SEMI-SUPERVISED TRAINING FOR END-TO-END MODELS VIA WEAK DISTILLATION

Bo Li, Tara N. Sainath, Ruoming Pang, Zelin Wu

Google LLC, USA

{boboli,tsainath,rpang,zelinwu}@google.com

ABSTRACT

End-to-end (E2E) models are a promising research direction in speech recognition, as the single all-neural E2E system offers a much simpler and more compact solution compared to a conventional model, which has a separate acoustic (AM), pronunciation (PM) and language model (LM). However, it has been noted that E2E models perform poorly on tail words and proper nouns, likely because the end-to-end optimization requires joint audio-text pairs, and does not take advantage of additional lexicons and large amounts of text-only data used to train the LMs in conventional models. There has been numerous efforts in training an RNN-LM on text-only data and fusing it into the end-to-end model. In this work, we contrast this approach to training the E2E model with audio-text pairs generated from unsupervised speech data. To target the proper noun issue specifically, we adopt a Part-of-Speech (POS) tagger to filter the unsupervised data to use only those with proper nouns. We show that training with filtered unsupervised-data provides up to a 13% relative reduction in word-error-rate (WER), and when used in conjunction with a cold-fusion RNN-LM, up to a 17% relative improvement.

Index Terms— semi-supervised training, sequence to sequence

1. INTRODUCTION

End-to-end models provide a simple yet effective way for automatic speech recognition (ASR). Traditionally, an ASR system consists of an AM, PM and LM, while end-to-end models fold these three components into a single neural network that is jointly optimized. Listen, Attend and Spell (LAS) [1] is one such end-to-end model, that has shown promising results compared to a strong conventional ASR system [2]. However, while the LM in a conventional system can be independently trained on a large amount of text-only data, training an LAS model requires audio-text pairs, which are much more expensive to collect and much smaller in scale. Thus, LAS performs poorly compared to conventional models in recognizing rare words or phrases, such as song names, contacts, etc [2–4].

There have been many efforts to improve end-to-end model performance using unpaired text data. One popular research direction looks to integrate an external LM, trained on the text-only data, with an end-to-end model. For example, [5] initializes the end-to-end model with a pre-trained LM from text-only data and then jointly optimizes the end-to-end model and the LM through mutli-task training. In addition, interpolating independently trained end-to-end and LM models via shallow fusion has been explored, both for neural machine translation [6] and ASR [4, 7]. Furthermore, integrating an RNN-LM trained on text-only data jointly into the end-to-end decoder has been explored, via both cold and deep fusion [3, 4, 8]. Overall leveraging text-only data has shown between 3% to 7% relative improvement in WER for ASR [3].

[9] explored *backtranslation* to improve machine translation with monolingual training data. The authors found that this improved the BLEU score by 2.8∼3.7. This idea has also been applied to speech recognition [10], where synthetic audio generated from unpaired text data was used to expand the audio-text pairs for training end-to-end models. While the use of TTS data gives dramatic improvements on TTS test sets, degradation has been observed on real test sets.

In addition, conventional ASR systems make use of unlabelled audio data to improve performance. Confidence scores from an existing ASR system is commonly used to select unsupervised data for training with more data [11–15]. For example, [16] selects unsupervised speech data using a combination of the recognition word confidence score and the MLP posteriogram-based phoneme occurrence confidence for low resource languages. For the YouTube speech caption task [17], an "island of confidence" approach was developed to largely increase the amount of training data to improve WER performance. To our knowledge, there has not been any existing work using unsupervised speech data for end-to-end speech recognition.

