FEW-SHOT LEARNING WITH ATTENTION-BASED SEQUENCE-TO-SEQUENCE MODELS

Bertrand Higy[1,2*] *and Peter Bell*[3]

[1] iCub Facility, Istituto Italiano di Tecnologia, Italy
[2] Università degli studi di Genova, Italy
[3] Centre for Speech Technology Research, University of Edinburgh, UK

ABSTRACT

End-to-end approaches have recently become popular as a means of simplifying the training and deployment of speech recognition systems. However, they often require large amounts of data to perform well on large vocabulary tasks. With the aim of making end-to-end approaches usable by a broader range of researchers, we explore the potential to use end-to-end methods in small vocabulary contexts where smaller datasets may be used. A significant drawback of small-vocabulary systems is the difficulty of expanding the vocabulary beyond the original training samples – therefore we also study strategies to extend the vocabulary with only few examples per new class (few-shot learning).

Our results show that an attention-based encoder-decoder can be competitive against a strong baseline on a small vocabulary keyword classification task, reaching 97.5% of accuracy on Tensorflow's Speech Commands dataset. It also shows promising results on the few-shot learning problem where a simple strategy achieved 34.8% of accuracy on new keywords with only 10 examples for each new class. This score goes up to 80.3% with a larger set of 100 examples.

Index Terms— Automatic speech recognition, end-to-end models, keyword recognition, small vocabulary, few-shot learning

1. INTRODUCTION

Vocal interfaces are becoming more and more popular as our devices (e.g. smartphones, tablets or more recently smart speakers) are becoming more intelligent. Speech is an intuitive and effective way to transmit commands, which makes it very appealing. However, the complexity of modern speech recognition technology and the difficulty of gathering the necessary data can make it hard for single individuals or small companies to develop their own systems, even for small vocabulary recognition. This paper explores strategies to train a keyword/command recognition system by trading off the size of the vocabulary against the quantity of data available. In other words, we focus on low resource, small vocabulary tasks, with a strong bias toward simplicity. Our motivation is similar to that behind the Google Tensorflow team's release of the Speech Commands (SC) dataset [1] and the organization of an accompanying challenge[1]

Together with the SC dataset, Google released a baseline classification system[2] which was used as a starting point by many challenge participants. To enable a simple classification system to be used directly without the use of time-warping or other dynamic programming algorithms, every input file in the dataset is constrained to a fixed length, something that would not be required by the more

flexible standard HMM-based approaches to speech recognition. Although the fixed-length constraint is not unreasonable for a small vocabulary keyword recognition task, we were motivated to consider a more recent end-to-end (E2E) approach – specifically the attention-based encoder-decoder architecture – as a means of allowing input of arbitrary length, whilst retaining the simplicity of a single DNN-based discriminative classifier. This approach also allows us to readily switch between sub-word (phoneme or grapheme) and word modeling by just changing the target output.

In this paper, we experiment with the use of a sequence-to-sequence (S2S) model for a modified version of the Speech Commands task, comparing it with the traditional deep neural network (DNN)-HMM approach. In the literature, S2S models are usually applied on large vocabulary tasks with large datasets and it is not obvious that they will work well in our setup.

The obvious limitation of the small vocabulary approach we take is that trained system is confined to the list of commands defined in the original data. To alleviate this constraint, we also explore strategies to extend the set of commands requiring very few examples (the few-shot learning problem [2, 3, 4]).

The remainder of the paper is organized as follows: relevant literature is presented in section 2, methodology and experiments in sections 3 and 4 respectively, and we conclude in section 5.

2. RELATED WORK

E2E training has attracted much attention recently. One of the first breakthroughs came from the connectionist temporal classification (CTC) loss [5], which allows an acoustic neural model to be trained directly on unsegmented data. While the original technique is not E2E, it has later been extended to train models that predict grapheme sequences [6] or in conjunction with a language model (LM) based on recurrent neural networks (RNNs), an architecture refered to as the RNN-transducer [7]. More recently, the attention-based encoder-decoder model has been applied to automatic speech recognition (ASR) (see e.g. [8, 9]).

