The 3rd International Symposium on Linguistic Patterns in Spontaneous Speech (LPSS2019)
Institute of Linguistics, Academia Sinica, Taipei
November 21-22, 2019

Automatic Detection of Fillers in Mandarin Conversational Speech

Yeh-Sheng Lin[1,2], *Hen-Hsen Huang*[3], *Shu-Chuan Tseng*[2], *Hsin-Hsi Chen*[1,4]

[1]Department of Computer Science and Information Engineering,
National Taiwan University, Taiwan
[2]Institute of Linguistics, Academia Sinica, Taiwan
[3]Department of Computer Science, National Chengchi University, Taiwan
[4]MOST Joint Research Center for AI Technology and All Vista Healthcare, Taiwan

yehsheng@gate.sinica.edu.tw, hhhuang@nccu.edu.tw, tsengsc@gate.sinica.edu.tw,
hhchen@ntu.edu.tw

Abstract

Fillers are related to discourse planning, signaling interactive speech acts and turn management. This article presents experiments on automatic detection of fillers which are the most common nonverbal cues in spontaneous conversation. We propose an attention-based long-short term memory (LSTM) neural network and modern Universal Transformer model to detect fillers in frame level. Experimental results show that F1 scores up to 0.81 in test set for fillers can be achieved. Additionally, the character error rate (CER) in the automatic speech recognition experiment (ASR) decreases given that filler occurrences are eliminated in speech signal and both automatic detection output results and golden scripts information are used for the experiment.

Index Terms: filler detection, speech recognition, sequence labeling

1. Introduction

Fillers [1] are often used by the speakers for expressing hesitation, uncertainty, or simply keeping the floor in conversation, unlike ordinary spoken words that usually signify lexical meaning. Detection of fillers is an important task in decoding semantic interpretation of conversations (e.g. [2-5]). They are often signals for hesitation of the speaker and can be used to help detect the flow of conversation [6].

In recent years, neural network based approaches such as Deep Neural Network (DNN) [7], Convolutional Neural Network (CNN) [8], and Long-Short Term Memory (LSTM) [9, 10] have been applied to filler detection, in which they outperform other conventional machine learning approaches. In particular, LSTM has been widely used in time series forecasting, often called sequence labeling or classification task.

LSTM is a specific recurrent neural network (RNN) architecture. Simple RNN has the ability to process contextual information from past inputs (and future inputs in the bidirectional RNNs) [11]. However, the vanishing and the exploding gradient problems limit the capability of long sequence process in RNN [12]. LSTM is a redesigned RNN architecture that addresses vanishing gradient problems around special 'memory cell' units [13]. In addition, LSTM with attention mechanism has been proved to be more powerful for sequence to sequence prediction problems [14].

However, sequential computation of LSTM prevents parallelization across elements of the input sequence. Therefore, the Universal Transformer [15], a modern sequence-to-sequence attention-based model depending entirely on self-attention without using CNN or RNN, achieves state-of-the-art results on a wide range of sequence modeling tasks. This model is a generalization of the Transformer model [16] that extends its theoretical capabilities. The Transformer model is more parallelizable and it requires significantly less time for training. In this study, we use vanilla LSTM, attentive LSTM, and Universal Transformer for the task of detecting fillers in a Mandarin conversational speech corpus.

We used the Sinica Mandarin Conversational Dialogue Corpus (Sinica MCDC8) [17] for the present study. The corpus contains eight free conversations with sophisticated lexical transcriptions and manually edited word boundary information. Paralinguistic sounds (Para) and discourse-related items such as long pauses, laughter, fillers, particles, and discourse marker are annotated in the transcripts. Examples of paralinguistic sounds and discourse-related items occurred in the Sinica MCD8 are shown in Table 1.

