TMUDS at the NTCIR-15 DialEval-1 Task

Yen Chun Huang
Graduate Institute of Data Science,
Taipei Medical University, Taiwan
m946108006@tmu.edu.tw

Yi Hsuan Huang
Graduate Institute of Data Science,
Taipei Medical University, Taiwan
m946108002@tmu.edu.tw

Yu Ya Cheng
Professional Master Program in Data
Science, Taipei Medical University,
Taiwan
i906108009@tmu.edu.tw

Jung Yi Liao
Professional Master Program in Data
Science, Taipei Medical University,
Taiwan
i906108007@tmu.edu.tw

Yung Chun Chang
Graduate Institute of Data Science,
Taipei Medical University, Taiwan
changyc@tmu.edu.tw

ABSTRACT

In this paper, we present our approaches to the Nugget Detection (ND) subtask at the NTCIR-15 STC-3 task. The purpose of this subtask is to automatically identify the state of dialogue sentences in logs of a dialogue system. The proposed model integrates BERT Embeddings and BiLSTM through a concatenated attention mechanism. The results demonstrate that BERT Embeddings are effective in capturing the semantic relationship between pieces of the dialogue in the context. Therefore, our models are capable of surpassing two baseline models (i.e., BL-uniform and BL-popularity). In addition, according to our final evaluation results, the attention mechanism plays a crucial role in model optimization.

TEAM NAME

TMUDS

SUBTASKS

Nugget detection (Chinese)

KEYWORDS:

Nature Language processing, Short text conversation, Dialogue system, Nugget detection, Neural network, Long-short-term-memory, Attention mechanism

1 INTRODUCTION

In recent years, artificial intelligence (AI) and natural language processing (NLP) research are booming, and business organizations tend to improve the efficiency of customer service in the hope of increasing customer satisfaction. There are more and more online customer service systems that provide customers with help. However, how to use machines to effectively recognize the state of the dialogue sentence in a large amount of dialogue messages, such as: asking questions, problem solving, etc., is still an unsolved problem. The Nugget Detection (ND) subtask in the Short Text Conversation (STC) competition aims at alleviating the burden. This paper introduces the approaches of our team, TMUDS, to tackle this problem.

The dialogue evaluation task (DialEval-1) hosts two subtasks, dialogue quality (DQ) and nugget detection (ND). The main goal of DQ is to test the completion of the dialogue, the effectiveness of the dialogue, and the satisfaction of the customer. On the other hand, the goal of ND is to predict the situational state of the dialogue. We participated in the Chinese ND task, which involves processing logs from customer service desk dialogues and predict the distribution of predefined labels over nugget types from the dialogue content of each round. There are four labels for a customer's turn: "Customer trigger" (CNUG0), "Customer goal" (CNUG*), "Customer regular" (CNUG), and "Customer Not-a-Nugget" (CNaN). Besides, there are three labels for a helpdesk's turn: "Helpdesk goal" (HNUG*), "Helpdesk regular" (HNUG), and "Helpdesk Not-a-Nugget" (HNaN). A total of 19 annotators perform labeling and convert the labelled results into probability distribution. The organizers use evaluation measures including the Root Normalised Sum of Squares (RNSS) and Jensen-Shannon Divergence (JSD) for the performance evaluation. [11]

2 RELATED WORK

In recent years, research on general domain, task-oriented dialogue agents has become more and more popular. However, there are very few methods for evaluating such systems. Therefore, the task of STC-3 was added to NTCIR-14, in which the ND subtask is similar to dialogue act (DA) labeling problem. This task could be solved using sequence labeling techniques or modeled as a classification problem. The major difference from traditional DA labeling is that the output is not a single label but a distribution of label probability for each sentence.

The literature on STC includes a variety of approaches, e.g., Hidden Markov Model [12], Naïve Bayes [8], Conditional Random Fields (CRF) [12, 15], and deep learning methods [1, 4, 6, 7, 9]. Early deep learning models rely on convolutional neural network (CNN) and bidirectional long short-term memory (Bi-LSTM) modules [7]. Later, hierarchical CNN and Bi-LSTM models were applied to the sentence and dialogue representation used for DA labeling to better represent sentences [1]. Recently, several authors [4, 6, 9] have proposed CRF-enhanced DNN models.

