KNOWLEDGE DISTILLATION ACROSS ENSEMBLES OF MULTILINGUAL MODELS FOR LOW-RESOURCE LANGUAGES

Jia Cui, Brian Kingsbury, Bhuvana Ramabhadran, George Saon, Tom Sercu,
Kartik Audhkhasi, Abhinav Sethy, Markus Nussbaum-Thom, Andrew Rosenberg

IBM Watson, 1101 Kitchawan Rd,
Yorktown Heights, NY, 10598, U.S.A.

ABSTRACT

This paper investigates the effectiveness of knowledge distillation in the context of multilingual models. We show that with knowledge distillation, Long Short-Term Memory(LSTM) models can be used to train standard feed-forward Deep Neural Network (DNN) models for a variety of low-resource languages. We then examine how the agreement between the teacher's best labels and the original labels affects the student model's performance. Next, we show that knowledge distillation can be easily applied to semi-supervised learning to improve model performance. We also propose a promising data selection method to filter un-transcribed data. Then we focus on knowledge transfer among DNN models with multilingual features derived from CNN+DNN, LSTM, VGG, CTC and attention models. We show that a student model equipped with better input features not only learns better from the teacher's labels, but also outperforms the teacher. Further experiments suggest that by learning from each other, the original ensemble of various models is able to evolve into a new ensemble with even better combined performance.

Index Terms— Multilingual, Keyword search, Low-resource language, LSTM, VGG, CTC, Attention Models

1. INTRODUCTION

The concept of teacher-student for neural network was proposed in [1] to help investigate why deep neural networks performed better than shallow neural networks. It was derived from the model compression proposed in [2]: training one model (student) to mimic another model or an ensemble of models (teacher). The methodology was modified and applied in [3] to compress a large deep neural network (DNN) into a smaller one so that the latter can be fit into handheld devices without much loss in performance. Hinton et al. [4] called this kind of methodology as *distilling knowledge*. Knowledge distillation is an active area of research [5, 6, 7, 8, 9, 10]. Some research has focused on whether one type of complex neural network is necessary or can be replaced by simpler networks [5, 6, 7], while other research [11] investigates whether a complex model could benefit from a simple one given limited training resources.

Our research on knowledge distillation can be largely divided into two parts. The first part investigates knowledge transfer from long short-term memory (LSTM) models [12, 13] to deep neural network (DNN) [14] models. We take the same strategy as described in [6], where the top soft labels from LSTM are used to train a DNN by minimizing the cross-entropy of frame based labels. We evaluate the models for both recognition and keyword search performance. In order to better understand which kind of training data yields the most benefit from the teacher's labels, we divide the training data into two categories. We show that most of the gains from teacher models are actually from the *difficult* frames where the teacher *disagrees* with the original labels. Moreover, we find that it is important to balance the *easy* and *difficult* data in each mini batch during DNN training to avoid significant performance loss. We show how this observation can be used to improve semi-supervised learning of acoustic models.

The second part of our research focuses on knowledge transfer between models of the same structure yet different input features. We use five different multilingual (ML) features, either extracted from different multilingual models directly or derived through different monolingual models trained from the same multilingual features. We show that a student with better ML features is able to outperform the teacher model just by learning from the teacher's soft labels. We will also show that this learning leads to better performance than feature fusion and sometimes model combination. Moreover, the combination of student model with the teacher model is even better than that of baseline model and teacher model. That is surprising given that the models in the new ensemble are more similar to each other than the original ensemble. It is in a way similar to the observation in [7]: if each model in the ensemble tries to mimic the average of the ensemble, each of them can match the ensemble performance. Our experiments go a step further, showing that the new ensemble might be able to evolve by encouraging models to learn from each other.

The rest of the paper is organized as follows. In Section 2, we explain our implementation of knowledge distillation in detail and make comparisons to related work. Section 2, gives a background of multilingual training and the Babel project. In Section 3, experiments on DNN models which learn from LSTM models on various data sets are be reported in terms of recognition and keyword search performance. Further analysis on data selection and semi-supervised learning are also discussed. In Section 4, we turn our attention to knowledge transfer between models of same structure but different input features, comparing results with different model combination methods and investigating possible self-improvement of an ensemble of models.