Table 1: Paralinguistic and discourse-related items

Para	Filler	Particle	Marker	
(breathe)	UHN	A	NA	
(clear_throat)	UHNN	BA	NE	(那)
(cough)	UHM	E	NEI	
(exhale)	UHMM	EP	NA GE	(那個)
(inhale)	NHN	EN	SHE ME	(什麼)
(laugh)	NHNN	LA	SHEN ME	
(pause)	MHM	WA	ZHE	(這)
(silence)	MHMM	YA	ZHE GE	(這個)

Mandarin has a relative large number of utterance-final discourse particles. Some of them share similar phonetic representation with fillers that are often used in other languages, e.g., *uh* and *uhn* in English. For instance, ***EN*** in Table 2 can appear in the utterance-initial and -final positions. The variant ***MHM*** with a bilabial nasal onset and coda used within the utterance is more close to conventional filler definition. In our study, we include feedback uses in our filler data. Table 3 is an excerpt of the filler ***MHMHM*** as a feedback. A sequential labeling model was proposed to detect discourse markers (the right column of Table 1) in the Sinica MCDC8 by employing both acoustic and word information [18]. Different from the previous work, the objective of this study is to automatically detect fillers by only using acoustic information from the signal.

Table 2: Different locations of fillers.

Location	Utterance
Utterance-initial	***EN*** 陽明山 那邊 也 有 露天的 游泳池 ***EN*** Yang-Ming-Mountain there also have open-air swimming-pool (There is also an open air swimming pool on the Yang-Ming Mountain.)
Utterance-medial	因為 有 一些 女孩子 在 花費上 ***MHM*** 不是 會 很 節制 Because have some girls in spending ***MHM*** NEGATION can very self-control (Because some girls are not very self-control in spending.)
Utterance-final	他們 本來 都 在 台北 後來 就 ***EN*** They originally all live Taipei later then ***EN*** (They were all live in Taipei)

Table 3: Example of filler use in conversation

Speaker	Turn
A	譬如 說 你 買到 電影 票 比較慢 的話 For-example say you buy movie ticket slower if (For example, if you need to wait for movie to start.)
B	***MHMHM***
A	你 可以 在 旁邊 逛一逛 You can in nearby walk-around (You can go shopping nearby.)
B	對 對 還有 對面 還有 新光三越 百貨 Yes yes still-more opposite still-more Shin-Kong-Mitsukoshi Department-Store (That's right! There is also the Shin Kong Mitsukoshi Department Store across the street.)

The structure of this paper is as follows: Section 2 describes the detection method in detail. Section 3 describes the experimental setup. Then we present and analyze our experimental results. Finally, we draw some conclusions and discuss further applications.

2. Methods

In this study, we consider the detection task as a time series sequence labeling. Each frame in the utterance is represented in time steps. We use the filler detection result of the vanilla LSTM as our baseline. Then, we apply attention-based LSTM and modern Universal Transformer and compare the results.

2.1. Input Feature

Mel-frequency cepstral coefficients (MFCCs) are widely used for speech processing tasks [19]. This feature extraction mimics the human auditory system. The advantage of the MFCCs is that it captures main characteristics of human speech with a relatively low complexity. In our experiment, we use the MFCCs as our speech feature set. The feature extraction process has been implemented in Python using the librosa library [20]. The length of frame is 32 milliseconds.

2.2. Attentive LSTM

LSTM may use forget/input/output gate and cell memory to memorize time dependent information when processing sequential data. While applying LSTM to the encoder-decoder architecture for a sequence to sequence model, the input sequence is compressed in a fixed-length internal representation (context vector). This design limits the performance for long input sequences. Thus, attention mechanism [14] is purposed to address this problem by paying selective attention to the inputs and relate them to items in the output sequence.

2.3. Universal Transformer

The architecture of the Universal Transformer is shown in Figure 1. It is based on the encoder-decoder architecture commonly used in sequence-to-sequence models. The recurrent encoder block and decoder block are responsible for input and output sequences, respectively. Each block will iteratively compute T steps. After T steps, the output of the encoder is an intermediary vector for multi-head attention layer in the decoder, and the decoder computes output probabilities by Softmax layer.