Most ND methods regard this problem as a classification task, and extract features from text through machine learning-based methods. However, some mistakes may occur in this process, which may lead to final classification errors. In order to solve this problem, neural networks that have strong feature and semantic learning capabilities are proposed, and it can automatically learn text representations from data. In the ND subtask of NTCIR-14, the team that won the first place [5] utilize the Bidirectional Encoder Representation from Transformers (BERT) [2] as the embedding layer of the LSTM baseline. Unlike the original BERT, only the first four layers are used as feature extractors. In addition, other teams [14] have used the Bi-LSTM model to extract the context dependency between dialogues, and they adopt attention mechanism to learn the key sentences or phrases in the dialogue, which can improve the identification of nugget detection ability.

3 METHODOLOGY

3.1 Data Pre-processing

First, we separate the training data set into two according to the sender (customer and helpdesk). For the word segmentation of Simplified Chinese text data, we adopt Jieba to conduct the tokenization and drop stop words. Some of the input are in the format of non-text emojis, which are replaced with '*'. Notably, according to our preliminary data analysis, we found that for each service case, the cumulative distribution of labels is skewed based on the dialogue round order. Therefore, we also extract the dialogue sequence (round) as an important feature.

3.2 Model Structure

In this work, we use BiLSTM as the basis, and add an attention layer at the end of the model to more accurately capture context semantics and relationships. LSTM consists of input X_t at time t, cell state C_t, temporary cell state $\widetilde{C}_t$, hidden state h_t, forget gate f_t, input gate i_t, and output gate o_t. The calculation process of LSTM is to learn information that is useful for subsequent calculations through the forgetting of information in the cell state and memorizing new ones. And, the hidden layer state h_t is output through the output gate at each step. Gates and output are calculated from the hidden state h_{t-1} at the previous timestep and the current input X_t jointly.

The first step in our LSTM is to decide what information will be forgotten from the cell state. In this step, the previous hidden layer h_{t-1} and the current input X_t are inputs. The calculation is as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

The next step is to determine what new information is to be stored in the cell state. There are two parts here. First, the input gate determines what value we will be received. Then, a new candidate value vector, $\widetilde{C}_t$, will be constructed.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\widetilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

We multiply the old state by f_t so that unwanted information is discarded. Then, we add $i_t \times \widetilde{C}_t$, the candidate cell value modulated by the input gate. This completes the update of the cell state.

$$C_t = f_t \times C_{t-1} + i_t \times \widetilde{C}_t \tag{4}$$

Finally, we need to determine what values to output. The output is based on the cell state, but is also a filtered version. First, we use a sigmoid function to determine which part of the cell state will be output. Next, we process the cell state through tanh and multiply it with the output of the sigmoid gate. In the end, we will only output the part of the output that comes through of the output gate.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \times \tanh(C_t) \tag{6}$$

Interestingly, LSTM can operate in a forward fashion (left to right) as well as backward (right to left) when learning NLP data. It is straightforward to imagine that in language, the current state is not only affected by the left state, but also by the next state. Regarding this problem, Graves et al. [3] proposed a Bidirectional LSTM (BiLSTM) to capture this phenomenon.

This work also adopts the attention mechanism to enhance the ability of the model to extract important information from the corpus. Attention mechanism [13] is an attempt to formalize the action of selectively focusing on some related content and ignoring others. In NLP, the attention model is mainly used to find the correlation between words, so that the model can focus on important information only, and then more effectively extract them in text and improve overall training performance. The calculation method is to first obtain the correlation between each encoder hidden states $h_1 \cdots h_T$ and decoder hidden state s_t, and perform softmax normalization operation to obtain the weight of each hidden layer vector a_{ij}, the calculation formula is as follows:

$$e_j = a(s_{i-1}, h_j) = v_a^T \tanh(W_a s_{i-1} + U_a h_j) \tag{7}$$

$$a_{ij} = \frac{\exp(e_{ij})}{\Sigma_t \exp(e_{ij})} \tag{8}$$

e_{ij} represents the correlation between the previous hidden layer state $s_i - 1$ of the i output and the j input hidden layer vector h_j. Then, the weighted sum of $h_1 \cdots h_T$ is performed to obtain the encoding vector c_i