2. BACKGROUND AND RELATED WORK

Various algorithms have been proposed for transferring knowledge from teacher models to student models. In [1], a student shallow net is trained to mimic teacher models by minimizing the L2 loss of logits (input of the softmax layer). Li et al. [3] proposes to minimize the Kullback-Leibler (KL) divergence between the output distributions of the small-size DNN and a large-size DNN by utilizing large amounts of un-transcribed data. [5, 6] follows the KL-divergence criterion but used only a small portion (about 1%) of the top labels

which cover a large(about 98%) probability mass from the teacher model as targets. [6] investigated the number of labels and the interpolation weight of teacher's labels with original labels. In this paper, we took a strategy similar to [6] and use posteriors of top 50 most likely labels for each prediction of the teacher. We do not interpolate the teacher's label with the original labels. The KL-divergence criterion used for training the student model is equivalent to minimizing the cross entropy of the soft target labels [5].

The work reported in this paper is focused on the IARPA Babel OP3 evaluation. The system performance is evaluated by term-weighted value (TWV): a measure that summarizes system performance for a specific assignment of costs to misses and false alarms [15, 16]. We report results in term of both word error rate (WER) and the maximum term-weighted value (MTWV) which is the TWV achieved at the optimal setting of the decision threshold.

The focus of this paper is on improving the performance of multilingual models. Multilingual models are trained using Multilingual (ML) features and have been the key for achieving good performance in ASR and keyword search (KWS) tasks in the Babel program [17, 18, 19, 20, 21, 22]. In this work, we use three types of ML models. The baseline is a two-stage model, using convolutional neural network (CNN) [23] and DNN components. The bottleneck (BN) layer of the second DNN is extracted as the ML feature directly or being used as the input for monolingual CTC [24] and Attention models [25]. The encoder activations of those models are then extracted for our experiments. The other two ML models are one-stage models: very deep convolutional neural network (VGG) [26] and long short-term memory (LSTM) RNN network.

3. KNOWLEDGE TRANSFER FROM LSTM TO DNN

3.1. Experimental Setup

All experiments were conducted on Babel OP3 languages: the development data includes 305 Guarani, 307 Amharic, 403 Dholuo, 104 Pashto, 306 Igbo, 401 Mongolian and 402 Javanese and the surprise language 404 Georgian. The development languages are included in multilingual modeling but for the surprise language we only do fine-tuning of the network. The baseline multilingual (ML) features are extracted using a BUT-style [27, 17, 28] stacked CNN+DNN structure. The first-level CNN takes an 11-frame context window containing 40-dimensional log-Mel static+Δ + Δ2 features. The CNN has two convolutional layers plus 7 fully connected sigmoid layers. The DNN has 6 sigmoid layers with a [400, 1024, 1024, 80, 1024, 3000, 3000] architecture.

The input of the second stage DNN is the BN layer from CNN with expanded context. The BN layer extracted from the second DNN is used as baseline ML features without fine tuning. The ML model is trained with 24 Babel full language packs and the additional four LDC languages (English, Spanish, Mandarin, and Arabic). This feature (called ML28) will be used to train all models in this section.

We build two types of monolingual models with the ML input features. One of them is a regular DNN model. It has a structure [80,1024,1024,1024,1024,256,3000], three ReLU layers followed by one sigmoid and one linear layer. The baseline DNN is tuned with different learning rates and the best one selected. The student DNN has the same structure except with 2048 nodes in each hidden layer instead of 1024.

The teacher model we use is a LSTM with 4 bi-directional layers and a projection layer of size 256 at the end. Each layer has 1024 hidden units. The LSTM is trained only with cross-entropy criterion and implemented with Theano. All DNN models on target languages are speaker independent models. They are trained with The IBM Attila Speech Recognition Toolkit [29] in the following three steps [30, 31, 32, 33]: (1) layer-wise discriminative pre-training using stochastic gradient and cross-entropy loss, (2) training using stochastic gradient and cross-entropy (XENT) loss, and (3) training using distributed Hessian-free (HF) optimization and state-level minimum Bayes risk (sMBR) loss. Teacher-student learning in this paper only happens at the pre-training and XENT training step. All sMBR training uses the original labels. We settled on using the top 50 labels empirically from experiments on Pashto where the true label was in the 50-best 97% of the time.

3.2. Student Models Evaluated by WER and MTWV

We begin with Pashto experiments. Both teacher and student models take ML28 as input features. Table 1 shows the performance of baseline and student models without sMBR training. The LSTM gives WER better than baseline model (53.7% vs. 54.9%). The student model improves on both XENT and Phone error rate (PER), as well as the WER (-0.9%). In order to check if the gain comes from the larger model size, we also tried different model sizes (1K or 2K hidden units) for baseline and students (reported in the last two rows in Table 1). We find that increasing parameters doesn't affect baseline model but improves student model significantly, which is consistent with the observation from [1].

