COMPARING GRU AND LSTM FOR AUTOMATIC SPEECH RECOGNITION

Shubham Khandelwal, Benjamin Lecouteux, Laurent Besacier

COMPARING GRU AND LSTM FOR AUTOMATIC SPEECH RECOGNITION

Shubham Khandelwal *Benjamin Lecouteux* *Laurent Besacier*

LIG/GETALP, Univ Grenoble Alpes, France

ABSTRACT

This paper proposes to compare Gated Recurrent Unit (GRU) and Long Short Term Memory (LSTM) for speech recognition acoustic models. While these recurrent models were mainly proposed for simple read speech tasks, we experiment on a large vocabulary continuous speech recognition task: transcription of TED talks. In addition to be simpler compared to LSTM, GRU networks outperform LSTM for all network depths experimented.

We also propose a new model termed as DNN-BGRU-DNN. This model uses Deep Neural Network (DNN) followed by a Bidirectional GRU and another DNN. First DNN acts as a feature processor, BGRU is used to store temporal contextual information and final DNN introduces additional non-linearity. Our best model achieved 13.35% WER on TEDLIUM dataset which is a 16.66% & 17.84% relative improvement on baseline HMM-DNN and HMM-SGMM models respectively.

***Index Terms*—** Speech Recognition, Acoustic Models, LSTM, GRU, RNN.

1. INTRODUCTION

In recent years, artificial neural networks (ANNs) have been deployed rapidly for Automatic Speech Recognition (ASR) systems. Several factors like huge data availability and advancements in computing power (notably GPUs) helped neural networks to become popular (again). To improve ASR performance, many methods have been proposed such as hybrid Deep Neural Network-Hidden Markov Model (DNN-HMM) systems, Recurrent Neural Networks (e.g. Long Short Term Memory (LSTM), Gated Recurrent Units (GRU)), etc.

Contributions. This paper proposes to investigate ASR acoustic modeling using several recurrent neural networks (RNNs). More precisely, we systematically compare LSTMs and GRUs in several architectures. Our findings (for an English transcription task) are that GRUs are not only simpler but also more efficient than LSTMs for ASR. Our best architecture, based on a deep GRU, obtains a WER of 13.35% on the first version of TED-LIUM data set [1], which is (to our knowledge) the best result reported so far on TED-LIUM v1.

Outline. The rest of this paper is organized as follows: section 2 is dedicated to related works. We present our proposed method in section 3 while section 4 reports experiments. Finally, section 5 concludes this work and gives some perspectives.

2. RELATED WORK

Feed forward neural networks have been used as feature extractors in HMM-based speech recognition systems [2, 3, 4]. Lately, Recurrent Neural Networks (RNNs) have also been introduced for speech recognition because of their modeling capabilities for sequences. RNNs allow the model to store temporal contextual information directly without explicitly defining the length of temporal contexts. Among several implementations of RNNs, Long Short Term Memory (LSTM) [5] networks have the capability to memorize sequences with long range temporal dependencies and they started to be used for end-to-end speech recognition.

Graves et al [6] proposed the first use of deep Long Short Term Memory (LSTM) for speech recognition. They have shown that bidirectional LSTM (BLSTM) has more advantage over unidirectional LSTM and that depth is more important than layer size. Their combination of deep BLSTMs achieved a phoneme error rate of 17.7% on TIMIT benchmark (test set) which was comparable to state-of-the-art results. To overcome the difficulty of integrating deep BLSTM networks with existing large vocabulary speech recognition systems, Graves et al. [7] proposed an hybrid HMM-BLSTM architecture. The proposed model outperformed both GMM and DNN benchmarks on a subset of the Wall Street Journal (WSJ) corpus.

