Analyzing ASR Pretraining for Low-Resource Speech-to-Text Translation

ANALYZING ASR PRETRAINING FOR LOW-RESOURCE SPEECH-TO-TEXT TRANSLATION

Mihaela C. Stoian, Sameer Bansal, Sharon Goldwater

School of Informatics, University of Edinburgh, UK

{c.mihaela.stoian, sameer.bansal}@ed.ac.uk, sgwater@inf.ed.ac.uk

ABSTRACT

Previous work has shown that for low-resource source languages, automatic speech-to-text translation (AST) can be improved by pretraining an end-to-end model on automatic speech recognition (ASR) data from a high-resource language. However, it is not clear what factors—e.g., language relatedness or size of the pretraining data—yield the biggest improvements, or whether pretraining can be effectively combined with other methods such as data augmentation. Here, we experiment with pretraining on datasets of varying sizes, including languages related and unrelated to the AST source language. We find that the best predictor of final AST performance is the word error rate of the pretrained ASR model, and that differences in ASR/AST performance correlate with how phonetic information is encoded in the later RNN layers of our model. We also show that pretraining and data augmentation yield complementary benefits for AST.

***Index Terms*—** speech-to-text translation, transfer learning, pretraining, speech recognition, data augmentation.

1. INTRODUCTION

Low-resource automatic speech-to-text translation (AST) has recently gained traction as a way to bring NLP tools to under-represented languages. An end-to-end approach [1–7] is particularly appealing for source languages with no written form, or for endangered languages where translations into a high-resource language may be easier to collect than transcriptions [8]. However, building high-quality end-to-end AST with little parallel data is challenging, and has led researchers to explore how other sources of data could be used to help.

A number of methods have been investigated. Several of these use transcribed source language audio and/or translated source language text in a multitask learning scenario [4, 6, 9] or to pre-train parts of the model before fine-tuning on the end-to-end AST task [4]. Others assume, as we do here, that no additional source language resources are available, in which case transfer learning using data from language(s) other than the source language is a good option. In particular, several researchers have shown that low-resource AST can be improved by pretraining on an ASR task in some other language, then transferring the encoder parameters to initialize the AST model. For example, Bansal et al. [5] showed that pre-training on either English or French ASR improved their Spanish-English AST system (trained on 20 hours of parallel data) and Tian [10] got improvements on an 8-hour Swahili-English AST dataset using English ASR pretraining.

Overall these results show that pretraining helps, but leave open the question of what factors affect the degree of improvement. For example, does language relatedness play a role, or simply the amount of pretraining data? Bansal et al. showed bigger AST gains as the amount of English pretraining data increased from 20 to 300 hours, and also found a slightly larger improvement when pretraining on 20 hours of English versus 20 hours of French, but they pointed out that the Spanish data contains many English code-switched words, which could explain the latter result. In related work on multilingual pretraining for low-resource ASR, Adams et al. [11] showed that pre-training on more languages helps, but it is not clear whether the improvement is due to including more languages, or just more data.

To begin to tease apart these issues, we focus here on monolingual pretraining for low-resource AST, and investigate two questions. First, can we predict what sort of pretraining data is best for a particular AST task? Does it matter if the pretraining language is related to the AST source language (defined here as part of the same language family, since phonetic similarity is difficult to measure), or is the amount of pretraining data (or some other factor) more important? Second, can pretraining be effectively combined with other methods, such as data augmentation, in order to further improve AST results?

To answer these questions, we use the same AST architecture and Spanish-English parallel data as Bansal et al. [5], but pretrain the encoder using a number of different ASR datasets: the 150-hour AISHELL corpus of Chinese as well as seven GlobalPhone languages, each with about 20 hours of data. We find that pretraining on a larger amount of data from an unrelated language is much better than pretraining on a smaller amount of data from a related language. Moreover, even when controlling for the amount of data, the WER of the ASR model from pretraining seems to be a better predictor of final AST performance than does language relatedness. Indeed, we show that there is a very strong correlation between the WER of the pretraining model and BLEU score of the final AST model—i.e., the best pretraining strategy may simply be to use datasets and methods that will yield the lowest ASR WER during pretraining. However, we also found that AST results can be improved further by augmenting the AST data using standard speed perturbation techniques [12]. Our best results using non-English pretraining data improve the test set BLEU scores of an AST system trained on 20 hours of parallel data from 10.2 to 14.3, increasing to 15.8 with data augmentation.

