SPOKEN LANGUAGE UNDERSTANDING WITHOUT SPEECH RECOGNITION

*Yuan-Ping Chen**

Department of Computer Science
University of California, Santa Cruz
ychen294@ucsc.edu

Ryan Price, Srinivas Bangalore

Interactions, LLC.
Murray Hill, NJ, USA
{rprice,sbangalore}@interactions.com

ABSTRACT

While conventional approaches to spoken language understanding involve cascading a speech recognizer with a language understanding system, in this paper, we describe a novel approach for deriving semantics directly from the speech signal without the need for an explicit speech recognition step. We evaluate this approach in the context of a customer care dialog system and demonstrate its effectiveness in comparison to the conventional approach.

***Index Terms*—** Spoken Language Understanding, end-to-end, intent determination, speech recognition

1. INTRODUCTION

With the increasing availability of robust speech recognition on mobile devices as seen in consumer devices such as Alexa, Siri, GoogleNow, there is a resurgence of speech-driven conversational systems that allow users to accomplish their desired tasks by interacting with virtual agents. Such systems are typically modeled as a sequence of components chained together – (a) a speech recognizer that transforms the speech signal into words, (b) a language understanding component that transforms the words into application semantics, typically a machine interpretable and actionable sequence of labels, and (c) a dialog manager that maintains the context of the conversation and interfaces to information sources to accomplish users' requests.

A speech recognizer itself is a sequence of transductions that transforms a sequence of acoustic events to words, mediated by an acoustic model, a pronunciation lexicon that maps phones (a representation of units of speech) into words and a language model that ranks the likelihood of a sequence of words. While the acoustic model is trained from data consisting of the audio signal and its transcription and the language model is trained from a large corpus of words, the pronunciation lexicon is a language-specific resource that is typically created through manual supervision, a tedious and expensive process.

A language understanding system (NLU) extracts one or more semantic labels from a user's request. The model is derived through supervised classification techniques from corpora of user requests annotated with semantic labels. The classification techniques rely on an optimal combination of attributes extracted from user requests with the objective of minimizing the error in predicting the semantic label during supervision.

Spoken language understanding is accomplished through a chaining of a speech recognizer with an NLU system. Typically, a single best recognizer output is used to extract the semantic content of the user's request. In order to alleviate the impact of errors in speech recognition, there have been attempts to extract semantics from n-best and lattice outputs of a speech recognizer with limited success.

In this work, we present a novel *end-to-end* approach to extract semantics directly from the speech signal without the need for a speech recognition system. The benefits of this approach include (a) obviating the need for a pronunciation lexicon, (b) extracting semantics from mixed language speech and (c) potential for richer information transfer from speech to identify the semantics. In this approach, the lexical content of the speech signal is represented as distributions over continuous space representations that are tuned to optimize the discrimination loss of identifying the semantics.

The outline of the paper is as follows. We review previous approaches and compare them to our approach in Section 2. In Section 3, we describe the end-to-end approaches that we use for jointly training the acoustic and semantic classification models. We present the data and the experimental results in Section 4, followed by discussion in Section 5, and our conclusions in Section 6.

2. RELATED WORK

The components of conventional SLU systems tend to be trained independently with training criteria that are specific to the subsystem and may be different from the overall metric for the SLU system. For example, ASR systems are typically evaluated based on word error rate (WER) but an ASR component with the lowest WER may not provide the best translation performance as part of a spoken language translation system [1]. Many attempts have been made to overcome this inconsistency by no longer treating each subsystem in isolation. End-to-end optimization of the pipeline is one approach to overcome the inconsistency problem [2]. Yaman *et al.* [3] proposes joint optimization of the ASR language model and parameters of a log-linear model for text classification using discriminative training with n-best lists. He and Deng [2] extended that work by developing a generalized framework for jointly optimizing all parameters of a GMM-based ASR subsystem and a downstream subsystem modeled by a log-linear model for a variety of speech information processing tasks.

Unlike these earlier works which have maintained the full ASR component but improved performance by jointly optimizing ASR and NLU components, our approach does not require a conventional ASR system at any stage nor does it assume the NLU component takes the form of a log-linear model.