$$c_i = \sum_{j=1}^{T} a_{ij} h_j \tag{9}$$

3.3 Word Embedding

This work also utilizes Word2Vec and BERT embeddings as embedding methods. Word2Vec [10] uses a vector to represent the semantics of words through learning from a large amount of textual data. After embedding words into a space, those with similar meanings would be closer with each other, forming a cluster. In the Word2Vec model, there are mainly two approaches: continuous bag-of-words (CBOW) and Skip-gram (SG). As Figure 1 shows, the training goal of the SG model is to find word representations that can be used to predict surrounding words in a sentence or document. Given the training word sequence $w_1, w_2, w_3 ... w_T$, the goal of the SG model is to maximize the average logarithmic probability. On the other hand, Figure 2 depicts that CBOW attempts to model the surrounding words of a given target word to predict the input word representation.

BERT was proposed by Devlin et al. [2] in 2018. The full name is Bidirectional Encoder Representations from Transformers. It is a language representation model trained by Google using a large amount of unlabeled text in an unsupervised manner. Its architecture resembles the Encoder in a regular Transformer model. The aim of BERT is to train a basic representation model that can be applied to multiple NLP tasks, and then fine tune multiple downstream tasks on this basis. This work introduces Simplified Chinese pre-training corpus Word2vec and BERT embedding, as Figure 3 indicated, to be the input feature vector of the BiLSTM model.

4 EXPERIMENT

4.1 Settings

The dataset for this paper is from the Chinese Nugget Detection subtask of NTCIR-15 Tasks, which consists of Simplified Chinese helpdesk and customer dialogue data crawled from the Weibo website. The dataset contains 3,700 cases of training data, 390 cases of development data, and 300 cases of test data.

