INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Distilling Knowledge from an Ensemble of Models for Punctuation Prediction

Jiangyan Yi[1,2], *Jianhua Tao*[1,2,3], *Zhengqi Wen*[1], *Ya Li*[1]

[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China
[3]CAS Center for Excellence in Brain Science and Intelligence Technology, Institute of Automation, Chinese Academy of Sciences, Beijing, China

{jiangyan.yi, jhtao, zqwen, yli}@nlpr.ia.ac.cn

Abstract

This paper proposes an approach to distill knowledge from an ensemble of models to a single deep neural network (DNN) student model for punctuation prediction. This approach makes the DNN student model mimic the behavior of the ensemble. The ensemble consists of three single models. Kullback-Leibler (KL) divergence is used to minimize the difference between the output distribution of the DNN student model and the behavior of the ensemble. Experimental results on English IWSLT2011 dataset show that the ensemble outperforms the previous state-of-the-art model by up to 4.0% absolute in overall F_1-score. The DNN student model also achieves up to 13.4% absolute overall F_1-score improvement over the conventionally-trained baseline models.

Index Terms: transfer learning, knowledge distillation, ensemble, neural network, punctuation prediction

1. Introduction

The output word sequences of most automatic speech recognition (ASR) systems don't contain punctuation marks. Thus it degrades the readability of the generated word sequences and causes poor user experience in real-world ASR systems. Therefore, it is important to predict punctuation marks for the speech transcripts.

Many efforts have been made to predict punctuation automatically for speech transcripts. These approaches can be roughly divided into three categories in terms of the applied features: acoustic features, lexical features and the combination of the previous two features. Although the acoustic features are effective [1, 2, 3], it doesn't work well when users make pauses in unnatural places in real ASR systems [4, 5]. The combination of the acoustic and lexical features [6, 7, 8] can alleviate this problem. However, many of these works [8, 9, 10] need utilize the lexical data with the corresponding audio data for training. Thus there is a limitation to use any kind of textual data. To overcome this problem, this paper focuses on the lexical features based approach. There are many literatures [11, 12, 13, 14] that predict punctuation marks only with lexical features.

Previously, punctuation marks are treated as hidden inter-word events [15]. The n-gram language model (LM) is used to train on texts with punctuation marks [11]. Some researchers find that conditional random fields (CRFs) are well-suited to predict punctuation marks. Lu and Wang et al. [12, 16] use CRF with only token features. Ueffing et al. [13] propose to combine syntactic features with LM scores, token features and sentence length. These approaches obtain improvement only with lexical features.

More recently, neural networks are used to predict punctuation marks in speech transcriptions. Che et al. [14] propose to use deep neural network (DNN) and convolution neural network (CNN) to predict punctuation marks. This method outperforms the CRF based method over purely lexical features. The work in [6] describes a punctuation prediction model based long short-term memory (LSTM). Most recently, the bidirectional recurrent neural network with attention mechanism (T-BRNN) is proposed by Tilk et al. [17] to improve the performance of punctuation prediction. The overall F_1-score of this model on English IWSLT2011 dataset [14] is 64.4%. Although it has achieved the state-of-the-art performance on this dataset only using lexical features, there is still much room for improvement.

Previous studies [18, 19] show that an ensemble of models outperforms the single models obviously. However, it is cumbersome to deploy the ensemble to make predictions in real-world systems. Therefore, knowledge distillation has been proposed by Caruana et al. [20, 21] to address this problem. Hinton et al. [22] propose a more general framework to distill knowledge efficiently using high temperature. A new model is trained to imitate the behavior of a strong ensemble of models. Meanwhile, there are also some works use knowledge distillation to compress acoustic models or perform model adaptation in ASR tasks. Li et al. [23] utilize Kullback-Leibler (KL) divergence to train a small DNN model guided by a large DNN model. Chan et al. [24] propose to transfer knowledge from a recurrent neural network (RNN) model to a small DNN model using KL divergence. Most recently, Chebotar et al. [25] propose to distill ensembles of models into a single model by KL divergence. In [26], Asami et al. use knowledge distillation to perform domain adaptation. Experimental results show that these approaches are effective.