Other works have left the ASR component in place but replaced the text classifier component with deep learning for intent determination and slot filling [4, 5, 6, 7, 8, 9].

Lee *et al.* [10] provides an overview of approaches which aim to go beyond cascading ASR and information retrieval for spoken

*Work performed as an intern at Interactions, LLC.

information retrieval (SIR). When the queries are spoken, spoken content can be retrieved without ASR through direct matching of the signals at the acoustic level in the query and archive. They review two categories of approaches for content retrieval when queries are spoken. In the first category of approaches, matches are found by comparing the audio signals or feature vector sequences by template matching using dynamic time warping. The second category of approaches is model-based and involves unsupervised training of acoustic models with a set of automatically discovered acoustic patterns found in the target archive. For example, acoustic unit discovery (AUD) can be applied to discover phoneme-like or word-like patterns from data in the target archive. Then both the spoken queries and documents in the archive can be decoded with the set of automatically discovered acoustic patterns to obtain a set of acoustic pattern sequences which can be compared.

Liu *et al.* [11] proposes topic identification without the need for manual transcriptions and dictionaries for building an ASR system specifically for the target domain. In their approach, unsupervised tokenizations of speech are obtained through unsupervised term discovery (UTD) or AUD and input into a CNN to obtain a topic classification. However, the input features for UTD and AUD are bottleneck features extracted from a conventional ASR system trained with transcribed multilingual data. Furthermore, the system for UTD or AUD representations and the CNN classifier are trained with different objectives and are never jointly trained. Thus, the inconsistencies resulting from treating each subsystem in isolation still remain.

Audhkasi *et al.* [12] proposes an approach for keyword search (KWS) without an ASR system. A character-based CNN-RNN language model is trained on a text corpus for encoding text queries and an RNN autoencoder is trained with untranscribed speech for extracting a fixed length representation of utterances in the KWS database. At test time, utterance encodings and query embeddings are input to a KWS neural network to determine if the utterance contained the query. The system components can also be finetuned together for the KWS objective but that did not improve performance. Their approach suffers from a large degradation in performance relative to the baseline system with ASR. Additionally, the input features for the RNN autoencoder are bottleneck features extracted from a conventional ASR system trained with transcribed multilingual data.

3. METHODS

The SLU task we focus on in this work is intent determination. Intent determination is a classification task that can be defined as predicting an intent class $\hat{C}$ which maximizes the posterior probability given an utterance X,

$$\hat{C} = \underset{C}{\operatorname{argmax}} P(C|X). \quad (1)$$

The typical approach consists of two stages of processing to compute $\hat{C}$. First, an ASR system determines the most likely word sequence $\hat{W}$ as

$$\hat{W} = \underset{W}{\operatorname{argmax}} P(W|X) \quad (2)$$

through Viterbi decoding performed on a weighted finite state search graph which combines scores from an acoustic model, pronunciation lexicon, language model and transition or duration model[1]. Following that, n-gram features are extracted from $\hat{W}$ and an intent determination is made by computing

$$\hat{C} = \underset{C}{\operatorname{argmax}} P(C|\hat{W}) \quad (3)$$

[1]In practice, the Viterbi decoder returns the most likely state sequence.

using existing text classification algorithms such as support vector machines (SVMs) [13] trained to predict the intent. The conventional approach can be described as a "divide and conquer" strategy essentially treating ASR and NLU as separate components in a two-stage pipeline of processing. Variants of this two-stage strategy include passing n-best output of the ASR or a lattice subgraph of the search space [14] to intent determination, in order to alleviate the potential errors in the 1-best ASR output sequence.

In contrast, our proposed approaches train the entire pipeline end-to-end by either jointly optimizing the two stages for intent classification (Section 3.1) or directly predicting the intent given the audio input signal (Section 3.2).

3.1. Finetuning Pretrained Acoustic and Text Classifiers

In the first proposed approach, the model can be thought of as consisting of two components which are pretrained and then finetuned jointly to minimize the intent classification error (Eq. (1)) in the final stage of training. The first of the two components consists of an end-to-end acoustic model, so unlike the conventional approach, beam search decoding is never done and a pronunciation dictionary and language model are not required.

