC9E			88			는		

The vocabulary size of an ASR model depends on the choice of its tokenization strategy. In the context of multilingual ASR, several strategies have been proposed and used, some common ones are grapheme-based [10], BPE-based [11], unicode byte-based [10] etc. Byte-based BPE (BBPE) is another excellent strategy introduced for multilingual machine translation [13]. However, it is still unexplored in the ASR domain. Each of these techniques has it's pros and cons, and an extensive study is required to compare these on a similar baseline. Therefore, in this work we are first introducing BBPE in the context of E2E ASR. Second, we give an exhaustive comparison of different tokenization strategies covering three main aspects of an on-device ASR solution

i.e. word error rate, model size, and decoding time. There are several variants of E2E ASR architecture - Listen, Attend and Spell [14], Connectionist Temporal Classification (CTC) based models [15], and Recurrent Neural Network Transducers (RNN-T) [16]. In this work we have used RNN-T as the baseline ASR architecture owing to its streaming capability.

Some related studies includes [10, 13, 17]. [10] is one of the few studies which explore unicode bytes for multilingual ASR. It highlights some key benefits of byte-based tokenization: a single compact vocabulary that can be used across multiple languages, a significantly small vocabulary size, etc. However, it only compares the byte models against the character models, whereas we have covered comparison with BPE and BBPE. [13] is the closest work we could found; however it mainly analyzes BBPE for neural machine translation. Hence the results may not directly translate to speech domain plus it only compares models in terms of BLEU scores while we have examined other aspects like xRT, model size. [17] is another study which compares using phoneme vs graphemes and BPE but its limited to monolingual ASR.

2. MODEL ARCHITECTURE

Fig 1 illustrates the RNN-T architecture used in this work. It consists of three major components - a transcription network (encoder), a prediction network (decoder) and a joint network. Transcription network is analogous to an acoustic model of a conventional ASR. It takes acoustic features $X = [x_1, x_2, \ldots, x_T]$ and an optional Language ID (LID) vector as input and transform them into high level embeddings $\mathcal{F} = [f_1, f_2, \ldots, f_U]$ as follows

$$\mathcal{F}' = \mathcal{L}_n^{LID} \circ \cdots \circ \mathcal{L}_1^{LID}(X) \quad (1)$$

$$\mathcal{L}_i^{LID} = LSTM([\mathcal{L}_{i-1}^{LID}; LID]) \quad (2)$$

$$\mathcal{F} = W_{enc}\mathcal{F}' + b_{enc} \quad (3)$$

where n is the number of encoder layers and ; symbol indicates the concatenation of LID at each time step. Likewise, the decoder is comparable to a language model. It takes LID vector (optional) and the sequence of target tokens $Y = [y_1, \ldots, y_V]$ prepended with the start-of-sequence token ϕ i.e. $\hat{Y} = [\phi, y_1, \ldots, y_V]$ as input, and computes embeddings $\mathcal{G} = [g_0, g_1, \ldots, g_V]$.

$$\mathcal{G}' = \mathcal{L}_m^{LID} \circ \cdots \circ \mathcal{L}_1^{LID}(\hat{Y}) \quad (4)$$

$$\mathcal{G} = W_{dec}\mathcal{G}' + b_{dec} \quad (5)$$

where m is the number of decoder layers. During inference the decoder is fed with the previous non-null token y_{v-1} and computes g_v.

The joint network takes input the two embeddings $\mathcal{F}$ and $\mathcal{G}$ and combines them as follows