Inspired by these approaches, this paper proposes to distill knowledge from an ensemble of models to a single DNN model for punctuation prediction. The single DNN model is called the student model. The ensemble is called the teacher model. The teacher model consists of three single models: DNN, T-BRNN and bidirectional LSTM with a CRF layer (BLSTM-CRF). The BLSTM-CRF model is proposed to perform named entity recognition tasks [27]. KL divergence is used to minimize the difference between the output distribution of the student model and the behavior of the teacher model.

http://dx.doi.org/10.21437/Interspeech.2017-1079

Experimental results on English IWSLT2011 dataset show that the ensemble outperforms the previous state-of-the-art model by up to 4.0% absolute in overall F_1-score. The DNN student model also achieves up to 13.4% absolute overall F_1-score improvement over the conventionally-trained baseline models.

The rest of this paper is organized as follows. Section 2 describes ensembles of models. Section 3 introduces knowledge distillation from an ensemble. Section 4 presents the experiments. The results are discussed in Section 5. This paper is concluded in Section 6.

2. Ensembles of models

This section explains how ensembles of models can be used to improve the performance of punctuation prediction.

The ensemble consists of three single models: DNN, T-BRNN [17] and BLSTM-CRF [27]. Given M models that have already been trained on available data, the word-level predictions are combined by taking weighted average of their output probabilities over punctuation marks z. That is, for each word w of a sentence, the ensemble replaces a vector of output probabilities over z computed as

$$P_{ens}(z|w) = \sum_{j=1}^{M} \alpha_j \, P_j(z|w) \qquad (1)$$

where j denotes the index of a single model, $P_j(z|w)$ are the output probabilities of z computed from the j-th single model given w, $\alpha_j \in [0,1]$ is the weight of the j-th single model, $\sum_{j=1}^{M} \alpha_j = 1$, $P_{ens}(z|w)$ are the output probabilities of z computed from the ensemble given w. This method requires that all the single models have identical punctuation marks.

Equation (1) can be viewed as a simplified version of linear regression (LR) where the bias is just set to 0. Therefore, LR is used to choose the weight combination that makes the ensemble obtain the best performance of punctuation prediction on a given dataset.

3. Distilling knowledge from an ensemble

In this section, the distillation is described in detail. Then the framework of knowledge distillation for punctuation prediction is introduced.

3.1. Distillation

The distillation is to make the student model mimic the behavior of the teacher model. Thus the output distribution of the student model is forced to be close to the behavior of the teacher model. This can be achieved by minimizing the KL divergence [28] between the output distribution of the student model and the behavior of the teacher model. Letting P_t denotes the behavior of the teacher model, Q denotes the output distribution of the student model, we wish to minimize

$$D_{KL}(P_t||Q) = \sum_i P_t(z_i|w) ln\,(P_t(z_i|w)/Q(z_i|w)) \qquad (2)$$

where i denotes the index of punctuation marks, z_i denotes the i-th punctuation mark, w denotes a given word, $Q(z_i|w)$ is referred as the posterior probability of z_i computed from the student model given w, $P_t(z_i|w)$ is referred as the behavior of z_i computed from the teacher model given w.

The behavior of the teacher model contains the posterior probabilities of the teacher model and the correct probabilities of the training data. The posterior probabilities of the teacher model are called soft labels. The correct probabilities of the training data are called hard labels. Thus, $P_t(z_i|w)$ can be viewed as a linear interpolation of the hard labels and the soft labels, which can be defined

