Interspeech 2018
2-6 September 2018, Hyderabad

Subword and Crossword Units for CTC Acoustic Models

Thomas Zenkel[1], *Ramon Sanabria*[2], *Florian Metze*[2] *and Alex Waibel*[1,2]

[1]Karlsruhe Institute of Technology, Karlsruhe, Germany
[2]Carnegie Mellon University, Pittsburgh, PA, U.S.A.

thomas.zenkel@kit.edu, ramons@cs.cmu.edu, fmetze@cs.cmu.edu, ahw@cs.cmu.edu

Abstract

This paper proposes a novel approach to create a unit set for CTC-based speech recognition systems. By using Byte-Pair Encoding we learn a unit set of arbitrary size on a given training text. In contrast to using characters or words as units, this allows us to find a good trade-off between the size of our unit set and the available training data. We investigate both crossword units, which may span multiple words, and subword units. By evaluating these unit sets with decoding methods using a separate language model, we are able to show improvements over a purely character-based unit set.

Index Terms: automatic speech recognition, decoding, neural networks

1. Introduction

Traditional automatic speech recognition (ASR) systems consist of three components: a phoneme-based acoustic model (AM), a word-based language model (LM), and a pronunciation lexicon, which maps a sequence of phonemes to words [1]. This setup works well for systems based on hidden Markov models (HMMs) as well as for 'end-to-end' systems [2, 3], where the AM is trained towards a sequence loss, such as Connectionist Temporal Classification (CTC) [4]. An expert-created phone dictionary will contain accurate pronunciations, and often multiple variants for frequent words, which facilitates the AM's task.

A CTC model's 'spike-train' pattern marginalizes over all possible alignments of the output symbols, and bi-directional long-short term memory (LSTM) networks can learn temporal patterns well [5]. CTC models thus perform surprisingly well, even when using context-independent characters as the AM's units, and in languages with 'irregular' pronunciations, such as English. Because of CTC's independence assumption, the model cannot however explicitly learn co-articulation patterns, which should help improve performance.

If characters are used as the units of the AM, the LM can directly be implemented as a recurrent neural network (RNN) [6]. Together with a CTC AM, it allows the creation of 'all-neural' systems, which are not restricted to decoding within the weighted Finite State Transducer (WFST) framework.

Recent work shows that CTC models can directly predict word units, albeit on an extremely large training corpus [7]. [8] circumvent this problem by pre-training the AM with phoneme sequences. Unfortunately, using words as output units re-introduces a fixed vocabulary and only frequent words will be trained robustly.

Character n-gram units provide a trade-off between character and word-based unit sets. When using n-grams, the number of possible outputs increases exponentially with respect to the longest n-gram length. For this reason [9, 10] constrain the length of their n-gram units to a fixed length and only use selected subword n-grams.

Additionally, none of the mentioned grapheme-based approaches deal with crossword pronunciation phenomena like contractions and reductions. In conversational English, 'kind of' and 'going to' are often pronounced as 'kinda' and 'gonna.' It may be useful to treat such phenomena as their own units, since the AM could then learn these reduced pronunciations [11].

In this work we propose a method to create a unit set for CTC AMs based on the Byte-Pair Encoding (BPE) algorithm [12]. This tackles two shortcomings of n-gram units: we do not restrict our units to have a fixed length, and we only create units that frequently appear in a given training corpus. Due to the iterative construction of the unit set, it is easy to empirically determine the best trade-off between the size of our unit set and the number of training labels. Additionally, we do not restrict our units to be part of a word only. In this paper we investigate the use of subword and crossword BPE units for CTC speech recognition systems and compare them to character and word units. We compare WFST and RNN decoding approaches.

Compared to other approaches to learning the units of an ASR system, the main advantage of this method is that it does not rely on phonetics at all, and can be trained on text only, which greatly speeds up system-building efforts. Additionally, we still keep the ability of recognizing arbitrary words and do not constrain our system by using a fixed vocabulary.