The goal of this work is to investigate using unsupervised speech data for improving the end-to-end model accuracy. We place a specific emphasis on improving performance in rare words and proper nouns, such as contacts, song names and app names, which is very important for contextual biasing [18], an important component of any production-level ASR system. In this work, we propose to distill information captured by a conventional ASR system's output hypotheses to our LAS model. Specifically, we use the top hypothesis generated by a full-stack conventional production ASR system, which includes contextual-biasing, as the transcript truth to train our LAS model on the unsupervised speech data. This is referred to as *weak distillation*, a simplified sequence-level knowledge distillation [19, 20]. We also experiment different ways of using these unsupervised data. Most importantly, to ensure the training is targeted for proper noun cases, we adopt a POS tagger to identify utterances that contain proper nouns and use only those data.

We report results across 5 test sets, which include a generic Voice Search test set and 4 different test sets targeting at rare words. We find that the proposed weak distillation using unsupervised data with proper nouns is the most effective method. It reduces the supervised LAS model's WER by 4%-12% relatively, while with cold fusion we obtain 2%-6% relative WER reduction. When combining these two approaches to leverage both speech-only and text-only data, we achieve a relative 7%-13% WER reduction.

2. SEMI-SUPERVISED TRAINING

Training an all-neural E2E system such as LAS requires audio-text pairs to learn jointly an AM, PM and LM. While this joint training allows for potentially better optimization it also restricts to the use of paired audio-text data, resulting in E2E model performing poorly

on rare words and proper nouns. In this work, we explore techniques of utilizing untranscribed speech data to improve the performance of E2E models on these tail words.

2.1. Weak Distillation

Past work has shown that while LAS models outperform conventional ASR systems on generic test sets, performance degrades on tail words and proper nouns. A conventional ASR system includes a large hand-designed lexicon, a 1st and 2nd pass LM trained on a trillion-word text corpora, and a contextual biasing mechanism that boosts those tail words into top hypotheses. All of these contribute to improved performance on tail words.

In this work, we leverage the strength of a conventional model to fix the weakness of the E2E model. Specifically, we look to distill the information captured in the conventional ASR's output hypotheses to the end-to-end model. We use Google's Voice Search production model [21] as the teacher and decode millions of unsupervised voice search queries. The recognition hypotheses of the teacher model are used as the training targets for the student E2E model. This can be considered as simplified sequence distillation [19, 20] and is referred to as weak distillation in this work. This approach requires no extra model parameters and is more preferable when the size of model is critical on cases such as on-device applications.

2.2. LM On Text-Only Data

Recognition errors of the teacher model may cause mismatches between the audio and the target text transcript, thus causing issues for E2E model training. This problem can be alleviated by training with the text-only portion of the unsupervised data. A very common approach to utilize a large amount of text-only data is to train an RNN-LM and then perform fusion with an E2E model [4]. There have been numerous fusion approaches proposed in the literature [3]. Cold fusion [8] has been shown to be an effective strategy for Voice Search. Unlike weak distillation, this approach brings in an extra LM which increases the total number of model parameters. In this study we compare cold fusion with aforementioned weak distillation.

2.3. Synthesizing Audio from Transcripts

Another way to address possible mismatches in unsupervised audio-text pairs is to generate synthetic audio from the text hypotheses using a single-speaker TTS engine with parallel WaveNet vocoder [22]. This is similar to the "backtranslation" approach used in machine translation [9]. One potential problem with this approach is the acoustic differences between real speech and synthetic audio, particularly the limited speaker characteristics and clean speaking style. To address this concern, we compare backpropping the encoder and decoder of the LAS model, versus just the decoder. The intuition is that the encoder represents an AM and should be trained on realistic conditions. However, the decoder is akin to the LM and can be trained with less realistic conditions. Therefore, we explore if backpropping the decoder only could perhaps address the unrealistic audio concerns with TTS data.

2.4. Data Filtering

We have access to more than a billion unsupervised utterances. This comes with an advantage that with more unsupervised data, our model sees a much larger vocabulary during training. However, more data comes at a cost of longer model training time.