$$\mathcal{Q} = W_{joint}tanh(\mathcal{F} + \mathcal{G}) + b_{joint} \quad (6)$$

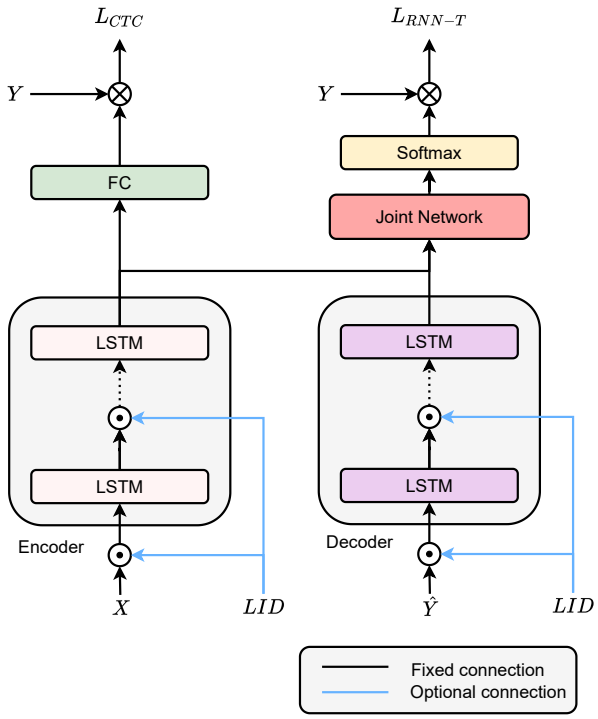

Fig. 1: Schematic diagram of RNN-T architecture with optional Language ID (LID) input.

where W_{joint} and b_{joint} are projection weights and biases and $\mathcal{Q} \in \mathbb{R}^{U \times (V+1) \times K}$ with K as the vocabulary size. Finally, a *softmax* activation is applied over $\mathcal{Q}$ to get the posterior probability distribution of each token symbol k as follows

$$P(k|u, v) = softmax(\mathcal{Q}_{u,v}^k) \quad (7)$$

During training the model optimizes a joint loss L_{Total}

$$L_{Total} = L_{RNN-T} + L_{CTC} \quad (8)$$

where L_{RNN-T} and L_{CTC} are RNN-T loss [16] and CTC loss [15] respectively. The L_{RNN-T} is defined as the negative log posterior of target token sequence Y given the acoustic features X.

$$L_{RNN-T} = -\ln P(Y|X) \quad (9)$$

$$P(Y|X) = \sum_{Y_{align} \in A} P(Y_{align}|X) \quad (10)$$

where A is set all possible alignments of Y (including blank labels). While the L_{CTC} is defined as follows

$$L_{CTC} = -\sum_{\pi \subset \Pi(Y)} \sum_{u}^{U} \ln P(y_u^\pi | x'_u) \quad (11)$$

here $\Pi(Y)$ is defined a set of all possible alignments of target sequence Y which include the blank symbol and repeated tokens from Y. $P(y_u^\pi|x_u)$ is described as the conditional probability of u^{th} label in the π alignment given the $x' = x[u \times t : u \times t + PF]$ (PF is the encoder pooling factor). Logits for the CTC loss are obtained by projecting embeddings F to the vocabulary size K (includes the blank token).

3. TOKENIZATION TECHNIQUES

Tokenization is defined as the process of splitting a word, a sentence or an entire text document into smaller chunks which are then converted to integer ids through a lookup table. There are many tokenization strategies available, but here we will discuss only three techniques, which are most popular in a multilingual context, namely unicode byte encoding [10], byte pair encoding [18] and byte-level byte pair encoding [13, 19].

3.1. Unicode Byte Encoding

Unicode is a standardized mapping from a unicode character to a variable length sequence of bytes. Therefore any sequence of unicode characters can be uniquely represented as a sequence of bytes e.g. 멋있는→ EB A9 8B EC 9E 88 EB 8A 94. There are multiple unicode encoding formats available. In this work, we have used UTF-8 encoding which consists of total 256 bytes. Unicode bytes are language agnostic, thus the same unicode vocabulary can be used for tokenizing almost all languages. This unique property brings several advantages (i) A small softmax layer (ii) The same model can be reused when adding more languages (iii) Bytes get shared among multiple characters for character-rich languages e.g. Korean, thus have a better coverage compared to the corresponding char and subword tokens.

The above advantages make byte-based tokenization a suitable candidate for a multilingual ASR model. However, it has some salient disadvantages as well (i) The length of a tokenized sequence can get long (up to $4\times$ the original length) (ii) Since it segments a text sequence at a finer level compared to char or subword tokenizers, a trained model is more likely to make word-level errors.