2. Related Work

The problem of selecting a unit set is closely related to finding a decomposition of the target sequence into basic units. The straightforward variant is to rely on a fixed decomposition of the target sequence, hence a given target sequence will always be split into the same units.

Many approaches using fixed decomposition have been proposed to solve different problems. An approach used for language models is to keep frequent words as units and split the remaining units into syllables [13]. In machine translation systems the Byte-Pair Encoding algorithm is used to split infrequent words into subword units [14]. This can improve the translation quality of compounds as well as cognates and loanwords. For speech recognition applications, [11] deals with crossword pronunciation phenomena. By adding frequent multiwords like 'sort of' and 'kind of' and their corresponding pronunciations to the pronunciation lexicon they were able to improve recognition performance.

Instead of using a fixed decomposition, it is also possible to learn a variable decomposition of the target sequence. The decomposition can for example depend on speaking style for speech recognition tasks. This approach was successfully applied to neural speech recognition systems [9, 10]. [10] extends

10.21437/Interspeech.2018-2057

the CTC loss function to learn the alignment between the input and the target sequence as well as a suitable decomposition of the target sequence. The unit set consists of character n-grams up to a fixed length and infrequent n-grams are omitted in the unit set. Because of the additional task of learning a decomposition, more training data is necessary.

Another recent trend is to combine word and character-based acoustic models. Word-based models use a finite vocabulary and model words which do not appear in the vocabulary as a special output token, normally called the OOV token. Instead of outputting the OOV token, [15, 16] output characters from a different acoustic component for OOVs.

3. Unit Selection

For creating our units we use Byte-Pair Encoding (BPE) [12]. BPE is a compression algorithm that iteratively replaces the most frequent pair of units (or bytes) with an unused unit. Our initial unit set for all our experiments consists of alphanumerical characters, tokens to represent various noises, and some special characters (-'/&). With each step the unit set grows by one and thus we can create an arbitrary number of units. For example, if the units 'AB' and 'CDE' are already in the unit set and 'AB' frequently appears before 'CDE', the new unit 'ABCDE' will be created and added to the unit set. We use the BPE algorithm to create subword units as well as crossword units.

3.1. Subword Units

To create subword units we use the method proposed for machine translation in [14]. We start the algorithm using our initial unit set. A special token (in our case '@') is used to denote if the unit appears within a word. We do not use a dedicated space character, as the word boundaries are defined by the absence of the token '@' within a unit. Crossword boundaries are not considered in this case; we only count the co-occurrences of units within a word. Since BPE creates a new unit at each iteration, very frequent words are eventually merged into a single unit.

3.2. Crossword Units

To create crossword units we slightly change the algorithm used for the subword units. In place of an end of word symbol, we mark the beginning of each word with a capital letter. This idea is inspired by the unit set used for the speech recognition system in [17]. For example, the utterance 'i don't know' will be preprocessed to 'IDon'tKnow'. Notice that we do not use a dedicated space character, since the word boundaries are modeled using capital letters. We now apply BPE and also merge across word boundaries. We argue that this method is superior to directly applying BPE to utterances containing spaces. When doing this, it is possible that the unit set will now include 'know', '_know', 'know_' as well as '_know_' (for visibility spaces are replaced by '_').

3.3. Comparison

We compare the decomposition of a given target sequence in Table 1. We show an example of both small and large unit sets for each approach, subword and crossword. The small set is created by applying 300 merge operations, while the large set is created by applying 10,000 merge operations. For small subword units, all words except very frequently appearing words are split into subword units. Large subword models almost resemble word units; however, very infrequent words or words that did not appear in the training corpus are still split into multiple subword units.

In contrast, the crossword units also include units consisting of multiple words. Small subword units already contain multi-words for very frequent expressions. The units can become longer for the large model and also less frequent expressions are merged into a single unit. The length of both the subword and the crossword sequences is significantly lower than the length of the character sequences.

