Subword Regularization: An Analysis of Scalability and Generalization for End-to-End Automatic Speech Recognition

Egor Lakomkin, Jahn Heymann, Ilya Sklyar, Simon Wiesler

Amazon

{egorlako, jahheyma, lisklyar, wiesler}@amazon.de

Abstract

Subwords are the most widely used output units in end-to-end speech recognition. They combine the best of two worlds by modeling the majority of frequent words directly and at the same time allow open vocabulary speech recognition by backoff to shorter units or characters to construct words unseen during training. However, mapping text to subwords is ambiguous and often multiple segmentation variants are possible. Yet, many systems are trained using only the most likely segmentation. Recent research suggests that sampling subword segmentations during training acts as a regularizer for neural machine translation and speech recognition models, leading to performance improvements. In this work, we conduct a principled investigation on the regularizing effect of the subword segmentation sampling method for a streaming end-to-end speech recognition task. In particular, we evaluate the subword regularization contribution depending on the size of the training dataset. Our results suggest that subword regularization provides a consistent improvement of 2–8% relative word-error-rate reduction, even in a large-scale setting with datasets up to a size of 20k hours. Further, we analyze the effect of subword regularization on recognition of unseen words and its implications on beam diversity.

Index Terms: end-to-end speech recognition, regularization, subword units

1. Introduction

Open vocabulary Automatic Speech Recognition (ASR) systems with daily user interactions are constantly challenged by new words unseen during training. This issue is especially pronounced for end-to-end neural models which only have a loose decomposition of acoustic and language modeling components opposed to the hybrid ASR approach. It also makes it hard to model whole words directly since 1) the English vocabulary, for example, can exceed hundreds of thousands of words, which would dramatically increase the network size 2) the training data is sparse and word frequencies follow Zipfian distribution, making it difficult to get enough training samples covering each word, 3) it does not allow for out-of-vocabulary (OOV) words like named entities. Consequently, the first examples of large vocabulary end-to-end ASR systems used characters as output units which in principle allow to emit any word 4) it does not allow for out-of-vocabulary (OOV) words like named entities. Consequently, the first examples of large vocabulary end-to-end ASR systems used characters as output units which in principle allow to emit any word 5) it does not allow for out-of-vocabulary (OOV) words like named entities. Consequently, the first examples of large vocabulary end-to-end ASR systems used characters as output units which in principle allow to emit any word. However, mapping text to subwords is ambiguous and often multiple segmentation variants are possible. Yet, many systems are trained using only the most likely segmentation. Recent research suggests that sampling subword segmentations during training acts as a regularizer for neural machine translation and speech recognition models, leading to performance improvements. In this work, we conduct a principled investigation on the regularizing effect of the subword segmentation sampling method for a streaming end-to-end speech recognition task. In particular, we evaluate the subword regularization contribution depending on the size of the training dataset. Our results suggest that subword regularization provides a consistent improvement of 2–8% relative word-error-rate reduction, even in a large-scale setting with datasets up to a size of 20k hours. Further, we analyze the effect of subword regularization on recognition of unseen words and its implications on beam diversity.

Subwords are the most widely used output units in end-to-end speech recognition. They combine the best of two worlds by modeling the majority of frequent words directly and at the same time allow open vocabulary speech recognition by backoff to shorter units or characters to construct words unseen during training. However, mapping text to subwords is ambiguous and often multiple segmentation variants are possible. Yet, many systems are trained using only the most likely segmentation. Recent research suggests that sampling subword segmentations during training acts as a regularizer for neural machine translation and speech recognition models, leading to performance improvements. In this work, we conduct a principled investigation on the regularizing effect of the subword segmentation sampling method for a streaming end-to-end speech recognition task. In particular, we evaluate the subword regularization contribution depending on the size of the training dataset. Our results suggest that subword regularization provides a consistent improvement of 2–8% relative word-error-rate reduction, even in a large-scale setting with datasets up to a size of 20k hours. Further, we analyze the effect of subword regularization on recognition of unseen words and its implications on beam diversity.

In the context of ASR, subword regularization has only been employed in a few recent publications. Hannun et al. [10] obtained a 4% relative Word Error Rate (WER) reduction on LibriSpeech by sampling segmentations on word-level with a Listen-Attend-Spell (LAS) model. Drexler et al. [11] investigated subword regularization on two end-to-end ASR models: Connectionist Temporal Classification (CTC) [12] and LAS. Their results presented a 3–7% relative WER improvement on the small-scale WSJ dataset (50 hours) but experiments with subword regularization on the larger LibriSpeech dataset (thousand hours) did not yield improvements.