3.2. Byte Pair Encoding.

Byte pair encoding (BPE) tokenizes a word by splitting it into smaller subword units. These subwords are learned in a greedy manner using the following algorithm -

- Initialize a vocabulary with a set of all the characters and set a value N for the total number of merge operations
- Split each word in the text corpus into its constituents characters (initial subwords) with an additional end-of-word symbol $< /w >$ appended to the last character

Table 2: Statistics of 4 datasets used in different experiments. Only Spanish ASR data is used from Multilingual LibriSpeech and Librispeech test and dev sets refers to the cleaner versions only

Dataset	Data (hr)		
	train	**dev**	**test**
LibriSpeech [21]	960	10	10
Multilingual LibriSpeech [22]	918	10	10
English	1000	10	10
Korean	1000	10	10

- Next, iteratively count the frequency of all the consecutive pairs of current subwords and find the most frequent pair
- The most frequent pair is then added to the vocabulary and all its occurrences in the text corpus are replaced with the merged pair
- Repeat the above three steps for N-1 times

The final size of this BPE vocabulary is the no of characters plus N.

A BPE subword unit can range from a single character to a complete word, which allows the segmentation of any unseen or rare words as a sequence of subwords and thus alleviates the out of vocabulary issue. However, since a BPE vocabulary is initialized with the complete character set it can get very large for character-rich languages like Korean, even without a single merge operation. This issue gets further aggravated for multilingual text corpus.

3.3. Byte level Byte Pair Encoding.

Byte level subwords or Byte based byte pair encoding (BBPE) was introduced in [13, 20] as a scalable alternative of large BPE vocabularies. The generation logic for BBPE is similar to BPE with some key differences.

- BBPE vocabulary is initialized with bytes instead of chars
- A BBPE subword is made of a sequence of bytes, and thus it may start or end with full or partial characters

A BBPE vocabulary nicely interpolates from complete words for frequent words to bytes for infrequent words/characters. Furthermore, its small initial size (256) enforces a compact representation of tokens and optimal usage of model capacity [20].

Table 3: Performance (WER) of different monolingual and bilingual models. Below en results correspond to LibriSpeech dataset

Output Unit	Experiment	WER ↓	
		en	**sp**
Byte	Monolingual en	8.73	–
	Monolingual sp	–	24.54
	Bilingual en-sp	10.25	21.88
	Bilingual en-sp + LID	9.41	**17.64**
BPE	Monolingual en	**7.52**	–
	Monolingual sp	–	19.57
	Bilingual en-sp	7.55	19.89
	Bilingual en-sp + LID	7.55	**15.47**
BBPE	Monolingual en	7.67	–
	Monolingual sp	–	19.87
	Bilingual en-sp	**7.64**	19.98
	Bilingual en-sp + LID	7.81	**15.52**

Table 4: Performance (WER) of different monolingual and bilingual models. Below en results correspond to in-house English data

Output Unit	Experiment	WER ↓	
		en	**ko**
Byte	Monolingual en	**30.62**	–
	Monolingual ko	–	30.33
	Bilingual en-ko + LID	33.06	**27.72**
BPE	Monolingual en	20.14	–
	Monolingual ko	–	33.69
	Bilingual en-ko + LID	**17.80**	**23.31**
BBPE	Monolingual en	20.31	–
	Monolingual ko	–	24.93
	Bilingual en-ko + LID	**17.24**	**20.11**

4. EXPERIMENTAL SETUP

4.1. Datasets

In this work, we have used two public datasets and our in-house data which is for handling voice requests for TV and mobile etc. English data (960h) from LibriSpeech [21] and Spanish data (918h) from Multilingual LibriSpeech [22] are used for en-sp experiments. Around 1000h of English and Korean data each from our in-house corpus are used for en-ko experiments. These datasets are chosen such that there are languages from the same as well as different language families. Additionally, the sizes of these datasets are comparable hence does not pose data imbalance issues. Further details of these datasets are listed in Table 2.

4.2. Model

Base architecture in all the experiments is an RNN-T network, as illustrated in section 2. The encoder network consists of 6 stacked unidirectional LSTM [23] layers, each with a cell size of 1024 units, followed by a projection layer with an output dimension of 512. In Byte models, each of the first two LSTM layers was followed by a pooling layer to down-sample the number of audio time steps. Similarly, for BPE and BBPE models, three pooling layers were added in a similar pattern. Higher pooling for BBPE and BPE models is proportional to the longer token length of a BBPE/BPE token compared to a byte token. The pooling factor for each pooling layer is kept to 2 in all the architectures. The decoder network consists of a stack of an embedding layer with an embedding size of 512 units, a single unidirectional LSTM layer with the cell size of 1024 units, and a projection layer with an output dimension of 512. The joint network then combines the outputs $\mathcal{F}$ and $\mathcal{G}$ from the two networks using equation (6) to get the logits $\mathcal{Q}$. Finally, a softmax activation is applied to get the posterior probability distribution. Dropout [24] of 0.2 has also been used after each layer except the last softmax layer.