Data	XENT	PER	WER
Baseline	2.23	50.1	54.9
Teacher	-	-	53.7
Student	2.20	49.3	54.0
Baseline w/ 2K	2.27	50	54.7
Student w/ 1K	2.23	49.8	54.7

Table 1. *Comparing DNN baseline and student models learned from LSTM on Pashto*

After sequence training, the student Pashto model is still 0.6% better than the baseline model, and the MTWV also improves 0.5%. Figure 1 shows the improvement of all student models on Babel OP3 development languages in terms of WER in blue (the lower the better) and MTWV in yellow (the higher the better). The gains are not even over all languages. For Javanese, the WER drops 1.1% and MTWV increases 1%, while for Igbo, student model is almost the same as the baseline. It might make sense given that the teacher LSTM model itself doesn't improve over the baseline XENT model for that language while all other teacher models are better than the baseline XENT models. Igbo also has the highest WER(about 58

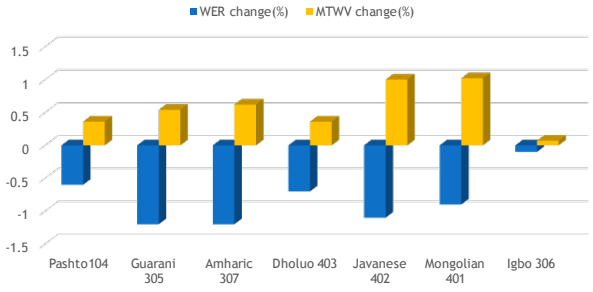

Fig. 1. *Improvement from student models in terms of WER and MTWV*

3.3. Training Data Analysis

In this set of experiments, we divide all Pashto training data into two categories: an *easy* one where original labels are consistent with the teacher's best choices (Data-easy) and a *difficult* one where they differ (Data-diff). About 60% of the training data frames are *easy* with ∼40% difficult as showed in Figure 2. We build baseline and student models respectively on different categories of data. We find that the student model improves more on Data-diff partition. If we continue training with all of the data and teacher labels, the model that started with Data-easy soft labeled data yields the same results as the original student model, while the one started with Data-diff and original labels ends up 2% absolute worse in WER. We may need larger models to learn well from the difficult data.

Fig. 2. *Build baseline and student models on different data categories on Pashto*

3.4. Semi-supervised Learning

Traditional semi-supervised learning transcribes the unlabeled data automatically with a trained ASR model, then filters the data or down scales it before mixing it with the original transcribed data. There we use the student sequence trained model (row 3 in Table(2), with WER 40.8%) to label the un-transcribed Georgian data (about 40 hours), we then build a model on this data only (U-auto), we end up with WER 46.3% which is worse than the baseline 45.8%. Results from these semi-supervised experiments are reported in Table2.

When we use LSTM to assign labels to these data and only select those whose teacher's best labels are consistent with the auto assigned labels (Uc), we can improve the baseline by simply adding these data into the transcribed data (45.6% v.s. 45.8%). This shows that choosing the *easy* frames is a promising method for data selection of unlabeled data.

Model and Data	WER(XENT/sMBR/)/MTWV
DNN: T-original	45.8/41.6/0.6947
LSTM: T-original	43.8
Student1: T-teacher	44.6/40.8/0.6997
DNN: U-auto	46.3
DNN: T-original+Uc-original	45.6
Student2: U-teacher	44.8
Student3: (T+U)-teacher	44.2/40.2/0.7029
Student1: (L+Uc)-teacher	44.2

Table 2. *Semi-supervised learning on Georgian*

If we only apply teacher-student model to unlabeled data as in [3], we can see significant gains from un-transcribed data. Table 2 shows that using un-transcribed data alone yield 44.8% WER, better than the baseline 45.8% and close to the student model trained on transcribed data 44.6%. When we combine both data, the WER improves to 44.2% and with sequence training yields 40.2%. The keyword search performance is also improved from 0.6997 to 0.7029%. However, removing the *difficult* automatically transcribed data doesn't seem to help (last row).