To the best of our knowledge, all the published experimental results evaluating subword regularization on speech recognition tasks were obtained on relatively small data and it remains unknown if this technique scales to larger datasets. At the same time, there is also a lack of analysis on the effect of subword regularization on the ASR model, especially how it affects the
The RNN-T model defines the conditional probability distribution of the next output wordpiece:\n\[ h_{y_{j+1}}^{\text{dec}} = \operatorname{LSTM}(y_{j-1}) \quad (i) \]
\[ h_{y_{j+1}}^{\text{dec}} = \operatorname{LSTM}(y_{j-1}) \quad (ii) \]
where \((i)\) is used during training and conditions on the ground truth previous output label (teacher-forcing) and \((ii)\) is used during inference where we use the hypothesized previous output label \(y_{j-1}\).

The joint network is a function combining transcription and prediction network representations. The common choice is a single layer feedforward network followed by a non-linearity:
\[ \mathcal{F}_{\text{joint}}(h_{enc}^t, h_{dec}^j) = \tanh(W h_{enc}^t + V h_{dec}^j + b) \]  
(3)

Finally, the conditional distribution of a next output label given the observed feature vectors and output labels is obtained by a projection to the vocabulary size and the application of softmax:
\[ p(y_{j+1}|x_1, \ldots, x_t, y_1, \ldots, y_j) = \operatorname{softmax}(W' \mathcal{F}_{\text{joint}}(h_{enc}^t, h_{dec}^j) + b'). \]  
(4)

RNN-T is typically trained to maximize the log-likelihood of the training data. In ASR, a training example is a pair of an acoustic feature sequence \(x = (x_1, \ldots, x_T)\) and a corresponding word sequence \(w = (w_1, \ldots, w_N)\). In this work, we assume that the output labels of the RNN-T system are wordpieces. Usually, only a single wordpiece segmentation corresponding to the word sequence is considered in training, e.g. the longest match for the BPE model or the most likely one with a unigram wordpiece model. We denote this mapping of a word sequence to its first best subword segmentation as \(\sigma_{\text{1best}}\).

The training objective function is then
\[ L(\theta) = \mathbb{E}_{(x, w) \sim \mathcal{D}_{\text{data}}} \mathbb{E}_{y \sim \sigma_{\text{1best}}(w)} \log P(y|x; \theta) \]  
(5)

Computation of the log-likelihood requires marginalizing over all possible alignments, which can be carried out by introducing a special blank output symbol and efficiently using the forward-backward algorithm [13]. Since the objective function is composed of differentiable functions, the ASR system can be trained end-to-end maximizing \(L\).

### 3. Subword Regularization

The idea of subword regularization is to define a probability distribution \(P(y|w)\) over the wordpiece segmentations corresponding to a word sequence \(w\). In the objective function with subword regularization, the conditional log-likelihood of a training example is replaced by an expectation w.r.t. the distribution over all segmentations:
\[ L(\theta) = \mathbb{E}_{(x, w) \sim \mathcal{D}_{\text{data}}} \mathbb{E}_{y \sim \sigma_{\text{1best}}(w)} \log P(y|x; \theta) \]  
(6)

Following Kudo [8], we define \(P(y|w)\) using a wordpiece unigram language model. Typically, the segmentations are restricted to an \(N\)-best list. Further, the wordpiece distribution can be smoothed or sharpened using a temperature parameter \(\alpha\):
\[ P_{\alpha, N}(y|w) = P(y|w)^{\alpha} / Z(w) \quad \text{if } y \in \text{N-best} \]
\[ 0 \quad \text{else} \]  
(7)
where $Z(w)$ is the normalization constant. When $\alpha = 0$, segmentations are sampled from the uniform distribution. In the case of $\alpha = \infty$, the distribution is a delta-function peaked at the most probable segmentation.

Exact computation of the expectation $\mathbb{E}_{w \sim P(y|w)}$ is intractable, because the number of valid segmentations increases exponentially with the length of the word sequence. Instead, the marginalization is included in the stochastic optimization of the objective function, i.e. for every mini-batch we draw a single segmentation $y$ from $P(y|w)$.

4. Experiments

4.1. Data

We evaluate subword regularization on two datasets: an in-house collection of real recordings of natural human interactions with voice-controlled far-field devices and the publicly available LibriSpeech corpus. Our focus is on the far-field speech recognition task.

