SELF-ATTENTION BASED MODEL FOR PUNCTUATION PREDICTION USING WORD AND SPEECH EMBEDDINGS

Jiangyan Yi[1], *Jianhua Tao*[1,2,3]

[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[2]CAS Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences, Beijing, China
{jiangyan.yi, jhtao}@nlpr.ia.ac.cn

ABSTRACT

This paper proposes to use self-attention based model to predict punctuation marks for word sequences. The model is trained using word and speech embedding features which are obtained from the pre-trained Word2Vec and Speech2Vec, respectively. Thus, the model can use any kind of textual data and speech data. Experiments are conducted on English IWSLT2011 datasets. The results show that the self-attention based model trained using word and speech embedding features outperforms the previous state-of-the-art single model by up to 7.8% absolute overall F_1-score. The results also show that it obtains performance improvement by up to 4.7% absolute overall F_1-score against the previous best ensemble model.

Index Terms— Self-attention, transfer learning, word embedding, speech embedding, punctuation prediction

1. INTRODUCTION

In general, automatic speech recognition (ASR) systems don't generate output sequences with punctuation marks. However, punctuation marks will affect the readability of speech transcripts. Therefore, it is very important to predict punctuation marks for speech transcripts.

A lot of efforts have been made to restore punctuation automatically. There are three main approaches to predict punctuation marks in terms of the modeling technology. First, punctuation marks are treated as hidden inter-word events [1]. The n-gram language models [2, 3] or hidden Markov models [4] are used to restore punctuation marks. Second, predicting punctuation is viewed as a sequence labeling task [5, 6], in which a punctuation mark is assigned to each word. Previous studies [7, 8, 5] show that conditional random fields (CRFs) are well-suited to predict punctuation marks. The overall F_1-score of the best CRF based model on the English IWSLT2011 datasets [9] is 53.5%. Recently, Che et al. [9] propose to use deep neural network and convolution neural network to predict punctuation marks. The results show that the neural network based method outperforms the CRF based method. More recently, Tilk et al. use long short-term memory (LSTM) [10] and bidirectional recurrent neural network with attention mechanism (T-BRNN) [11] to improve the performance. Most recently, Yi et al. [12] propose to use bidirectional LSTM with a CRF layer (BLSTM-CRF) and an ensemble of models to predict punctuation. The overall F_1-score of the best ensemble model on the IWSLT2011 datasets [9] is 68.4%. Finally, restoring punctuation is treated as a monolingual machine translation problem in which the source is unpunctuated text and the target is punctuated text [13, 14, 15] or sequences of punctuation marks [16, 17]. Klejch et al. [16, 17] propose a RNN encoder-decoder architecture with an attention layer to restore punctuation marks. This architecture is similar to the model used for machine translation task. The overall F_1-score of the model in [17] on the MGB Challenge dataset [18] is 62.63%. Although the results show that the above mentioned approaches are effective and promising, there is still much room for improvement.

Inspired by the success of self-attention mechanisms for machine translation tasks [19], this paper proposes to use the self-attention based model to improve the performance of punctuation prediction tasks. Previously, three kinds of features are used to predict punctuation marks: acoustic features, lexical features and the combination of acoustic and lexical features. Although the acoustic features are more effective than the lexical features [4, 20], they don't work well when users make pauses in unnatural places in real ASR systems [21]. The combination of acoustic and lexical features [16, 17] can alleviate this problem. However, many of the studies [10, 22, 17] need utilize the lexical data with the corresponding speech data for training. So the use of textual data and speech data is limited. Motivated by the success of the Speech2Vec applied on word similarity tasks [23], this paper also proposes to train self-attention based model using the word and speech embedding from the pre-trained Word2Vec [24] and Speech2Vec [23]. Therefore, the self-attention based model can use any kind of textual data and speech data.

The main contributions of this paper are as follows. (1)

Self-attention based model is used to predict punctuation marks. (2) Speech2Vec and Word2Vec are both used to obtain word embedding and speech embedding features, respectively. Experiments are conducted on English IWSLT2011 datasets [9]. The results show that the self-attention based model trained using word and speech embedding features outperforms the previous state-of-the-art single model. The results also show that it achieves performance improvement against the previous best ensemble model.