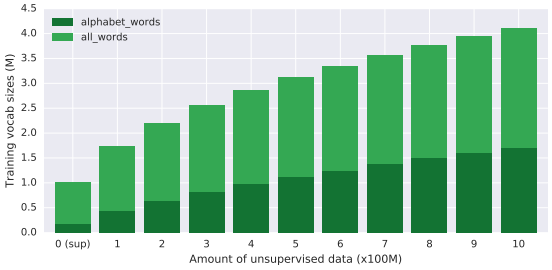

Fig. 1: The amount of unique words added to training by using the unsupervised data.

Table 1: Details of test sets used for evaluation.

Test Set	Size	Source
Voice Search (VS)	15K	Real
Apps	16K	TTS
Songs	15K	TTS
Contacts-TTS	15K	TTS
Contacts-Real	5K	Real

Therefore, we explore selecting a subset of data to train the LAS model. Specifically, because our model does poorly on proper nouns, we explore if filtering the unsupervised data to include these utterances allows us to obtain quality improvements with unsupervised data, with smaller training time compared to using all of the data. The decision whether an utterance contains proper nouns is made by running a Part-of-Speech (POS) tagger [23] on the text hypothesis.

3. EXPERIMENTS

3.1. Data Sets

Our experiments are conducted on a human transcribed supervised training set and an unlabelled unsupervised training set. The supervised training set consisting of 35 million English utterances ($\sim$ 27,500 hours). These utterances are anonymized and hand-transcribed, and are representative of Google's voice search and dictation traffic. These utterances are further artificially corrupted using a room simulator [24], adding varying degrees of noise and reverberation such that the overall SNR is between 0dB and 30dB, with an average SNR of 12dB. The noise sources are from YouTube and daily life noisy environmental recordings. For each utterance, we generated 25 different noisy versions for training.

The unsupervised training set consists of 1 billion English utterances. These utterances are randomly collected from Google's voice search traffic without human transcriptions. We constrain the time frame during which those utterances are logged to be different from the supervised training set. This ensures that the supervised and unsupervised training sets do not overlap with each other. We use the recognition hypotheses from Google's voice search production system [21] as the transcripts for the unsupervised data. We lowercase all the transcripts and treat each non-empty character sequence as "words", which may contain invalid words. The number of "words" vs. the amount of unsupervised data is plotted in Fig. 1. As the amount of unsupervised data increases, so does the number of unique words. With more data, there are large number of numeric words added into the training.

For evaluation, we test our models on 5 test sets, which are detailed as follows. A summary of the test sets are given in Table 1. The *Voice Search* test set contains queries from Google's voice search traffic that are collected from a time frame different

from both the supervised and unsupervised training sets. This set has a matched data distribution compared to the training data and is used to ensure the performance on the matched domain does not degrade when training with unsupervised data.

The *Apps*, *Songs* and *Contacts-TTS* test sets are artificially created using the aforementioned TTS system. The synthesized samples are also corrupted with noise similarly to the way we corrupt the training data. The *Apps* test set contains requests to talk with one of many chat-bots such as "*talk to trivia game*". The *Songs* test set contains requests to play music such as "*play rihanna music*" and the *Contacts-TTS* test set contains call requests such as "*call demetri mobile*". The *Contacts-Real* test set is similar to *Contacts-TTS* but from Google's voice search traffic.

3.2. Modeling

We use 80-dimensional log-Mel features, computed with a 25ms window and shifted every 10ms. Similar to [21, 25], at each current frame, these features are stacked with 3 consecutive frames to the left and then down-sampled to a 30ms frame rate.

The experiments are conducted with the LAS [1] model, following the set-up outlined in [2]. Specifically, the encoder network consists of 10 unidirectional long short-term memory (LSTM) [26] layers, with each layer having $2,048$ hidden units followed by a 384 dimensional projection layer. After the 2nd layer of the encoder network, we concatenate each frame with its adjacent left neighboring frame and stride by 2 before passing them to the following layers. This stacking layer further reduces the frame rate to 60ms. Layer normalization [27] is adopted for encoder layers to stabilize the training. Additive attention [28] with 4 attention heads are used. The decoder network consists of 4 unidirectional LSTM layers with $2,048$ hidden units and output projection size of 384. The LAS model outputs a vocabulary of 16K word pieces. The models are trained with label smoothing and cross-entropy loss using TensorFlow [29]. We use 8×8 Tensor Processing Units (TPU) slices with global batch size of 4,096 and train the models for around 200K steps.