For the far-field recognition task, we perform experiments on subsets with 2.4k, 5k, 10k, and 20k hours of transcribed audio data. Models are evaluated on two different test sets: GENERAL - a general test set following the same distribution as the training data (247k utterances), and RARE - a sample of utterances containing rare words (52k utterances).

The LibriSpeech corpus consists of 960 hours of English read speech. Models are evaluated on two test sets: test-clean and test-other.

4.2. Experimental setup

Our RNN-T model consists of a five-layer LSTM encoder with 1,024 units each and a two-layer LSTM prediction network with the same number of units. We use the Adam optimizer with a warm-up, hold, and exponential learning rate decay policy and a mini-batch size of 1,536. We extract 64-dimensional Log-Mel-Frequency features with a filterbank size of 64 every 10ms from the input speech signal. Three consecutive frames are stacked, resulting in a 192-dimensional feature vector, which is used as input to the encoder. If not mentioned otherwise, we use an output vocabulary of 4,000 wordpieces. SpecAugment data augmentation is performed on the un-stacked features during training. In our experiments on the far-field dataset, we apply two frequency masks with a maximum size of 24 and one time mask with a maximum adaptive size of ten percent of the utterance length. In our LibriSpeech experiments, we use the LibriFullAdapt augmentation policy proposed in [21]. In all our experiments with subword regularization, we used $\alpha = 0.25$ and $N = 200$ sampling parameters (Eq. 7). Figure 3 illustrates the normalized average edit distance between the sampled output and the most likely segmentation with different values for $\alpha$ and $N$. With our settings, roughly every forth subword is changed.

All results are obtained with a pure RNN-T system without shallow fusion, language model rescoring, or context biasing as would be used in a production setup. We report results with beam search decoding with a beam width of 16.

4.3. Results

In this section, we present results on the far-field and the LibriSpeech dataset. In addition to overall results, we measure the effect of subword regularization when varying the number of wordpieces.

Firstly, we evaluate subword regularization on four subsets of the far-field task with 2.4k, 5k, 10k, and 20k hours of training data. The results are summarized in Table 1. Subword regularization consistently improves WER for the general test set and the rare word test set. The gains are largest for the smaller datasets with up to 8% rel. WER, but even with 20k hours we still observe a two percent relative WER improvement. Our LibriSpeech experiments are presented in Table 2. For reference, the table provides another result with RNN-T on LibriSpeech. We observe a 2.3% and 1.3% relative WER improvement on test-clean and test-other splits.

To measure the sensitivity of subword regularization to the size of the wordpiece vocabulary, we conduct trainings with the output size of 2.5k, 4k, and 10k wordpieces. The results in Table 3 show that the improvement is largely independent of the vocabulary size.

Table 1: Relative Word Error Rate (WERR) reductions with respect to the 2.4khrs model trained without subword regularization on the GENERAL and RARE test sets. We set WER to 1.0 for the baseline model performance as a reference (trained on 2.4k hours of data with 4k wordpieces). Four experiments were conducted with 2.4k, 5k, 10k, and 20k training subsets of Alex data. Relative Word Error Reduction (%) due to subword regularization is reported in parentheses.

| Experiment | GENERAL | RARE |
|------------|---------|------|
| 2.4khrs    | 1.00    | 2.16 |
| +subword reg. | 0.939 (6.1%) | 2.03 (6.3%) |
| 5khrs      | 0.893   | 1.94 |
| +subword reg. | 0.817 (8.43%) | 1.77 (8.5%) |
| 10khrs     | 0.743   | 1.63 |
| +subword reg. | 0.711 (4.29%) | 1.57 (3.75%) |
| 20khrs     | 0.653   | 1.41 |
| +subword reg. | 0.641 (1.85%) | 1.38 (1.85%) |

Figure 3: Expected numbers of edits per wordpiece for different number of $N$ best segmentations depending on $\alpha$. The black dashed lines indicates our chosen operating point which has a 26% chance of an edit for a single wordpiece.
calculate precision

tp
score

fn
do not emit the correct unseen word (false negative,

a correct (true positive,

tp

cation problem. We calculate how many times the model emits

ingly).

0.85% overall for 5k, 10k, and 20k experiments correspond-

a relatively large fraction of unseen words (1.93%, 1.27%, and

ization on the rare word test set of the far-field task. This has

ing results of models trained with and without subword regular-

izes that the ASR model is more capable of producing words

to a larger variety of wordpiece sequences. Hence, we hypoth-

abled in RNN-T.

because the acoustic and language model are not fully decou-

guage model using shallow fusion, its effectiveness is limited

words. Even though it is in principle possible to apply a lan-

be beneficial to infer the language model context for unseen

s are present in the audio training data. The model hasn’t observed

lary, we expect a lower accuracy for words that haven’t been

Even though our subword-based ASR system is open vocabu-

larization across all metrics. The overall F-score improvement

observe gains in unseen word recognition with subword regu-

5. Analysis

5.1. Modeling unseen words

Even though our subword-based ASR system is open vocabu-

ary, we expect a lower accuracy for words that haven’t been

in the audio training data. The model hasn’t observed

the pronunciation of the full word and, in general, will assign

a low probability for unseen combinations of subword units.