The rest of this paper is organized as follows. Section 2 describes the model architecture of punctuation prediction. Section 3 presents the experiments and results. This paper is concluded in Section 4.

2. MODEL ARCHITECTURE OF PUNCTUATION PREDICTION

The self-attention neural network is proposed by Vaswani et al. [19] to perform machine translation tasks. This network has an encoder-decoder structure, which relies solely on an attention mechanism. The self-attention based model is used to predict punctuation marks in this paper as shown in Fig.1. The inputs are word sequences, e.g. "*Amy where is the national theatre*". The outputs are punctuation marks, such as "*O COMMA O O O O O QUESTION*".

2.1. Model architecture

The encoder consists of a stack of N identical layers as shown in the left of Fig.1. Each layer has two sub-layers. The first is a multi-head self-attention mechanism. The second is a fully connected feed-forward network. A residual connection is employed around each of the two sub-layers, followed by layer normalization [19].

The decoder is also composed of a stack of N identical layers as shown in the right of Fig.1. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, namely a masked multi-head attention mechanism.

Positional encodings are used to make use of the order of the input or output sequence. Similarly to other sequence transduction models, learned embeddings are used to convert output tokens to vectors of dimension d_{model}.

Different from input embeddings learned in [19], we use word and speech embeddings from the pre-trained Word2Vec and Speech2Vec as the input embeddings, as shown in the bottom of the left of Fig.1. Since 50-dimensional word and speech embedding features are used in this paper, a linear transformation is used to convert the 50-dimensional features to d_{model}-dimensional features .

2.2. Multi-head self-attention

An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors.

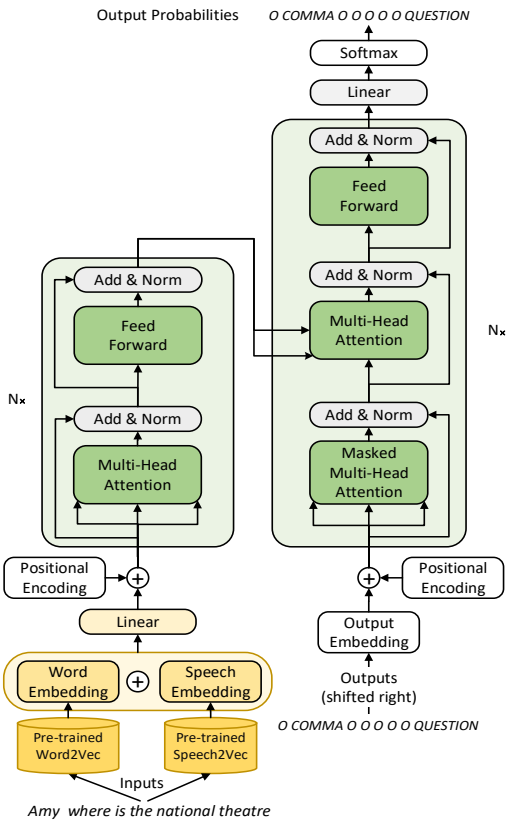

Fig. 1. The architecture of the self-attention based model for punctuation prediction.

The input consists of queries and keys of dimension d_k, and values of dimension d_v. The attention function is computed on a set of queries simultaneously, packed together into a matrix Q. The keys and values are also packed together into matrices K and V. The matrix of outputs is computed as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

Instead of performing a single attention function with d_{model}-dimensional keys, values and queries, Vaswani et al. [19] found it beneficial to linearly project the queries, keys and values h times with different, learned linear projections to d_k, d_k and d_v dimensions, respectively. Multi-head attention allows these projected versions of queries, keys and values to perform the attention function in parallel, yielding d_v-dimensional output values.

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \quad (2)$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (3)$$

Where the projections are parameter matrices $W_i^Q \in R^{d_{model} \times d_k}$, $W_i^K \in R^{d_{model} \times d_k}$, $W_i^V \in R^{d_{model} \times d_v}$ and $W^O \in R^{hd_v \times d_{model}}$.