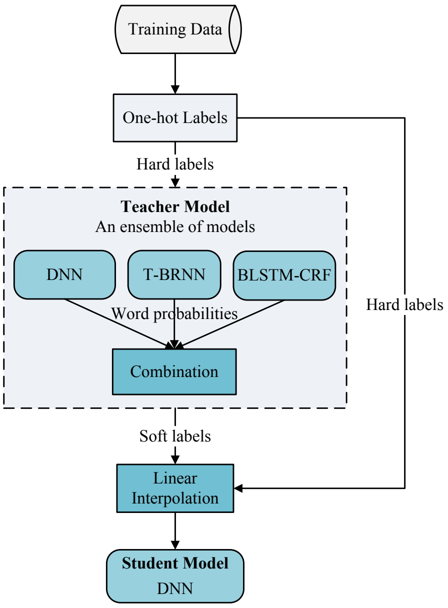

Figure 1: *Framework of knowledge distillation for punctuation prediction.*

$$P_t(z_i|w) = (1-\beta)T(z_i|w) + \beta P_{ens}(z_i|w) \qquad (3)$$

where $\beta \in [0,1]$ is the interpolation weight, $T(z_i|w)$ is the hard label of z_i given w, $P_{ens}(z_i|w)$ is the soft label of $z_i$computed from the teacher model given w.

$D_{KL}(P_t||Q)$ is also defined as follow

$$D_{KL}(P_t||Q) = H(P_t,Q) - H(P_t) \qquad (4)$$

$$H(P_t,Q) = \sum_i P_t(z_i|w) lnQ(z_i|w) \qquad (5)$$

$$H(P_t) = \sum_i P_t(z_i|w)\, lnP_t(z_i|w) \qquad (6)$$

where $H(P_t)$ is only related to the teacher model but has no influence on the training of the student model. So $D_{KL}(P_t||Q)$ can be defined

$$D_{KL}(P_t||Q) \triangleq \sum_i -P_t(z_i|w) lnQ(z_i|w) \qquad (7)$$

By Equation (7), we can see that minimizing the KL divergence is equivalent to minimize the Cross Entropy (CE) loss. Thus, the Equation (7) can be viewed as the standard CE loss function. We only need to replace the hard labels with $P_t(z_i|w)$.

$P_t(z_i|w)$ is computed by Equation (3). β is a hyper parameter which can be adjusted using a development set. When $\beta = 1$, the student model is trained only using the soft labels. When $\beta = 0$, the student model is trained only using the hard labels.

3.2. Framework of knowledge distillation

The teacher model has the same training data with the student model. Moreover, the punctuation marks of the teacher model are identical to the student model. The framework of knowledge distillation for punctuation prediction is shown in Fig. 1.

The distillation procedure has four stages. The first stage is to generate hard labels from the training data. The hard labels are one-hot vectors, such as [0 1 0 0] is referred as the hard labels of one word. The probability of this word belonging to label 2 is 1. The probability of this word belonging to other labels is 0. The second stage is to train the teacher model using the hard labels. The teacher model is an ensemble of DNN, T-BRNN and BLSTM-CRF models. The third stage is to compute soft labels from the teacher model. The soft labels are computed using forward algorithm, such as [0.01 0.87 0.02 0.1] denotes the soft labels of one word. The probability of this word belonging to label 2 is 0.87. The probability of this word belonging to label 1 is 0.01. Finally, the student model is trained to mimic the behavior of the teacher model. The behavior of the teacher model is realized by a linear interpolation of the hard labels and the soft labels using Equation (3).

The student model has the same architecture with the DNN model of the ensemble models. Before the training, the student model is initialized with a copy of the parameters from the DNN model of the ensemble. This will make the student model to converge faster.

4. Experiments

In this section, a series of experiments are conducted to evaluate the proposed approach for punctuation prediction.

4.1. Dataset

Our experiments are conducted on an English IWSLT dataset which contains TED talks. This dataset is reorganized by Che et al. [14]. It contains three datasets: training set, development set and test set. The training set and development set are from the training data of IWSLT2012 machine translation track. The training set contains 2.1 M words. The development set has 296K words. The test set is from the IWSLT2011 reference (Ref.) and ASR test set, which contains 14K words respectively. More details of this dataset can be found in [14]. The state-of-the-art performance on this dataset is achieved by 64.4% in overall F_1-score [17].