Finally, we analyze the representations learned by the models and show that better performance seems to correlate with the extent to which phonetic information is encoded in a linearly separable way in the later RNN layers.

2. METHODOLOGY

For both ASR and AST tasks we use the same end-to-end system architecture shown in Figure 1: the encoder-decoder model from [5], which itself is adapted from [2], [4] and [3]. Details of the architecture and training parameters are described in Section 3.4.

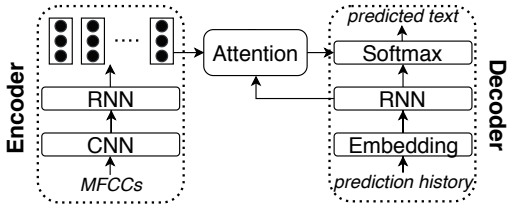

Fig. 1: Encoder-decoder architecture used for both ASR and AST.

After pretraining an ASR model, we transfer only its encoder parameters to the AST task. Previous experiments [5] showed that the encoder accounts for most of the benefits of transferring the parameters. Transferring also the decoder and attention mechanism does bring some improvements, but is only feasible when the ASR pretraining language is the same as the AST target language, which is not true in most of our experiments.

In addition to pretraining, we experimented with data augmentation. Specifically, we augmented the AST data using Kaldi's [13] 3-way speed perturbation, adding versions of the AST data where the audio is sped down and up by a factor of 0.9 and 1.1, respectively.[1]

To evaluate ASR performance we compute the word error rate (WER).[2] To evaluate AST performance we calculate the 4-gram BLEU score [14] on four reference translations.[3]

3. EXPERIMENTAL SETUP

3.1. Parallel data

For the AST models, we use Spanish-English parallel data from Fisher corpus [15], containing 160 hours of Spanish telephone speech translated into English text. To simulate low-resource settings, we randomly downsample the original corpus to 20 hours of training data. Each of the dev and test sets comprise 4.5 hours of speech.

3.2. Pretraining data

Since we focus on investigating factors that might affect the AST improvements over the baseline when pretraining, we have chosen ASR datasets for pretraining that contrast in the number of hours and/or in the language similarity with Spanish. Statistics for each dataset are in the left half of Table 1, with further details below.

To look at a range of languages with similar amounts of data, we used **GlobalPhone corpora from seven languages** [16], each with around 20 hours of speech: Mandarin Chinese (zh), Croatian (hr), Czech (cs), French (fr), Polish (pl), Portuguese (pt), and Swedish (sv). French and Portuguese, like the source language (Spanish), belong to the Romance family of languages, while the other languages are less related—especially Chinese, which is not an Indo-European language. GlobalPhone consists of read speech recorded using similar conditions across languages, and the transcriptions for Chinese are Romanized, with annotated word boundaries.

[1] In principle, we can augment the ASR pretraining data, the AST data, or both. However, we only augmented the AST data because in a preliminary experiment on AISHELL, we found that augmenting the ASR pretraining data did not improve its WER or the performance of the final AST system. Other researchers have reported ASR improvements using speed perturbation, and given the strong correlation we report below between ASR WER and AST BLEU, we would expect other data augmentation methods that do improve WER in pre-training to also improve AST.