More recently, it has been shown that recurrent neural networks can outperform feed-forward neural networks on larger scale speech recognition tasks [8][9]. Geiger et al [10] trained BLSTM with HMM states as training targets for acoustic modeling. Their experimental results showed that the hybrid system (using state prediction networks) achieves competitive results on a medium-vocabulary ASR task on read speech data. (from the CHiME challenge). Further improvements were obtained by combining different LSTM acoustic models. Graves et al [11] proposed a speech recognition system that directly transcribes the audio data into text with minimal preprocessing using spectrograms and no explicit phonetic representation. They described a novel objective function that allows the network to be directly optimized for WER and directly integrated the network outputs with a language model during decoding. Chan et al [12] presented *Listen, Attend and Spell* (LAS), an attention based neural network which can directly transcribe acoustic signals into characters. The *Listener* (first component), is a pyramidal acoustic RNN encoder that transforms the input sequence into a high level feature representation. The *Speller* (second component) is a

RNN decoder that attends to the high level features and spells out the transcript one character at a time. LAS achieved a WER of 14.1% without any dictionary nor language model, and WER of 10.3% with language model rescoring. By comparison, the state-of-the-art CLDNN-HMM model achieved a WER of 8.0% on the same dataset. Miao et. al [13] proposed a framework called EESEN: a single RNN is learned to predict context-independent targets (phonemes or characters). Connectionist temporal classification (CTC) [14] objective function is used to infer the alignments between speech and label sequences so that they do not need pre-generated frame labels. 7.87% and 7.34% WER were obtained using phoneme based and character based systems respectively on TIMIT corpus using a trigram LM which is a relative improvement of 15% compared with [11].

A novel TC-DNN-BLSTM-DNN acoustic model architecture was proposed by Chan et al [15]. The model combines a Deep Neural Network (DNN) with Time Convolution (TC), followed by a Bidirectional Long Short Term Memory (BLSTM), and a final DNN. The first DNN acts as a feature processor to their model, the BLSTM then generates a context from the sequence acoustic signal, and the final DNN takes the context and models the posterior probabilities of the acoustic states. This model achieved 3.47% WER on the Wall Street Journal (WSJ) eval92 task (8% relative improvement over the baseline DNN models).

Finally, Gated Recurrent Unit (GRU) was proposed by [16] for machine translation. Chung et al [17] evaluated the performance of tanh, LSTM and GRU models on several NLP datasets. They demonstrated the superiority of both LSTM and GRU models over tanh unit. However, no concrete conclusion was drawn on which one is better: LSTM or GRU? Amodei et al [18] tried to compare GRU with simple RNN for speech transcription and their experimental results showed that if the model size is scaled up for a fixed computational budget, then simple RNN performs slightly better than GRU.

This paper is one attempt to compare LSTM and GRU architectures on a large vocabulary continuous speech recognition (LVCSR) task which is more complex than read speech transcription on TIMIT or WSJ. For this, we experiment on the transcription of TED Talks using TED-LIUM corpus [1].

3. PROPOSED METHOD

The proposed model is summarized as DNN-BGRU-DNN acoustic model shown in figure 1. This model uses fixed window context of acoustic features (Feature-space maximum likelihood linear regression (fMLLR) transformed features [19]) as an input.

To project the original acoustic features into a high dimensional feature space, 2 layers of DNN (2048 ReLU neurons) are used. Then high dimensional features are consumed by a single layered BGRU which models the temporal dependencies of the speech signal. The output of BGRU is then consumed by another DNN (2048 ReLU neurons) to add additional non-linear transformations before softmax layer to classify the context dependent acoustic states.

Our BGRU is implemented similarly to [16]. GRU is used to make each recurrent unit adaptively capture dependencies of different time scales. Similar to the LSTM, GRU has gating units used to deal with the flow of information inside the unit without having separate memory cells. The architecture is simpler than LSTMs.

The activation h_t of GRU at time t is a linear interpolation between the previous activation h_{t-1} and the candidate activation $\tilde{h}_t$:

$$h_t = (1 - z_t)h_{t-1} + z_t\tilde{h}_t$$

where the update gate z_t decides the update weight. This update is computed by:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$

The candidate activation $\tilde{h}_t$ is computed similar to the standard RNN:

$$\tilde{h}_t = f(W_h x_t + r_t \odot U_h h_{t-1} + b_h)$$

where r_t is a set of reset gates and $\odot$ is an element-wise multiplication. The reset gate r_t is computed similarly to the update gate:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$

Bidirectional GRU (BGRU) consumes the input acoustic window. BGRU output is a concatenation of two vectors (one for each direction: forward and backward).

$$c = \begin{bmatrix} h_T^f \\ h_T^b \end{bmatrix}$$

Here, c represents the context of the acoustic signal generated by the BGRU.