In this work, the acoustic model component is a grapheme-based network with convolutional and recurrent layers trained with Connectionist Temporal Classification (CTC) [15]. CTC is a sequence based objective [16] for recurrent neural networks (RNNs) that does not require a predetermined alignment of input frames to output labels. An alignment between inputs and output label sequences is learned during training, obviating the need for a frame-level alignment generated by an existing ASR system. Other potential choices for the acoustic model component include RNN transducers [17] and attention-based networks [18].

Once the acoustic model component has been pretrained with CTC, the softmax probabilities over graphemes output by the acoustic model component are fed directly into the second component which processes them to make an intent prediction. With the parameters of the acoustic model component fixed, a deep text classifier is trained with a set of intent labeled data to predict posterior probabilities over the set of intents $P(C|Y)$ given the softmax probabilities output by the acoustic model component Y. In this work, the text classifier component is a grapheme-based CNN [19].

After pretraining the text classifier component on the softmax outputs from the pretrained acoustic model component, we finetune the entire architecture using the negative log-likelihood criterion with the full set of intent labeled data. The proposed end-to-end architecture is illustrated in Figure 1. During this final finetuning step all parameters of the model are optimized for the intent classification objective, Eq. (1), unlike the conventional approach in which the acoustic model, language model, and NLU classifier are all trained independently with different objectives.

One advantage of this approach over the direct training approach described in Section 3.2 is that it leaves open the possibility of extracting named entities like account numbers, a type of service or product, or a persons name by further processing the sequence of softmax probabilities over graphemes extracted from the output layer of the acoustic model component.

3.2. Direct Training of Audio-to-Intent

The approach proposed in Section 3.1 initializes an audio-to-intent model with a pretrained end-to-end ASR system and a grapheme-based text classifier. While this is likely to provide a well-initialized

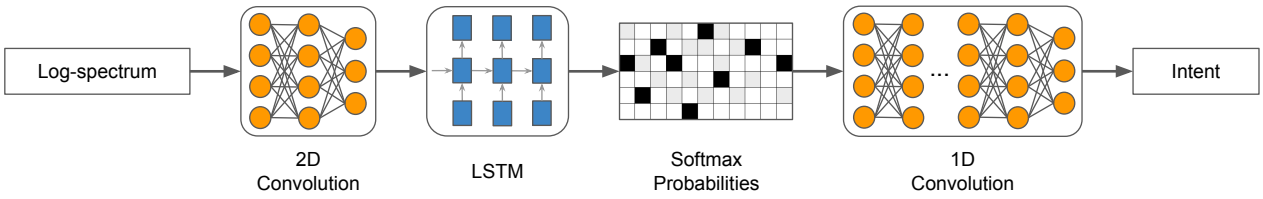

Fig. 1. Diagram of the proposed audio-to-intent architecture for semantic classification. The acoustic model component and text classification component can be pretrained separately and then finetuned together as a single model as described in Section 3.1. Alternatively, the entire model can be trained directly from random initialization to predict the semantic class given the audio input as described in Section 3.2. In that case, the model is not interpreted as having acoustic model and text classification components explicitly.

model for a potentially difficult to optimize objective of predicting the intent given the audio signal input, it still requires significant amounts of transcribed data for training the end-to-end acoustic model. Reducing the training data requirements to include only intent labeled data is highly desirable since human transcription is expensive.

We address this issue in a straightforward way by proposing to directly train the model to predict the intent given the audio input signal starting from a random initialization. This approach performs spoken language understanding entirely without speech recognition. While a variety of architectures can be conceived of, and there is no constraint to use the same architecture proposed in Section 3.1, we have chosen to do so for this initial exploration. The architecture is illustrated in Figure 1 and layers are described in Table 1. Note that the softmax function after the LSTM layers is optional and not required when CTC pretraining is not applied. Further details of the architecture we experiment with are described in Section 4 along with the rest of experimental setup.

4. EXPERIMENTS

We evaluate our approach on a customer care call classification task. Specifically, this task is from a prompt asking a customer to confirm they are calling about the account number on record. The user's responses are not restricted to contain a predetermined set of responses. This is in contrast to a Keyword Search (KWS) task for intent where the utterances are expected to contain predetermined phrases (e.g. [20]). The system is text-independent and the user is encouraged to speak naturally.