2.3. Speech embedding

Inspired by Word2Vec, Yu-An [23] et al. propose to train Speech2Vec. The proposed Speech2Vec aims to learn a fixed length embedding of an audio segment that captures semantic information of the spoken word directly from audio. It can be viewed as a speech version of Word2Vec. Learning speech embedding directly from speech enables Speech2Vec to make use of the acoustic information carried by speech that does not exist in plain text.

In this paper, we use word and speech embedding as the input features of the self-attention based model for punctuation prediction tasks. The word and speech embedding features are obtained from the pre-trained Word2Vec and Speech2Vec, respectively. Therefore, the proposed model can use any kind of textual data and speech data.

3. EXPERIMENTS

3.1. Datasets

Our experiments are conducted on English IWSLT datasets which contain TED talks. The datasets are reorganized by Che et al. [9]. There are 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 about 2.1M words, 144K sentences. The development set has about 296K words, 21K sentences. There are two test sets: reference (Ref.) and ASR, which are from the IWSLT2011. The test set contains about 13K words, 860 sentences. The datasets contain three kinds of punctuation marks (COMMA, PERIOD and QUESTION) and a non-punctuation mark "O". More details of the datasets can be found in [9].

3.2. Experimental setup

The self-attention based models are implemented with the TensorFlow toolkit [25].

The basic architecture of the models is shown in Fig.1. The encoder consists of a stack of $N = 6$ identical layers. The decoder is also composed of a stack of $N = 6$ identical layers. There are $h = 8$ parallel attention layers, or heads. For each of these heads, we use $d_k = d_v = d_{model}/h = 64$. Due to the reduced dimension of each head, the total computational cost is similar to that of single-head attention with full dimensionality. In order to use residual connections, d_{model} is set to 512. The positional encodings have the same dimension d_{model} as the embeddings layers, so that the two can be sumed. The dimensionality of the inner-layer is 2048.

In order to compare with other models in [10, 11, 12], we choose pre-trained word vectors from the Glove [1] to obtain word embeddings. The GloVe.6B.50d vector has 50 dimensions. The pre-trained speech vectors from the Speech2Vec [2] are used to get speech embeddings. Motivated by the work in [23], the 50.vec vectors are used to obtain speech embedding features, which has 50 dimensions. The Adam algorithm [26] with gradient clipping and warmup is used for optimization. The initial learning rate is 0.0001. The learning rate is varied over the course of training, according to the formula in [19]. The $warmup\_steps$ is set to 4000. During training, label smoothing of value $\epsilon = 0.1$ is employed, and the $batch\_size$ is 32. The learning rate is exponentially decayed during training. The rate of dropout is set to 0.1 for all the self-attention based models.

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). More details of metrics can be found in [9].

3.3. Results

A series of self-attention based models are trained to predict punctuation marks in our experiments.

Self-attention: the model is trained using the input embeddings learned similarly to the usual sequence transduction models in [19].

Self-attention-word: the model is trained using the word embeddings obtained from the pre-trained Word2Vec Glove.

Self-attention-speech: the model is trained using the speech embeddings obtained from the pre-trained Speech2Vec.

Self-attention-word-speech: the model is trained using the word embeddings and speech embeddings obtained from the pre-trained Word2Vec Glove and Speech2Vec, respectively. The word and speech embeddings are summed.

The results of our models on Ref. and ASR test sets are reported in the last four rows in Table 1 and 2, respectively. In Table 1 and 2, "Overall" denotes three kinds of punctuation marks. The results show that the Self-attention model obtains the worst results among our models. The Self-attention-speech model outperforms the Self-attention-word model. The Self-attention-word-speech model obtains the best performance.

We also compare our models with previous models. The previous results are listed in Table 1 and 2, respectively. T-LSTM and T-BRNN-pre are the best attention model in [10] and [11], respectively. BLSTM-CRF is the best single model in [12]. Teacher-Ensemble is the best ensemble model in [12]. All our models outperform the previous

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

[2] https://github.com/iamyuanchung/speech2vec-pretrained-vectors/tree/master/speech2vec

Table 1. The results of the models in terms of $P(\%)$,$R(\%)$, $F_1(\%)$ on the Ref. test set.