4. LEARNING WITHIN ENSEMBLES

4.1. Learning from the Baseline sMBR Model

Since the baseline sMBR model yields better performance, we wonder whether it can be used as a teacher and if yes, what knowledge will it pass to the student. Figure 3) shows the experimental result. The first bar is the baseline sMBR model, gives WER 52.0%. The student model starts from random initialization, learning from the soft labels provided by the baseline sMBR. After cross-entropy training, the student model gives a very good WER (52.3%), almost matches the teacher's performance and much better than the student learning from LSTM (54.0%). However, the follow-up sMBR training only yields a slight improvement, while the student learning from LSTMs can improve all the way to 51.3%. There are two observations: first, a model can be trained with cross-entropy criterion to reach the point that only sequence training can reach by learning from soft labels. This might suggest that the soft labels include some sequence information; second, the model learning from its own sMBR model could not outperform its own best result without new knowledge.

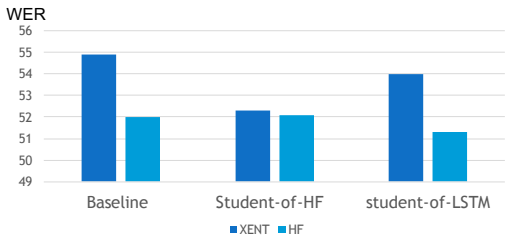

Fig. 3. *Learning from the baseline sMBR model on Pashto*

4.2. Various Multilingual Features for Georgian

In the following part of this section, we focus on DNN models with different input features. We will compare performance of single model as well as system combination implemented by unweighted posterior averaging. We will check if the models, by learning from one another, can improve the combination performance.

There are three multilingual (ML) models. The first one is the baseline as introduced in the previous section. The other two are LSTM and VGG ML models, both trained with 24 Babel languages only. The LSTM model has 4 bidirectional LSTM layers with 512 units per direction, a linear bottleneck layer with 256 units and a 3000-unit output layer. It is fined tuned with Georgian training data only. The VGG model [26, 34, 35, 36] comprises 12 convolutional layers, with a max-pooling layer inserted after every 3 convolutional layers, followed by 5 fully connected layers. All hidden layers are ReLU. It is fine-tuned by a mixture of Georgian and the original training data. There are also two derived ML features, CTC[37] and Attention[25]. Both are monolingual Georgian models trained with ML24 (CNN+DNN but with only 24 languages) features fused with ML features from RWTH. The encoder activations from CTC and Attention models are extracted as derived ML features. Again, all those features are used as input for DNN training on Georgian data.

4.3. DNN Learning From DNN

Table 3 shows a set of experiments with DNN models learning from each other. The sequence trained models are used as teachers. The student models are randomly initialized and trained with teacher labels for pre-training and XENT training. The first three rows are baseline models trained with original hard labels and with different input features. The fourth and fifth rows show the student models with ML28 as input, learning from soft labels generated by DNN sMBR models trained with CTC and Attention features respectively. We can see that both student models improve WER at the XENT step, beating the baseline performance but not that of the teachers. Sequence training of students gives further improvement in terms of WER, but not always on MTWV. As in Section 3.2 where Igbo's teacher model is not as good as the baseline, here we notice that the Attention sMBR model doesn't outperform the baseline sMBR model.

To investigate whether the teacher has to be better than the student, we conduct further experiments as showed in the second group of experiments in Table 3. fVGG is a strong ML feature. It's own baseline is better than the student1 model (cf. Section 4.2): 39.8% v.s. 40.8%. To our surprise, the student model outperforms the teacher model even before doing sequence training, reaching WER 40.0%, almost the same as the original sMBR model performance (39.8%), 0.8% better than the teacher performance (40.8%). In order to check if this gain is only from the sequence training, we continue sMBR training on the student model and it yields another 1.1% WER improvement. The impact on keyword search performance is also significant from 0.7055 to 0.7163.

Model	XENT WER	sMBR (WER/MTWV)
ML28	45.1	41.6/0.6947
CTC	-	40.6/0.6889
ATT	-	43.9/0.6541
ML28-learn-from-CTC	41.6	40.0/0.7086
ML28-learn-from-ATT	42.8	41.1/0.6930
ML28-learn-from-LSTM(stu1)	44.2	**40.8**/0.6997
fVGG	42.7	39.8/0.7055
fVGG-learn-from-stu1	**40.0**	38.9/0.7163

Table 3. *DNN learning from DNN on Georgian*

4.4. Ensemble Evolution

Encouraged by our previous observation that a student can outperform the teacher, we continue to investigate if the ensemble of the teacher and student models could perform better than the ensemble of the teacher and the baseline models. In other words, when a model in an ensemble improves itself by learning from models in the same ensemble, will the combined performance of the ensemble be improved or not?