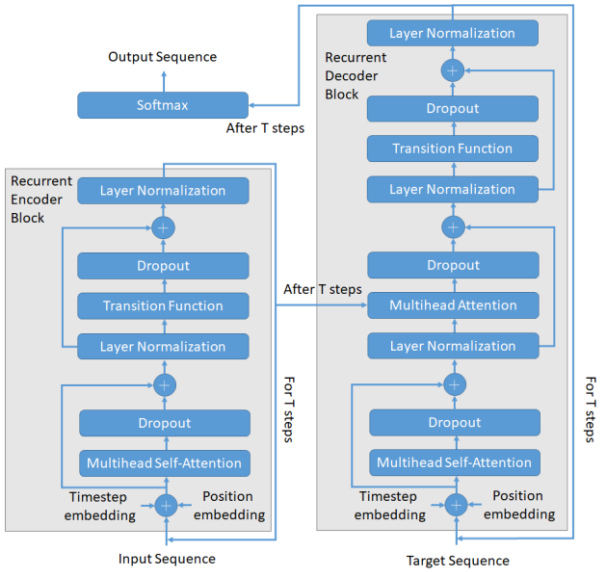

Figure 1: Architecture of Universal Transformer

The two-dimensional coordinate embedding consists of position and time-step embedding. In order to make use of the order of the sequence, positional encoding is added to input sequence. Universal Transformer employs an Adaptive Computation Time (ACT) [21] mechanism to implement dynamic halting by using time-step embedding to modulate the number of computational steps needed for processing.

Universal Transformer depends entirely on self-attention without using CNN or RNN. Multihead Self-Attention enables the model to jointly attend to information from different representation subspaces at different positions of the same input sequence. Scaled dot-product attention which combines queries Q, keys K and values V is adopted as basic attention function as follows

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d}}\right)V, \quad (1)$$

where d is the dimension of queries, keys and values. Multi-Head Self-Attention with k heads is purposed in [16],

$$MultiHeadSelfAttention\left(H^t\right)$$

$$= Concat(head_1, \dots, head_k)\, W^O \quad (2)$$

$$\text{where } head_i = Attention\left(Q W_i^Q, K W_i^K, V W_i^V\right) \quad (3)$$

where the state H^t is map to queries, keys and values with affine projections using matrices $W^Q \in \mathbb{R}^{d\times d/k}$, $W^K \in \mathbb{R}^{d\times d/k}$, $W^V \in \mathbb{R}^{d\times d/k}$, $W^O \in \mathbb{R}^{d\times d}$.

Layer normalization [22] is a normalization method that computes the mean and variance from all of the summed inputs to the neurons in a layer to improve the training speed for various neural network models. Dropout [23] layer provides a simple regularization method to avoid overfitting by randomly dropping out nodes during training. Transition function is applied to fully-connected layer with shared weights across position and time. Because the recurrent transition function can be applied an arbitrary number of times, Universal Transformer can have variable depth, while the standard Transformer only comprises of fixed stack of Transformer blocks.

3. Experimental Setup

3.1. Sinica MCDC8

Our experiment was conducted by using speech data of the Sinica MCDC8. The conversation partners (9 females and 7 males, aged between 16 and 46) met each other for the first time. Speakers were free to choose and change topics during their conversation in about one hour each.

The conversations in the Sinica MCDC8 were segmented into speaker turns. The speech content was orthographically transcribed with annotations of discourse-related items and paralinguistic sounds. There are four different sub-groups of filler variants, as shown in Table 4, each with/o nasal onset and coda. Multi-syllabic fillers are transcribed with repeated 'H', corresponding to the number of syllables. Table 4 also lists the group and the numbers of fillers in the Sinica MCDC8. As mentioned above, we also include filler-like particles, listed in Table 5 in our experiments.

Table 4: Occurrences *of fillers in the Sinica MCDC8*

Group1	#	Group2	#	Group3	#	Group4	#
UHN	74	UHM	1	NHN	161	MHM	836
UHNN	88	UHMM	4	NHNN	49	MHMM	326
UHNHN	65	UHMHM	2	NHNHN	188	MHMHM	453
						MHMHMHM	114
						MHMHMHMHM	29

Table 5: Filler-like *Particles.*

Particles	#
E	206
EP	3
EN	249
EIN	95
EI	419

The audio files of the Sinica MCDC8 are segmented into speaker turns, with information about Inter-Pause Units (IPU) boundaries, separated by silent pauses and paralinguistic sounds. The IPUs are regarded as the utterance unit in this study. There are 16,423 utterances, where 20% of the overall utterances are used as testing data and the remaining 80% of the data are used as training data.