Table 1: *An example utterance split into characters and processed with the subword and crossword BPE algorithm when creating 300 and 10,000 additional units, respectively. In the subword model the character '@' denotes that the unit is not the end of the word.*

Method	Utterance
Original	you know it's no not even cold weather
Character	y o u k n o w i t ' s n o n o t e v e n c o l d w e a t h e r
Subword 300	you know it's no not even co@ ld w@ ea@ ther
Subword 10k	you know it's no not even cold weather
Crossword 300	YouKnow It's No Not E ven Co ld We at her
Crossword 10k	YouKnowIt's No NotEven ColdWeather

Since we always keep single characters in the unit set, we are able to model arbitrary words with both approaches. This enables our speech recognition system to be open vocabulary.

4. Acoustic Model

The AM of our system is composed of four bidirectional LSTM layers [18] with 320 units in each direction followed by a softmax layer. The size of the softmax layer depends on the unit set we use. We jointly train the whole model under the CTC loss function [4].

To train the AM we use the 300h Switchboard data set (LDC97S62). We perform data augmentation as in [19], creating 3 subsamples with a reduced frame rate (*i.e.* from 10ms to 30ms) from each original sample. The code is open-sourced in EESEN [20]; the `tf_clean` branch was used to train our models. To improve training stability, we pre-trained each model using only character labels until its convergence. Afterwards, we train with one of the unit sets described in section 3. We optimize the parameters of the network using stochastic gradient descent. We halve the learning rate after each epoch where the validation accuracy does not improve.

5. Decoding Strategies

In this section we briefly summarize the different approaches to generate a transcription given the static sequence of probabilities generated by the acoustic model. For a more detailed description we refer readers to [19].

5.1. Greedy Decoding

To estimate the quality of the AM we perform a greedy search without adding any linguistic knowledge by selecting the most probable unit at each frame [4]. By removing repeated units and the blank token, we are able to create a string of units. For the subword units, we insert a space to seperate words between each decoded unit (e.g. 'oh ye@ ah') and remove the sequence

'@ ' to get a sequence of words (e.g. 'oh yeah'). In the crossword case, we simply concatenate the decoded units (e.g. 'OhYeah') and replace all uppercase character with their lowercase counterpart and a space (e.g. 'oh yeah').

5.2. Weighted Finite State Transducer

To improve over simple greedy search, the Weighted Finite State Transducer (WFST) approach adds linguistic information at the word level [20].

The search graph of the WFST is composed of a token WFST, which applies the CTC squash function (i.e. removing repeated characters and the blank token), a lexicon WFST that maps a sequence of units to words, and a grammar WFST that is modeled by a word based n-gram language model.

The search graph is used to find the most probable word sequence. We use a trigram and a 4-gram LM smoothed with Kneser Ney discounting. We train the LMs with cleaned Switchboard and Fisher transcripts. We do not use WFST decoding for crossword units. However, this would be possible by introducing multi-words into the lexicon.

5.3. Beam Search with a RNN Language Model

In contrast to the WFST decoding, we combine the information of the AM and the RNNLM directly at each frame. This is possible because we train the LM on the same unit set as the AM. We additionally add symbols denoting the start and the end of a sequence to the LM unit set. To find the most probable output sequence, we apply beam search similarly to [21]. For a mathematical formulation of the search, we refer to [19].

We use a two-layer LSTM network with 1024 hidden units at each layer and a 256-dimensional embedding layer as our LM. We use the same cleaned Switchboard and Fisher transcripts as our training corpus for the RNNLM. We optimize the network with adam [22]. We halve the learning rate and restart adam whenever our validation cost does not decrease [23].

6. Results

The 2000 HUB5 'Eval2000' (LDC2002S09) set is used for evaluation. The corpus consists of English telephone conversations and is divided into the 'Switchboard' subset, which more closely resembles the training data, and the 'Callhome' subset. We evaluate both the crossword and subword models with both beam search using an RNNLM and greedy decoding, which does not use additional linguistic knowledge from the LM. For subword models we also apply WFST decoding. For each approach, we evaluate with 300, 600, 1k, 3k, 6k and 10k units.