Table 4 presents four groups of experimental results A denotes the teacher model, B denotes a baseline model and B' denotes the student model. B and B' take the same input features but learned from original labels and teacher labels respectively. The '+' symbol denotes system combination with posterior interpolation. In each of these experiments, the student model outperforms its own baseline and the new ensemble A+B' outperforms the old ensemble A+B in terms of MTWV. An interesting observation is that fVGG learns better from the CTC model than from a better teacher fLSTM or stu1 model (0.7194 vs 0.7163 and 0.7147). Also, the student model itself can sometimes beat the combination of the two baseline models as showed in the third row of Table 4.

A	B	A+B	B'	A+B'
stu1 0.6997	fVGG 0.7055	0.7203	0.7163	0.7228
fLSTM 0.6993	fVGG 0.7066	0.7268	0.7147	**0.7323**
CTC 0.6889	fVGG 0.7066	0.7182	**0.7194**	0.7212
CTC 0.6889	ML28 0.6947	***0.7040***	***0.7086***	0.7114

Table 4. *Student v.s. baseline ensemble evaluated by MTWV*

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have investigated knowledge distillation as applied to different types of NN models and models trained with different input features. We have conducted our experiments with Babel OP3 development and surprise language, evaluating performance in terms of both recognition and keyword search.

We found that across all Babel languages student models improved performance in terms of both WER and MTWV by learning from soft labels provided by teachers. Our experiments showed that the gain was mostly from difficult frames where the teacher's best label was not consistent with the original label. We also conducted a simple experiment of using unlabeled data in this framework and also got positive gains.

After observing student models outperforming teacher models, we conducted a group of experiments where two different models were paired and one is learning from the other. We found that in all cases, the new pair (student+teacher) outperformed the old pair (baseline+teacher). Sometimes, the student model itself yielded better performance than the combination of baseline and teacher. Further investigation on larger ensembles will be conducted, together with learning from the un-labeled data set.

.

6. ACKNOWLEDGEMENT

This work is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD/ARL) contract number W911NF-12-C-0012. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government. This effort uses the IARPA Babel Program language collection release IARPA-babe{101-v0.4c, 102b-v0.5a, 103b-v0.4b, 104b-v0.4bY, 105bv0.4, 106-v0.2f, 107b-v0.7, 201b-v0.2b, 202b-v1.0d, 203b-v3.1a, 204b-v1.1b, 205b-v1.0a,206bv0.1d, 207b-v1.0b,301b-v2.0,302b-v1.0a,303b-v1.0a,304b-v1.0b,305b-v1.0c, 306b-v2.0c,307b-v1.0b,401b-v2.0b,402b-v1.0b,403b-v1.0b,404b-v1.0a,}.

7. REFERENCES

[1] Jimmy Ba and Rich Caruana, "Do deep nets really need to be deep?," in *Advances in Neural Information Processing Systems 27*, pp. 2654–2662. Curran Associates, Inc, 2014.

[2] Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil, "Model compression," in *KDD*, 2006.

[3] Jinyu Li, Rui Zhao, Jui-Ting Huang, and Yifang Gong, "Learning small-size dnn with output-distribution-based criteria," in *Interspeech*, September 2014.

[4] Jeff Dean Geoffrey Hinton, Oriol Vinyals, "Distilling the knowledge in a neural network," in *arXiv:1503.02531v1*, 2015.

[5] William Chan, Nan Rosemary Ke, and Ian Lane, "Transferring knowledge from a rnn to a dnn," in *INTERSPEECH*, 2015.

[6] Krzysztof J. Geras, Abdel-rahman Mohamed, Rich Caruana, Gregor Urban, Shengjie Wang, Ozlem Aslan, Matthai Philipose, Matthew Richardson, and Charles Sutton, "Blending lstms into cnns," in *ICLR Workshop*, 2016.

[7] Xiaohui Zhang, Daniel Povey, and Sanjeev Khudanpur, "A diversity-penalizing ensemble training method for deep learning," in *INTERSPEECH*, 2015.

[8] Yevgen Chebotar and Austin Waters, "Distilling knowledge from ensembles of neural networks for speech recognition," in *Interspeech*, 2016.

[9] Zhiyuan Tang, Dong Wang, and Zhiyong Zhang, "Recurrent neural network training with dark knowledge transfer," in *ICASSP*, 2016.

