ALBERT-BiLSTM for Sequential Metaphor Detection

Shuqun Li, Jingjie Zeng, Jinhui Zhang, Tao Peng, Liang Yang and Hongfei Lin
Information Retrieval Laboratory of Dalian University of Technology,
Department of Computer Science, Dalian University of Technology
{1397023717,jjwind,wszjh,1316785071}@mail.dlut.edu.cn
{liang,hflin}@dlut.edu.cn

Abstract

In our daily life, metaphor is a common way of expression. To understand the meaning of a metaphor, we should recognize the metaphor words which play important roles. In the metaphor detection task, we design a sequence labeling model based on ALBERT-LSTM-softmax. By applying this model, we carry out a lot of experiments and compare the experimental results with different processing methods, such as with different input sentences and tokens, or the methods with CRF and softmax. Then, some tricks are adopted to improve the experimental results. Finally, our model achieves a 0.707 F1-score for the all POS subtask and a 0.728 F1-score for the verb subtask on the TOEFL dataset.

1 Introduction

As a common rhetorical device, we often use metaphors to express our feelings and ideas vividly and concisely in our daily life. Detecting metaphors in texts is of great significance for analyzing the meaning and polarity of sentences. It can also be used to generate sentences that are more suitable for human expression, and promote the development of chat robots, machine translation and other fields.

Metaphor detection generally recognizes metaphorical words or phrases from the metaphorical sentence, such as "she is a woman with a stone heart", in which "stone" is a metaphorical word modified "heart". However, the task of metaphor detection is very challenging. Firstly, metaphor detection is a sequence labeling task, and every word in a sentence needs to be classified. Secondly, the boundaries between metaphors and non metaphors are sometimes vague. Moreover, due to the different identities of authors, some metaphorical words involve knowledge in specific fields and are difficult to recognize directly (Tsvetkov et al., 2014). The traditional lexicon based method cannot cover all possible words occurred in metaphors. It is difficult to recognize metaphors when certain words are out-of-vocabulary. Although the traditional machine learning method needs to extract features manually (Heintz et al., 2013), its performance is still insufficient.While with the further development of language model, different kinds of end-to-end pre-trained models almost dominate the field of natural language processing, and also improve the prediction accuracy of various tasks to a higher level. Hence, in this paper we use pre-trained models to deal with metaphor detection task.

The purpose of this metaphorical shared task is to identify the whole words and verbs in given sentences. In this paper, we design an ALBERT-BiLSTM structure to recognize metaphorical words in TOEFL dataset. Firstly, we conduct an experimental comparison on the form of input sentence, and then select the form of inputting the single sentence directly. Secondly, we compare the application of BERT on this sequence labeling problem, and extract the input form of the first part after the BPE word segmentation of BERT. Finally, the effect of conditional random field (CRF) and softmax with class weights in the output layer is compared and the result shows that softmax with class weights is better. At the same time, we also adopt some tricks in the training process, including semantic merge and loss with class weight. The final result in the test set achieves a 0.707 F1-score for the all POS subtask, and a 0.728 F1-score for the verb subtask on TOEFL dataset.

2 Related works

At present, researchers in the field of natural language processing have made a lot of effort in metaphor detection task. Shutova et al. (2016)

Proceedings of the Second Workshop on Figurative Language Processing, pages 110–115
July 9, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

used unsupervised learning to detect metaphors, and applied the syntactically perceived distribution word vectors. Gong et al. (2017) used metaphorical language detection as a method to explore the composition of word vectors, and calculated cosine distance to distinguish metaphor from non-metaphor: words that are out of context in sentences may be metaphorical. Gao et al. (2018) proposed a model to connect the expression of Glove and Elmo for solving the sequence labeling task, which is also transferred to the following metaphor task. Gutiérrez et al. (2016) used the flexibility of word vectors to study metaphor and its possibility of modeling in semantic space. Mao et al. (2019) designed an end-to-end model based on Glove and Elmo, which could identify metaphors conveniently.

For metaphor often contains emotions, some researchers tended to carry on emotion analysis on metaphors. Veale (2012) constructed an lexicon based model for analyzing emotions of metaphors. Kozareva (2013) proposed a new method, which integrated the trigger factors of cognition, emotion and perception.

Verb metaphor recognition is also an important subtask of metaphor recognition. Jia and Yu (2008) used conditional random fields(CRF) model and maximum entropy(ME) model to recognize verb metaphor, and they pointed out that there were no mature syntactic and semantic tools for metaphor analysis in Chinese. Beigman Klebanov et al. (2016) studied the effectiveness of semantic generalization and classification in capturing the rules of verb behavior, and tried to analyze their metaphors from the orthographic words unigrams.