6.1. Evaluation of Subword and Crossword Units

Figure 1 summarizes the results. For the smallest unit set (300), the crossword model outperforms the subword model. While the word error rate (WER) of the subword model constantly decreases, the WER of the crossword model increases. One problem of the crossword model is that the length of the units constantly increases (e.g. "You'reNotGoingTo"), while for the subword model units cannot be longer than a single word. We argue that it is more difficult for the AM to recognize long expressions. The other drawback of a bigger unit set is that we have fewer training examples per unit. This holds for both approaches; however, with the subword model frequent words will still have a higher number of training examples, as words cannot be represented as longer units. As an example, consider the word '*kinds*': in the subword 10k model, this unit has 539 training examples, while for the crossword 10k model, these are scattered between multiple units ("AllKindsOf" (203), "KindsOf" (158), "KindsOfThings" (75) etc). Because of a lower number of training examples, we argue that the crossword model is not able to learn robust representations for these units. This is also supported by the fact that large crossword models output blanks in place of a reasonable unit in many situations. The deletion rate consistently increases with the size of the unit set, reaching 14.3% for the crossword 10k model.

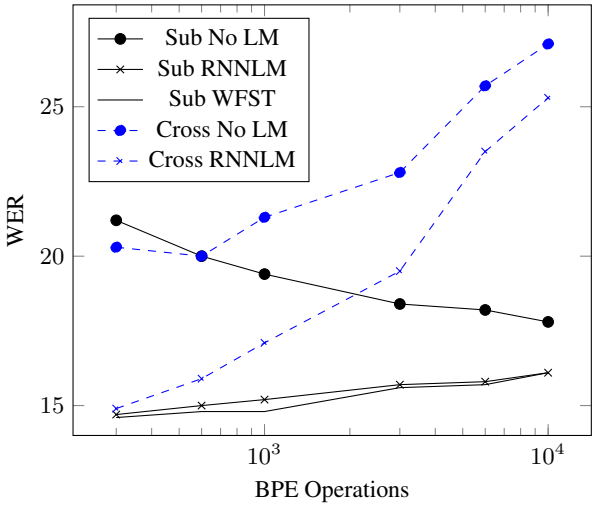

Figure 1: *Comparison of Subword (Sub) and Crossword (Cross) unit sets using no LM, a word-based LM (WFST), and an RNNLM. We report the Word Error Rate on the Switchboard subset of Eval2000.*

Adding linguistic knowledge during decoding always improves our results. For the large crossword models, the RNNLM is not able to fix the discussed drawbacks; even with a tuned insertion bonus, the deletion rate remains high (Table 2). We achieve the best results when using models with small unit sets. We argue that by using smaller units we can include the linguistic knowledge earlier in the search process. We do not have to wait until the AM has recognized a word, but in most cases can combine the information from the AM and the LM already at the subword level.

Table 2: *Substitution (S), Insertion (I), Deletion (D) and Word Error Rates (WER) for different unit sets using an RNNLM during decoding on the Switchboard subset of Eval2000.*

Method	S	D	I	WER
Subword 300	8.8%	3.5%	2.4%	14.7%
Subword 10k	8.0%	6.0%	2.1%	16.1%
Crossword 300	8.8%	4.0%	2.1%	14.9%
Crossword 10k	9.9%	12.2%	3.2%	25.3%

We also found that the training time of each model varies according to the number of units used. For instance, a training epoch of the subword AM with 300 output units, 4 layers, and 320 cells takes 42 minutes, and the same model with 10k output units takes 166 minutes using an Nvidia GeForce GTX 1080 Ti. This is most likely due to the final projection layer which maps the hidden units towards a desired number of units.