Each customer's answer corresponds to one of 15 intent categories, which are annotated by either human (*HAU*, human assisted understanding) or our previous spoken understanding (two-stage pipelined ASR+Intent Classification) system (*SLU*). Intent classifications made with high confidence are handled by the virtual assistant. When the system is unable to determine the customer's intent with a high enough confidence score, the call will be routed to a human agent and the customer's intent would be annotated manually without the utterance being transcribed (HAU). This implies that utterances annotated using the SLU system are typically simpler than those that fail over to the HAU system. We selected utterances labeled with one of 15 different intents which are mainly comprised of responses confirming the account is correct ("true") or stating it is incorrect ("false"). Other examples of classes include "live agent", "noise", "Spanish", "garbled", "don't know", "account correction", "live agent + true", "confused", "no match" and "not talking to me".

Log-spectrum features are extracted from the 8kHz speech signal. Utterances longer than 30 seconds are truncated to facilitate batch processing on the GPU. We use a combination of HAU and SLU data for training, including 100k HAU, 500k HAU, 500k HAU + 500k SLU, and 1 million HAU. The validation and test sets consist of 2987 and 8080 utterances that have been human annotated with the correct intent label, respectively.

We compare to a baseline built following the conventional approach of cascading an ASR system and SVM text classifier. The ASR system consists of an n-gram language model and hybrid DNN acoustic model trained with the cross-entropy criterion followed by the state-level Minimum Bayes Risk (sMBR) objective. The SVM text classifier was trained with word n-gram features from ASR hypotheses and hinge loss. Decoding was performed with a wide beam setting. The baseline system obtains an accuracy of 96.45% on the test set.

4.1. Finetuning With Pretrained Models

For pretraining the acoustic model component with CTC, each transcription is preprocessed as follows: 1) lowercase all characters 2) convert English numerals to words 3) separate all words with spaces 4) remove all characters outside the predefined alphabet which includes lowercase characters, space, hyphen, apostrophe, and a "blank" symbol for CTC. We use approximately 400 hours of transcribed data collected from customer care applications that has been anonymized for pretraining the CTC acoustic model component and a validation set of approximately 5000 utterances. We further augment the training data with two publicly available datasets, Switchboard and Fisher, which contain approximately 300 and 2000 hours of data respectively.

Details of the entire audio-to-intent (A2I) network architecture are given in Table 1. For details of the layers corresponding to the CTC acoustic model component, refer to layers 1-7. Convolutional layers 1 and 2 have "same" padding and batch normalization [21]. Bidirectional LSTM layers 3-6 are followed by sequence-wise batch normalization [22]. We used the deepspeech.torch [23] implementation for pretraining the acoustic model component with the CTC objective. Stochastic gradient descent (SGD) with a fixed initial learning rate of 0.00015 and Nestrov momentum of 0.9 were used. The learning rate was annealed twice by a factor of 2 when the character error rate on the validation data stopped decreasing. The batch size was 20.

For pretraining the grapheme-based CNN text classifier component we use 100k HAU labeled utterances. Inputs are 30-dimensional outputs from the pretrained CTC acoustic model component and the input feature vector length is truncated at 1014 for each utterance. For details of the layers corresponding to the text classifier component, refer to layers 8-16 in Table 1. The text classifier component was implemented in PyTorch. The network was trained with intent labeled data and negative log-likelihood loss for 50 epochs with a learning rate of 0.0005 and batch size of 20.