These studies also provided some guidance to our work. For example, the word vector concatenation in LSTM is similar to RNN_HG (Mao et al., 2019).

3 Task definition

The dataset of this metaphorical shared task includes two kinds: VUA (Steen, 2010) and TOEFL (Klebanov et al., 2018). This paper mainly conducts experiments on TOEFL data. TOEFL dataset contains 180 articles written by non-native English speakers in the TOEFL test, and 60 articles in the test set. Each article is divided into several parts by sentence. At the same time, the corresponding examination questions of each article are provided, and there are 8 kinds of questions. The details of the dataset are as follows in table 1.

We make statistics on the sentence length distribution in the data set, and the following is shown by the box chart.

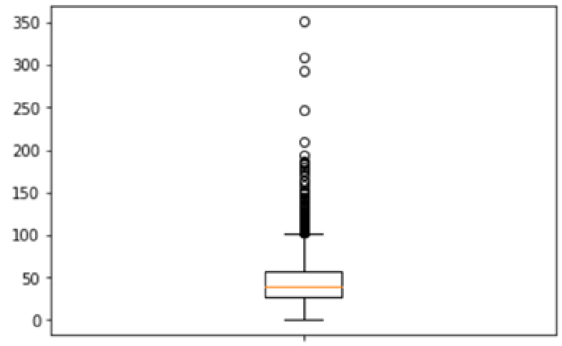

Figure 1: Sentence length distribution of train set.

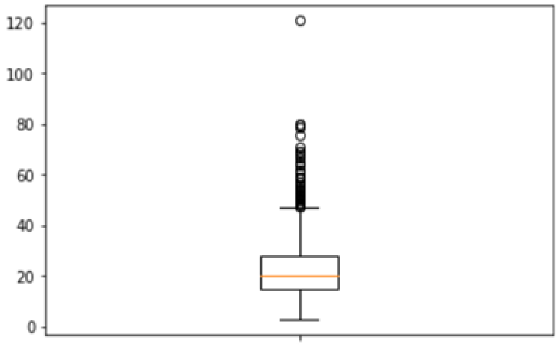

Figure 2: Sentence length distribution of test set.

It can be seen that the sentence length of training set is longer than that of test set, but most of them are distributed between 0 and 100, no more than 350 tokens. It is suitable for BERT model, because the maximum sentence length that BERT can support is 512.

This shared task subtask is divided into all POS recognition and verb recognition. It also provides some tokens' ID in sentences of the test set, and finally submits the recognition result corresponding to token ID. Final ranking and results are reported by Leong et al. (2020).

4 Method

In this section, we aim to introduce the method conducted on TOEFL dataset in this shared task. We use 'BERT+BiLSTM+softmax' as the baseline model, in which BERT (Devlin et al., 2019) is a pretrained language model proposed by Google in 2018, and BiLSTM is a bidirectional Recurrent Neural Network. The details of our method are described below.

	Number of articles	Total number of sentences	Average sentence length	Proportion of positive samples
Train	180	2741	45.5	0.03356
Test	60	968	22.9	/

Table 1: The details of the data set.

4.1 Data processing

Our data preprocessing method mainly includes two parts: data alignment and data augmentation. In data alignment part, we associate each word with its label in the sentence to transform the task into a sequential labeling task. In data augmentation part, we introduce context information and topic information of the sentences to expand the training data. There are three forms of our processed data: (1) single sentence; (2) the form of "sentence pair": considering that some metaphors are related to the context, we process the sentence into "sentence pair" form; (3) the form of "sentence-prompt pair": similar to form (2), we convert sentences into sentence pair, but we use prompt information instead of context information.The specific form is shown in table 2.

Format	Input	Label
single sentence	*B*	*L*
sentence pair	*A*+*B*	*L*
sentence-prompt pair	*B*+*P*	*L*

Table 2: Three different data input format. *B* is a sentence to be recognized and *L* is the label of *B*; *A* is the previous sentence of *B* in the text; *P* is the prompt of the text to which the sentence *B* belongs given in the TOEFL dataset.

The reason for this is that we believe the prompt information of the sentence will influence the prediction results of the model. In order to find the best form of data, we train the baseline model on three forms of data respectively. The results is shown in table 3.

Format	F1
single sentence	**0.687**
sentence pair	0.673
sentence-prompt pair	0.665

Table 3: Three different data input format preprocessing methods with baseline model.

The results show that the data format (1) performs best. After observing the dataset, we believe that the poor performance of data format (2) is due to the fact that the metaphor contained in the second sentence is not closely related to the first sentence. Additional input leads to the increased difficulty in model training. And the reason for the poor performance of data format (3) is that the sentence is less related to the given prompt. In conclusion, metaphors are more related to the local information in the sentence, and we use data format (1) as the input of our method.