Context c is further consumed and projected by a second DNN. This DNN is further used to add more non linear transformations. The output of the second DNN is fed to the softmax layer to model the context dependent state priors. The proposed model is trained using backpropagation minimizing the cross entropy. Proposed architecture is inspired from [15].

4. EXPERIMENTS

4.1. Experimental setup

The proposed approach is evaluated on a lecture transcription task (TED talks in English). TEDLIUM dataset [1] was used for the experimentation of all models. It was developed for large vocabulary continuous speech recognition (LVCSR). The *train* part of the dataset is composed of 774 talks, representing 118 hours of speech. Evaluation is performed on the *dev* part of the dataset (19 talks, 4h).

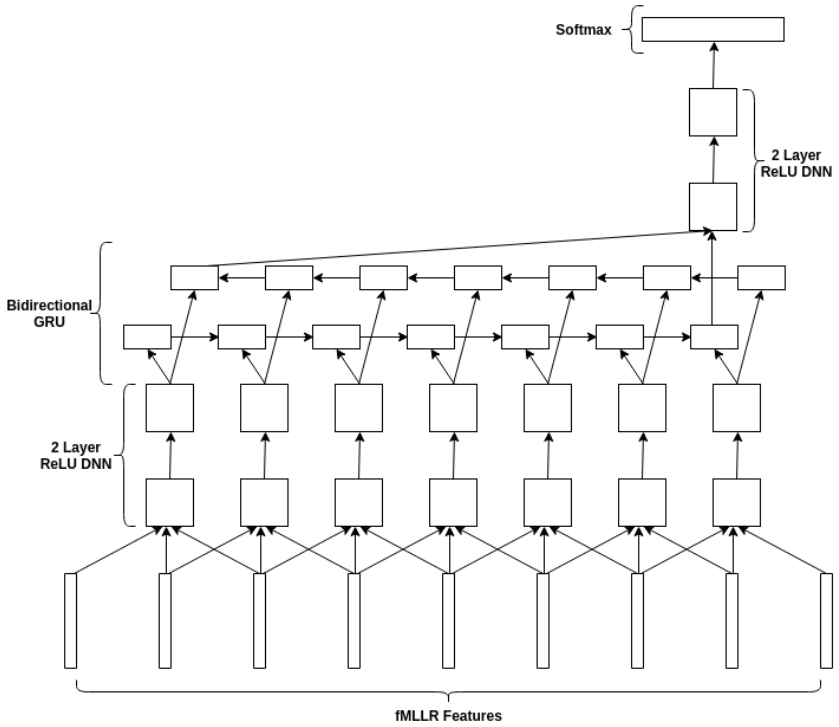

Fig. 1. Proposed architecture: DNN-BGRU-DNN. The model has 3 parts: ReLU DNN used to project the original fMLLR acoustic features to vectors which are then consumed by a BGRU before a final ReLU DNN uses the BGRU output for additional non linear projections before softmax classification. Figure modified from [15].

We use Kaldi [20], an open-source speech recognition toolkit distributed under a free license. The baseline GMM system is based on mel-frequency cepstral coefficient (MFCC) acoustic features (13 coefficients expanded with delta and double delta features and energy : 40 features) with various feature transformations including linear discriminant analysis (LDA), maximum likelihood linear transformation (MLLT), and feature space maximum likelihood linear regression (fMLLR) with speaker adaptive training (SAT). The GMM acoustic model makes initial phoneme alignments of the training data set for the following DNN (or RNN) acoustic model training.

The speech transcription process is carried out in two passes: an automatic transcript is generated with a GMM-HMM model of 12000 states and 200000 Gaussians. Then word graphs outputs obtained during the first pass are used to compute a fMLLR-SAT transform on each speaker. The second pass is performed using DNN or SGMM acoustic model trained on acoustic features normalized with the fMLLR matrix.

The English language model is trained with MIT language model toolkit [1] using following corpora:

- News commentary 2007-2012 [21]
- Gigaword version 5 [22]
- TDT 2-4 [23]
- TED Train (80%)

Then, linear interpolation is applied between the LM trained on the above corpora by tuning the perplexity on the remaining 20% data of TED/train. We will use 3 and 5-gram language models for our experiments.

