Using Multi-level Attention Based on Concept Embedding Enrichen Short Text to Classification

Ben You, Xiaohong Li, Qixuan Peng, Ruihong Li

Using Multi-level Attention based on Concept Embedding Enrichen Short Text to Classification

Ben You, XiaoHong Li*, QiXuan Peng, RuiHong Li

College of Computer Science and Engineering, Northwest Normal University, Lanzhou,China
xiaohongli@nwnu.edu.cn

Abstract. Aiming at the defects of short text, which lack context information and weak ability to describe topic, this paper proposes an attention network based solution for enriching topic information of short text, which can leverage both text information and concept embedding to represent short text. Specifically, short text encoder is used to enhance the representation of short texts in the semantic space. The concept encoder obtains the distribution representation of the concept through the attention network composed of *C-ST* attention and *C-CS* attention. Finally, Concatenating outputs from the two encoders creates a longer target representation of short text. Experimental results on two benchmark datasets show that our model achieves inspiring performance and outperforms baseline methods significantly.

Keywords: Short Text Representation, Knowledge Base, Conceptualization, BERT, Attention Mechanism.

1 Introduction

The task of categorizing short texts is one of the important methods for a wide range of applications, including web search, news classification. Short texts lack enough contextual information, which poses a great challenge for short text classification. An essential intermediate step for text classification is text representation. According to the different ways of leveraging external sources, previous text representation methods can be divided into two categories: explicit representation and implicit representation [1].

For explicit approaches, a short text is represented as a sparse vector by labeling, POS tagging, and syntactic parsing. Researchers develop effective features from many aspects, such as knowledge base. Although explicit models are easily understandable by humans, it is difficult for the models to capture deep semantic information from the contexts. Besides, they also suffer from the data sparsity problem.

In terms of implicit representation, a short text is usually mapped to an implicit space and represented as a dense vector [2] which is called embedding. Encoder-decoder framework is also frequently adopted to capture the semantics of texts [3]. An implicit representation model can capture rich information from context and facilitate text understanding with the help of deep neural networks. However, implicit representation model ignores semantic relations such as *isA* and *isPropertyOf* that exist in knowledge

bases. Such information is helpful for understanding short texts, especially when dealing with previously unseen words.

It is ineffective to use either explicit or implicit representations independently for short text representation or classification. Previous work combined the two and used a rich knowledge base to enrich the prior knowledge of short texts by conceptualizing [4]. However, there are still two major problems. First, when conceptualizing the short text, improper concepts are easily introduced due to the ambiguity of entities or the noise in knowledge bases. For example, "*Steve Jobs established Apple*", the conceptual *fruit* and *company* of *Apple* were extracted, but obviously *fruit* is not an appropriate concept which is caused by the ambiguity of *Apple*. Second, it is necessary to consider the relative importance of the concepts. For the same example, we extract the concepts *individual* and *entrepreneur* of *Steve Jobs* from the knowledge base. Despite the fact that they are both correct concepts, *entrepreneur* is more specific than *individual*.

In this work, we proposes an attention network based solution for enriching topic information of short text, which can leverage both concept embedding and text information to represent short text. Specifically, short text encoder is used to enhance the representation of short texts in the semantic space. The concept encoder obtains the distribution representation of the concept through the attention network composed of two-level attentions. Finally, Concatenating output from the two encoders creates a longer target representation of short text.