If the simplicity of the training procedure of E2E systems is attractive, they generally show reduced performance over traditional HMM-based systems, especially so when used without an external LM, a good example being [9]. Using a much bigger dataset [10] managed to reach competitive results on a dictation task, but was still performing worse on voice-search data. This doesn't mean though that E2E models will necessarily be bad in lower resource conditions. For example, [11] achieved competitive results on several languages, even though it failed to surpass a DNN-HMM baseline. To the best of our knowledge, E2E models have never been applied to small vocabulary speech recognition tasks before. The work closest to ours is probably [12] where an attention-based E2E architecture is applied to keyword spotting. Though, despite the vocabulary being reduced to one word, a very large dataset is used.

Between the different E2E approaches, the attention-based encoder-decoder architecture has been shown to give better results [10]. While the original model [13] was proposed for machine translation, several ways to adapt it for speech recognition have since been proposed. A first difference with machine translation resides in the ratio between the length of the input and output sequences: in speech recognition, the input sequence tends to be much longer than the output sequence. [9] proposed to use pyramidal layers to downsample the input. This reduces the number of hidden states the attention has to attend to, thus improving both the accuracy and the computational performance. Similarly, convolutional neural networks (CNNs) have been shown to be effective [14], leading to further improvement. Another concern pertains to the global attention mechanism which is a bit too flexible for speech recognition (an essentially monotonic left-to-right process). Ways to encourage monotonicity [8] or ensure the local and monotonous nature of the attention system [15] have thus been proposed. Taking a different approach, a hybrid CTC/Attention architecture trained in a multi-task fashion has been proposed in [16, 17]. The idea there is to use the monotonous and left-to-right properties of CTC to find better alignments, which compensate for the over-flexibility of the attention-based decoder.

3. METHODOLOGY

The main task considered here corresponds to the one proposed in Tensorflow's Speech Commands challenge mentioned earlier, that is keyword classification. In addition to the keywords, two additional classes are considered: (i) a `_silence_` category corresponding to records free of speech, and (ii) an `_unknown_` category for records containing speech that is none of the keywords.

3.1. CNN-HMM baseline

In previous experiments on the SC task, we obtained our best performance with a DNN-HMM system using CNNs. The CNN-HMM model has been trained using a standard Kaldi recipe[3] [18]. It is composed of two 2D (across both time and frequency) convolutional layers followed by 4 fully connected layers. The network is trained with cross-entropy, followed by 1 iteration of discriminative training with the state-level minimum Bayes risk (sMBR) objective [19]. We use as input 11 frames (5 from both sides) of 40 filterbank coefficients, augmented with Δ and $\Delta\Delta$ features.

3.2. End-to-end model

We opted for the attention-based encoder-decoder approach, and more precisely the hybrid CTC/Attention model from [16, 17] which showed promising results and for which the code was available[4]. We used a CNN-based encoder that is composed of 4 convolutional layers, with 2 max pooling layers (after the second and fourth convolutions). Each pooling layer has a reduction factor of 2, thus downsampling the timescale of the input by 4 overall. Four layers of 320 bidirectional long-short term memory (BiLSTM) units sit on top of the CNN part.

We used the location-aware attention mechanism [20] and a layer of 300 LSTM cells for the decoder. Default hyperparameters from the `voxforge` recipe were used unless stated otherwise. The input was composed of 80 fbanks and we experimented with 3 different types of labels: phonemes, graphemes and words.

[3]egs/rm/s5/local/nnet/run_cnn2d.sh
[4]https://github.com/espnet/espnet

3.3. Strategies for few-shot learning

The main limitation of our small vocabulary approach is its flexibility. The ASR system is limited to the set of keywords it was trained on and no guarantee is given that it will generalize to new ones (in fact we expect it to recognize them poorly if at all). This is a limitation that is hardly manageable in practical usage. To alleviate it, we propose to explore strategies for few-shot learning, where one can gather few examples of a new word and use them to retrain or adapt the existing system, so that it will perform better on this new word.

The simplest strategy we tried consists in adding the examples of the new keywords to the training set from the beginning and train a new model on it (a method referred to as `retrain` hereafter). One issue with this method is that the new keywords will be underrepresented compared to the original ones. To improve on that, we propose to try oversampling the few-shot examples, that is we will see these examples k times during an epoch (where k is the oversampling factor) when the original examples will be seen only once.

