INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Combining CNN and BLSTM to Extract Textual and Acoustic Features for Recognizing Stances in Mandarin Ideological Debate Competition

Linchuan Li[1,2], *Zhiyong Wu*[1,2,3], *Mingxing Xu*[1,2], *Helen Meng*[1,3], *Lianhong Cai*[1,2]

[1] Tsinghua-CUHK Joint Research Center for Media Sciences, Technologies and Systems,
Shenzhen Key Laboratory of Information Science and Technology,
Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China
[2] Tsinghua National Laboratory for Information Science and Technology (TNList),
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[3] Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong SAR, China

lilinchuan318@gmail.com, zywu@sz.tsinghua.edu.cn, xumx@tsinghua.edu.cn,
hmmeng@se.cuhk.edu.hk, clh-dcs@tsinghua.edu.cn

Abstract

Recognizing stances in ideological debates is a relatively new and challenging problem in opinion mining. While previous work mainly focused on text modality, in this paper, we try to recognize stances from both text and acoustic modalities, where how to derive more representative textual and acoustic features still remains the research problem. Inspired by the promising performances of neural network models in natural language understanding and speech processing, we propose a unified framework named C-BLSTM by combining convolutional neural network (CNN) and bidirectional long short-term memory (BLSTM) recurrent neural network (RNN) for feature extraction. In C-BLSTM, CNN is utilized to extract higher-level local features of text (n-grams) and speech (emphasis, intonation), while BLSTM is used to extract bottleneck features for context-sensitive feature compression and target-related feature representation. Maximum entropy model is then used to recognize stances from the bimodal textual acoustic bottleneck features. Experiments on four debate datasets show C-BLSTM outperforms all challenging baseline methods, and specifically, acoustic intonation and emphasis features further improve F1-measure by 6% as compared to textual features only.

Index Terms: Stance recognition, intonation, emphasis, convolutional neural network (CNN), bidirectional long short-term memory (BLSTM)

1. Introduction

Understanding stance and opinion in debates can provide critical insight into the theoretical underpinnings of discourse, argumentation and sentiment. In ideological debate competition, affirmative and negative sides express their opinions towards the given resolution. In addition, the competition also contains remarks or comments from presenter and jury, which we attribute as the neutral side. The goal of stance recognition in debate is then to determine the side (i.e. affirmative, negative or neutral) one participant is taking.

Previous approaches mainly fell into two categories: text modality only and multimodal based methods. The former ones just extract various kinds of textual features to classify stances in online debate forums [1]-[4]. [1] and [2] consider opinion expressions and their targets to capture sentiment towards debate resolution. [5] partitions the debate posts based on the dialogue structure of the debate and assigns stance to a partition using lexical features of candidate posts. These approaches cannot make full use of the information from every word in a sentence. More advanced methods use bag-of-words model or word embedding to produce semantic sentence vector. However, such methods may not perform well without considering context dependencies. This problem is overcome by using recurrent neural networks (RNNs). In addition, convolutional neural networks (CNNs) have been demonstrated to achieve excellent results in extracting higher-level local features on sentence classification tasks [6].

For multimodal based methods, speech and text have been analyzed jointly for the purpose of opinion identification in [7]. [8] uses deep convolutional neural network (DCNN) and multiple kernel learning for utterance-level sentiment analysis. These approaches first extract textual and acoustic features respectively and then try to integrate them together to obtain better performance by taking the advantage of complementary relationships between textual and acoustic features. However, which features on earth are helpful for recognizing stances are still not well investigated.

This paper focuses on extracting more representative textual and acoustic features for the task of stance recognition with convolutional neural network (CNN) and bidirectional long short-term memory (BLSTM) recurrent neural network (RNN). In addition to textual features, we further investigate whether acoustic intonation and emphasis information is useful in recognizing stances. We propose a unified framework named C-BLSTM by combining CNN and BLSTM to extract textual, intonation and emphasis related bottleneck features for stance recognition. The architecture of our proposed method is shown in Figure 1. We first perform convolution to textual and acoustic features respectively to extract higher-level sequential features. The extracted higher-level textual or acoustic features are then fed into BLSTM recurrent neural networks to capture long and short-term dependencies of preceding and succeeding contexts. We set the last hidden layer of BLSTM to be bottleneck layer. Using bottleneck features offers the advantages in context-sensitive feature compression and target-related feature extraction. Maximum entropy (MaxEnt) method is finally used to recognize stances as affirmative, negative or neutral from textual, intonation and emphasis bottleneck features.