3.2. Evaluation

One of the main challenges of the experiments is that the classes are unbalanced. Fillers account for less than 6% of the overall utterances. Therefore, we use precision, recall, and F1 score (F1) to evaluate the performance of filler detection.

3.3. Observation the influence of fillers on ASR

As we are also interested in the influence of fillers on ASR performance, we eliminated all filler occurrences in the Sinica MCDC8 by using the golden transcripts. The filler eliminated utterances were input to Google cloud speech API with the default setting for our ASR experiment. In addition, we also used Praat [24] to eliminate automatic detected filler and applied their ASR results to compare with the golden scripts. Figure 2 shows the design of our experiment.

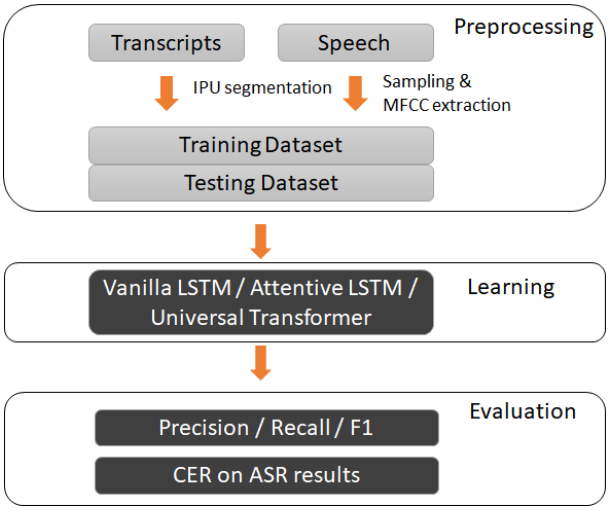

Figure 2: Experimental flow chart

4. Results

Table 6 shows the performance of filler detection with vanilla LSTM, attentive LSTM and Universal Transformer on testing data on frame level.

Table 6: Results of fillers detection on frame level

Model	Precision	Recall	F1
Vanilla LSTM	0.8239	0.6447	0.7234
Attentive LSTM	0.8468	0.7528	0.7970
Universal Transformer	0.8481	0.7758	0.8103

The results in Table 7 clearly reflect the impact of fillers on ASR performance. Fillers are noisy, especially when they occur in the middle of utterances. In addition, we also compared the results of fillers detection with vanilla LSTM, attentive LSTM and Universal Transformer. The results show the CER only slightly decreased on attentive LSTM and Universal Transformer while vanilla LSTM did not improve the performance at all.

Table 7: CER on filler-eliminated speech data (IPU)

Eliminating fillers by	Overall	Initial	Medial	Final
Golden Transcripts	29.86%	23.73%	36.38%	30.05%
Vanilla LSTM	31.94%	26.24%	38.20%	32.17%
Attentive LSTM	31.90%	26.21%	38.14%	31.17%
Universal Transformer	31.66%	26.29%	37.71%	31.72%
Original speech data	31.93%	26.73%	37.95%	31.47%

5. Conclusions and future work

Experiments on automatic filler detection were conducted to Mandarin conversational speech by using the attention based LSTM and modern Universal Transformer. The preliminary results reported in this paper will be further elaborated to develop efficient models that can be used to automatically detect fillers from spontaneous speech flow. Applications can be accordingly implemented to detect hesitation event and discover useful cues of memory recall processes in conversational speech.

6. Acknowledgments

The data collection and processing projects were financially supported by the Ministry of Science and Technology of Taiwan, under grant MOST 106-2410-H-001 -045 -MY2, granted to the third author.

7. References

[1] H. H Clark and J. Fox Tree, "Using uh and um in spontaneous dialog," *Cognition*, 2002, vol. 84, pp. 73-111.

[2] L. Tóth, G. Gosztolya, V. Vincze, I. Hoffmann, G. Szatlóczki, E. Biró, F. Zsura, M. Pákáski, and J. Kálmán. "Automatic detection of mild cognitive impairment from spontaneous speech using ASR," in *Proceedings of Interspeech*, 2015, pp. 2694-2698.