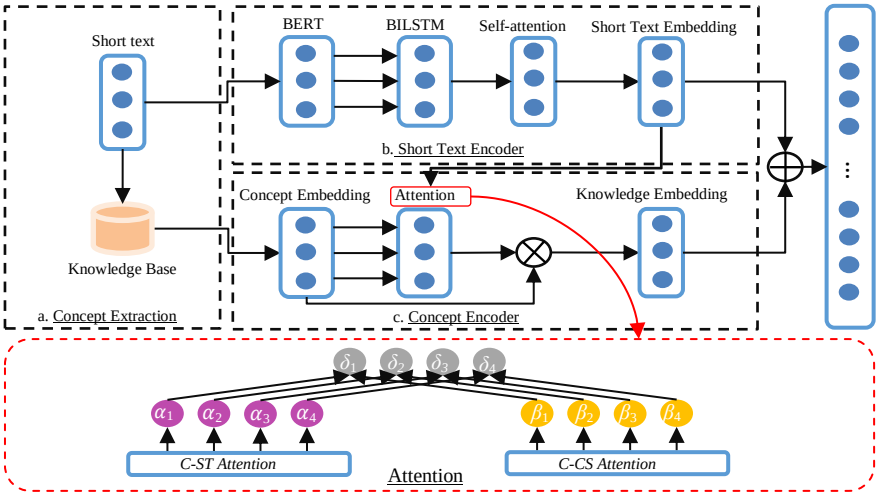

Fig. 1. Model Architecture

2 Proposed Model

The overall architecture of our model (MACE) is shown in Figure 1, which can be divided into three parts: concepts extraction, short text encoder and concept encoder.

We are given a set of documents D with a set of document labels Y where each document $d \in D$ is composed of multiple words $W^d = \{w_0^d, w_1^d, ..., w_n^d\}$. The input of our model is a short text d, where w_i^d represents i-th word in the short text d.

2.1 Concepts Extraction

The task of this module is to extract relevant conceptual knowledge from the external knowledge base. We use the *IsA* relationship to define the relationship between entities and concepts. Specifically, given a short text d, the goal is to find a concept set $C^d = \{c_1^d, c_2^d, ..., c_m^d\}$ related to the entities in the short text, where c_i^d is one of the concepts. First, entity linking technology is needed to identify entities in short text. Then for each entity, it needs to be conceptualized and its conceptual information is obtained from the external knowledge base. We utilize the existing entity-concept knowledge base (Microsoft Concept Graph) [5] to obtain conceptual information. It is a huge entity-concept knowledge base and has excavated *IsA* data from billions of web pages, with tens of millions of entities and millions of concepts, which is of great help to the understanding of short texts.

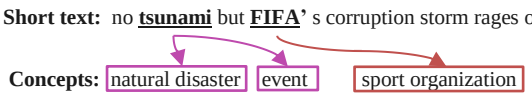

Fig. 2. Example of short text conceptualization.

Figure 2 shows an example of short text conceptualization. For this short text of sport classification, it can be seen that there may be specialized special words in a certain field or situation, such as the word *FIFA* in the sport text. It does not exist in either the implicit or explicit representation. Our model can obtain the prior knowledge of short texts by combining with the knowledge base to solve this problem.

2.2 Short Text Encoder

The goal of this module is to generate a short text representation $\boldsymbol{q}$ for a given short text sequence d. The short text is regarded as a sentence and used as the input of the BERT [6], and the word vector corresponding to each word of the sentence is calculated. Here we use the average of word-level hidden states in the last layer of BERT as the abstract semantics representation. The output of BERT is a representation of the sequence $(\boldsymbol{x}_1^d, \boldsymbol{x}_2^d, ..., \boldsymbol{x}_n^d)$ where $\boldsymbol{x}_i^d$ is the word vector in 768-dimension of w_i^d.

Then a layer of BiLSTM [7] is used on top of BERT. BiLSTM includes both forward and backward networks, which solves the problem of traditional LSTM model that cannot be processed due to serialization. Let the number of hidden units of each unidirectional LSMT be u. We denote the short text representation sequence as $\boldsymbol{H} = (\boldsymbol{h}_1^d, \boldsymbol{h}_2^d, ..., \boldsymbol{h}_n^d)$ where $\boldsymbol{h}_i^d \in \mathbb{R}^{2u}$ is the vector representation for word w_i^d after BERT and BiLSTM.

After that, the self-attention mechanism is adopted to solve the problem of vanishing gradient of BiLSTM. We use the scaled dot-product attention mechanism [3], which distinguishes the importance of different features, ignores unimportant features, and focuses attention on important features. Finally, the module obtains word vector $\boldsymbol{h}_i^* \in \mathbb{R}^{2u}$ corresponding to word w_i^d.