4. RESULTS

4.1. Baselines

First the performance of the LAS model trained with only the supervised training data (denoted as `B0`) is presented in Table 2. We also present the performance of the full stack conventional model [30] we used as the teacher model for weak distillation. The teacher model is a conventional context-dependent phoneme based low frame rate acoustic model, a 4M word pronunciation lexicon and a 5-gram language model. This model is referred to as `B1`. The teacher model is trained using the same supervised training data. The table shows that the LAS model outperforms the conventional model on most of the test sets. However, the conventional model uses context information in practice to prune the search space [18], which helps reduce WER on sets with many proper nouns (songs, contacts, apps). The performance of the teacher model with context biasing is denoted as `B2` in Table 2.

Table 2: WER performance (%) of baseline experiments.

Exp	VS	Apps	Songs	Contacts	
				TTS	Real
B0	5.4	9.2	13.5	24.8	15.0
B1	6.8	9.0	13.1	26.0	16.8
B2	-	-	2.2	3.7	6.3

4.2. Weak Distillation

To distill the knowledge encoded in the recognized hypotheses, we start with training `B0` on the 1 billion unsupervised data. We use the hypotheses generated by `B2` as the reference transcripts, regardless of the errors in those transcripts. Training on 1 billion (1B) unsupervised data for 450K steps (`E0`), we obtain good improvements on all the TTS sets but see degradation for the *Voice Search* and *Contacts-Real*. The wins on TTS sets mainly come from the more word variations brought by the data, but the loss is most likely due to the errors in decoded hypotheses. To reduce the degradation on *VS* and *Contacts-Real*, we further fine-tune `E0` with the supervised data for 150K steps (`E1`). It improves over `B0` on all the test sets.

Training with 1B data takes a long time. To understand whether this amount of data is needed, we randomly down-sample the unsupervised data to 500 million (500M) and 100 million (100M) respectively. We train on the unsupervised data alone first (`E2` and `E4`) and then fine-tune them on the supervised data (`E3` and `E5`). We are able to get gains with both 100M and 500M unsupervised data across test sets, but using 1B data offers slightly better performance.

Table 3: WER performance (%) of two-stage training with unsupervised data.

Exp	uns Data	VS	Apps	Songs	Contacts	
					TTS	Real
B0	0	5.4	9.2	13.5	24.8	15.0
E0	1B	6.7	9.2	12.9	**23.3**	18.5
E1		**5.0**	8.9	12.9	23.9	**14.5**
E2	500M	6.8	9.5	13.3	23.6	19.4
E3		5.2	8.8	**12.3**	24.0	15.1
E4	100M	6.7	9.6	13.6	24.6	16.9
E5		5.2	**8.7**	12.9	24.1	14.7

4.3. Mixed training

Experiments in Table 3 showed that after training the LAS model with unsupervised data, we needed to fine-tune the model with supervised data again. To simplify the training procedure, we experiment with mixing the supervised and unsupervised data together during training. Specifically, whenever creating a batch of utterances for training, we randomly select from the two training sets with a fixed ratio. For example, with the mixing ratio of 8:2, a training batch comes from the supervised data 80% of the time and from unsupervised data 20% of the time. From the results in Table 4, mixing the supervised and unsupervised data is an effective way of utilizing the unsupervised data. Among the three different ratios, 8:2 gives the best performance across board with marginal differences. When comparing `E8` to `E1` we achieve much lower WERs on test sets with more proper nouns (*Apps, Songs, Contacts*) although the gain on *Voice Search* is smaller compared to `E1`.