In addition, we find that some sentences in TOEFL data are mainly written by people from non-native English speaking countries, and there are many spelling errors. So we try to use the SpellChecker package of Python to correct the spelling of words, the F1-score of the whole tokens in cross-validation before and after correction are 0.687 and 0.681 respectively. We initially thought word correction may be a useful method. However, the results show that the corrected data is not as good as expected, so we skip this step.

4.2 Our Model

The target of this evaluation is to identify metaphorical words in sentences, and we regard this task as a sequence labeling task. Our model consists of three layers: the pre-trained model layer, the contextual layer and the classification layer.

In pre-trained model layer, we use BERT for sequence labeling task. We find that the word segmentation algorithm BPE will divide the input words into smaller semantic units, i.e. subword, which leads to that the length of output sequence is greater than the length of input. To keep the length of the input and output in the same way, we propose three model input structures: (1) only the first subword of word is taken as input; (2) the input is unchanged, and only the first embedding of each word in output is taken as the representation of the current word; (3) the input is unchanged, and the embeddings of a word in output are merged into one embedding by convolutional neural networks which is taken as the representation of the current word. The results are as shown in table 4.

The results show that the structure (1) performances best, so we use only the first subword as the input of each word. We think the reason for the poor performance of structures (2) and (3) is

Input	F1
First segmentation	**0.687**
First vector	0.674
Aggregate vector	0.681

Table 4: The result of the three different word vector representation methods, where we use softmax as the classification layer.

that the provided TOEFL dataset is small and easy to be affected by input noise. The first subword is often the main part of a word, which can better express the semantic of the word compared to the rest subwords. Non-first subwords preserved by structure (2) and structure (3) will increase the input length of BERT, which brings noise while training, and makes it more difficult to learn from a small dataset.

In contextual layer, we use BiLSTM to get the context representation of the word based on the output embedding of BERT. In classification layer, we compare the performance of CRF and softmax. The cross-validation F1-score of the whole tokens are 0.671 for CRF and 0.687 for softmax. The results show that the softmax is better than the CRF model. We believe the reason is that there is no hard relation between the metaphor words and other words, so the constraint of CRF does not work well.

Finally, we adopt ALBERT+BiLSTM+softmax as our model. As a new pre-trained model released by Google, the performance of ALBERT-xxLarge-v1(ALBERT for short) (Lan et al., 2019) on natural language understanding task is better than BERT. Since the output dimension of ALBERT model is as high as 4096 dimensions, we just concatenate the first 300 dimension output embedding of ALBERT with the output embedding of BiLSTM. Then let the merged representation go through a full connection layer to get the probability distribution. Finally, the probability distribution is classified in the softmax layer. The reason for concatenating two parts of embedding is that we hope our model can predict by combining the context meaning and the word meaning. Table 5 shows that the concatenation method used performs better.

4.3 Tricks

In this subsections, we will introduce some useful tricks used in this evaluation.

Concatenation Format	F1
300-d	**0.709**
0-d	0.696
Linear mapping	0.695

Table 5: The result of the three different concatenation method. 300-d means concatenating LSTM output and the first 300 dimensions of ALBERT output as linear layer input; 0-d means taking only LSTM output as linear layer input; Linear mapping means mapping ALBERT output through a linear layer to 300 dimensions and concatenating it with LSTM output as linear layer input.

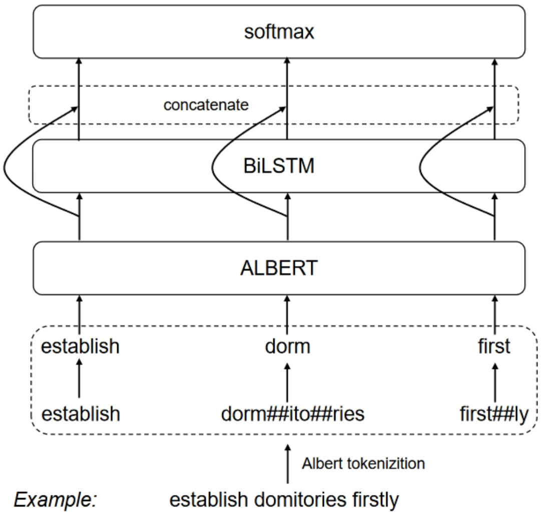

Figure 3: The architecture of our method.

4.3.1 Semantic merge

ALBERT has 12 layers in total. It is generally believed that each layer of language model learns different features of the text, so we make two attempts on the representation vectors for different layers: (1) we concatenate the average output of the last four layers as the final output; (2) we weighted sum the output of all 12 layers as the final output. The final online results show that method (1) is better. We believe that this is because the lower level of the language model is more inclined to learn the syntactic features of the text, while the higher level is more inclined to learn the semantic features of the text (Jawahar et al., 2019). The task of metaphor recognition is more challenging for the proposed model to understand the semantics. The addition of lower level feature representation will introduce noise information instead.