4.2. HMM-SGMM and HMM-DNN baselines

We have used two baseline systems based on Kaldi *s5* recipe using HMM-SGMM model and HMM-DNN. The performance using 3-gram and 5-gram language models are provided in Table 1 (HMM-SGMM) and Table 2 (HMM-DNN).

Model	dev (WER%)
subspace-GMM	18.64 (lm3) 16.25 (lm5)

Table 1. HMM-SGMM performance on TEDLIUM dev

For HMM-DNN, we experimented with deeper and wider networks, however we found that 8 layered DNN architecture was the best.

[1] https://github.com/mitlm/mitlm

Model	size	layers	dev (WER%)	params n.
DNN	512	3	22.98 (lm3)	4.8 M
	1024	5	19.96 (lm3)	12.76 M
	1024	8	19.45 (lm3), 16.02 (lm5)	15.91M

Table 2. HMM-DNN performance on TEDLIUM dev

Model	Size	Layers	dev (WER%)	params n.
BLSTM	128	1	23.2 (lm3)	4.6M
LSTM	128	5	25.91 (lm3)	1.2 M
LSTM	256	5	22.98 (lm3)	3M
BLSTM	512	1	23.73 (lm3)	5.2M
BGRU	512	1	22.44 (lm3)	4.8M
GRU	256	5	21.32 (lm3)	1.8M

Table 3. Small size LSTM and GRU performance on TEDLIUM dev. Preliminary experiments on TEDLIUM dev.

4.3. Preliminary RNN Experiments

Preliminary experiments were done to compare small size LSTM and GRU (similar or smaller number of parameters compared to our DNN-512-3 baseline).

The experimental results using trigram language model are shown in table 3. The results show that the GRU performs better than the LSTM for equivalent number of parameters (BGRU512-1 better than BLSTM128-1 while both models have the same number of parameters). Also, we observe that deeper GRUs perform better. The result obtained with only 1.8M parameters (GRU256-5, WER=21.32%) outperforms the DNN-512-3 baseline (4.8M parameters, WER=22.98%) as well as the LSTM with similar topology (LSTM256-5, 3M parameters, WER=22.98%).

4.4. Proposed Architecture

Different combinations of BLSTM and BGRU with DNN were evaluated. We report here the main results obtained with the architecture proposed in previous section.

We compared our proposed DNN-BGRU-DNN architecture with the DNN-BLSTM-DNN architecture proposed in [15]. The results are shown in table 4.

The results show that adding DNN before and after BLSTM and BGRU significantly improves the performances reported in Table 3 for BLSTM and BGRU. The DNN-BGRU-DNN model (2 layered DNN + 1 layer BGRU + 2 layer DNN ; dimension of the GRU recurrent and non-recurrent node: 512 & DNN: 1024) achieved 20.13% WER and 16.85% WER using 3-gram and 5-gram language models respectively which is better than the results obtained with the corresponding DNN-LSTM-DNN architecture, while having less parameters. Surprisingly, it seems that DNN-BGRU-DNN benefits more from a 5-gram LM which would mean that the obtained word graph contain better hypotheses.

Model	Size	Layers	dev (WER%)	params n.
[Existing] DNN-BLSTM-DNN	1024-512-1024	2-1-2	**20.24 (lm3), 19.11 (lm5)**	12.37M
[Proposed] DNN-BGRU-DNN	1024-512-1024	2-1-2	**20.13 (lm3), 16.85 (lm5)**	11.12M

Table 4. Proposed Model Results on TEDLIUM dev.

The proposed model is very easy to train and converge and is equivalent in performance to the deeper DNN baselines (5 and 8 layers) mentioned in Table 2 (which use slightly more parameters). The model is trained using stochastic gradient descent method using a minibatch size of 100. The learning rate was started from 0.0006 and it was decayed by an geometrical distribution in every epoch. The learning rate floor was 0.00006 (that mean the learning rate does not decay beyond this value). The proposed model took around 47 hours to train with GeForce Nvidia GTX Titan 970 GPU.