[10] Jeremy H. M. Wong and Mark J. F. Gales, "Sequence student-teacher training of deep neural networks," in *Interspeech*, 2016.

[11] Dong Wang, Chao Liu, Zhiyuan Tang, Zhiyong Zhang, and Mengyuan Zhao, "Recurrent neural network training with dark knowledge transfer," in *CSLT TECHNICAL REPORT-20150013*, 2015.

[12] Sepp Hochreiter and Jurgen Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, 1997.

[13] Alex Graves, Abdel rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *ICASSP*, 2013.

[14] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *Signal Processing Magazine*, 2012.

[15] Jonathan G. Fiscus, Jerome Ajot, and George Doddington, "The spoken term detection (std) 2006 evaluation plan," *NIST USA Sep*, 2006.

[16] Mary P. Harper, "http://www.iarpa.gov/index.php/research-programs/babel," .

[17] Zoltan Tuske, Ralf Schluter, and Hermann Ney, "Multilingual hierarchical MRASTA features for ASR," in *Interspeech*, 2013.

[18] Zoltan Tuske, David Nolden, Ralf Schluter, and Herman Ney, "Multilingual MRASTA features for low-resource keyword search and speech recognition systems," in *ICASSP*, 2014.

[19] Kate M. Knill, Mark J.F. Gales, Shakti P. Rath, Phil Woodland, Chao Zhang, and Shixiong Zhang, "Investigation of multilingual deep neural networks for spoken term detection," in *ASRU*, 2013.

[20] Mark J. F. Gales, Kate M. Knill, Anton Ragni, and Shakti P. Rath, "Speech recognition and keyword spotting for low resource languages: Babel project research at CUED," in *Spoken Language Technologies for Under-Resource Languages*, 2014.

[21] František Grézl and Martin Karafiát, "Combination of multilingual and semi-supervised training for under-resourced languages," in *Proc. Interspeech*, 2014.

[22] Jia Cui, Brian Kingsbury, Bhuvana Ramabhadran, and etc, "Multilingual representations for low resource speech recognition and keyword search," in *ASRU*, 2015.

[23] Yann LeCun and Yoshua Bengio, "Convolutional networks for images, speech, and time-series," *The Handbook of Brain Theory and Neural Networks*, 1995.

[24] Alex Graves, Santiago Fernndez, and Faustino Gomez, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *ICML 2006*, 2006, pp. 369–376.

[25] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, "End-to-end attention-based large vocabulary speech recognition," in *arXiv:1508.04395*, 2016.

[26] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015.

[27] Frantisek Grezl, Martin Karafiat, and Lukas Burget, "Investigation into bottle-neck features for meeting speech recognition," in *Interspeech*, 2009.

[28] Martin KARAFIAT, Frantisek GREZL, Karel VESELY, Mirko HANNEMANN, Igor SZOKE, and Jan CERNOCKY, "But 2014 babel system: Analysis of adaptation in nn based systems," in *Interspeech*, 2014.

[29] Hagen Soltau, George Saon, and Brian Kingsbury, "The IBM Attila speech recognition toolkit," in *Spoken Language Technology Workshop (SLT), 2010 IEEE*. IEEE, 2010, pp. 97–102.

[30] Yoshua Bengio, Pasca Lamblin, Dan Popovici, and Hugo Larochelle, "Greedy layer-wise training of deep networks," *Advances in Neural Information Processing Systems*, 2007.

[31] James Martens, "Deep learning via hessian-free optimization," in *ICML*, 2010.

[32] Brian Kingsbury, Tara N Sainath, and Hagen Soltau, "Scalable minimum bayes risk training of deep neural network acoustic models using distributed hessian-free optimization," in *Interspeech*, 2012.

[33] Tara N. Sainath, Brian Kingsbury, Bhuvana Ramabhadran, Petr Fousek, Petr Novk, and Abdel rahman Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *ASRU*, 2011.

[34] Tom Sercu, Christian Puhrsch, Brian Kingsbury, and Yann LeCun, "Very deep multilingual convolutional neural networks for lvcsr," in *ICASSP*, 2016.

[35] Tom Sercu and Vaibhava Goel, "Advances in very deep convolutional neural networks for lvcsr," in *Interspeech*, 2016.

[36] Sergey Ioffe and Christian Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[37] Yajie Miao, Mohammad Gowayyed, and Florian Metze, "Eesen: end-to-end speech recognition using deep rnn models and wfst-based decoding," in *CoRR, vol. abs/1507.08240*, 2015.