The output vocabulary of this dataset contains three kinds of punctuation marks (Comma, Period and Question mark) and a non-punctuation mark "O". "Overall" denotes three kinds of punctuation marks.

4.2. Metrics

In our experiments, all models are evaluated using precision (P), recall (R), F_1-score(F_1). We evaluate the performance for comma, period and question marks on two test sets (Ref. and ASR) respectively. More details of metrics can be found in [14].

4.3. Features

The input features of all the models are word embedding features. In order to compare with other works in [6, 14, 17], we choose pre-trained word vectors from the Glove[1] to obtain input features. The GloVe.6B.50d vector has 50 dimensions.

4.4. Ensemble teacher model

The teacher model is an ensemble of three single models. The combination of the models is performed at word-level.

The single DNN model has 3 hidden layers. Each hidden layer has 2048 nodes. These parameters are set motivated by the work in [14]. The DNN model is implemented using Theano [29] and trained on GPUs. The initial learning rate is set to 2×10^{-3} for the DNN model. The T-BRNN model is trained using the public available source code[2]. We also use the same hyper parameters for the T-BRNN-pre model that are used in [17]. The T-BRNN-pre model denotes that the T-BRNN model is trained with pre-trained Glove word vectors. The BLSTM-CRF model is trained using the public available source code[3]. The teacher model is trained by linear combination of the above three single models. The teacher model is referred as Teacher-Ensemble. The best weight combination is listed in Table 1.

Table 1:*The best weight combination for the teacher model.*

Model	DNN	T-BRNN-pre	BLSTM-CRF
Weight α	0.19	0.38	0.43

From Table 1, we can see that the BLSTM-CRF model makes the most contribution to the teacher model. The T-BRNN-pre makes more contribution than the DNN model.

The performance of the teacher model and the three single models on Ref. and ASR test sets are listed in Table 2.

From Table 2, we can see that the single model BLSTM-CRF obtains the highest overall F_1-score among all the single models. The main reason is that the BLSTM-CRF model not only uses past and future information, but also utilizes the sentence-level label knowledge. Furthermore, the Teacher-Ensemble achieves the best performance on both Ref. and ASR test sets. The overall F_1-score improves absolutely by 4.0% on Ref. test set and by 2.7% on ASR test set when comparing the Teacher-Ensemble model with the previous state-of-the-art model T-BRNN-pre in [17]. The Teacher-Ensemble model also outperforms the best single model BLSTM-CRF on Ref. test set by 3.3% absolute in overall F_1-score and on ASR test set by 2.6% absolute in overall F_1-score.

The student model has the same architecture with the DNN model from the ensemble models. So we select the DNN model as our baseline model to compare the performance with the student model. The DNN model in Table 2 is also referred as Baseline-DNN in Table 3.

4.5. Single student model

The student model has the same architecture and identical number of the parameters with the Baseline-DNN model. The student model is denoted as Student-DNN. The initial learning rate is set to 1×10^{-5} for the training of the Student-DNN model. The interpolation weight β is adjusted on the development set. When β is set to 0.3, we can obtain the best Student-DNN model. The performance of the student model and the other best models on Ref. and ASR test sets are listed in Table 3.

From Table 3, we can find that the Student-DNN model obtains obvious improvement when compared with the Baseline-DNN model and the other best models in [14, 6]. The overall F_1-score improves absolutely by 10.3% on Ref. test set and by 9.4% on ASR test set when comparing the Student-DNN model with the Baseline-DNN model.