4.4. Leveraging Text-Only Data

In this section, we compare different approaches of incorporating the unsupervised data. For all experiments, we explore performance with a randomly sampled 100M subset of the unsupervised data, for fast experiment turn-around. `E9` is trained exactly the same way as `E8` but with less unsupervised data. The results in Table 5 show that with less unsupervised data we get slightly better performance on the generic *Voice Search* test set but higher WERs on test sets with more tail words. Next, we explore performance by synthesizing audio from the unsupervised transcripts, where we use the aforementioned

Table 4: WER performance (%) of mixed training with unsupervised data. "Ratio" corresponds to the percentage of using supervised vs. unsupervised data.

Exp	Ratio	VS	Apps	Songs	Contacts	
					TTS	Real
B0	-	5.4	9.2	13.5	24.8	15.0
E1	-	**5.0**	8.9	12.9	23.9	14.5
E6	6:4	5.4	8.0	11.5	22.9	**13.7**
E7	7:3	5.3	**7.8**	**11.3**	22.9	**13.7**
E8	8:2	5.3	**7.8**	**11.3**	**22.8**	**13.7**

Table 5: WER performance (%) of using audio-only data vs. text-only data.

Exp	Info	VS	Apps	Songs	Contacts	
					TTS	Real
B0	-	5.4	9.2	13.5	24.8	15.0
E9	8:2(100M)	5.2	8.2	11.9	23.6	13.8
E10	TTS(enc+dec)	5.2	**3.1**	**5.2**	**14.2**	**14.5**
E11	TTS(dec)	5.3	3.3	**5.2**	**14.2**	14.7
E12	LM fusion	**5.1**	9.0	12.7	24.1	14.7

TTS system that is used to create the rare word test sets. We replace the unsupervised data used in E9 with this TTS training set and the results are presented in Table 5 as E10. It achieves a large WER reduction for all the TTS test sets but degrades the performance on Contacts-Real. This huge error reduction on TTS sets mainly comes from the matched acoustics between the added unsupervised data and the test sets. To avoid the potential mismatched audio conditions between real and synthetic data, we disable the update of the encoder network parameters and only update the decoder network of the LAS model during training. The results (E11) are similar to E10 with slightly degradation on Apps. Despite the large error reductions on TTS sets, we believe that the degradation on more realistic test sets compared to E9 tells the real story. We hence prefer E9 over E10 and E11.

Another way of utilizing the unsupervised data is to integrate an LM into the LAS system. Specifically, we train an RNN-LM on the supervised and 100M unsupervised data transcripts, and then integrate it into the LAS model training using cold fusion [3, 8]. The result (E12) shows 2%-6% relative WER reduction over the supervised baseline (B0), but the gain is much smaller compared to E9.

4.5. Filtering

In this section, we explore how to better utilize the unsupervised data. First, instead of random selection (E9) of 100M unsupervised utterances, we filter the unsupervised data to use only those with proper nouns (E13 in Table 6) for training, as that allows us to select utterances where the LAS model does poorly. The selection is done with a proper noun tagger [23,31,32]. We mix the 100M unsupervised data focusing on proper nouns with the supervised training data at the same 8:2 ratio for training. With the same amount of data, training with the proper noun filtered speech gives us 6%-13% relative WER reduction compared to the 4%-12% relative reduction using random selection. Finally, we extend the filtering idea to the entire 1B unsupervised training data, which leaves us with around 500M utterances with proper nouns. With more data (E14), we see slightly gains on TTS sets but slightly degradation on VS. We then combine the weak distillation with cold fusion (E15), which is much better than using all the 1B data and it reduces the WER of the baseline system on all the four test sets by 6%-17% relatively.

Table 6: WER performance (%) of using proper noun filtered unsupervised data.