[2] `https://github.com/belambert/asr-evaluation`

[3] `https://www.nltk.org/_modules/nltk/translate/bleu_score.html`

	DATA		RESULTS	
Dataset	**Hrs.**	**Spks.**	**ASR (WER)**	**AST (BLEU)**
ast-20h	20		—	10.3
zh-ai-small	20	81	38.7	12.4 (+2.1)
zh-ai-large	150	340	22.5	14.6 (+4.3)
zh-ai-hanzi	150	340	25.3	13.2 (+2.9)
hr-gp	12	72	71.5	10.7 (+0.4)
sv-gp	18	79	59.4	12.3 (+2.0)
pl-gp	19	79	59.6	10.8 (+0.5)
pt-gp	23	86	80.5	10.5 (+0.2)
fr-gp	25	84	31.1	12.5 (+2.2)
zh-gp	26	111	51.5	12.0 (+1.7)
cs-gp	27	82	53.7	11.1 (+0.8)
multilin6	124	482	44.2	13.3 (+3.0)

Table 1: Dataset statistics (left); dev set results from ASR pretraining and from the final AST system (right). AST results in all rows except the first are from pretraining using the dataset listed in that row, followed by fine-tuning using *ast-20h*. Numbers in brackets are the improvement over the baseline.

To explore the effects of using a large amount of pretraining data from an unrelated language, we used the **AISHELL-1 corpus of Mandarin Chinese** [17], which contains 150 hours of read speech. Transcriptions with annotated word boundaries are available in both Hanzi (Chinese characters) and Romanized versions, and we built models with each. To compare to the GlobalPhone data, we also created a 20-hour subset of the Romanized AISHELL (*zh-ai-small*) by randomly selecting utterances from a subset of the speakers (81, roughly the number present in most of the GlobalPhone datasets).

Finally, to reproduce one of the experiments from [5], we pretrained one model using 300 hours of **Switchboard English** [18]. This data is the most similar to the AST speech data in terms of style and channel (both are conversational telephone speech). However, as noted by [5], the Fisher Spanish speech contains many words that are actually in English (code-switching), so pretraining on English may provide an unfair advantage relative to other languages.

3.3. Preprocessing

We compute 13-dim MFCCs and cepstral mean and variance normalization along speakers using Kaldi [13] on our ASR and AST audio. To shorten the training time, we trimmed utterances from the AST data to 16 seconds (or 12 seconds for the 160h augmented dataset).

To account for unseen words in the test data, we model the ASR and AST text outputs via sub-word units using byte-pair encoding (BPE) [19]. We do this separately for each dataset as BPE works best as a language-specific tool (i.e. it depends on the frequency of different subword units, which varies with the language). We use 1k merge operations in all cases except Hanzi, where there are around 3000 symbols initially (vs around 60 in the other datasets). For Hanzi we ran experiments with both 1k and 15k merge operations. For Chinese Romanized transcriptions we removed tone diacritics.

3.4. Model architecture and training

Following the architecture and training procedure described in [5], input speech features are fed into a stack of two CNN layers. In each CNN layer we stride the input with a factor of 2 along time, apply

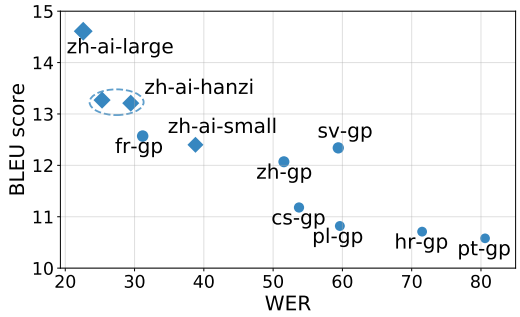

Fig. 2: WER of each ASR model vs BLEU score of the corresponding pre-trained AST model, computed in both cases on dev sets. Diamond markers are AISHELL data sets; circles are from GlobalPhone. The points in the circled group come from different runs on the same dataset but with different BPE or learning rate schedules. The Spearman rank correlation of these points is -0.97; the correlation is -0.92 when using test sets to compute both ASR and BLEU.