http://dx.doi.org/10.21437/Interspeech.2016-324

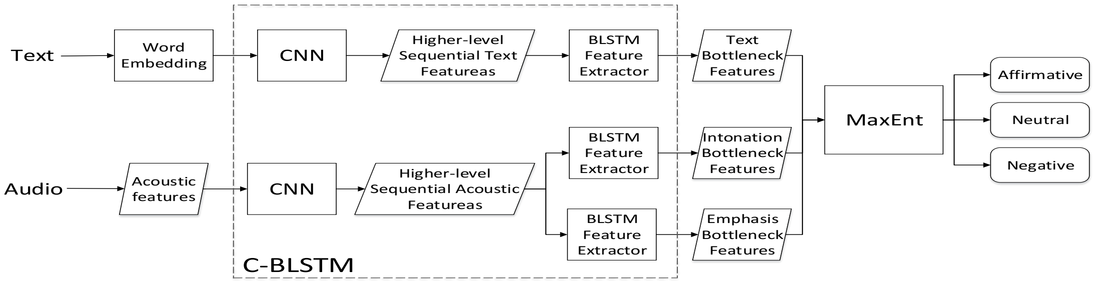

Figure 1: *The architecture of the proposed method*

2. Motivation

2.1. Considering acoustic intonation and emphasis

Much previous work has demonstrated that combining textual and acoustic features could yield better performance in stance recognition [7]-[9]. In this work, we further try to figure out which acoustic features are helpful. By observing ideological debate datasets, we discover two interesting phenomena. One is participants prefer employing rhetorical question to state their opinions in debates. In Mandarin, the same utterance spoken in question intonation may have totally different meaning opposite to the statement intonation. The other interesting phenomenon is speaker usually tends to emphasize the key words when taking a stance. Hence, in this work, we take intonation and emphasis information into account for stance recognition.

2.2. Combining CNN and BLSTM

CNN is able to learn local higher-level features from temporal or spatial data but lacks the ability of learning sequential correlations. While the RNN is specialized for sequential modeling. Combination of CNN and RNN in image caption [10] and speech recognition [11] tasks has already achieved excellent performance. Furthermore, in ideological debates, participants usually speak long, logical and complicated sentences, recognizing stances at current time step needs both preceding and succeeding contexts. BLSTM offers an elegant solution. In this work, we combine CNN and BLSTM to extract textual and acoustic bottleneck features for stance recognition.

3. Methods

As shown in Figure 1, our proposed method consists of two workflows for text and audio respectively. For textual modality, we transform words into *d*-dimensional vector representations by using word2vec [12]. For speech, we use openSMILE [13] to extract acoustic features of each frame. C-BLSTM framework is then used to further extract bottleneck features from word-embedding or acoustic features. C-BLSTM consists of two components: convolution layer and BLSTM layer.

3.1. Extracting higher-level features by convolution

At convolution layer of the above C-BLSTM, we follow the one-dimensional convolution method [14] that involves a filter vector sliding over a sequence and detecting features at different positions. In our method, we utilize C-BLSTM on both textual and acoustic features. An example of C-BLSTM architecture for text modality is illustrated in Figure 2, whose details are elaborated as follows. Let $\boldsymbol{s}_i \in \mathbb{R}^{1\times d}$ be the *d*-dimensional word vectors for the *i*-th word in a sentence, $\boldsymbol{s} \in \mathbb{R}^{L\times d}$ denote the sentence with L words, k be the length of filter vector $\boldsymbol{m} \in \mathbb{R}^{k\times d}$. To extract k-gram features, we design a window vector $\boldsymbol{w}_j \in \mathbb{R}^{k\times d}$ with k consecutive word vectors:

$$\boldsymbol{w}_j = [\boldsymbol{s}_j, \boldsymbol{s}_{j+1}, \ldots, \boldsymbol{s}_{j+k\text{-}1}],\ 1 \le j \le L\text{-}k+1 \tag{1}$$