Exp	Info	VS	Apps	Songs	Contacts	
					TTS	Real
B0	-	5.4	9.2	13.5	24.8	15.0
E8	1B random	5.3	7.8	11.3	22.8	13.7
E9	100M random	5.2	8.2	11.9	23.6	13.8
E13	100M filtered	5.1	8.0	12.0	22.8	13.6
E14	500M filtered	5.2	8.0	11.9	22.7	-
E15	E14 + fusion	**5.0**	**7.7**	**11.2**	**21.9**	**13.2**

Table 7: Comparisons of OOV rates (%), In-vocab (IV) error rates (%), OOV error rates (%) and proper noun error rates (%) between the baseline model ("B0") and the best system ("E15").

Analysis	Exp	VS	Apps	Songs	Contacts	
					TTS	Real
OOV Rates	B0	2.3	0.6	0.9	2.5	1.9
	E15	2.1	0.1	0.6	0.4	0.7
IV Errors	B0	19.0	23.0	23.7	26.3	32.6
	E15	17.3	19.4	19.6	23.8	28.6
OOV Errors	B0	81.6	88.7	88.6	90.2	88.1
	E15	78.2	73.1	74.8	72.0	79.3
Proper Noun Errors	B0	30.2	38.7	30.8	68.1	60.8
	E15	27.7	31.5	26.2	62.8	53.6

4.6. Analysis

To understand the improvements brought by the unsupervised data, we compare the two systems B0 and E15 in this section. B0 uses only the supervised training data, while E15 uses additional unsupervised training data. First, the use of unsupervised data reduces the out-of-vocabulary ("OOV") rates across all the test sets (row "OOV Rates" in Table 7). Computing the errors on both in-vocab ("IV") words and OOV words (row "IV Errors" and row "OOV Errors" in Table 7 respectively), the use of unsupervised data helps reduce errors for both cases. For OOV words, there are still many cases not fixed by E15 yet. Furthermore, proper nouns are the main reason for OOV words, we compute the error rates on proper nouns only. From row "Proper Noun Errors" in Table 7, similarly, E15 does better than B0 but there are still room for improvement, which maybe addressed by using more unsupervised data.

5. CONCLUSIONS

In this paper, we investigate the use of unsupervised speech data to improve the performance of the LAS model on long tail words. We use a conventional ASR system with contextual biasing as the teacher model to generate text hypotheses as transcript truth for a large amount of unsupervised data. We then mix these machine-labeled data with human-labeled data to train an end-to-end LAS model. To focus on LAS model's weakness on rare words, we apply proper-noun-based filtering for the unsupervised data. With the filtered data, experimental results have shown that up to 17% relative WER reduction could be achieved by introducing unsupervised data. In future, we plan to investigate further filtering techniques to improve training data efficiency and increase coverage on rare words and proper nouns.

6. REFERENCES

[1] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *CoRR*, vol. abs/1508.01211, 2015.

[2] Chung-Cheng Chiu, Tara N Sainath, et al., "State-of-the-art speech recognition with sequence-to-sequence models," *arXiv preprint arXiv:1712.01769*, 2017.

[3] Shubham Toshniwal, Anjuli Kannan, Chung-Cheng Chiu, Yonghui Wu, Tara N Sainath, and Karen Livescu, "A comparison of techniques for language model integration in encoder-decoder speech recognition," *arXiv preprint arXiv:1807.10857*, 2018.

[4] Anjuli Kannan, Yonghui Wu, Patrick Nguyen, Tara N Sainath, Zhifeng Chen, and Rohit Prabhavalkar, "An analysis of incorporating an external language model into a sequence-to-sequence model," *arXiv preprint arXiv:1712.01996*, 2017.

[5] Prajit Ramachandran, Peter J Liu, and Quoc V Le, "Unsupervised pretraining for sequence to sequence learning," *arXiv preprint arXiv:1611.02683*, 2016.

[6] Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "On using monolingual corpora in neural machine translation," *arXiv preprint arXiv:1503.03535*, 2015.

[7] Jan Chorowski and Navdeep Jaitly, "Towards better decoding and language model integration in sequence to sequence models," *arXiv preprint arXiv:1612.02695*, 2016.