This is a particularly important problem for end-to-end ASR

systems in comparison to HMM-based ASR systems as pronun-

ciations cannot be inferred from a lexicon. Further, the end-to-

tend system doesn’t make use of text training data, which could

be beneficial to infer the language model context for unseen

words. Even though it is in principle possible to apply a lan-

guage model using shallow fusion, its effectiveness is limited

because the acoustic and language model are not fully decou-

ded in RNN-T.

The model trained with subword regularization is exposed

to a larger variety of wordpiece sequences. Hence, we hypothe-

size that the ASR model is more capable of producing words

unseen during training. We verify this by analyzing the decoding

results of models trained with and without subword regular-

ization on the rare word test set of the far-field task. This has

a relatively large fraction of unseen words (1.93%, 1.27%, and

0.85% overall for 5k, 10k, and 20k experiments correspond-

ingly).

In the analysis, we compute precision, recall, and F-score

metrics by treating unseen word detection as a binary classifi-

cation problem. We calculate how many times the model emits a

correct (true positive, tp) or incorrect (false positive, fp) un-

seen word on the test set, and the number of times the model
does not emit the correct unseen word (false negative, fn). We

calculate precision \( \frac{tp}{tp + fp} \), recall \( \frac{tp}{tp + fn} \) and F-

score \( \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \) metrics to evaluate the unseen word

recognition performance. The results are shown in Table 3. We

observe gains in unseen word recognition with subword regu-

larization across all metrics. The overall F-score improvement

ranges from 9% to 28%. As expected, the largest gains are ob-

tained on the smallest dataset.

Table 2: Results with RNN-T models on LibriSpeech test-clean and test-other splits. Numbers in parenthesis are n-best WERs.

| Experiment | test-clean | test-other |
|------------|------------|------------|
| Yeh et al. [16] | 12.31 | 23.16 |
| Our RNN-T | 8.16 (4.40) | 20.22 (14.67) |
| + subword reg. | 7.97 (4.63) | 19.94 (14.81) |

Table 3: WERR reductions of the models with 2.5k, 4k, and 10k wordpiece vocabulary size and subword regularization with respect to the baseline model without subword regularization (trained on 2.4khrs with 4k wordpieces).

| Experiment | GENERAL | RARE |
|------------|---------|------|
| 2.4khrs (4k wp) | 1.0 | 2.16 |
| + subword reg. | 0.939 (6.1%) | 2.03 (6.3%) |
| 2.4khrs (2.5k wp) | 1.05 | 2.16 |
| + subword reg. | 0.99 (5.8%) | 2.04 (5.7%) |
| 2.4khrs (10k wp) | 1.03 | 2.17 |
| + subword reg. | 0.97 (6.2%) | 2.03 (6.3%) |

5.2. N-best WER and beam diversity

In our experiments, we observed that the n-best WER does not improve in contrast to the first best WER as a side effect of subword regularization. We analyzed the hypotheses in the decoding beam and discovered that there is a significant amount of du-
plicate hypotheses. For example, for the 10khrs, 20khrs experi-
ments without sampling, there are on average 97.5% and 96.8% unique hypotheses in the 16-best beam respectively, while for the models with subword regularization the ratio of unique hypo-
theses drops to 56.9% and 54.8%. This directly translates to a degra-
dation in the n-best WER which degrades by 9.6% and 14.0% rel. WER for 10khrs and 20khrs experiments, even though the first best WER improves. We observe a similar
change in our LibriSpeech experiments: with subword regu-
larization, the ratio of unique hypotheses on test-clean reduces from 94% to 71.3%. This behavior is expected for the default beam decoding implementation as the model with subword regu-
larization is trained to explore alternative segmentations and
therefore assigns a more balanced probability mass to segmen-
tations corresponding to the same word sequence.

6. Conclusions

We presented a study on subword regularization applied to streaming end-to-end ASR. Our results suggest that sampling wordpiece segmentations on-the-fly during training improves word error rate across several setups and dataset sizes from 1k to 20k hours. We demonstrate that subword regularization improves recognition accuracy of unseen words which is driven by the enhanced ability of an ASR model to utilize character units.