[1] http://nlp.stanford.edu/projects/glove

[2] https://github.com/ottokart/punctuator2

[3] https://github.com/marekrei/sequence-labeler

Table 2: *The performance of the ensemble teacher model and the three single models on Ref. and ASR test sets. T-BRNN-pre is the best model in [17], and DNN, BLSTM-CRF, Teacher-Ensemble are our models.*

	Model	**Comma**			**Period**			**Question**			**Overall**		
		P(%)	*R(%)*	F_1*(%)*	*P(%)*	*R(%)*	F_1*(%)*	*P(%)*	*R(%)*	F_1*(%)*	*P(%)*	*R(%)*	F_1*(%)*
Ref.	DNN	58.1	35.8	44.3	62.1	64.8	63.4	60.5	48.9	54.1	60.2	49.8	53.9
	T-BRNN-pre [17]	65.5	47.1	54.8	73.3	72.5	72.9	70.7	63.0	66.7	70.0	59.7	64.4
	BLSTM-CRF	58.9	59.1	59.0	68.9	72.1	70.5	71.8	60.6	65.7	66.5	63.9	65.1
	Teacher-Ensemble	**66.2**	**59.9**	**62.9**	**75.1**	**73.7**	**74.4**	**72.3**	**63.8**	**67.8**	**71.2**	**65.8**	**68.4**
ASR	DNN	47.5	32.3	38.5	58.3	60.5	59.4	57.1	46.8	51.4	54.3	46.5	49.8
	T-BRNN-pre [17]	59.6	42.9	49.9	70.7	72.0	71.4	60.7	48.6	54.0	66.0	57.3	61.4
	BLSTM-CRF	55.7	56.8	56.2	68.7	71.5	70.1	63.8	53.4	58.1	62.7	60.6	61.5
	Teacher-Ensemble	**60.6**	**58.3**	**59.4**	**71.7**	**72.9**	**72.3**	**66.2**	**55.8**	**60.6**	**66.2**	**62.3**	**64.1**

Table 3: *The performance of the DNN student model and the other best models on Ref. and ASR test sets. DNN, DNN-A and CNN-2A are the best models in [14], T-LSTM is the first stage model in [6], and Baseline-DNN and Student-DNN are our models.*

	Model	**Comma**			**Period**			**Question**			**Overall**		
		P(%)	*R(%)*	F_1*(%)*	*P(%)*	*R(%)*	F_1*(%)*	*P(%)*	*R(%)*	F_1*(%)*	*P(%)*	*R(%)*	F_1*(%)*
Ref.	DNN [14]	58.2	35.7	44.2	61.6	64.8	63.2	-	-	-	60.3	48.6	53.8
	DNN-A [14]	48.6	42.4	45.3	59.7	68.3	63.7	-	-	-	54.8	53.6	54.2
	CNN-2A [14]	48.1	44.5	46.2	57.6	69.0	62.8	-	-	-	53.4	55.0	54.2
	T-LSTM [6]	49.6	41.4	45.1	60.2	53.4	56.6	57.1	43.5	49.4	55.0	47.2	50.8
	Baseline-DNN	58.1	35.8	44.3	62.1	64.8	63.4	60.5	48.9	54.1	60.2	49.8	53.9
	Student-DNN	**60.2**	**55.4**	**57.7**	**70.6**	**71.2**	**70.9**	**69.1**	**59.5**	**63.9**	**66.6**	**62.0**	**64.2**
ASR	DNN [14]	47.2	32.0	38.1	59.0	60.9	60.0	-	-	-	54.4	45.6	49.6
	DNN-A [14]	41.0	40.9	40.9	56.2	64.5	60.1	-	-	-	49.2	51.6	50.4
	CNN-2A [14]	37.3	40.5	38.8	54.6	65.5	59.6	-	-	-	46.4	51.9	49.1
	T-LSTM [6]	41.8	37.8	39.7	56.4	49.3	52.6	55.6	42.9	48.4	49.1	43.6	46.2
	Baseline-DNN	47.5	32.3	38.5	58.3	60.5	59.4	57.1	46.8	51.4	54.3	46.5	49.8
	Student-DNN	**56.1**	**51.4**	**53.6**	**66.8**	**68.5**	**67.6**	**61.2**	**51.9**	**56.2**	**61.4**	**57.3**	**59.2**

The Student-DNN model outperforms the best DNN-A model in [14] on Ref. test set by 10.0% absolute in overall F_1-score and on ASR test set by 8.8% absolute in overall F_1-score. Additionally, the Student-DNN model outperforms the T-LSTM model in [6] on Ref. test set by 13.4% absolute in overall F_1-score and on ASR test set by 13.0% absolute in overall F_1-score.