2.3 Concept Encoder

Given a concept representation set $\boldsymbol{T}^d$ of size *m* denoted as $\{\boldsymbol{t}_1^d, \boldsymbol{t}_2^d, ..., \boldsymbol{t}_m^d\}$, where $\boldsymbol{t}_i^d$ is the *i*-th concept vector, the goal of this module is to generate vector representation $\boldsymbol{p}$ for $\boldsymbol{T}^d$.

In order to reduce the ambiguity of entities or the noise of external knowledge and the bad influence on incorrect concepts, we adopt the *Concept towards Short Text* (*C-ST*) attention [8] which is used to measure semantic similarity between the *i*-th concept and the short text $\boldsymbol{q}$. *C-ST* attention is given by the following formula:

$$\alpha_i = softmax(\boldsymbol{w}_1^T \tanh(\boldsymbol{W}_1 \times \text{concat}[\boldsymbol{t}_i^d; \boldsymbol{q}] + b_1)) \tag{2}$$

Here α_i represents the attention weight of the *i*-th concept to the short text. A larger α_i means that the *i*-th concept is more similar to the short text in semantics. $\boldsymbol{W}_1 \in \mathbb{R}^{d_a \times (2u+d)}$ is the parameter matrix and $\boldsymbol{w}_1 = \mathbb{R}^{d_a}$ is the parameter vector where d_a is a hyperparameter, and b_1 is the bias. It should be noted that an entity may correspond to more than one concept. Therefore, for multiple concepts, we set the hyperparameter K as the maximum number of concepts that an entity can obtain.

In addition, based on consideration of the relative importance of concepts, *Concept towards Concept Set* (*C-CS*) is defined to measure the importance of each concept c_i^d:

$$\beta_i = softmax(\boldsymbol{w}_2^T \tanh(\boldsymbol{W}_2 \boldsymbol{t}_i^d + b_2)) \tag{3}$$

Here β_i denotes the attention weight from the *i*-th concept to the entire concept set C_i^d. $\boldsymbol{W}_2 \in \mathbb{R}^{d_b \times d}$ is a weight matrix and $\boldsymbol{w}_2 \in \mathbb{R}^{d_b}$ is a weight vector where d_b is a hyperparameter, and b_2 is the bias.

The final attention weight of each concept is obtained by combining α_i and β_i with:

$$\delta_i = softmax(\gamma \alpha_i + (1-\gamma)\beta_i) \tag{4}$$

Here δ_i represents the final attention weight from the *i*-th concept towards the short text, and $\gamma \in [0,1]$ is the hyperparameter that manually adjusts the importance of α_i and β_i.

In the end, the final attention weight is used to calculate the weighted sum of the concept vector to obtain the semantic vector which represents the concepts:

$$\boldsymbol{p} = \sum_{i=1}^{n} \delta_i^d \boldsymbol{t}_i^d \tag{5}$$

After obtaining the semantic concept representation, we combine it with short text representation by concatenating them. Then we apply an output layer on the join vector to convert the output numbers into probabilities for classification.

3 Experiments

In this section, we conduct extensive experiments to evaluate our method.

3.1 Dataset

We use two benchmark short text classification datasets for evaluation. TagMyNews [9], a news dataset, and Snippets [10], contains Google search snippets. The details about such corpora are shown in Table 1.

Table 1. Details of the experimental datasets.

	Classes	Docs	Avg len per doc
TagMyNews	7	32,567	8
Snippets	8	12,332	17

In experiment, each dataset is randomly split into 80% for training and 20% for testing. 20% of the randomly selected training examples are used to development set.

3.2 Compared Methods

We compared our proposed method with the following methods. two feature-based methods, two deep learning methods and BERT.

SVM+BOW and SVM with unigram characteristics [11]. SVM+LDA, which is characterized by LDA [12]. Bidirectional long short-term memory (BiLSTM) with attention mechanism (AttBiLSTM) [13]. Convolutional neural network (CNN) [14]. BERT (bert-base-uncased) with fine-tuning and BERT without fine-tuning.