The idea behind one-dimensional convolution is to take the element-wise multiplication of filter vector $\boldsymbol{m}$ with each window vector $\boldsymbol{w}_j$ in the sentence $\boldsymbol{s}$ to obtain a feature map $\mathbf{c} \in \mathbb{R}^{L\text{-}k+1}$, where each element c_j is produced as:

$$c_j = f(\boldsymbol{w}_j \circ \boldsymbol{m} + b), \tag{2}$$

where ∘ is element-wise multiplication, $b \in \mathbb{R}$ is bias term and f is nonlinear transformation function. In our case, we follow the work in [15] to choose ReLU as the nonlinear function. A feature map $\mathbf{c}$ is produced given a filter vector $\boldsymbol{m}$. We use multiple filter vectors to generate different feature maps and then concatenate them together to produce new features. Let n be the numbers of filter vectors, we have:

$$\mathbb{C} = [\boldsymbol{c}_1; \boldsymbol{c}_2; \ldots; \boldsymbol{c}_n]. \tag{3}$$

Semicolons represent column concatenation and $\boldsymbol{c}_i$ is the feature map generated by the *i*-th filter. Each row of $\mathbb{C} \in \mathbb{R}^{(L\text{-}k+1)\times n}$ is the new higher-level feature representation for the k-grams at each position. Dynamic k-max pooling is often applied to features maps after the convolution to select the k-most important features. However, BLSTM is specified for modeling sequential input, pooling will break such sequential organization. Hence, we feed the outputs of convolution layer into BLSTM directly without applying pooling.

Similarly, we perform convolution on acoustic features to obtain acoustic higher-level features which can capture short and long-range relations. Different from textual modality, $\boldsymbol{s}_i$ denotes the acoustic features (F0, MFCCs, etc.) of the *i*-th frame instead of word vectors.

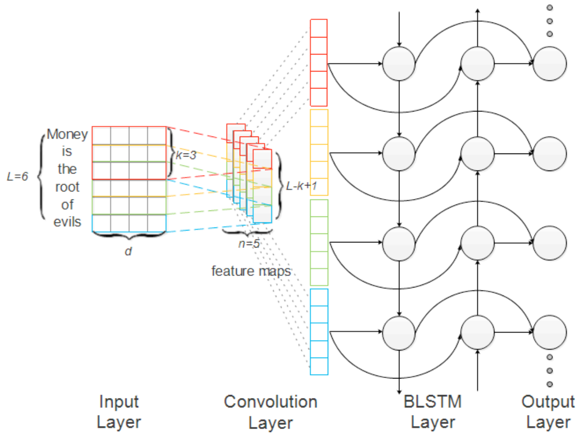

Figure 2: *An example of C-BLSTM for text modality. Blocks of the same color in feature maps correspond to higher-level k-gram features extracted at the same position.*

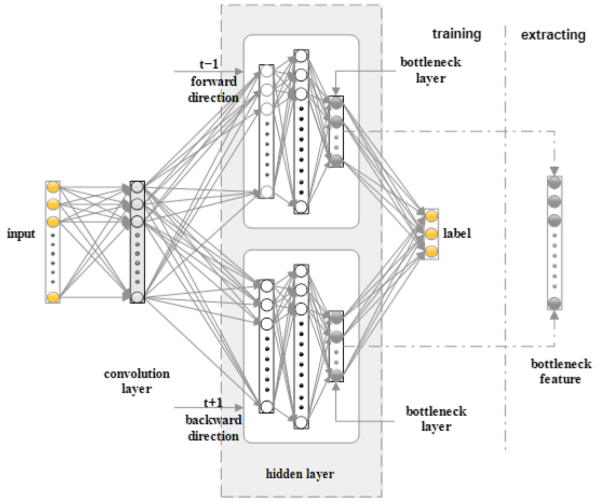

Figure 3: *The architecture of C-BLSTM bottleneck feature extractor.*

3.2. Extracting bottleneck features with BLSTM

Given that BLSTM is able to retrieve long short-term contexts of inputs to the current output in both forward and backward directions, we propose to use BLSTM recurrent neural network to extract bottleneck features. Figure 3 illustrates the network structure in detail.