The main limitation of the `retrain` strategy is that it requires to retrain the model from scratch every time. Alternatively, we propose a method based on adaptation (referred to as `adapt`) where we start from a model trained on the 12 original categories (see section 3.2). We then adapt all its weights by training it for a few more epochs on the few-shot examples, keeping the same training procedure otherwise. To avoid performance deteriorating on the original keywords, we also include some of their examples, with the same number of examples per class as for the few-shot classes. The overall number of examples being very small, few updates are made per epoch. We thus expect higher learning rates to be useful. We also optimized the number of epochs which plays a complementary role.

One drawback of this strategy, however, is that the model we start from may not contain all the output labels required for the new keywords (limited to the phonemes or graphemes present in the 10 original keywords). We solve this problem by replacing the missing phonemes (resp. graphemes) by the `UNK` model (resp. the character `_`) initially introduced for the `_unknown_` category (see section 4.1). This can result in a dramatic change as exemplified by the word `backward` for which most phonemes are absent from the pretrained models output. The original transcription "B AE K W ER D" becomes "UNK UNK UNK UNK UNK D" after replacement. Similarly for graphemes, replacing missing characters leads to the transcription "????w?rd". In view of this limitation and in order to compare the `adapt` strategy more fairly with the `retrain` approach, we introduced the `retrain_replace` strategy. This strategy uses the same training procedure as the `retrain` method, but with the modified labels.

4. EXPERIMENTS

4.1. Experimental setup

Our experimental setup is derived from the second version of the SC dataset [1]. It contains 105,800 recordings of 35 different keywords. Each record has a fixed duration of 1 second. The results in [1] correspond to the task that was proposed for the original challenge, that is the classification of 10 keywords (out of the 35 available), the remaining ones being used to populate the `_unknown_` category.

This setup has been slightly modified here. A limitation of the original design is that the same collection of words are used for the `_unknown_` category at both training and test time. In order to better evaluate the generalization capability of the model to unseen words (which will inevitably happen given the number of words we use to train this category), we decided to exclude some of them from the

Set	Words
org_kwd	down, go, left, no, off, on, right, stop, up, yes
org_unk	bed, bird, cat, dog, happy, house, marvin, sheila, tree, visual, wow
new_kwd	forward, four, one, three, two, zero
new_unk	eight, five, follow, learn, nine, seven, six

Table 1: List of keywords assigned to the different sets.

Model	org_kwd	unk	new_kwd
Phoneme-based S2S	**96.6**	**62.0**	-
Grapheme-based S2S	**96.6**	57.6	-
Word-based S2S	96.4	59.5	-
retrain-10	**96.1**	63.6	36.5
retrain_replace-10	96.0	**64.1**	**36.6**
adapt-10	86.8	42.3	23.6
retrain-100	**96.1**	54.2	82.1
retrain_replace-100	**96.1**	**56.8**	81.9
adapt-100	92.3	39.6	**89.9**

Table 2: Validation accuracy (%) of the S2S model with different types of outputs (first set of rows, trained on 12 classes only) or with different few-shot learning strategies (second and third set of rows, trained with 7 additional classes).

Model	org_kwd	unk	new_kwd
CNN-HMM (baseline)	95.8	45.5	-
Phoneme-based S2S	**97.5**	**59.5**	-
retrain-10	**96.9**	**60.3**	**34.8**
adapt-10	86.6	42.5	22.6
retrain-100	**97.0**	**51.8**	80.3
adapt-100	92.4	38.1	**88.6**

Table 3: Test accuracy (%) of the baseline and the main S2S models. The first set of rows correspond to models trained on 12 categories only, while second and third set of rows correspond to few-shot experiments.

training set. Also, as mentioned earlier, we are interested in exploring strategies for few-shot learning. Hence, we also kept a few words aside for use in those experiments.

The 10 original keywords (from here on referred to as the org_kwd set of words) are kept identical. The 25 remaining ones however are split into two main categories: 7 are used as new keywords in the few-shot experiments (refered to as new_kwd) and 18 as unknowns (the unk set). This later group is further split into 11 words (org_unk) that are used for training and evaluation, while the remaining 7 (new_unk) are seen at evaluation time only. Table 1 gives the list of words assigned to each category.