After the acoustic model and text classifier components had been

	Layer	Configuration
1	Conv 2D	# maps:32, K:(21,11), S:(2,2), BN
2	Conv 2D	# maps:32, K:(11,11), S:(2,1), BN
3	Bi-LSTM	# hidden units:500, BN
4	Bi-LSTM	# hidden units:500, BN
5	Bi-LSTM	# hidden units:500, BN
6	Bi-LSTM	# hidden units:500, BN
7	Fully-connected	Output dim:30
8	Conv 1D	# maps:256, K:7, S:1, MP:3
9	Conv 1D	# maps:256, K:7, S:1, MP:3
10	Conv 1D	# maps:256, K:3, S:1,
11	Conv 1D	# maps:256, K:3, S:1,
12	Conv 1D	# maps:256, K:3, S:1,
13	Conv 1D	# maps:256, K:3, S:1, MP:3
14	Fully-connected	Output dim:1024, DO
15	Fully-connected	Output dim:1024, DO
16	Fully-connected	Output dim:15

Table 1. Specification of the network architecture used with the following abbreviations - K:kernel, S:stride, MP: MaxPooling, BN:BatchNorm, DO:Dropout

	Model	Test Acc.
	ASR+SVM Baseline	96.45%
Finetuning	A2I 100k HAU Train	97.80%
	A2I 500k HAU Train	98.07%
	A2I 500k HAU 500k SLU Train	97.70%
	A2I 1m HAU Train	98.02%
Direct training	A2I 100k HAU Train	96.56%
	A2I 500k HAU Train	97.41%
	A2I 500k HAU 500k SLU Train	97.03%
	A2I 1m HAU Train	97.51%

Table 2. Test Set Accuracies for audio-to-intent models after finetuning of pretrained acoustic and text classifier components, and for models directly trained from random initialization to predict intent given audio input. 100k HAU labeled training samples were used for pretraining the text classifier component of the finetuned models.

pretrained separately, all layers in the A2I network were jointly optimized in a finetuning phase using intent labeled data and negative log-likelihood loss. The network was finetuned with SGD using a learning rate of 0.0005 until accuracy on the validation set stopped improving. The batch size was 15. The results for finetuning A2I models starting from pretrained acoustic and text classifier components are shown in Table 2. Prior to beginning finetuning, the combination of the pretrained acoustic model and text classifier components obtained an accuracy of 96.68% on the test set.

4.2. Direct Training

We used the same architecture and training data for direct training of A2I models as was used for finetuning of pretrained component models. All model parameters, layers 1-16 in Table 1, started from random initialization. No transcribed speech data was used for training. We followed the same training procedure that was used when finetuning the pretrained models with intent labeled data described in Section 4.1. Figure 2 shows the accuracy on the validation set as the network learns with different amounts of training data. The results on the test set for directly training the A2I models are shown in Table 2.

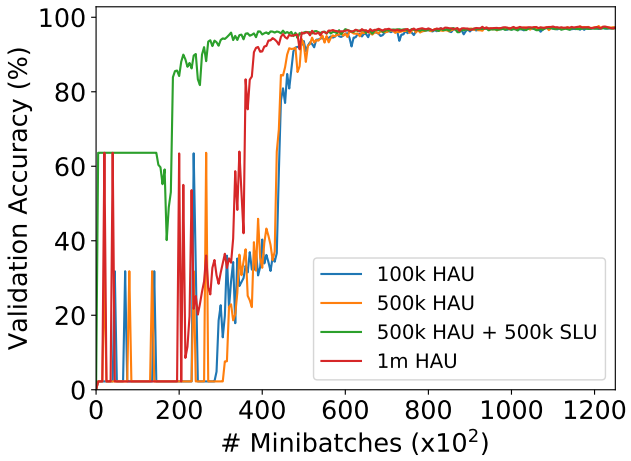

Fig. 2. Validation accuracy for A2I models directly trained from random initialization with different amounts of training data. The plot is truncated at 125k minibatches though training continued.

5. DISCUSSION

As expected, the finetuned models outperformed the A2I models directly trained from random initialization. Both methods outperformed the baseline. Finetuning was very effective at further reducing the intent classification error rate compared to simply stacking the pretrained components without any further training. With the 100k HAU labeled training set, intent accuracy was increased from 96.68% for the combined pretrained models to 97.8% after finetuning. We also observe that, increasing the training data from 100k to 500k samples improves accuracy for both the finetuning and direct training approaches, however the improvements in accuracy are larger for the direct training case. We believe the reason increasing the training data for the directly trained model is more beneficial is because it starts from a random initialization, where as the finetuned model has already benefited from the initialization obtained through CTC pretraining with a significant amount of transcribed data. Results reported for the experiments with 1 million samples are from partial runs and may have benefited from further training if time had allowed. When SLU labeled data is included in training, the randomly initialized network rapidly shifts to predicting the majority class for all inputs at the beginning of learning and we observe a higher initial accuracy for the 500k HAU + 500k SLU line in Figure 2 than for the strictly HAU labeled data. We suspect this is due to the SLU labels being noisy and ultimately the network trained with both SLU and HAU data was not as accurate as the one trained only on HAU data.