In all our experiments, we have used 40 dimensional *mfcc* features computed with a 25ms window and shifted every 10ms. For faster convergence, we adopted the pre-training scheme used in [1, 25] and trained all the models for 12 epochs (further training does not improve the results). We used a learning rate of 0.0003 with exponential decay strategy and optimized the combined loss L_{Total} using Adam optimizer [26]. We fed an additional one-hot language vector for multilingual experiments with LID during training and inference time. The LID vector is concatenated with the input features and outputs of intermediate layers in both the encoder and the decoder as shown in Fig 1.

5. RESULTS AND ANALYSIS

5.1. Monolingual and Bilingual Experiments

5.1.1. Monolingual Models

All monolingual models use vanilla RNN-T architecture i.e. without LID. Vocabulary sizes for Byte, BPE and BBPE experiments are 256, 10k and 10k respectively for all the languages. All models were evaluated with beam search using a beam size of 8 and results are listed in Tables 3 and 4. We can make some observations: For Korean the order of WERs is $BPE > Byte > BBPE$, while for English and Spanish this order is $Byte > BBPE \approx BPE$.

The order for Korean can be understood since Korean is a character-rich language and its BPE vocabulary mainly com-

prises single characters. The frequency distribution of Korean characters is highly skewed and consequently due to insufficient data, many of it's less frequent Korean characters are not learned optimally. In contrast, a byte vocabulary represents a single Korean character as a sequence of multiple bytes (3-4) and these bytes are shared among multiple characters. This sharing of bytes induces a more uniform distribution of tokens and hence the byte models perform better on infrequent Korean characters. However, a byte model has to learn both intra-char and inter-char or inter-word dependencies, thus is more prone to word-level errors. A BBPE model alleviates this issue by segmenting a word into byte subwords, which have fewer dependencies to learn.

Unlike Korean, tokens of a BPE/BBPE vocabulary of English/Spanish range from a single character to a complete word, with a relatively uniform distribution. Therefore, their BPE/BBPE models give lower WER than their char/byte counterparts. Furthermore, for English and Spanish, each of their characters is mapped with a single unique byte. Thus, their BBPE and BPE subwords are equivalent, and hence both the models give a similar performance.

5.1.2. Bilingual Models

Augmenting multilingual ASR models with a LID has repeatedly been shown to improve model accuracy [13, 10, 27]. Therefore, our bilingual en-sp experiments in Table 3 include both with and without the LID experiments. Due to LID's improvement in earlier experiments, we only conducted en-ko experiments in Table 4 with LID. We have kept the vocabulary size constant to 10k for all the BPE and BBPE bilingual experiments to maintain consistency. All the models were evaluated using beam search with a beam size of 8. Based on the results we can make the following observations - First, we see that providing additional LID information consistently improves model performance over monolingual baseline for at least one language. Second, even though the vocabulary size of bilingual models is smaller than the sum of their monolingual vocabularies, the bilingual models perform similar or better than their monolingual counterparts. Third, for English and Spanish the order of their WERs is the same as for the monolingual models - $Byte > BBPE \approx BPE$ while for Korean, it's slightly changed $Byte > BPE > BBPE$.

Interestingly, we can observe that the en-ko BPE model shows the highest relative improvement for Korean, such that it surpasses the en-ko byte model. However, BBPE models still outperform the BPE model with 3% and 14% reduction in WER for English and Korean, respectively.

5.2. xRT

For an ASR model, its xRT is defined as the ratio of the time it takes to decode an audio to the length of the audio itself. It is one of the significant criteria while comparing two streaming models. Since xRT can vary over different devices, we evaluated all our models on a single machine with Tesla M40 GPU. Table 5 list the xRT values for different languages with different output units. Listed evaluations correspond to the monolingual baseline experiments with 10k vocabulary for BPE and BBPE and 256 tokens for the byte vocabulary. As expected, we observe on average 44% higher xRT for byte models compared to BPE models across all languages, while xRTs are comparable for BPE and BBPE. This can be attributed to more decoding steps with the byte model because of the longer byte sequences. Likewise, we can also observe that for Korean there is a highest relative increase in xRT value around 48%, which is because in UTF-8 encoding each Korean character is made up of 3-4 bytes while for English and Spanish its just 1 byte. Since with bytes, vocabulary size is fixed there is no way to mitigate this increased latency. However, for BPE and BBPE with variable vocabulary lengths, xRT values can be tuned while considering trade-offs like model size and model performance.