ReLU activation [20] followed by batch normalization [21]. The CNN output is fed into a three-layer bi-directional long short-term memory network (LSTM) [22], with 512 hidden layer dimensions. For decoding, we use the predicted token 20% of the time and the training token 80% of the time [23] as input to a 128-dimensional embedding layer followed by a three-layer LSTM, with 256 hidden layer dimensions, and combine this with the output from the attention mechanism [24] to predict the word at the current time step.

We use code and hyperparameter settings from [5][4]: the Adam optimizer [25] with an initial learning rate of 0.001 and decay it by a factor of 0.5 based on the dev set BLEU score. When training AST models, we regularize using dropout [26] with a ratio of 0.3 over the embedding and LSTM layers [27]; weight decay with a rate of 0.0001; and, after the first 20 epochs, 30% of the time we replace the predicted output word by a random word from the target vocabulary. At test time we use beam decoding with a beam size of 5 and length normalization [28] with a weight of 0.6.

4. RESULTS AND DISCUSSION

4.1. Baseline and ASR results

Our baseline 20-hour AST system obtains a BLEU score of 10.3 (Table 1, first row), 0.5 BLEU point lower than that reported by [5]. This discrepancy might be due to differences in subsampling from the 160-hour AST dataset to create the 20-hour subset, or from Kaldi parameters when computing the MFCCs.

WERs for our pre-trained models (Table 1) vary from 22.5 for the large AISHELL dataset with Romanized transcript to 80.5 for Portuguese GlobalPhone. These are considerably worse than state-of-the-art ASR systems (e.g., Kaldi recipes can achieve WER of 7.5 on AISHELL and 26.5 on Portuguese GlobalPhone), but we did not optimize our architecture or hyperparameters for the ASR task since our main goal is to analyze the relationship between pretraining and AST performance (and in order to use pretraining, we must use a seq2seq model with the architecture as for AST).

4.2. Pretraining the AST task on ASR models

AST results for our pre-trained models are given in Table 1. Pretraining improves AST performance in every case, with improvements ranging from 0.2 (*pt-gp*) to 4.3 (*zh-ai-large*). These results make it clear that language relatedness does not play a strong role in predicting AST improvements, since on the similar-sized GlobalPhone datasets, the two languages most related to Spanish (French and Portuguese) yield the highest and lowest improvements, respectively. Moreover, pretraining on the large Chinese dataset yields a bigger improvement than either of these—4.3 BLEU points. This is nearly as much as the 6 point improvement reported by [5] when pretraining on 100 hours of English data, which is especially surprising given not only that Chinese is very different from Spanish, but also that the Spanish data contains some English words.

This finding seems to suggest that data size is more important than language relatedness for predicting the effects of pretraining. However, there are big differences even amongst the languages with similar amounts of pretraining data. Analyzing our results further, we found a striking correlation between the WER of the initial ASR model and the BLEU score of the AST system pretrained using that model, as shown in Figure 2. Therefore, although pretraining data size clearly influences AST performance, this appears to be mainly due to its effect on WER of the ASR model. We therefore hypothesize that WER is a better direct predictor of AST performance than either data size or language relatedness.

4.3. Multilingual pretraining

Although our main focus is monolingual pretraining, we also looked briefly at multilingual pretraining, inspired by recent work on multilingual ASR [29, 30] and evidence that multilingual pretraining followed by fine-tuning on a distinct target language can improve ASR on the target language [11, 31, 32]. These experiments did not directly compare pretraining using a similar amount of monolingual data, but such a comparison was done by [33, 34] in their work on learning feature representations for a target language with no transcribed data. They found a benefit for multilingual vs monolingual pretraining given the same amount of data.

Following up on this work, we tried pretraining using 124 hours of multilingual data (all GlobalPhone languages except Chinese), roughly the amount of data in our large Chinese models. We combined all the data together and trained an ASR model using a common target BPE with 6k merge operations, then transferred only the encoder to the AST model. However, we did not see a benefit to the multilingual training (Table 1, final row); in fact the resulting AST model was slightly worse than the *zh-ai-large* model (BLEU of 13.3 vs 14.6). Other configurations of multilingual training might still outperform their monolingual counterparts, but we leave this investigation as future work.