We also find that the prediction of comma mark is more challenging when compared with the prediction of period and question mark in English. It is more difficult to predict punctuation marks on ASR test set than on Ref. test set. The reason is that the texts in ASR test set have some errors. These conclusions are consistent with the conclusions in [6, 14, 17].

5. Discussion

We can make some interesting observations from the experimental results.

The ensemble of DNN, T-BRNN and BLSTM-CRF models outperforms any of the single models. The ensemble can utilize the strengths of different model architectures to improve performance. The DNN model can learn high-level representation. The T-BRNN model can use past and future context information. In addition, BLSTM-CRF can utilize sentence-level label information. Therefore, the ensemble can obtain improvement over the single models.

The DNN student model achieves obvious improvement when compared with the conventionally-trained baseline models. The reason is that the student model is trained to imitate the teacher model with a linear interpolation of hard labels and soft labels. The soft labels have more additional rank information about the non-target labels than the hard labels. The knowledge from the strong teacher model can be transferred to the student model. Thus the student model can learn better using more rank information and more accurate knowledge.

6. Conclusions

This paper proposes to distill knowledge from an ensemble of models to a DNN student model for punctuation prediction. The ensemble consists of DNN, T-BRNN and BLSTM-CRF models. Experimental results on English IWSLT2011 dataset show that the ensemble outperforms the previous state-of-the-art model by up to 4.0% absolute in overall F_1-score. The DNN student model also achieves up to 13.4% absolute overall F_1-score improvement over the conventionally-trained baseline models. In addition, Although the ensemble outperforms the student model, the student model is more suitable to deploy than the ensemble. Future work includes training on a large dataset, comparing with Chinese models and using acoustic features.

7. Acknowledgements

This work is supported by the National High-Tech Research and Development Program of China (863 Program) (No.2015AA016305), the National Natural Science Foundation of China (NSFC) (No.61425017, No.61403386), the Strategic Priority Research Program of the CAS (GrantXDB02080006). We would like to thank Che et al. [14] for the support with the IWSLT dataset.

8. References

[1] H. Christensen, Y. Gotoh, and S. Renals, “Punctuation annotation using statistical prosody models,” in ISCA Tutorial and Research Workshop (ITRW) on Prosody in Speech Recognition and Understanding, 2001.

[2] T. Levy, V. Silber-Varod, and A. Moyal, “The effect of pitch, intensity and pause duration in punctuation detection,” in Electrical & Electronics Engineers in Israel (IEEEI), 2012 IEEE 27th Convention of. IEEE, 2012, pp. 1–4.

[3] J.-H. Kim and P. C Woodland, “The use of prosody in a combined system for punctuation generation and speech recognition.,” in Proc. Interspeech, 2001.

[4] J. Kolar and L.Lamel, “Development and evaluation of automatic punctuation for French and English speech-to-text,” in Proc. Interspeech, 2012, Portland, OR, USA, 2012.

[5] F. Batista, H. Moniz, I. Trancoso, and N. Mamede, “Bilingual experiments on automatic recovery of capitalization and punctuation of automatic speech transcripts,” Audio, Speech, and Language Processing, IEEE Transactions on, vol. 20, no. 2, pp. 474–485, 2012.

[6] O. Tilk and T. Alumäe, “LSTM for punctuation restoration in speech transcripts,” in Proc. Interspeech, 2015.

[7] X. Che, S. Luo, H. Yang and C. Meinel, “Sentence boundary detection based on parallel lexical and acoustic models,” in Proc. Interspeech, 2016.