Table 5: xRT values for different output units and different languages

Output Unit	xRT ↓		
	en	**ko**	**sp**
Byte	0.315	0.362	0.318
BPE	**0.218**	**0.244**	**0.225**
BBPE	0.226	0.251	0.229

5.3. Model Size

With a growing inclination towards on-device ASR [1, 28], model size is another key characteristic that needs to be optimized. Since we have kept the base architecture fixed, the only differentiator in the model size is the size of the softmax layer or the vocabulary size. Table 6 lists the model sizes for different output units with varying vocabulary lengths. Clearly, the byte model is smaller than the other two models with a significant margin. Interestingly, we can also observe that just by increasing the vocabulary size for BPE and BBPE, the model size increases significantly. In the context of multilingual ASR where even a character-based vocabulary can go

Table 6: Comparison of model sizes for different output units with varying vocabulary sizes

Output Unit	Vocabulary Size	Model Size (in MB) ↓
Byte	256	**211**
BPE	10k	308
BBPE	10k	308
BPE	100k	659

up to 100K length, a BPE vocabulary might not be an optimal strategy. In comparison, a BBPE vocabulary may make a better alternative since by varying its length we can adjust the trade-off between model size and its performance (xRT and WER).

6. CONCLUSION

In this work, we compared the three popular tokenization strategies namely bytes, BPE, and BBPE in the context of monolingual and multilingual ASR. Our experiments compared them over several key criteria like WER, xRT, and memory footprint. We observed that though BPE (or its variants like word-pieces) is the default choice in current monolingual ASR systems, it has some trade-offs when scaled for a multilingual system. On the contrary, despite byte units being a sub-optimal choice for a monolingual ASR, it presents several advantages in different contexts like monolingual ASR for a character-rich language or a multilingual ASR.

In the context of multilingual ASR we are restricted to use a single tokenization strategy. To that end, we empirically showed that the BBPE could be a better alternative than just using bytes or a BPE vocabulary since it encompasses the benefits of both, besides allowing control over different aspects e.g. model size and WER trade-off.

The behavior of the three tokenization strategies may likely differ for truly large multilingual models [11] compared to the bilingual models. Thus, in future we would like to study the different tokenization schemes with larger multilingual ASR models with diverse language sets.

7. REFERENCES

[1] Kwangyoun Kim, Kyungmin Lee, Dhananjaya Gowda, Junmo Park, Sungsoo Kim, Sichen Jin, Young-Yoon Lee, Jinsu Yeo, Daehyun Kim, Seokyeong Jung, et al., "Attention based on-device streaming speech recognition with large speech corpus," in *Proc. ASRU*. IEEE, 2019, pp. 956–963.

[2] Yanzhang He, Tara N Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, et al., "Streaming end-to-end speech recognition for mobile devices," in *Proc. ICASSP*, 2019, pp. 6381–6385.

[3] Tara N. Sainath, Ruoming Pang, David Rybach, Yanzhang He, Rohit Prabhavalkar, Wei Li, Mirk'o Visontai, Qiao Liang, Trevor Strohman, Yonghui Wu, I. Mcgraw, and Chung-Cheng Chiu, "Two-pass end-to-end speech recognition," in *Proc. INTERSPEECH*, 2019.

[4] Abhinav Garg, Dhananjaya Gowda, Ankur Kumar, Kwangyoun Kim, Mehul Kumar, and Chanwoo Kim, "Improved Multi-Stage Training of Online Attention-based Encoder-Decoder Models," in *Proc. ASRU*, 2019.

[5] Chanwoo Kim, Sungsoo Kim, Kwangyoun Kim, Mehul Kumar, Jiyeon Kim, Kyungmin Lee, Changwoo Han, Abhinav Garg, Eunhyang Kim, Minkyoo Shin, Shatrughan Singh, Larry Heck, and Dhananjaya Gowda, "End-to-end training of a large vocabulary end-to-end speech recognition system," in *Proc. ASRU* , 2019.