The split of the data in training, validation and test sets was done using the procedure provided in Tensorflow's example code with 80%, 10% and 10% for each set respectively. The _unknown_ category being the combination of several keywords, it is overrepresented in the dataset. To prevent it from dominating the learning procedure, we downsample it for the training set, randomly selecting a number of examples corresponding to the mean number of examples we have for the keywords (org_kwd). Finally, for all experiments on few-shot learning, we randomly sample f examples of each new class from the training records we have kept aside.

In the phoneme-based experiments, we introduce a special phoneme, labelled UNK, to model the words of the _unknown_ category. Similarly, for grapheme-based or word-based experiments, all the words are transcribed with the unique character ?. In all conditions, we further map all output not corresponding to one of the keywords or _silence_ to the _unknown_ category.

4.2. End-to-end approach for small vocabulary ASR

We first report results on the original classification task trained on 12 categories, comparing traditional and E2E pipelines. Table 3 summarizes the results obtained with the S2S model for different types of outputs, where we see that accuracy of the three S2S models on the keywords are very close. Looking at the performance of the same models on the unk set, we see they all show a big drop in accuracy, as expected with only 11 different words to populate the _unknown_ category for training. It can be noticed though that the

phoneme-based S2S model generalizes significantly better than its two competitors.

Hence, the greater simplicity of the grapheme- or word-based approaches, which don't require a pronunciation dictionary, can be traded off for better performance. Moreover, it has to be highlighted that in our small vocabulary context, building the pronunciation dictionary is greatly simplified compared to the large vocabulary context.

In table 3, we compare the test accuracy of the best E2E model with the CNN-HMM baseline. On the main task, the E2E approach beats the baseline by 1.7% absolute. This is very promising as it shows that E2E models are a competitive alternative to more traditional approaches for our task. The results on the unk set shows that they also generalize much better, the E2E approach beating the baseline by 14% absolute on this subset.

Finally, we give some insight on the behavior of the S2S models. As can be seen from figure 1, and confirmed by manual inspection, the attention tends to focus on a single portion of each input and doesn't shift as the output tokens are produced. It appears that the model representation is more akin to word than sub-word modeling, as is usually observed. With our small vocabulary, the model is apparently able to discriminate between the different keywords with a single "glance" at the data. For example, in the case of the word stop, the model seems to attend to the phoneme AA (left part of figure 1). More surprisingly, in the case of the word yes, the model seems to seek information from a fixed position, even if it falls in the silence preceding the word. A more quantitative analysis would be required to better understand those dynamics, which maybe related to the effective window size of the encoder.

4.3. Few-shot learning

For low values of f, the variability introduced by the random selection of the few-shot examples is high. We thus report here the mean accuracy over 3 runs for all experiments on few-shot learning.

We experiment with $f \in \{10, 100\}$. While 100 examples may seem a lot for few-shot learning, it allows to test how the different strategies behave when the number of examples increase. It is also to be noted that 100 examples is only 2.6% of the number of examples available for the original classes ($\sim$ 3850 on average).