[8] Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates, "Cold fusion: Training seq2seq models together with language models," *arXiv preprint arXiv:1708.06426*, 2017.

[9] Rico Sennrich, Barry Haddow, and Alexandra Birch, "Improving neural machine translation models with monolingual data," *arXiv preprint arXiv:1511.06709*, 2015.

[10] Yanzhang He, Tara N. Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, Deepti Bhatia, Yuan Shangguan, Qiao Liang, Bo Li, Golan Pundak, Khe Chai Sim, Tom Bagby, Shuo-yiin Chang, Kanishka Rao, and Alexander Gruenstein, "Streaming end-to-end speech recognition on mobile devices," in *submitted to ICASSP*. IEEE, 2019.

[11] Lori Lamel, Jean-Luc Gauvain, and Gilles Adda, "Lightly supervised and unsupervised acoustic model training," *Computer Speech & Language*, vol. 16, no. 1, pp. 115–129, 2002.

[12] Jeff Ma and Richard Schwartz, "Unsupervised versus supervised training of acoustic models," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[13] Kai Yu, Mark Gales, Lan Wang, and Philip C Woodland, "Unsupervised training and directed manual transcription for lvcsr," *Speech Communication*, vol. 52, no. 7-8, pp. 652–663, 2010.

[14] Yan Huang, Dong Yu, Yifan Gong, and Chaojun Liu, "Semi-supervised gmm and dnn acoustic model training with multi-system combination and confidence re-calibration.," in *Interspeech*, 2013, pp. 2360–2364.

[15] Olga Kapralova, John Alex, Eugene Weinstein, Pedro J Moreno, and Olivier Siohan, "A big data approach to acoustic model training corpus selection," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[16] Samuel Thomas, Michael L Seltzer, Kenneth Church, and Hynek Hermansky, "Deep neural network features and semi-supervised training for low resource speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6704–6708.

[17] Hank Liao, Erik McDermott, and Andrew Senior, "Large scale deep neural network acoustic modeling with semi-supervised training data for youtube video transcription," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 368–373.

[18] Petar Aleksic, Mohammadreza Ghodsi, Assaf Michaely, Cyril Allauzen, Keith Hall, Brian Roark, David Rybach, and Pedro Moreno, "Bringing contextual information to google speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[19] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[20] Yoon Kim and Alexander M Rush, "Sequence-level knowledge distillation," *arXiv preprint arXiv:1606.07947*, 2016.

[21] G. Pundak and T. N. Sainath, "Lower Frame Rate Neural Network Acoustic Models," in *Proc. Interspeech*, 2016.

[22] Aaron van den Oord, Yazhe Li, et al., "Parallel wavenet: Fast high-fidelity speech synthesis," *arXiv preprint arXiv:1711.10433*, 2017.

[23] Google, "Cloud natural language," `https://cloud.google.com/natural-language/`, 2018, [Online].

[24] C. Kim, A. Misra, K. Chin, T. Hughes, A. Narayanan, T. N. Sainath, and M. Bacchiani, "Generated of Large-scale Simulated Utterances in Virtual Rooms to Train Deep-Neural Networks for Far-field Speech Recognition in Google Home," in *Proc. Interspeech*, 2017.

[25] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition," in *Proc. Interspeech*, 2015.

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

[27] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[28] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[29] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," Available online: http://download.tensorflow.org/paper/whitepaper2015.pdf, 2015.

[30] Golan Pundak and Tara N Sainath, "Lower frame rate neural network acoustic models.," in *Interspeech*, 2016, pp. 22–26.

[31] Z. Huang and W. Xu and K. Yu, "Bidirectional LSTM-CRF Models for Sequence Tagging," *CoRR*, vol. abs/1508.01991, 2015.

[32] D. Andor, C. Alberti, D. Weiss, A. Severyn, A. Presta, K. Ganchev, S. Petrov, and M. Collins, "Globally Normalized Transition-Based Neural Networks," *CoRR*, vol. abs/1603.06042, 2016.