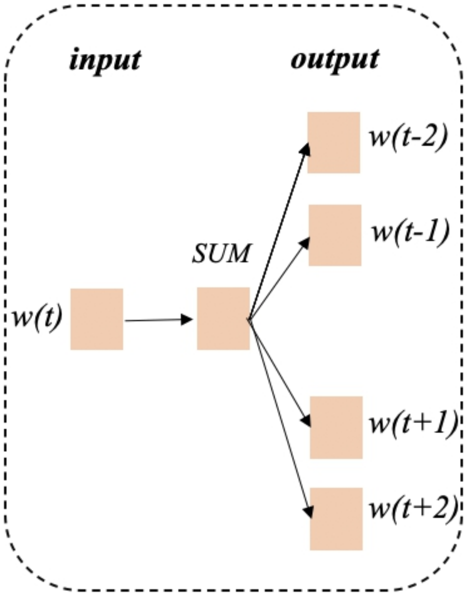

Figure 1: Skip-gram model structure

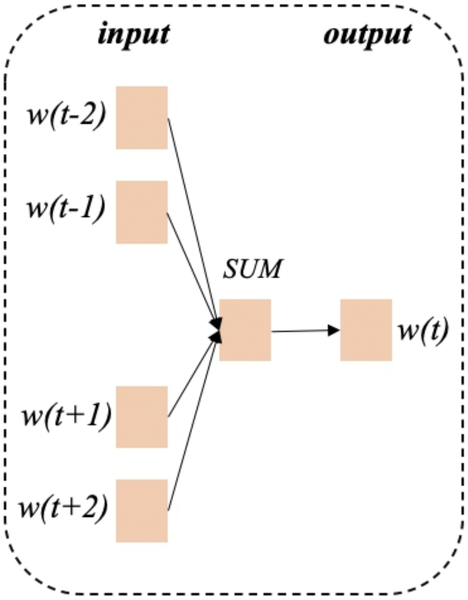

Figure 2: CBoW model structure

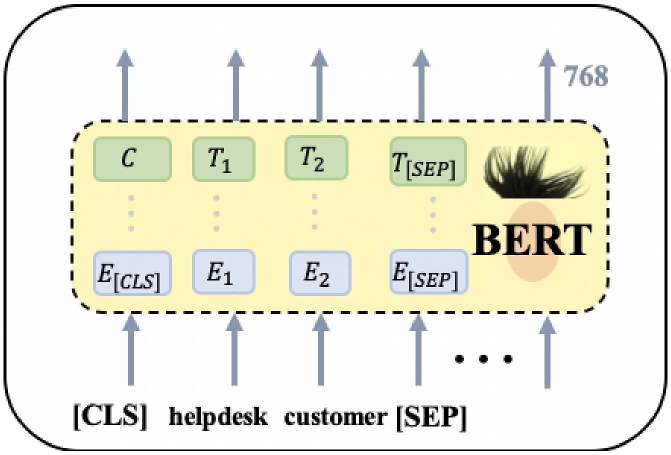

Figure 3: Bert embedding structure

Table 1: Hyper-parameter of experiment setting.

Hyper-parameter	Value
word embedding	word2vec 300/ Bert (100,768)
learning rate	0.001
dropout rate	0.35
batch-size	128

Table 1 lists the hyper-parameters used in the training of our neural model. In addition, the labels of this experiment are nominal, and the evaluation methods are the Jensen-Shannon Divergence (JSD) and Root Normalizes Sum of Square (RNSS), proposed by Sakai et. al. [11], to evaluate bin-by-bin probability distribution. The formal calculation is as follows:

$$JSD(p, p^*) = \frac{KLD(p||pM) + KLD(p^*||pM)}{2} \tag{10}$$

$$KLD(p_1||p_2) = \sum P_1(i) \log_2 \frac{p_1(i)}{p_2(i)} \tag{11}$$

$$RNSS(p, p^*) = \sqrt{\frac{SS(p, p^*)}{2}} \tag{12}$$

$$SS(p, p^*) = \sum (p(i) - p^*(i))^2 \tag{13}$$

4.2 Results

Below are the results of the three different runs that we submitted for evaluation via the online evaluation tool provided by the organizer. The results in Table 2 are the score from the official evaluation. We can see that, in terms of the JSD score, TMUDS-run-1 is the best. Meanwhile, RNSS is the highest in TMUDS-run-2. Finally, both JSD and RNSS of the TMUDS-run-0 submission is the lowest.

Table 2: Results of each run of experiment.

Run	JSD	RNSS
TMUDS-run-0	0.0906	0.1995
TMUDS-run-1	**0.0883**	0.1953
TMUDS-run-2	0.0887	**0.1948**

Table 3: Model structure of different Experiment runs from the TMUDS team.

Run	Structure
TMUDS-run-0	Bert emb + 2 BiLSTM
TMUDS-run-1	Bert emb + 2 BiLSTM + Att
TMUDS-run-2	Bert emb + 1 BiLSTM + Att

As shown in Table 4, TMUDS-run-0 is based on the BERT Embedding layer with two layers of BiLSTM, while TMUDS-run-1 and TMUDS-run-2 both have an Attention layer that is stacked on top of the recurrent model. When comparing these results, we can conclude that whether the attention mechanism is added is the key to model optimization. The difference between TMUDS-run-1 and TMUDS-run-2 lies in the number of recurrent layers. Namely, there are two layers and one single layer in the BiLSTM in these runs, respectively. There is no difference in their hyper-parameters. In addition, we also experimented with using word2Vec embedding layer at the early stage of the pilot study. However, the result was not satisfactory and could not surpass the outcome of the model based on BERT Embeddings. It again indicates that BERT can more effectively capture the semantic relationship of the dialogue in the context.