Take text modality as example, during the training stage, we train the BLSTM as a stance recognition classifier. The word embedding features (i.e. word vectors as described in 3.1) serve as the input and are converted to higher-level textual features through convolution layer. The higher-level features are then sent to hidden layers of BLSTM. The stance labels (affirmative, negative or neutral) serve as the output. Three hidden layers are used for both forward and backward directions. Following previous work [16], we use the last hidden layer as the bottleneck layer. During the extracting stage, the activations of the output layer (i.e. stance labels) are ignored, as we focus on the output of bottleneck layers only. The final bottleneck feature is generated by combining the outputs of the two directional bottleneck layers. As BLSTM generates outputs of bottleneck layers by taking into account both target stance labels and context information, extracting bottleneck features this way offers the advantages in context-sensitive feature compression and target-related feature extraction.

Similarly, for extracting intonation bottleneck features, the input of C-BLSTM is acoustic features and the output label of C-BLSTM is 1 or 0, which denotes whether current utterance is spoken in question intonation or not.

Different from labeling stance or intonation at utterance level, acoustic emphasis is the local prominence and should be determined at frame-level. Hence, to extract emphasis related bottleneck features, the input of C-BLSTM is acoustic features while the output is binary valued frame-wise label indicating if current frame is acoustically emphasized.

4. Experiments

4.1. Data description and preprocessing

The datasets are composed of 4 debate competitions available online. We download the debate videos of International Varsity Debate. Unlike other stances recognition work, our debates are triple-sided. Besides affirmative and negative sides, we attribute presenter and jury to neutral side. There are 10 participants in each debate, 4 of affirmative, 4 of negative, 1 presenter and 1 jury. The resolutions of the 4 debates are “Whether money is the root of evils,” “Human nature is good or evil,” “Whether is starting business more good than harm to college students,” and “Difficult to know and easy to do or difficult to do and easy to know” respectively. The duration of each debate amounts about 52 minutes.

For acoustic data, we first extract audio streams from the downloaded video. Since the alternate statement of each side, we further segment the audio into pieces, each of which is an utterance spoken by only one speaker. Meanwhile, we annotate the segmented utterances to their corresponding stances. We also remove the noisy segments of applause and laugh. Finally we get 1,254 effective utterances in total. We randomly select 4/5 as training set and the rest as test set for each debate.

For text data, with the benefit from automatic speech recognition (ASR), speech to text is easy to do with the IBM Speech-to-Text interface [17]. The transcribed text scripts are further manually checked. Unlike English word, Chinese words don't have space between each other, so we divide the utterance into tokens using Jieba tokenizer [18].

Table 1. *3-class stance classification accuracy on test set of different methods with text modality only*

Methods	Accuracy
LSTM	0.816
C-LSTM	0.825
BLSTM	0.832
C-BLSTM	0.845

4.2. Feature extraction experiments

The purpose of our work is to extract more representative textual and acoustic features for stance recognition. Hence, we conduct a set of experiments to validate whether the proposed C-BLSTM framework is capable of extracting appropriate text, intonation and emphasis related bottleneck features.

4.2.1. Text bottleneck features extraction

For word embedding, we use a Chinese data corpus [19] to train word vectors with publicly available tool word2vec [12]. The dimension of word vectors is set to be 300.

For experimental setting, we use one convolutional layer and three BLSTM hidden layers. For hidden layers, the number of LSTM blocks is 64, 128 and 32 respectively. For convolution layer, [15] conducts a series of experiments and concludes that one convolutional layer with filter length 3 and filter number 150 can reach the best balance between performance and speed. We follow the same settings in our work. As the convolution layer requires fixed-length input, after examining the statistics of the dataset, the $maxlen$ of sentence is set to be 50 words. For sentences with fewer words, we align the input by padding zero vectors; but for those with length longer than $maxlen$ we simply cut extra words. We utilize dropout technique to prevent over-fitting and dropout rate is set to be 0.5.