The two classes corresponding to either confirming an account number is correct or incorrect make up approximately 90% of the test data, so models still have to perform well on the remaining classes to achieve accuracies in the high nineties.

6. CONCLUSIONS

We have proposed an approach for end-to-end semantic classification without ASR. We evaluated the proposed approach on a customer care call classification task and compared to a conventional setup in which ASR and an SVM text classifier are chained together. The proposed methods outperformed the baseline even though they do not require a conventional ASR system.

7. REFERENCES

[1] Xiaodong He, Li Deng, and Alex Acero, "Why word error rate is not a good metric for speech recognizer training for the speech translation task?," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5632–5635.

[2] Xiaodong He and Li Deng, "Speech-centric information processing: An optimization-oriented approach," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1116–1135, 2013.

[3] Sibel Yaman, Li Deng, Dong Yu, Ye-Yi Wang, and Alex Acero, "An integrative and discriminative technique for spoken utterance classification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 6, pp. 1207–1214, 2008.

[4] Gokhan Tur, Li Deng, Dilek Hakkani-Tür, and Xiaodong He, "Towards deeper understanding: Deep convex networks for semantic utterance classification," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 5045–5048.

[5] Li Deng, Gokhan Tur, Xiaodong He, and Dilek Hakkani-Tur, "Use of kernel deep convex networks and end-to-end learning for spoken language understanding," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 210–215.

[6] Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al., "Using recurrent neural networks for slot filling in spoken language understanding," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 3, pp. 530–539, 2015.

[7] Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi, "Spoken language understanding using long short-term memory neural networks," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 189–194.

[8] Ngoc Thang Vu, Pankaj Gupta, Heike Adel, and Hinrich Schütze, "Bi-directional recurrent neural network with ranking loss for spoken language understanding," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 6060–6064.

[9] Ngoc Thang Vu, "Sequential convolutional neural networks for slot filling in spoken language understanding," *arXiv preprint arXiv:1606.07783*, 2016.

[10] Lin-shan Lee, James Glass, Hung-yi Lee, and Chun-an Chan, "Spoken content retrieval beyond cascading speech recognition with text retrieval," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 9, pp. 1389–1420, 2015.

[11] Chunxi Liu, Jan Trmal, Matthew Wiesner, Craig Harman, and Sanjeev Khudanpur, "Topic identification for speech without asr," *arXiv preprint arXiv:1703.07476*, 2017.

[12] Kartik Audhkhasi, Andrew Rosenberg, Abhinav Sethy, Bhuvana Ramabhadran, and Brian Kingsbury, "End-to-end asr-free keyword search from speech," *arXiv preprint arXiv:1701.04313*, 2017.

[13] Patrick Haffner, Gokhan Tur, and Jerry H Wright, "Optimizing svms for complex call classification," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*. IEEE, 2003, vol. 1, pp. I–I.

[14] Gokhan Tur, Jerry Wright, Allen Gorin, Giuseppe Riccardi, and Dilek Hakkani-Tur, "Improving Spoken Language Understanding Using Word Confusion Networks.," in *Proceedings of ICSLP'02*, September 2002.

[15] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International Conference on Machine Learning*, 2016, pp. 173–182.

[16] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen 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.

[17] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (ICASSP), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[18] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4945–4949.

[19] Xiang Zhang, Junbo Zhao, and Yann LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, 2015, pp. 649–657.

[20] Tara N Sainath and Carolina Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Interspeech*, 2015.

[21] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[22] César Laurent, Gabriel Pereyra, Philémon Brakel, Ying Zhang, and Yoshua Bengio, "Batch normalized recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2657–2661.

[23] Sean Naren, "deepspeech.pytorch," `https://github.com/SeanNaren/deepspeech.pytorch`, 2017.