Model	COMMA			PERIOD			QUESTION			Overall		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
T-LSTM [10]	49.6	41.4	45.1	60.2	53.4	56.6	57.1	43.5	49.4	55.0	47.2	50.8
T-BRNN-pre [11]	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 [12]	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 [12]	66.2	59.9	62.9	75.1	73.7	74.4	72.3	63.8	67.8	71.2	65.8	68.4
Self-attention	64.1	59.3	61.6	78.1	73.7	75.8	74.3	63.2	68.3	72.2	65.4	68.6
Self-attention-word	64.9	58.7	61.6	79.1	74.6	76.8	75.4	64.9	69.8	73.1	66.1	69.4
Self-attention-speech	66.9	60.5	63.5	80.1	75.2	77.6	79.1	68.8	73.6	75.4	68.2	71.6
Self-attention-word-speech	**67.4**	**61.1**	**64.1**	**82.5**	**77.4**	**79.9**	**80.1**	**70.2**	**74.8**	**76.7**	**69.6**	**72.9**

Table 2. The results of the models in terms of $P(\%)$,$R(\%)$, $F_1(\%)$ on the ASR test set.

Model	COMMA			PERIOD			QUESTION			Overall		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
T-LSTM [10]	41.8	37.8	39.7	56.4	49.3	52.6	55.6	42.9	48.4	49.1	43.6	46.2
T-BRNN-pre [11]	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 [12]	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 [12]	60.6	58.3	59.4	71.7	72.9	72.3	66.2	55.8	60.6	66.2	62.3	64.1
Self-attention	60.7	57.8	59.2	71.1	72.1	71.6	66.8	58.9	62.6	66.2	62.9	64.5
Self-attention-word	61.5	57.2	59.3	72.1	73.0	72.5	67.9	60.6	64.0	67.2	63.6	65.3
Self-attention-speech	63.5	59.0	61.2	73.1	73.6	73.3	71.6	64.5	67.9	69.4	65.7	67.5
Self-attention-word-speech	**64.0**	**59.6**	**61.7**	**75.5**	**75.8**	**75.6**	**72.6**	**65.9**	**69.1**	**70.7**	**67.1**	**68.8**

state-of-art models. When comparing our Self-attention-word-speech model with the previous state-of-the-art single model BLSTM-CRF in [12], the overall F_1-score improves absolutely by 7.8% and 7.3% on Ref. and ASR test set, respectively. Our Self-attention-word-speech model also outperforms the best ensemble model Teacher-Ensemble by 4.5% and 4.7% absolute overall F_1-score on Ref. and ASR test set, respectively.

3.4. Discussions

The above experimental results show that the proposed method is effective. All self-attention based models outperform the previous state-of-art models on English IWSLT datasets. The possible reason is that the self-attention mechanism can draw global dependencies between input and output. The combination of word and speech embedding features outperforms any of the single embedding. This is because that the model can not only obtain lexical information but also utilize acoustic features from the combination. The speech embedding features make more contribution to the self-attention model than the word embedding features. The main reason is that the speech embeddings carries acoustic information in speech that does not exist in plain text. The speech embeddings are learned from an audio segment. So the embeddings may contain pauses, pitch and intonation information which are useful for predicting punctuation marks. In summary, all our models benefit from self-attention mechanism, word and speech embedding features.

4. CONCLUSIONS

Self-attention mechanism is used to predict punctuation marks. The self-attention based model is trained using word and speech embedding features from the pre-trained Word2Vec and Speech2Vec. The model can learn lexical and acoustic features using any kind of text data without corresponding audio and speech data without corresponding text. The experimental results on the English IWSLT2011 datasets show that the proposed method is effective. The self-attention based model trained using word and speech embedding features outperforms the previous state-of-the-art single model. It also obtains performance improvement compared to the previous best ensemble model. Future work includes applying the proposed method to Chinese datasets and other punctuation marks, such as the exclamation mark.

5. ACKNOWLEDGMENTS

This work is supported by the National Key Research & Development Plan of China (No.2017YFC0820602) and the National Natural Science Foundation of China (NSFC) (No.61425017, No.61831022, No.61773379, No.61771472), and Inria-CAS Joint Research Project (No.173211KYSB20170061)

6. REFERENCES

[1] E. Liu, Y. nd Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. Harper, "Enriching speech recognition with automatic detection of sentence boundaries and disfluencies," *IEEE Trans Audio Speech Language Process*, vol. 14, no. 5, pp. 1526–1540, 2006.