3.3 Evaluation Results and Analysis

For the parameters of all the compared models, we performed grid-search on their appropriate ranges. Glove [15] is used for concept embeddings with the number of concepts at K=5. Only the final two layers of BERT and our model are fine-tuned and the maximum input length is 512. We adopt the standard cross-entropy loss function and Adam algorithm with learning rate 2e-4 to train our model. The epoch of each dataset is 15 and the batch size is 128.

To evaluate the performance, we adopted two popular metrics: accuracy and F1 score, which are widely used to evaluate the performance of classification.

Text Classification Performance. In the first set of experiments, we compare the classification performance of our method against all the compared methods on the two datasets. As shown in Table 2.

The proposed method achieves the best results on both datasets. Because our model considers the knowledge base and attention. On both datasets, the SVM model (SVM+LDA) that uses topic information produces better results than the model that does not use topic features (SVM+BOW). This observation shows that the topic representation captured at the corpus level helps to alleviate the data sparseness problem in short text classification [16]. The neural models based on CNN or AttBILSTM produce better results than traditional methods show the effectiveness of representation learning in neural networks for short texts.

Compared with traditional methods and deep learning methods, the pre-training model BERT is better than the previous two. BERT with fine-tuning is more effective than it without fine-tuning proves that fine-tuning can be used to adapt to specific tasks. Although the effect of the latter is worse (BERT(wo/fine-tuing)), it still performs better than traditional methods, which shows the strong effect of BERT.

Table 2. Reslults of compared models on different datasets.

	TagMyNews		Snippets	
	Accuracy	F1	Accuracy	F1
SVM+BOW	0.259	0.058	0.210	0.080
SVM+LDA	0.616	0.593	0.689	0.694
CNN	0.843	0.843	0.944	0.944
Attn+BILSTM	0.820	0.821	0.944	0.943
BERT(wo/fine-tuing)	0.700	0.700	0.762	0.754
BERT(fine-tuing)	0.890	0.876	0.965	0.954
MACE	**0.908**	**0.894**	**0.977**	**0.971**

Effects of Hyperparameters. We further analyze the impact of the two main hyperparameters in our model, i.e., the number of concepts K and the value of γ.

We conducted experiments on the impact of the number of concepts K corresponding to an entity in a short text on classification performance. The classification accuracy with the number of concepts on the test sets are shown in Figure 3. We can clearly see that achieves the highest accuracy with K=5. When K=0, no conceptual information is used, and the effect of using the BERT model alone is not as good as the model of using conceptual knowledge. It shows that a reasonable number of concepts will enable the model to achieve the best results on different datasets. However, an excessive number of concepts will result in a decrease in accuracy. The possible reason is that the increase in the number of concepts will confuse the semantics of short texts.

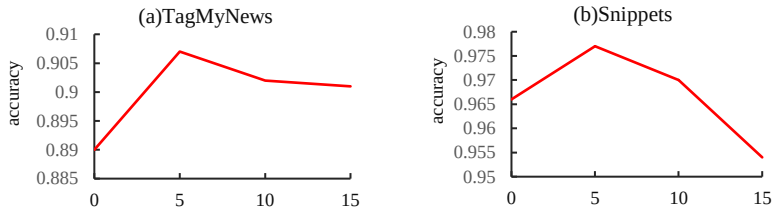

Fig 3. Influence of different K on accuracy.

To verify the effectiveness of the two attention mechanisms, we studied the influence of the parameter γ that adjusts the two attention weights on the results. Manually adjusting the parameter γ from 0 to 1 and the step size is 0.25. The experimental results are shown in Table 3. It can be seen from Table 3 that when γ = 0.50, the model effect achieves the best effect on both datasets. When the parameter γ is set to 0 or 1, the effects are both worse on two data sets.

Table 3. The effect of different γ on the accuracy.