For the retrain strategy, we tried oversampling the new keywords up to 3000 simulated examples ($k = 300$ for $f = 10$ and $k = 30$ for $f = 100$), so as to reach similar frequency in training and test sets. As figure 2 shows, the retrain strategy gives very variable results on the new_unk set without oversampling ($k = 1$). In contrast, higher values of k give better scores and lower variability. The best results are obtained with $k = 100$ for $f = 10$, where the grapheme-based model achieves 36.5.% of accuracy. Conversely with $f = 100$, best results are obtained for $k = 30$ with the phoneme-based model, for an accuracy of 82.1%.
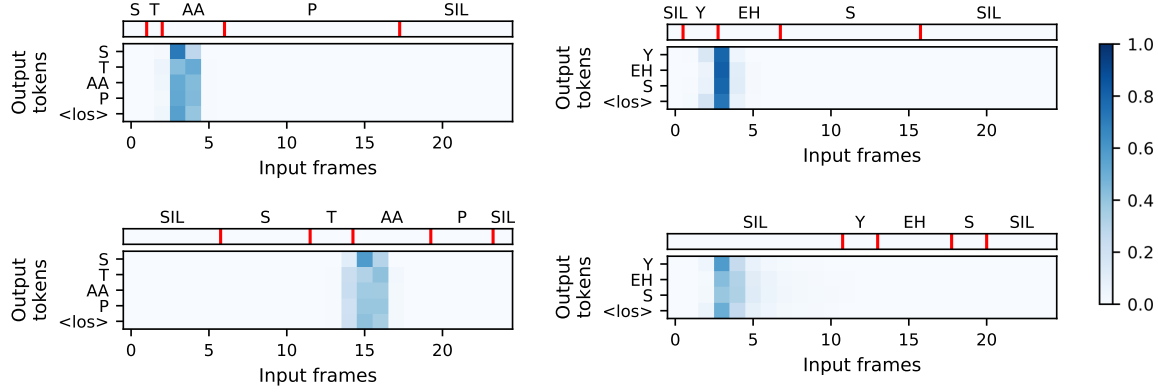
Fig. 1: Attention weights produced by the phoneme-based S2S model for two examples of the words "stop" (left) and "yes" (right), with their respective alignment on top.
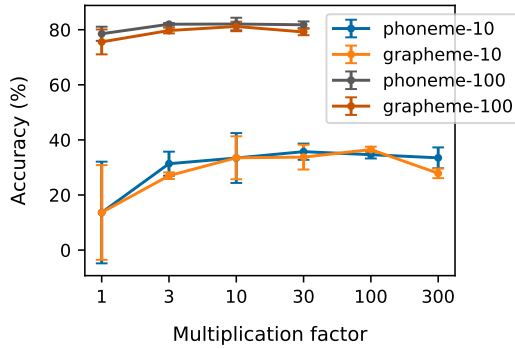

Fig. 2: Validation accuracy for the new keywords with the `retrain` strategy. We compare phoneme- and grapheme-based outputs, for f = 10 or 100. The bars represent the standard deviation over 3 runs.
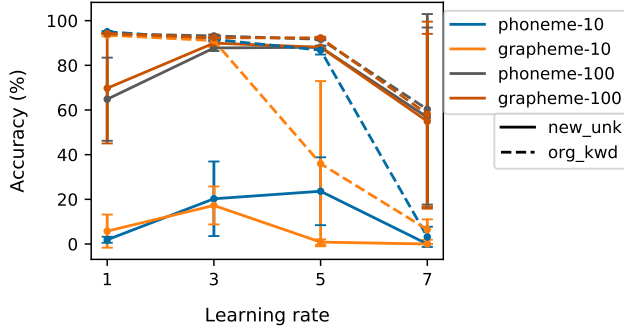

Fig. 3: Validation accuracy for the new keywords (plain) and the original keywords (dashed) with the `adapt` strategy, as a function of the learning rate. Phoneme- and grapheme-based outputs are compared for 10 and 100 fewshot-examples (f).

Figure 3 shows the accuracy with the `adapt` strategy on the `new_kwd` set as a function of the learning rate (lr). For each experiment, the best number of epochs is selected based on the validation set. We see that increasing the learning rate is necessary to get good result, but too high a value and the results deteriorate as we overfit the small training set. The best results are achieved with the phoneme-based output, with a learning rate of 5 (resp. 3) for $f = 10$ (resp. $f = 100$). We also reported the accuracy on the `org_kwd` set to show how the original training is progressively undone as the

learning rate increases. As one can see, the performance degrades consistently and at some point drops suddenly as the model overfits the adaptation set.

Table 2 summarizes the accuracy of the different strategies on the validation set for the two values of f (10 and 100) with the hyperparameters giving optimal scores on the `new_kwd` set. A first and surprising observation is that the phoneme/grapheme replacement rules introduced in section 3.3 for the `adapt` strategy doesn't seem to penalize the performance on the `new_kwd` set. The `retrain_replace` strategy gives results very close to the `retrain` one overall. Comparing the performance of the `adapt` and `retrain` strategies now, we see that adaption is not only much faster to train but is also the most performant on the new keywords for $f = 100$. Though, this is is achieved at the expense of the `org_kwd` and `unk` sets. Table 3 summarizes the test accuracy of the best models for both strategies (`retrain` and `adapt`).