[8] O. Klejch, P. Bell, and S. Renals, “Sequence-to-sequence models for punctuated transcription combining lexical and acoustic features,” in Proc. ICASSP, 2017, pp. 5700–5704.

[9] A. Lee and J. R. Glass, “Sentence detection using multiple annotations.” in Proc. Interspeech, 2012, pp. 1848–1851.

[10] E. Cho, K. Kilgour, J. Niehues, and A. Waibel, “Combination of nn and crf models for joint detection of punctuation and disfluencies,” in Sixteenth Annual Conference of the International Speech Communication Association, 2015.

[11] A. Gravano, M. Jansche, and M. Bacchiani, “Restoring punctuation and capitalization in transcribed speech,” in Proc. ICASSP, 2009, pp. 4741–4744.

[12] W. Lu and H. T. Ng, “Better punctuation prediction with dynamic conditional random fields,” in EMNLP 2010, Cambridge, MA,USA, 2010.

[13] N. Ueffing, M. Bisani, and P. Vozila, “Improved models for automatic punctuation prediction for spoken and written text,” in Proc. Interspeech, 2013.

[14] X. Che, C. Wang, H. Yang, and C. Meinel, “Punctuation prediction for unsegmented transcript based on word vector,” in The 10th International Conference on Language Resources and Evaluation (LREC), 2016.

[15] A. Stolcke, E. Shriberg, R. Bater, M. Ostendorf, D. Hakkani, M. Plauche, and Y. Lu, “Automatic detection of sentence boundaries and disfluencies based on recognized words,” In Proc. ICSLP, 1998.

[16] X. Wang, H.T. Ng, and K.C. Sim, “Dynamic conditional random fields for joint sentence boundary and punctuation prediction,” in Proc. Interspeech, 2012.

[17] O. Tilk and T. Alumäe, “Bidirectional Recurrent Neural Network with Attention Mechanism for Punctuation Restoration,” in Proc. Interspeech, 2016.

[18] Y. Zhao, J. Xue, and X. Chen, “Ensemble learning approaches in speech recognition,” in Speech and Audio Processing for Coding, Enhancement and Recognition, Springer New York, 2015, pp. 113–152.

[19] L. Deng and J. C. Platt, “Ensemble deep learning for speech recognition.” in Proc. Interspeech, 2014, pp. 1915–1919.

[20] C.Bucila, R. Caruana, and A. Niculescu-Mizil, “Model Compression,” in ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006.

[21] J. Ba and R. Caruana, “Do deep nets really need to be deep?” in Advances in Neural Information Processing Systems, 2014, pp. 2654–2662.

[22] G. Hinton, O. Vinyals, and J. Dean, “Distilling the Knowledge in a Neural Network,” in Neural Information Processing Systems: Workshop Deep Learning and Representation Learning Workshop, 2014.

[23] J. Li, R. Zhao, J.T. Huang, and Y. Gong, “Learning small-size DNN with output-distribution-based criteria,” in Proc. Interspeech, 2014.

[24] W. Chan, N.R. Ke, and I. Lane, “Transferring knowledge from a RNN to a DNN,” in Proc. Interspeech, 2015.

[25] Y. Chebotar and A. Waters, “Distilling knowledge from ensembles of neural networks for speech recognition,” in Proc. Interspeech, 2016, pp. 3439-3443.

[26] T. Asami, R. Masumura, Y. Yamaguchi, H. Masataki, and Y. Aono, “Domain Adaptation Of DNN Acoustic Models Using Knowledge Distillation,” in Proc. ICASSP, 2017, pp. 5185-5189.

[27] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural architectures for named entity recognition,” Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016 , pp.260-270.

[28] S. Kullback and R. A. Leibler, “On information and sufficiency,” Ann. Math. Statist., vol. 22, no. 1, pp. 79–86, 1951.

[29] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow,A. Bergeron, N. Bouchard, and Y. Bengio, “Theano: new features and speed improvements,” Deep Learning andUnsupervised Feature Learning NIPS 2012 Workshop, 2012.