[6] Dhananjaya Gowda, Abhinav Garg, Kwangyoun Kim, Mehul Kumar, and Chanwoo Kim, "Multi-task multi-resolution char-to-bpe cross-attention decoder for end-toend speech recognition," in *Proc. Interspeech*, 2019.

[7] Abhinav Garg, Gowtham Vadisetti, Dhananjaya Gowda, Sichen Jin, Aditya Jayasimha, Youngho Han, Jiyeon Kim, Junmo Park, Kwangyoun Kim, Sooyeon Kim, Youngyoon Lee, Kyungbo Min, and Chanwoo Kim, "Streaming on-device end-to-end ASR system for privacy-sensitive voicetyping," in *Proc. Interspeech*, 2020.

[8] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech*, 2019.

[9] Gabriel Synnaeve, Qiantong Xu, Jacob Kahn, Tatiana Likhomanenko, Edouard Grave, 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.

[10] B. Li, Y. Zhang, T. Sainath, Y. Wu, and W. Chan, "Bytes are all you need: End-to-end multilingual speech recognition and synthesis with bytes," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5621–5625.

[11] Vineel Pratap, Anuroop Sriram, Paden Tomasello, Awni Hannun, Vitaliy Liptchinsky, Gabriel Synnaeve, and Ronan Collobert, "Massively multilingual asr: 50 languages, 1 model, 1 billion parameters," in *INTERSPEECH*, 2020.

[12] Anjuli Kannan, Arindrima Datta, Tara N. Sainath, Eugene Weinstein, Bhuvana Ramabhadran, Yonghui Wu, Ankur Bapna, Zhifeng Chen, and Seungji Lee, "Large-Scale Multilingual Speech Recognition with a Streaming End-to-End Model," in *Proc. Interspeech 2019*, 2019, pp. 2130–2134.

[13] Changhan Wang, Kyunghyun Cho, and Jiatao Gu, "Neural machine translation with byte-level subwords,"

Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 9154–9160, 04 2020.

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

[15] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31st International Conference on Machine Learning*, Eric P. Xing and Tony Jebara, Eds., Bejing, China, 22–24 Jun 2014, vol. 32 of *Proceedings of Machine Learning Research*, pp. 1764–1772, PMLR.

[16] Alex Graves, "Generating sequences with recurrent neural networks," 2014.

[17] Albert Zeyer, Wei Zhou, Thomas Ng, Ralf Schlüter, and Hermann Ney, "Investigations on phoneme-based end-to-end speech recognition," *arXiv preprint arXiv:2005.09336*, 2020.

[18] 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)*, Berlin, Germany, Aug. 2016, pp. 1715–1725, Association for Computational Linguistics.

[19] Junqiu Wei, Qun Liu, Yinpeng Guo, and Xin Jiang, "Training multilingual pre-trained language model with byte-level subwords," *ArXiv*, vol. abs/2101.09469, 2021.

[20] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al., "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, pp. 9, 2019.

[21] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[22] Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert, "Mls: A large-scale multilingual dataset for speech research," *ArXiv*, vol. abs/2012.03411, 2020.

[23] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[25] Albert Zeyer, Kazuki Irie, Ralf Schlüter, and Hermann Ney, "Improved training of end-to-end attention models for speech recognition," *Interspeech 2018*, Sep 2018.

[26] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[27] S. Punjabi, Harish Arsikere, Zeynab Raeesy, Chander Chandak, Nikhil Bhave, Ankish Bansal, M. Müller, S. Murillo, A. Rastrow, S. Garimella, R. Maas, Mat Hans, A. Mouchtaris, and S. Kunzmann, "Streaming end-to-end bilingual asr systems with joint language identification," *ArXiv*, vol. abs/2007.03900, 2020.

[28] Yanzhang He, T. Sainath, Rohit Prabhavalkar, Ian McGraw, R. Alvarez, Ding Zhao, David Rybach, A. Kannan, Y. Wu, R. Pang, Qiao Liang, Deepti Bhatia, Yuan Shangguan, Bo Li, G. Pundak, K. Sim, Tom Bagby, Shuo-Yiin Chang, K. Rao, and A. Gruenstein, "Streaming end-to-end speech recognition for mobile devices," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6381–6385, 2019.