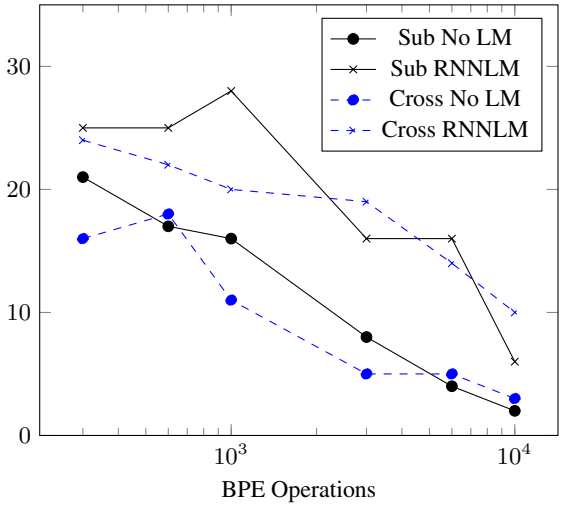

Figure 2: *Number of words which were correctly recognized during greedy and RNNLM decoding (y-axis), but did not appear in the acoustic training corpus.*

6.2. Recognition of unseen words

While the RNNLM is slightly inferior to WFST decoding, it is still able to output arbitrary words and is not restricted by a fixed vocabulary. In Figure 2, we analyze the number of words which did not appear in the training set of the acoustic model but were nonetheless correctly recognized in the test set. We report these numbers for greedy decoding, which means without adding any LM information. We notice that we are able to recognize more previously unseen words when using a smaller unit set. With an increasing unit set size, fewer unseen words are recognized correctly. This situation also holds for decoding with the RNNLM. When using the same unit set for both the AM and the LM, we are still able to output arbitrary words.

Most new words we are now able to recognize consist of previously seen words with different prefixes or suffixes. Examples for the subword 300 unit set include 'interactions', 'unofficial', 'humiliated', 'decency' and 'clunker'. We argue that a small unit set allows us to learn prefixes and suffixes more easily. This does not hold for the bigger unit sets, because most of the tokens seen during training are complete words (97% when using the subword 10k model). When using smaller unit sets, words are more often split into subword tokens.

6.3. Comparison to previous work

We compare our results to previous work using the 300h Switchboard training set in Table 3. We focus the comparison on CTC models with grapheme-based unit sets. For decoding strategies without using linguistic information, we report gains compared to our previous character based system trained on the same architecture. Highly tuned models which mainly focus on words as their output unit, such as [16], yield better results. However, word-based models do not profit as much from using an LM during decoding. Using LM information during decoding can be an advantage if for example the acoustic training data does not match the test domain. Another advantage of BPE units is that they are able to recognize arbitrary words. When including LM information during decoding, we still improve by more than 2% compared to the same system using a character unit set, and report slight improvements compared to the character-based model of [17].

While attention-based methods only show modest improvements when using bigger unit sets [9, 24], selecting an appropriate unit set for CTC systems seems to be more important. This is also in line with the results from [25], who report significant gains when switching from characters to words as output units. We argue that it is more difficult for CTC models to learn an implicit LM, which makes it hard to produce the correct spelling of words when only characters are used as output units.

Table 3: *Comparison of our results to related work on grapheme-based CTC ASR systems using no LM at all ('No LM') and using a LM during decoding ('LM'). We report the WER on the Switchboard (SW) and Callhome (CH) subsets of Eval2000. [19] represents the character baseline for our BPE experiments [ours].*

Category	Unit set	SW	CH
No LM [19]	Character	30.4%	44.0%
No LM [ours]	Subword 10k	17.8 %	29.0%
No LM [16]	Words & Characters	14.4%	24.0%
LM [19]	Character	17.0%	30.2%
LM [ours]	Subword 300	14.7%	26.2%
LM [17]	Characters	15.1%	26.3%

Another advantage of BPE-based approaches to create the units is that we can create a unit set of an arbitrary size. Thus, we can adapt the size of our unit set to the size of the training data. Furthermore, we can easily create a number of diverse models which are encouraged by their distinctive unit sets to learn different concepts. We combined the crossword and subword systems decoded with the RNNLM with ROVER [26] using majority voting without any confidence scores. This yields a word error rate of 11.2% on the Switchboard test subset, which represents a 3.5% improvement compared to our best single system.

7. Conclusions

In this paper we discussed two different methods to create a unit set for CTC-based speech recognition systems. Our method creates unit sets of any desired size, thus providing a method to conveniently adjust the size of the unit set to the amount of the available training data.