We leave for a future work an investigation of changes to the beam decoding required to utilize the potential of subword regularization. As we observed that almost 50% of hypotheses in the final beam are non-unique which impacts oracle WER which limits improvements from rescoring methods. Also, we plan to investigate combining several segmentations of an utter-
ance transcription in the loss function to get a better estimation of the probability of a word sequences given the acoustic input.

Table 4: Precision, Recall and F-score of unseen word recognition with 5k, 10k, and 20k hour training datasets.

| Model | Precision | Recall | F-score |
|-------|-----------|--------|---------|
| 5khrs | 0.06 | 0.09 | 0.07 |
| 5khrs + sampling | 0.07 | 0.12 | 0.09 (+28.5%) |
| 10khrs | 0.07 | 0.12 | 0.08 |
| 10khrs + sampling | 0.08 | 0.14 | 0.10 (+25.0%) |
| 20khrs | 0.09 | 0.15 | 0.11 |
| 20khrs + sampling | 0.09 | 0.18 | 0.12 (+9%) |
7. References

[1] H. Soltau, H. Liao, and H. Sak, “Neural Speech Recognizer: Acoustic-to-Word LSTM Model for Large Vocabulary Speech Recognition,” in Proc. Interspeech, 2017, pp. 3707–3711.

[2] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition,” in Proc. ICASSP. IEEE, 2016, pp. 4960–4964.

[3] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen et al., “Deep Speech 2: End-to-end Speech Recognition in English and Mandarin,” in Proc. ICML, 2016, pp. 173–182.

[4] R. Sennrich, B. Haddow, and A. Birch, “Neural Machine Translation of Rare Words with Subword Units,” in Proc. ACL, 2016.

[5] K. Rao, H. Sak, and R. Prabhavalkar, “Exploring Architectures, Data and Units for Streaming End-to-End Speech Recognition with RNN-Transducer,” in Proc. ASRU, 2017, pp. 193–199.

[6] M. Schuster and K. Nakajima, “Japanese and Korean voice search,” in Proc. ICASSP. IEEE, 2012, pp. 5149–5152.

[7] R. Chitnis and J. DeNero, “Variable-length Word Encodings for Neural Translation Models,” in Proc. EMNLP, 2015, pp. 2088–2093.

[8] T. Kudo, “Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates,” in Proc. ACL, 2018, pp. 66–75.

[9] I. Provilkov, D. Emelianenko, and E. Voita, “BPE-Dropout: Simple and Effective Subword Regularization,” arXiv preprint arXiv:1910.13267, 2019.

[10] A. Hannun, A. Lee, Q. Xu, and R. Collobert, “Sequence-to-Sequence Speech Recognition with Time-Depth Separable Convolutions,” in Proc. Interspeech, 2019, pp. 3785–3789.

[11] J. Drexler and J. R. Glass, “Subword Regularization and Beam Search Decoding for End-to-End Automatic Speech Recognition,” in Proc. ICASSP, 2019, pp. 6266–6270.

[12] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks,” in Proc. ICML, 2006, pp. 369–376.

[13] A. Graves, “Sequence Transduction with Recurrent Neural Networks,” arXiv preprint arXiv:1211.3711, 2012.

[14] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang et al., “Streaming End-to-End Speech Recognition for Mobile Devices,” in Proc. ICASSP, 2019, pp. 6381–6385.

[15] J. Li, R. Zhao, H. Hu, and Y. Gong, “Improving RNN Transducer Modeling for End-to-End Speech Recognition,” in Proc. ASRU, 2019, pp. 114–121.

[16] C.-F. Yeh, J. Mahadeokar, K. Kalgaonkar, Y. Wang, D. Le, M. Jain, K. Schubert, C. Fuegen, and M. L. Seltzer, “Transformer-Transducer: End-to-End Speech Recognition with Self-Attention,” arXiv preprint arXiv:1910.12977, 2019.

[17] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “LibriSpeech: an ASR Corpus based on Public Domain Audio Books,” in Proc. ICASSP. IEEE, 2015, pp. 5206–5210.

[18] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in Proc. ICLR, 2015.

[20] D. S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition,” in Proc. Interspeech, 2019, pp. 2613–2617.

[21] D. S. Park, Y. Zhang, C.-C. Chiu, Y. Chen, B. Li, W. Chan, Q. V. Le, and Y. Wu, “SpecAugment on Large Scale Datasets,” ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), May 2020.