[2] D. Beeferman, A. Berger, and J. Lafferty, "Cyberpunc: a lightweight punctuation annotation system for speech," in *ICASSP*, 1998, pp. 689–692 vol.2.

[3] A. Gravano, M. Jansche, and M. Bacchiani, "Restoring punctuation and capitalization in transcribed speech," in *ICASSP*, 2009, pp. 4741–4744.

[4] Heidi Christensen, Yoshihiko Gotoh, and Steve Renals, "Punctuation annotation using statistical prosody models," *Proc Isca Workshop on Prosody in Speech Recognition and Understanding*, pp. 35–40, 2001.

[5] N. Ueffing, M. Bisani, and P. Vozila, "Improved models for automatic punctuation prediction for spoken and written text," in *INTERSPEECH*, 2013, pp. 3097–3101.

[6] P. elasko, P. Szymaski, J. Mizgajski, A. Szymczak, Y. Carmiel, and N. Dehak, "Punctuation prediction model for conversational speech," in *INTERSPEECH*, 2018, pp. 2633–2637.

[7] M. Hasan, "Noise-matched training of crf based sentence end detection models," in *INTERSPEECH*, 2015, pp. 349–353.

[8] W. Lu and H.T. Ng, "Better punctuation prediction with dynamic conditional random fields.," in *EMNLP*, 2010, pp. 177–186.

[9] X. Che, C. Wang, H. Yang, and C. Meinel, "Punctuation prediction for unsegmented transcript based on word vector," in *LREC*, 2016, pp. 654–658.

[10] O. Tilk and T. Alumae, "Lstm for punctuation restoration in speech transcripts," in *INTERSPEECH*, 2015, pp. 683–687.

[11] O. Tilk and T. Alumae, "Bidirectional recurrent neural network with attention mechanism for punctuation restoration," in *INTERSPEECH*, 2016, pp. 3047–3051.

[12] J. Yi, J. Tao, Z.i Wen, and Y. Li, "Distilling knowledge from an ensemble of models for punctuation prediction," in *INTERSPEECH*, 2017, pp. 2779–2783.

[13] J. Driesen, A. Birch, S. Grimsey, S. Safarfashandi, J. Gauthier, M. Simpson, and S. Renals, "Automated production of true-cased punctuated subtitles for weather and news broadcasts," pp. 2146–2147, 2014.

[14] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modeling punctuation prediction as machine translation," in *International Workshop on Spoken Language Translation*, 2011, pp. 238–245.

[15] E. Cho, J. Niehues, and A. Waibel, "Segmentation and punctuation prediction in speech language translation using a monolingual translation system," pp. 252–259, 2012.

[16] O. Klejch, P. Bell, and S. Renals, "Punctuated transcription of multi-genre broadcasts using acoustic and lexical approaches," in *Spoken Language Technology Workshop*, 2016, pp. 433–440.

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

[18] P. Bell and M. et al. Gales, "The mgb challenge: Evaluating multi-genre broadcast media recognition," in *Automatic Speech Recognition & Understanding*, 2015.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[20] J. Kim and P. C Woodland, "A combined punctuation generation and speech recognition system and its performance enhancement using prosody," *Speech Communication*, vol. 41, no. 4, pp. 563–577, 2003.

[21] J. Kol and L. Lamel, "Development and evaluation of automatic punctuation for french and english speech-to-text," in *INTERSPEECH*, 2012, pp. 1376–1379.

[22] X. Che, S. Luo, H. Yang, and C. Meinel, "Sentence boundary detection based on parallel lexical and acoustic models," in *INTERSPEECH*, 2016, pp. 2528–2532.

[23] C. Yu-An and J. Glass, "Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech," in *INTERSPEECH*, 2018, pp. 811–815.

[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Computer Science*, 2013.

[25] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and M. Isard, "Tensorflow: a system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation*, 2016, pp. 265–283.

[26] D. Kingma and Ba J., "Adam: A method for stochastic optimization," in *ICLR*, 2015.