4.4. Augmenting the parallel data

Table 2 (top) shows how data augmentation affects the results of the baseline 20h AST system, as well as three of the best-performing pretrained models from Table 1. For these experiments only, we changed the learning rates of the augmented-data systems so that all models took about the same amount of time to train (see Figure 3). Despite a more aggressive learning schedule, the performance of the augmented-data systems surpasses that of the baseline and pretrained models, even those trained on the largest ASR sets (150-hr Chinese and 300-hr English).

For comparison to other work, Table 2 (bottom) gives results for AST models trained on the full 160 hours of parallel data, including models with both pretraining and data augmentation. For the latter, we used the original learning schedule, but had to stop training early due

[4] https://github.com/0xSameer/ast.

		dev set		test set	
	Pretrain	**No aug.**	**With aug.**	**No aug.**	**With aug.**
20h	–	10.3	13.0 (+2.7)	10.2	13.3 (+3.1)
	fr-gp	12.5	13.7 (+1.2)	12.6	14.3 (+1.7)
	zh-ai-lrg	14.6	15.5 (+0.9)	14.3	15.8 (+1.5)
	en-300h	19.5	20.1 (+0.6)	20.1	20.2 (+0.1)
160h	–	34.1	36.3 (+2.2)	34.6	37.3 (+2.7)
	en-300h	36.3	37.9 (+1.6)	36.4	37.8 (+1.4)

Table 2: BLEU scores on dev and test sets for models trained with and without data augmentation. We used either 20h of AST training data (top block) or 160h (bottom block), with various pretraining.

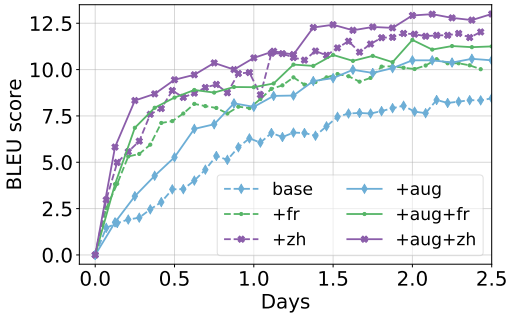

Fig. 3: The AST performance over time (without beam-search) of baseline, pretrained, and pretrained+augmented models.

to time constraints (after 15 days, compared to 8 days for complete training of the non-augmented 160h models). We find that both pretraining and augmentation still help, providing a combined gain of 3.8 (3.2) BLEU points over the baseline on the dev (test) set.

5. ANALYZING THE MODELS' REPRESENTATIONS

Finally, we hope to gain some understanding into why pretraining on ASR helps with AST, and specifically how the neural network representations change during pretraining and fine-tuning. We follow [35] and [10], who built diagnostic classifiers [36] to examine the representation of phonetic information in end-to-end ASR and AST systems, respectively. Unlike [10,35], who used non-linear classifiers, we use a *linear* classifier to predict phone labels from the internal representations of the trained ASR or AST model.

Using a linear classifier allows us to make more precise claims: if the classifier performs better using the representation from a particular layer, we can say that layer represents the phonetic information in a more linearly separable way. Using a nonlinear classifier raises questions about how to choose the complexity of the classifier itself, and therefore makes any results difficult to interpret.

We hypothesized that pretraining allows the models to abstract away from nonlinguistic acoustic differences, and to better represent phonetic information: crucially, both in the trained language and in other languages. To test this hypothesis, we used two phone-labelled datasets distinct from all our ASR and AST datasets: the English TIMIT corpus (a language different to all of our trained models, with hand-labeled phones) and the Spanish GlobalPhone corpus (the same language as our AST source language, with phonetic forced-alignments produced using Kaldi). We randomly sampled utterances from these and passed them through the trained encoders, giving us a total of about 600k encoded frames. We used 400k of these to train