We believe that this work shows that there is still room for improvement in automatically selecting a unit set for a given dataset. We will continue to improve the automatic selection of units, especially to remedy the drawbacks of crossword models. Using more knowledge about the input sequence to directly benefit from the pronunciation might be a good starting point.

We argue that the performance of methods like Gram-CTC and Latent Sequence Decomposition could improve when using a BPE unit sets, but also methods relying on less training data usage might be beneficial for systems focusing on a variety of low-resource languages. Also, in situations where the acoustic training data does not match the test domain, one might want to rely more on the information from the language model. Our results suggest that this is possible by specifying a unit set of a smaller size. In the future, we will investigate the use of BPE units in more diverse training and testing scenarios.

8. References

[1] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[2] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1764–1772.

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

[4] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[5] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional lstm networks for improved phoneme classification and recognition," *Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005*, pp. 753–753, 2005.

[6] A. L. Maas, Z. Xie, D. Jurafsky, and A. Y. Ng, "Lexicon-free conversational speech recognition with neural networks." in *HLT-NAACL*, 2015, pp. 345–354.

[7] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word lstm model for large vocabulary speech recognition," *arXiv preprint arXiv:1610.09975*, 2016.

[8] K. Audhkhasi, B. Ramabhadran, G. Saon, M. Picheny, and D. Nahamoo, "Direct acoustics-to-word models for english conversational speech recognition," *arXiv preprint arXiv:1703.07754*, 2017.

[9] W. Chan, Y. Zhang, Q. Le, and N. Jaitly, "Latent sequence decompositions," *arXiv preprint arXiv:1610.03035*, 2016.

[10] H. Liu, Z. Zhu, X. Li, and S. Satheesh, "Gram-ctc: Automatic unit selection and target decomposition for sequence labelling," *arXiv preprint arXiv:1703.00096*, 2017.

[11] M. Finke and A. Waibel, "Speaking mode dependent pronunciation modeling in large vocabulary conversational speech recognition." in *EUROSPEECH*, 1997.

[12] P. Gage, "A new algorithm for data compression," *The C Users Journal*, vol. 12, no. 2, pp. 23–38, 1994.

[13] T. Mikolov, I. Sutskever, A. Deoras, H.-S. Le, S. Kombrink, and J. Cernocky, "Subword language modeling with neural networks," 2012.

[14] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[15] J. Li, G. Ye, A. Das, R. Zhao, and Y. Gong, "Advancing acoustic-to-word ctc model," *arXiv preprint arXiv:1803.05566*, 2018.

[16] K. Audhkhasi, B. Kingsbury, B. Ramabhadran, G. Saon, and M. Picheny, "Building competitive direct acoustics-to-word models for english conversational speech recognition," *arXiv preprint arXiv:1712.03133*, 2017.

[17] G. Zweig, C. Yu, J. Droppo, and A. Stolcke, "Advances in all-neural speech recognition," *arXiv preprint arXiv:1609.05935*, 2016.

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

[19] T. Zenkel, R. Sanabria, F. Metze, J. Niehues, M. Sperber, S. Stüker, and A. Waibel, "Comparison of decoding strategies for ctc acoustic models," in *Proceedings of the 17th Annual Conference of the International Speech Communication Association, Interspeech 2017*. International Speech Communication Association, 2017.

[20] Y. Miao, M. Gowayyed, and F. Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 167–174.

[21] K. Hwang and W. Sung, "Character-leel incremental speech recognition with recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5335–5339.

[22] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] M. Denkowski and G. Neubig, "Stronger baselines for trustable results in neural machine translation," *arXiv preprint arXiv:1706.09733*, 2017.

[24] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, K. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," *arXiv preprint arXiv:1712.01769*, 2017.

[25] J. Li, G. Ye, R. Zhao, J. Droppo, and Y. Gong, "Acoustic-to-word model without oov," *arXiv preprint arXiv:1711.10136*, 2017.

[26] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*. IEEE, 1997, pp. 347–354.