4.3.2 Loss with class weight

Due to the small proportion of metaphorical words in sentences, we consider to increase the loss value of positive metaphorical samples to balance the number difference between positive and negative samples. We try the weight value of positive sample loss between 0.8 and 4, based on the results we find that when weight value of positive samples is 2, we can get the best result.

The specific hyper-parameters of the model are as follows: ALBERT's learning rate is 1e-5, and weight decay is 0.01; BiLSTM has one layer, and the learning rate is 2e-3, hidden units are 256, dropout rate is 0.5; the optimizer is Adam, batch size is 2, early stopping is used. The loss weights corresponding to the positive and negative classes are set to 2 and 1 respectively. The results of our final model on the test sets are as shown in table 6:

	ALLPOS	VERB
TOEFL	0.707	0.728
VUA	0.712	0.755

Table 6: The F1-score of final model on TOEFL and VUA test sets.

Table 6 shows that our model performs well on the TOEFL dataset, and we also tests the results of the model on the VUA dataset. The results show that the proposed model in this paper can achieve good results on both datasets.

5 Conclusion

In this paper, we propose a method with ALBERT+BiLSTM+softmax to identify metaphor words in the sentence. We extract text features through ALBERT's learning ability, and use BiLSTM to get contextual representation, then get the final prediction results with softmax layers. We also try several data preprocessing methods and utilize three tricks to improve the performance of our proposed model. Besides, we analyze and explain the results of each method according to the characteristics of the metaphor detection task. The experimental results show the effectiveness of our method.

References

Beata Beigman Klebanov, Chee Wee Leong, E. Dario Gutierrez, Ekaterina Shutova, and Michael Flor. 2016. Semantic classifications for detection of verb metaphors. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 101–106, Berlin, Germany. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ge Gao, Eunsol Choi, Yejin Choi, and Luke Zettlemoyer. 2018. Neural metaphor detection in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 607–613, Brussels, Belgium. Association for Computational Linguistics.

Hongyu Gong, Suma Bhat, and Pramod Viswanath. 2017. Geometry of compositionality. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3202–3208. AAAI Press.

E.Dario Gutiérrez, Ekaterina Shutova, Tyler Marghetis, and Benjamin Bergen. 2016. Literal and metaphorical senses in compositional distributional semantic models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 183–193, Berlin, Germany. Association for Computational Linguistics.

Ilana Heintz, Ryan Gabbard, Mahesh P. Srivastava, Dave Barner, Donald Black, Majorie Friedman, and Ralph M. Weischedel. 2013. Automatic extraction of linguistic metaphors with lda topic modeling.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3651–3657. Association for Computational Linguistics.

Yuxiang Jia and Shiwen Yu. 2008. Unsupervised Chinese verb metaphor recognition based on selectional preferences. In *Proceedings of the 22nd Pacific Asia Conference on Language, Information and Computation*, pages 207–214, The University of the Philippines Visayas Cebu College, Cebu City, Philippines. De La Salle University, Manila, Philippines.

Beata Beigman Klebanov, Chee Wee Leong, and Michael Flor. 2018. A corpus of non-native written english annotated for metaphor. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New*

Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers), pages 86–91. Association for Computational Linguistics.

Zornitsa Kozareva. 2013. Multilingual affect polarity and valence prediction in metaphor-rich texts. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 682–691, Sofia, Bulgaria. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations.

Chee Wee Leong, Beata Beigman Klebanov, Chris Hamill, Egon Stemle, Rutuja Ubale, and Xianyang Chen. 2020. A report on the 2020 vua and toefl metaphor detection shared task. In *Proceedings of the Second Workshop on Figurative Language Processing*, Seattle, WA.

Rui Mao, Chenghua Lin, and Frank Guerin. 2019. End-to-end sequential metaphor identification inspired by linguistic theories. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3888–3898, Florence, Italy. Association for Computational Linguistics.

Ekaterina Shutova, Douwe Kiela, and Jean Maillard. 2016. Black holes and white rabbits: Metaphor identification with visual features. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 160–170, San Diego, California. Association for Computational Linguistics.

Gerard J. Steen. 2010. A method for linguistic metaphor identification: From mip to mipvu.

Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 248–258. The Association for Computer Linguistics.

Tony Veale. 2012. A context-sensitive, multi-faceted model of lexico-conceptual affect. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 75–79, Jeju Island, Korea. Association for Computational Linguistics.