To investigate in what sense convolutional and bidirectional structures can benefit feature extraction, we also implement 3 additional models including LSTM, BLSTM and C-LSTM for comparison. For BLSTM, the number of hidden layers and the number of LSTM blocks in each layer are set to be the same with C-BLSTM. For LSTM and C-LSTM, the same three hidden layers are used while the number of LSTM blocks are doubled to match the same scale. For all the models, maximum entropy (MaxEnt) classifier is used to recognize stance from bottleneck features extracted by different models. Table 1 shows that C-BLSTM outperforms all the other neural networks, which demonstrate C-BLSTM takes advantages of extracting higher-level local features (n-gram) of CNN and handling sequential long short-term context dependencies of RNN. As for C-BLSTM, we extract the 1×64 dimensional text related bottleneck features for each sentence.

4.2.2. Intonation bottleneck features extraction

For acoustic features, we follow [20] to extract acoustic features including F0 (fundamental frequency), ZCR (zero-crossing rate), MFCCs (Mel-frequency cepstral coefficients), etc. Finally, we obtain 130 acoustic features for each frame with frame length of 20ms and frame shift of 10ms.

For intonation feature extraction, we adopt the same network settings as in text. For convolution layer, since the number of acoustic frames of an utterance is much larger than that of words, we set filter length to be 10. Furthermore, the convolution layer requires fixed-length input, we intercept fixed-length speech segments from original speech utterances. It should be noted that question intonation is mostly carried by the end part of a speech utterance, hence the speech segments are cut from the rear end. We further conduct intonation recognition experiments to find the proper length of speech segments by testing the length of 3, 4 or 5 seconds. Intonation recognition experiments indicate that speech segments with 4 seconds length can best determine the acoustic intonation.

In the above way, each original utterance is represented by the rear end speech segment with 4 seconds length, from which 400×130 dimensional original acoustic features are extracted and serve as the input of C-BLSTM. To train bottleneck feature extractor, the output of C-BLSTM is “1” if the current utterance is with question intonation. We finally extract the 1×64 dimensional intonation bottleneck feature that better describes intonation characters with low dimensions.

4.2.3. Emphasis bottleneck features extraction

Emphasis is determined at frame-level. Each frame feature (1×130 dimensions) corresponds to a label that denotes if current frame is acoustically emphasized. We use emphasis labeled data [21] to train our bottleneck feature extractor. We have 108,448 emphasized frames and 105,749 non-emphasized frames. By using bottleneck features, we can achieve 80.6% in accuracy on the test set. Then we apply the pre-trained bottleneck feature extractor to our debate dataset and obtain the bottleneck feature for each frame. Because stance is recognized at sentence-level, we further compute the statistical results based on these frame features (including mean, variance, maximum and minimum) and apply them to all frame features of an utterance. The final extracted emphasis related bottleneck feature has the size of 4×64 for each utterance.

4.3. Stance recognition experiments

4.3.1. Unimodal experiment

We conduct experiments using each individual bottleneck features to investigate the stance recognition performance of unimodal features. The results are illustrated in Table 2. F1-measure can achieve 0.843 with text bottleneck features only, while intonation and emphasis merely reaches 0.477 and 0.498 individually. Experimental results indicate that textual modality plays the most important role in recognizing stances. It accords to our expectation because text bottleneck features are stance-related, while intonation and emphasis bottleneck features are independent of stance. However, the truth that they are both statistically over 0.333 (for 3 side stance recognition) suggests that intonation and emphasis bottleneck features should be helpful in recognizing stances.

Table 2. *3-class stance classification performance of unimodal bottleneck features in F1-measure*

Unimodal	F1-measure
Text	0.843
Intonation	0.477
Emphasis	0.498

4.3.2. Bimodal experiment

We further conduct experiments combining text, intonation and emphasis bottleneck features in different combination settings. F1-measures of stance classification results are shown in Table 3. Compared to text modality only, results indicate that both acoustic intonation and emphasis information are helpful in recognizing stances and can improve F1-measure by 3.4% and 4.4% respectively. Combining all bottleneck features achieves the best performance, where F1-measure has the improvement of 6% from textual modality only.