[3] H. Salamin, A. Polychroniou, and A. Vinciarelli. "Automatic detection of laughter and fillers in spontaneous mobile phone conversations," in *Proceedings of 2013 IEEE International Conference on Systems, Man, and Cybernetics*, 2013, pp. 4282-4287.

[4] I. Hoffmann, D. Nemeth, C.D. Dye, M. Pákáski, T. Irinyi, and J. Kálmán, "Temporal parameters of spontaneous speech in Alzheimer's disease," *International Journal of Speech-Language Pathology*, 2010, vol. 12(1), pp. 29-34.

[5] G. Gosztolya. "Optimized Time Series Filters for Detecting Laughter and Filler Events," in *Proceedings of INTERSPEECH*, 2017, pp. 2376-2380.

[6] E.E. Shriberg, "Preliminaries to a theory of speech disfluencies," Phd dissertation, 1994, University of California, Berkeley.

[7] R. Gupta, K. Audhkhasi, S. Lee, and S. Narayanan. "Speech paralinguistic event detection using probabilistic time-series smoothing and masking," in *Proceedings of INTERSPEECH*, 2013, pp. 173-177.

[8] L. Kaushik, A. Sangwan, and J. Hansen. "Laughter and Filler Detection in Naturalistic Audio," in *Proceedings of INTERSPEECH*, 2015, pp. 2509-2513.

[9] H. Inaguma, K. Inoue, M. Mimura, and T. Kawahara. "Social Signal Detection in Spontaneous Dialogue Using Bidirectional LSTM-CTC," in *Proceedings of INTERSPEECH*, 2017, pp. 1691-1695.

[10] R. Brueckner and B. Schulter. "Social signal classification using deep blstm recurrent neural networks," in *Proceedings of 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4823-4827.

[11] A. Graves, "Supervised sequence labelling with recurrent neural networks". Studies in Computational Intelligence. 2012: Springer-Verlag Berlin Heidelberg.

[12] R. Pascanu, T. Mikolov, and Y. Bengio. "On the difficulty of training recurrent neural networks," in *Proceedings of International conference on machine learning*, 2013, pp. 1310-1318.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, 1997, vol. 9(8), pp. 1735-1780.

[14] D. Bahdanau, K. Cho, and Y. Bengio "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[15] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and Ł. Kaiser "Universal Transformers," *arXiv e-prints arXiv:1807.03819*, 2018.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin "Attention Is All You Need," *arXiv:1706.03762*, 2017.

[17] S.-C. Tseng, "Lexical Coverage in Taiwan Mandarin Conversation," *International Journal of Computational Linguistics and Chinese Language Processing*, 2013, vol. 18(1), pp. 1-18.

[18] Y.-W. Wang, H.-H. Huang, K.-Y. Chen, and H.-H. Chen. "Discourse Marker Detection for Hesitation Events on Mandarin Conversation," in *Proceedings of INTERSPEECH*, 2018, pp. 1721-1725.

[19] S. Narang, M.D.J.I.J.o.C.S. Gupta, and M. Computing, "Speech feature extraction techniques: a review," 2015, vol. 4(3), pp. 107-114.

[20] B. McFee, C. Raffel, D. Liang, D.P. Ellis, M. McVicar, E. Battenberg, and O. Nieto. "librosa: Audio and music signal analysis in python," in *Proceedings of Proceedings of the 14th python in science conference*, 2015, pp. 18-25.

[21] A. Graves "Adaptive Computation Time for Recurrent Neural Networks," *arXiv e-prints arXiv:1603.08983*, 2016.

[22] J. Lei Ba, J.R. Kiros, and G.E. Hinton "Layer Normalization," *arXiv e-prints arXiv:1607.06450*, 2016.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, 2014, vol. 15(1), pp. 1929-1958.

[24] P. Boersma and D. Weenink. *Praat: doing phonetics by computer [Computer program]. Version 6.1.03, retrieved 1 September 2019 from http://www.praat.org/*.