5 DISCUSSION

In this study, in addition to the final prediction results, we also implement many basic methods to compare results. Table 4 below compares the results of different methods and the application of different features into the model. The results here represent the local score (only the training set and validation set are considered, and the test set is not included)

- We found that the performance of traditional basic methods is not better enough. Compared with other basic models, the ability of bidirectional LSTM to process short texts can capture contextual meaning better;
- Secondly, we use Wiki Simplified Chinese pre-train vector model to extract the dialogue text vector, and the result is better than the previous basic model. We found that based on the statistical results of the training set and the validation set, the nugget label in each round of the dialogue are not uniformly distributed, which has a specific tendency. Therefore, we believe that the dialogue round is also an important feature parameter that affects prediction;
- In addition, we also noticed that most of the labels in the first round (customer initiates a customer service dialogue) are CNUG0, so we separate the first round of dialogue independently from the prediction model, and use the average probability distribution of the training set to represent the verification result;
- Finally, we noticed that the dialogue will be affected by the previous round of dialogue. Therefore, we also regard the vector of the previous round as the current training feature and add it to the training.

Table 4: Comparisons among different implementations. The test only includes training dataset and development dataset. Noted that the underline refer to concatenate features to the original vector. * is adding round feature and fixing first round as mean from training data. ** is the concatenation of the previous vector as a feature with the current vector.

Method	Local JSD	Local RNSS
TextCNN	0.048	0.141
LSTM	0.043	0.130
BiLSTM	0.040	0.129
2 BiLSTM	0.042	0.131
W2V 2 BiLSTM	0.036	0.122
W2V 2 BiLSTM (*)	0.036	0.120
W2V 2 BiLSTM (*/ **)	0.037	0.121
W2V 2 BiLSTM (*/ **) + Att	0.035	0.120
Bert emb 2 BiLSTM (* / **) + Att	**0.034**	**0.110**

6 CONCLUSION

In this work, we introduce a novel approach using BERT Embeddings, which can capture the semantic relationship of the dialogue in the context, to the Nugget Detection (ND) subtask at the NTCIR-15 STC-3 task. We observe that using a recurrent neural network with bidirectional LSTMs can effectively tackle this task. In addition, we also incorporate an Attention layer to improve the performance of our approach. It can indeed assist the model by assigning different weights of the input words according to the end goal, and extract only critical and important information. Therefore, the final model can be more accurate than those without the attention information. The evaluation results of our submitted runs, TMUDS-run-1 and TMUDS-run-2, are examples of the effectiveness of our approach.

7 ACKNOWLEDGMENT

This research was supported by the Ministry of Science and Technology of Taiwan under grant MOST 107-2410-H-038-017-MY3, MOST 109-2410-H-038 -012 -MY2, and MOST 107-2634-F-001-005.

REFERENCES

[1] Phil Blunsom and Nal Kalchbrenner. 2013. Recurrent convolutional neural networks for discourse compositionality. In *Proceedings of the 2013 Workshop on Continuous Vector Space Models and their Compositionality*. Proceedings of the 2013 Workshop on Continuous Vector Space Models and their ….

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[3] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks* 18, 5-6 (2005), 602–610.

[4] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).

[5] Sosuke Kato, Rikiya Suzuki, Zhaohao Zeng, and Tetsuya Sakai. 2019. SLSTC at the NTCIR-14 STC-3 dialogue quality and nugget detection subtasks. *NTCIR14. p. to appear* (2019).

[6] Harshit Kumar, Arvind Agarwal, Riddhiman Dasgupta, Sachindra Joshi, and Arun Kumar. 2017. Dialogue act sequence labeling using hierarchical encoder with crf. *arXiv preprint arXiv:1709.04250* (2017).

[7] Ji Young Lee and Franck Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. *arXiv preprint arXiv:1603.03827* (2016).

[8] Piroska Lendvai and Jeroen Geertzen. 2007. Token-based chunking of turn-internal dialogue act sequences. In *Proceedings of the 8th SIGDIAL Workshop on Discourse and Dialogue*. 174–181.

[9] Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354* (2016).

[10] Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*. 746–751.

[11] Tetsuya Sakai. 2018. Comparing two binned probability distributions for information access evaluation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1073–1076.

[12] Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics* 26, 3 (2000), 339–373.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[14] Ming Yan, Maofu Liu, and Junyi Xiang. [n.d.]. WUST at the NTCIR-14 STC-3 Dialogue Quality and Nugget Detection Subtask. ([n. d.]).

[15] Matthias Zimmermann. 2009. Joint segmentation and classification of dialog acts using conditional random fields. In *Tenth Annual Conference of the International Speech Communication Association*.