5. CONCLUSION

In this paper, we proposed to study the adequacy of E2E approaches on a small vocabulary task, in order to simplify the process of training a keyword/command recognition system and make this technology more accessible. We found that they can be competitive in such a context, giving better results than a strong CNN-HMM baseline. We also proposed two few-shot strategies. By simply training a model from scratch on the combination of the original dataset and those new examples, we managed to reach 34.8% of accuracy with only 10 examples per new keyword and 80.3% with 100 examples. A faster adaptation strategy was also proposed which achieves even better results with 100 examples reaching 88.6% of accuracy, but at the expense of the performance on the original keywords. This results may be further improved by using more advanced strategies.

We have also shown that the dynamic of the hybrid CTC/Attention model in our task is quite different from what is usually observed with large vocabulary tasks. It would be interesting in the future to analyze more deeply the behavior of the model as one move from a small vocabulary keyword task to a large vocabulary one with complex sentences.

7. REFERENCES

[1] Pete Warden, "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition," *arXiv:1804.03209 [cs]*, Apr. 2018, arXiv: 1804.03209.

[2] Sachin Ravi and Hugo Larochelle, "Optimization as a Model for Few-Shot Learning," in *Proc. of the International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[3] Jake Snell, Kevin Swersky, and Richard Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 4077–4087.

[4] Flood Sung Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018.

[5] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *Proc. of the 23rd International Conference on Machine Learning*, New York, NY, USA, 2006, ICML'06, pp. 369–376, ACM.

[6] Alex Graves and Navdeep Jaitly, "Towards End-to-end Speech Recognition with Recurrent Neural Networks," in *Proc. of the 31st International Conference on Machine Learning*, Beijing, China, 2014, vol. 32 of *ICML'14*, pp. II–1764–II–1772.

[7] Alex Graves, "Sequence Transduction with Recurrent Neural Networks," in *presented at the International Conference of Machine Learning (ICML) Workshop on Representation Learning*, Edinburgh, Scotland, Nov. 2012, arXiv: 1211.3711.

[8] Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "End-to-end Continuous Speech Recognition using Attention-based Recurrent NN: First Results," in *presented at NIPS Workshop on Deep Learning and Representation Learning*, Montréal, Canada, Dec. 2014, arXiv: 1412.1602.

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

[10] Rohit Prabhavalkar, Kanishka Rao, Tara N. Sainath, Bo Li, Leif Johnson, and Navdeep Jaitly, "A Comparison of Sequence-to-Sequence Models for Speech Recognition," in *Proc. of Interspeech*, Stockholm, Sweden, 2017, pp. 939–943.

[11] A. Rosenberg, K. Audhkhasi, A. Sethy, B. Ramabhadran, and M. Picheny, "End-to-end speech recognition and keyword search on low-resource languages," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, Mar. 2017, pp. 5280–5284.

[12] Changhao Shan, Junbo Zhang, Yujun Wang, and Lei Xie, "Attention-based End-to-End Models for Small-Footprint Keyword Spotting," *arXiv:1803.10916 [cs, eess]*, Mar. 2018, arXiv: 1803.10916.

[13] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *Proc. of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015, arXiv: 1409.0473.

[14] Takaaki Hori, Shinji Watanabe, Yu Zhang, and William Chan, "Advances in Joint CTC-Attention based End-to-End Speech Recognition with a Deep CNN Encoder and RNN-LM," in *Proc. of Interspeech*, Stockholm, Sweden, June 2017, arXiv: 1706.02737.

[15] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura, "Local Monotonic Attention Mechanism for End-to-End Speech and Language Processing," in *Proc. of the 8th International Joint Conference on Natural Language Processing*, Taipei, Taiwan, May 2017, arXiv: 1705.08091.

[16] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, Mar. 2017, pp. 4835–4839.

[17] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid CTC/Attention Architecture for End-to-End Speech Recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, Dec. 2017.

[18] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, and others, "The Kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, Waikoloa Village, HI, US, 2011, IEEE Signal Processing Society.

[19] Matthew Gibson and Thomas Hain, "Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition," in *Proc. of the Ninth International Conference on Spoken Language Processing*, Pittsburgh, PA, USA, 2006.

[20] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *Proc. Advances in neural information processing systems*, Montréal, Canada, 2015, pp. 577–585.