Table 3. *3-class stance classification performance of bimodal bottleneck features in F1-measure*

Bimodal	F1-measure
Text+Into	0.877
Text+Emp	0.887
Text+Into+Emp	0.903

5. Conclusions

In this paper, we study the problem of recognizing stances in ideological debate competitions and put focus on extracting more representative textual and acoustic features. We propose C-BLSTM for feature extraction by combining convolutional neural networks (CNNs) and bidirectional long short-term memory (BLSTM) recurrent neural networks. CNN is utilized to extract higher-level local features. BLSTM is used to extract bottleneck features for context-sensitive feature compression and target-related feature representation. For acoustic features, we extract bottleneck features of intonation and emphasis to investigate if these two factors will help in stance recognition. Experiments confirm that the proposed C-BLSTM outperforms all other baseline methods and intonation and emphasis can improve the stance recognition accuracy by 6% in F1-measure.

6. Acknowledgement

This work is supported by National Basic Research Program of China (2012CB316401), National High Technology Research and Development Program of China (2015AA016305), Natural Science Foundation of China (NSFC) (61375027, 61433018, 61370023 and 61171116), joint fund of NSFC-RGC (Research Grant Council of Hong Kong) (61531166002, N_CUHK404/15) and Major Program for National Social Science Foundation of China (13&ZD189).

7. References

[1] Somasundaran, Swapna, and Janyce Wiebe. "Recognizing stances in online debates." Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1. Association for Computational Linguistics, 2009.

[2] Somasundaran, Swapna, and Janyce Wiebe. "Recognizing stances in ideological on-line debates." Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text. Association for Computational Linguistics, 2010.

[3] Anand, Pranav, et al. "Cats rule and dogs drool!: Classifying stance in online debate." Proceedings of the 2nd workshop on computational approaches to subjectivity and sentiment analysis. Association for Computational Linguistics, 2011.

[4] Walker, Marilyn A., et al. "That is your evidence?: Classifying stance in online political debate." Decision Support Systems 53.4 (2012): 719-729.

[5] Walker, Marilyn A., et al. "Stance classification using dialogic properties of persuasion." Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2012.

[6] Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).

[7] Raaijmakers, Stephan, Khiet Truong, and Theresa Wilson. "Multimodal subjectivity analysis of multiparty conversation." Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2008.

[8] Poria, Soujanya, Erik Cambria, and Alexander Gelbukh. "Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis." Proceedings of EMNLP. 2015.

[9] Murray, Gabriel, and Giuseppe Carenini. "Detecting subjectivity in multiparty speech." INTERSPEECH. 2009.

[10] Xu, Kelvin, et al. "Show, attend and tell: Neural image caption generation with visual attention." arXiv preprint arXiv: 1502.03044 (2015).

[11] Sainath, Tara N., et al. "Convolutional, long short-term memory, fully connected deep neural networks." Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on. IEEE, 2015.

[12] Word2vec tools [OL]. [20160301]. http://code.google.com/p/ word2vec/

[13] Eyben, Florian, Martin Wöllmer, and Björn Schuller. "Opensmile: the munich versatile and fast open-source audio feature extractor." Proc. ACM, 2010.

[14] Kalchbrenner, Nal, Edward Grefenstette, and Phil Blunsom. "A convolutional neural network for modelling sentences." arXiv preprint arXiv:1404.2188 (2014).

[15] Zhou, Chunting, et al. "A C-LSTM Neural Network for Text Classification." arXiv preprint arXiv:1511.08630 (2015).

[16] Y. Dong, ML. Seltzer, "Improved Bottleneck Features Using Pretrained Deep Neural Networks." Proc. INTERSPEECH, 2011.

[17] Speech-to-Text interface [OL]. [2015-11-10]. http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/speech-to-text.html

[18] Jieba tokenizer tool [OL]. [2015-11-10]. Source code at: https://github.com/fxsjy/jieba

[19] http://www.datatang.com/datares/go.aspx?dataid=603542

[20] Y. Tang, Y. Huang, Z. Wu, H. Meng, M. Xu, L. Cai, "Question detection from acoustic features using recurrent neural network with gated recurrent unit," in Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016.

[21] Ning, Yishuang, et al. "Using Tilt for Automatic Emphasis Detection with Bayesian Networks." Proc. INTERSPEECH, 2015.