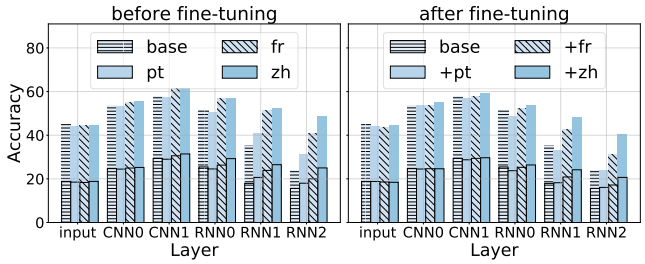

Fig. 4: Phonetic classification accuracy at different layers of our ASR (left) and AST (right) models. Different color bars indicate representations extracted from models (pre)trained on different datasets (*pt-gp*, *fr-gp*, or *zh-ai-large*). Results from the baseline AST model (without pretraining) are shown in both panels for comparison. The bars with black edges are results on TIMIT (majority baseline: 12.9%); the taller bars are for Spanish GlobalPhone (majority baseline: 15.2%).

logistic regression models to predict the phone labels, and tested on the remaining 200k frames.

Separate logistic regression models were trained on the representations from each layer of the encoder. Since convolutional layers have a stride of 2, the number of frames decreases at each convolutional layer. To label the frames after a convolutional layer we eliminated every other label (and corresponding frame) from the original label sequence. For example, given label sequence S_1 = *aaaaaaann* at input layer, we get sequence S_2 = *aaaan* at the first convolutional layer and sequence S_3 = *aan* at the second convolutional layer and at the following recurrent layers.

Results for the two classification data sets (Figure 4) show very similar patterns. In both the ASR and the AST models, the pretraining data seems to make little difference to phonetic encoding at the early layers, and classification accuracy peaks at the second CNN layer. However, the RNN layers show a clear trend where phone classification accuracy drops off more slowly for models with better ASR/AST performance (i.e., zh > fr > pt). That is, the later RNN layers more transparently encode language-universal phonetic information.

Phone classification accuracy in the RNN layers drops for both English and Spanish after fine-tuning on the AST data. This is slightly surprising for Spanish, since the fine-tuning data (unlike the pretraining data) is actually Spanish speech. However, we hypothesize that for AST, higher layers of the encoder may be recruited more to encode semantic information needed for the translation task, and therefore lose some of the linear separability in the phonetic information. Nevertheless, we still see the same pattern where better end-to-end models have higher classification accuracy in the later layers.

6. CONCLUSIONS

This paper explored what factors help pretraining for low-resource AST. We performed careful comparisons to tease apart the effects of language relatedness and data size, ultimately finding that rather than either of these, the WER of the pre-trained ASR model is likely the best direct predictor of AST performance. Given equivalent amounts of data, we did not find multilingual pretraining to help more than monolingual pretraining, but we did find an added benefit from using speed perturbation to augment the AST data. Finally, analysis of the pretrained models suggests that those models with better WER are transparently encoding more language-universal phonetic information in the later RNN layers, and this appears to help with AST.

7. ACKNOWLEDGEMENTS

The authors wish to thank Yusheng Tian for her work on her Master's thesis at the University of Edinburgh which inspired the analysis of the change in neural network representations during pretraining and fine-tuning. Also, thanks to Dr. Yevgen Matusevych and to Ramon Sanabria for useful discussions, proof-reading and providing feedback on the paper. This work was supported in part by a James S. McDonnell Foundation Scholar Award (220020374).

8. REFERENCES

[1] A. Bérard, O. Pietquin, C. Servan, and L. Besacier, "Listen and translate: A proof of concept for end-to-end speech-to-text translation," in *NIPS Workshop on end-to-end learning for speech and audio processing*, 2016.

[2] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly transcribe foreign speech," in *Proc. Interspeech*, 2017.

[3] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, "Low-resource speech-to-text translation," in *Proc. Interspeech*, 2018.

[4] A. Bérard, L. Besacier, A. C. Kocabiyikoglu, and O. Pietquin, "End-to-end automatic speech translation of audiobooks," in *Proc. ICASSP*, 2018.