4.5. Deeper RNN models

Model	Size	Layers	dev (WER%)	params n.
BLSTM	512	3	18.36 (lm3)	16.8M
BGRU	1024	2	17.80 (lm3)	30.3M
GRU	512	5	19.40 (lm3)	5.3M
GRU	1024	5	18.58 (lm3)	17.8M
BGRU	512	3	**16.37 (lm3)** **13.35 (lm5)**	13.8M

Table 5. Experiments on deeper and wider RNNs (LSTM and GRU) evaluated on TEDLIUM dev.

The model evaluated in previous subsection had only one layered bidirectional GRU. So, we experimented with increasing number of bidirectional layers in our model. Firstly, BGRU models with deeper and wider networks were trained and it was found that 3 layered BGRU model (dimension of the GRU recurrent and non-recurrent node: 512) achieved 16.37%, and 13.35% WER using 3-gram and 5-gram language models respectively which is the best performance found among all the models tested so far, as shown in table 5. The BGRU with 3 layers significantly outperforms our deep DNN baseline[2] with 8 layers, while having less parameters. It is also better than its BLSTM counterpart with 3 layers.

Now, to benefit from more BGRU layers, we should experiment this in our DNN-BGRU-DNN architecture (for instance, 2 layered DNN + 3 layer BGRU + 2 layer DNN; dimension of the GRU recurrent and non-recurrent node: 512 & DNN: 1024). However, such a model requires a massive amount of gpu memory. At the time of this submission, we could not get results with this model yet, but we plan to include them in final version of this paper if it is accepted[3].

[2] as well as the HMM-SGMM baseline

[3] Based on results observed in Tables 4 and 5, we believe that WER should be further reduced.

4.6. Discussion

Following are the main outcomes of our experiments:

- In general, it was observed that bidirectional RNNs are better than uni-directional RNNs.
- BGRU has much less computation time than BLSTM. So BGRU model could be more easily deployed on small devices like mobile, etc.
- GRU networks outperformed simple LSTM for all network depths and for fixed number of parameters. We also evaluated GRU networks with 5 or more recurrent layers but they did not improve the performance.
- Vanishing and exploding gradient problem is avoided by looking at threshold and clip the gradient to that threshold. On top of it, LSTM and GRU also helped to avoid such type of problems in following way:
  - LSTM allows disabling of writing to a cell by turning "off" the gate to prevent any changes in the content of the cell over many cycles. It means that longer term dependencies can be learned. Similarly, when the gate is "open", the update equation does not completely replace the contents of a cell, rather maintaining a weighted average of a new value and previous value.
  - In GRU, the update gate controls how much information from the previous hidden state will carry over to the current hidden state to maintain an averaged gradient. Also, when the reset gate is close to 0, the hidden state is forced to ignore the previous hidden state and reset with the current input only. Using this, the hidden state can drop any information which is irrelevant in the future.
- We will provide a *github* link to all our GRU scripts used, in the final version of this paper, so that other researchers can reproduce our experiments made on TED-LIUM dataset with *Kaldi*.

5. CONCLUSION AND PERSPECTIVES

5.1. Conclusion

The goal of this work was to evaluate and compare several RNN models for a large vocabulary continuous speech recognition task in English (TEDLIUM dataset). More precisely, RNN models were trained with different number of layers to compare the performance of LSTM and GRU. It was found that GRU outperforms LSTM in terms of (less) computation time and (better) WER[4].

A novel architecture (DNN-BGRU-DNN) was also proposed which achieves better performance than its (recently proposed) DNN-BLSTM-DNN counterpart. The proposed model is easy to implement using Kaldi. It does not get overfitting due to regularization and also avoid vanishing or exploding gradient problems observed for single LSTM/GRU.

[4]tested with roughly equivalent number of parameters

Intensive experiments were performed with deeper and wider recurrent networks. For 3-layered BGRU model, 16.37% and 13.35% WER were achieved using 3- and 5-gram language models respectively which is more than 16% relative improvement over the DNN baseline of 19.45% WER when 3-gram language model is considered. To the best of our knowledge, 13.35% WER is the best performance ever reported on *dev* set of this TEDLIUM experimental setup.