	TagMyNews	Snippets
$\gamma = 0$	0.891	0.953
$\gamma = 0.25$	0.903	0.970
$\gamma = 0.50$	**0.908**	**0.978**
$\gamma = 0.75$	0.900	0.954
$\gamma = 1.00$	0.883	0.960

4 Conclusion

In this paper, we propose an attention network which can leverage both text information and concept embedding to represent short text. First of all, short text encoder is used to enhance the representation of short texts in the semantic space. In addition, concept encoder obtains the distributed representation of the concept through the two attention networks. Finally, Concatenating outputs from the two encoders creates a final target representation of short text. On two short text classification datasets, the results show that our proposed model outperforms traditional methods, deep learning methods and original BERT.

ACKNOWLEDGEMENTS

This work was supported in part by National Natural Science Foundation of China (No. 61762078，61967013), University Innovation and entrepreneurship Fund Project (2020B-089), Supported by science and technology program of Province (20JR5RA518)， Natural Science Foundation of Province (20JR10RA076).

References

1. Wang, Z., Wang, H.: Understanding short texts. In: The Association for Computational Linguistics(Tutorial). ACL, Stroudsburg, Pennsylvania (2016).

2. Bengio, Y., Ducharme, R., Vincent, P., Janvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. 3, 1137–1155 (2003).
3. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., et al: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, pp. 5998–6008. NIPS, La Jolla, CA (2017).
4. Wang, J., Wang, Z., Zhang, D., Yan, J.: Combining knowledge with deep convolutional neural networks for short text classification. In: 26th International Joint Conference on Artificial Intelligence, pp. 2915–2921. IJCAI.org, USA (2017). doi: 10.24963/ijcai.2017/406
5. Wang, Z., Wang, H., Wen, J., Xiao, Y.: An Inference Approach to Basic Level of Categorization. In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management, pp. 653–662. ACM, New York (2015) doi: 10.1145/2806416.2806533.
6. Devlin, J., Chang, M., Lee, K., Toutanova, K.: 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, pp. 4171–4186. ACL, Stroudsburg, Pennsylvania (2019). doi: 10.18653/v1/n19-1423
7. Zhang, S., Zheng, D., Hu, X., Yang, M.: Bidirectional long short-term memory networks for relation classification. In: Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation. ACL, Stroudsburg, Pennsylvania (2015).
8. Chen, J., Hu, Y., Liu, J., Xiao, Y., Jiang, H.: Deep Short Text Classification with Knowledge Powered Attention. In: The Thirty-Third AAAI Conference on Artificial Intelligence, pp. 6252–6259. AAAI Press, Palo Alto (2019). doi: 10.1609/aaai.v33i01.33016252
9. Vitale, D., Ferragina, P., Scaiella, U.: Classification of short texts by deploying topical annotations. In: Baeza, R., Zaragoza, H., Cambazoglu, B. B., Murdock, V., Lepml, R., Silvestri, F. (eds) ECIR 2012, LNCS, vol. 7224, pp. 376–387. Springer, Heidelberg (2012). doi: 10.1007/978-3-642-28997-2_32
10. Xuan, H. P., Nguyen, M. L., Horiguchi, S.: Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: 17th International Conference on World Wide Web, pp. 91–100. ACM, New York (2008). doi: 10.1145/1367497.1367510
11. Wang, S., & Manning, C.: Baselines and Bigrams: Simple, good sentiment and topic classification. In: 50th Annual Meeting of the Association for Computational Linguistics, pp. 90–94. ACL, Stroudsburg (2012).
12. Blei, D., Ng, A., & Jordan, M.: Latent Dirichlet Allocation. J. Mach. Learn. Res. 3, 993–1022 (2003).
13. Zhang, D., Wang, D.: Relation Classification via Recurrent Neural Network. CoPR abs/1508.01006 (2015).
14. Kim, Y.: Convolutional Neural Networks for Sentence Classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1746–1751. ACL, Stroudsburg, Pennsylvania (2014). doi: 10.3115/v1/d14-1181
15. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing pp 1532-1543. ACL, Stroudsburg, Pennsylvania (2014). doi: 10.3115/v1/d14-1162
16. Zeng, J., Li, J., Song, Y., Gao, C., Lyu, M., King, I.: Topic memory networks for short text classification. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 3120-3131. ACL, Stroudsburg, Pennsylvania (2018). doi: 10.18653/v1/d18-1351