[5] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, "Pre-training on high-resource speech recognition improves low-resource speech-to-text translation," in *Proc. NAACL*, 2019.

[6] M. Sperber, G. Neubig, J. Niehues, and A. Waibel, "Attention-passing models for robust and data-efficient end-to-end speech translation," in *Trans. ACL*, 2019.

[7] E. Salesky, M. Sperber, and A. Waibel, "Fluent translations from disfluent speech in end-to-end speech translation," in *Proc. NAACL*, 2019.

[8] P. Godard, G. Adda, M. Adda-Decker *et al.*, "A very low resource language speech corpus for computational language documentation experiments," in *Proc. LREC*, 2018.

[9] A. Anastasopoulos and D. Chiang, "Tied multitask learning for neural speech translation," in *Proc. NAACL HLT*, 2018.

[10] Y. Tian, "How does pre-training improve low-resource speech-to-text translation? — a case study on a Swahili-English dataset," Master's thesis, University of Edinburgh, 2019.

[11] O. Adams, M. Wiesner, S. Watanabe, and D. Yarowsky, "Massively multilingual adversarial speech recognition," in *Proc. NAACL*, 2019.

[12] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proc. Interspeech*, 2015.

[13] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *Proc. ASRU*, 2011.

[14] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proc. ACL*, 2002.

[15] M. Post, G. Kumar, A. Lopez, D. Karakos, C. Callison-Burch, and S. Khudanpur, "Fisher and CALLHOME Spanish-English Speech Translation," 2014, https://catalog.ldc.upenn.edu/LDC2014T23.

[16] T. Schultz, "Globalphone: a multilingual speech and text database developed at Karlsruhe University," in *ICSLP*, 2002.

[17] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell-1: An open-source Mandarin speech corpus and a speech recognition baseline," in *O-COCOSDA*, 2017.

[18] J. Godfrey and E. Holliman, "Switchboard-1 Release 2 (LDC97S62)," 1993, https://catalog.ldc.upenn.edu/LDC97S62.

[19] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. ACL*, 2016.

[20] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010.

[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, 1997.

[23] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, 1989.

[24] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. EMNLP*, 2015.

[25] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, 2014.

[27] Y. Gal, "A theoretically grounded application of dropout in recurrent neural networks," in *Proc. NIPS*, 2016.

[28] Y. Wu, M. Schuster, Z. Chen *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016.

[29] S. Toshniwal, T. N. Sainath, R. J. Weiss, B. Li, P. Moreno, E. Weinstein, and K. Rao, "Multilingual Speech Recognition with A Single End-To-End Model," in *Proc. ICASSP*, 2018.

[30] S. Zhou, S. Xu, and B. Xu, "Multilingual end-to-end speech recognition with a single transformer on low-resource languages," in *arXiv*, 2018, preprint arXiv:1806.05059.

[31] J. Cho, M. K. Baskar, R. Li, M. Wiesner, S. Mallidi, N. Yalta, M. Karafiát, S. Watanabe, and T. Hori, "Multilingual sequence-to-sequence speech recognition: Architecture, transfer learning, and language modeling," in *Proc. SLT*, 2018.

[32] S. Dalmia, R. Sanabria, F. Metze, and A. W. Black, "Sequence-based multi-lingual low resource speech recognition," in *ICASSP*, 2018.

[33] E. Hermann, H. Kamper, and S. Goldwater, "Multilingual and unsupervised subword modeling for zero-resource languages," 2018, arXiv preprint arXiv:1811.04791.

[34] E. Hermann and S. Goldwater, "Multilingual bottleneck features for subword modeling in zero-resource languages," in *Proc. Interspeech*, 2018.

[35] Y. Belinkov and J. R. Glass, "Analyzing hidden representations in end-to-end automatic speech recognition systems," in *NIPS*, 2017.

[36] D. Hupkes, S. Veldhoen, and W. H. Zuidema, "Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure," *CoRR*, vol. abs/1711.10203, 2017.