5.2. Perspectives

Following ideas could be proposed in future work:

- **Short term:** at the time of this paper submission, we were not able to complete experiments on DNN-BGRU-DNN with 3 or more BGRU layers. We expect further improvements compared to our 13.35% WER obtained with the 3-layer BGRU in Table 5. Such experiments will be included in the final version of this paper if accepted.
- **Long term:** other architectures such as Highway Networks [24], High order RNNs [25], Multi-Function Recurrent Unit (MuFuRU) [26] could be studied and experimented on TEDLIUM. Connectionist Temporal Classification method [14] is another promising approach as it does not require presegmented (force-aligned) training data, or external post-processing to extract the label sequence from the network outputs. Combining convolutional neural networks (CNN) with deep LSTM and GRU models would be another interesting direction to explore.

Aknowledgment

Most of the computations presented in this paper were performed using the Froggy platform of the CIMENT infrastructure (https://ciment.ujf-grenoble.fr), which is supported by the Rhône-Alpes region (GRANT CPER07_13 CIRA) and the Equip@Meso project (reference ANR-10-EQPX-29-01) of the programme Investissements d'Avenir supervised by the Agence Nationale pour la Recherche.

6. REFERENCES

[1] Anthony Rousseau, Paul Deléglise, and Yannick Esteve, "Ted-lium: an automatic speech recognition dedicated corpus.," in *LREC*, 2012, pp. 125–129.

[2] Hervé Bourlard and Nelson Morgan, "Connectionist speech recognition. a hybrid approach," 1994.

[3] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[4] Abdel-rahman Mohamed, George E Dahl, and Geoffrey Hinton, "Acoustic modeling using deep belief networks," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 14–22, 2012.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," vol. 9, no. 8, pp. 17351780, 1997.

[6] Alan 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.

[7] Alan Graves, Navdeep Jaitly, and Abdel-rahman Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 273–278.

[8] Hasim Sak, Andrew W Senior, and Françoise Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling.," in *INTERSPEECH*, 2014, pp. 338–342.

[9] Hasim Sak, Oriol Vinyals, Georg Heigold, Andrew Senior, Erik McDermott, Rajat Monga, and Mark Mao, "Sequence discriminative distributed training of long short-term memory recurrent neural networks," *entropy*, vol. 15, no. 16, pp. 17–18, 2014.

[10] Jürgen T Geiger, Zixing Zhang, Felix Weninger, Björn Schuller, and Gerhard Rigoll, "Robust speech recognition using long short-term memory recurrent neural networks for hybrid acoustic modelling," in *INTERSPEECH*, 2014, pp. 631–635.

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

[12] William Chan, Navdeep Jaitly, Quoc V Le, and Oriol Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[13] Yajie Miao, Mohammad Gowayyed, and Florian Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," *arXiv preprint arXiv:1507.08240*, 2015.

[14] Alex Graves, "Connectionist temporal classification," in *Supervised Sequence Labelling with Recurrent Neural Networks*, pp. 61–93. Springer, 2012.

[15] William Chan and Ian Lane, "Deep recurrent neural networks for acoustic modelling," *arXiv preprint arXiv:1504.01482*, 2015.

[16] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[17] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[18] Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," *arXiv preprint arXiv:1512.02595*, 2015.

[19] Daniel Povey and George Saon, "Feature and model space speaker adaptation with full covariance gaussians.," in *INTERSPEECH*, 2006.

[20] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[21] Jörg Tiedemann, "Parallel data, tools and interfaces in opus.," in *LREC*, 2012, pp. 2214–2218.

[22] David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda, "English gigaword," *Linguistic Data Consortium, Philadelphia*, 2003.

[23] Mark et al Liberman, "Tdt2 multilanguage text version 4.0," *LDC2001T57 CD. Philadelphia: Linguistic Data Consortium,*, 2001.

[24] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber, "Highway networks," *arXiv preprint arXiv:1505.00387*, 2015.

[25] Rohollah Soltani and Hui Jiang, "Higher order recurrent neural networks," *arXiv preprint arXiv:1605.00064*, 2016.

[26] Dirk Weissenborn and Tim Rocktaschel, "Mufuru: The multi-function recurrent unit," *arXiv preprint arXiv:1606.03002*, 2016.