Joint Representations of Texts and Labels with Compositional Loss for Short Text Classification

Ming Hao[1], Weijing Wang[2] and Fang Zhou[1,*]

[1]*School of computer and communication engineering, University of science and technology Beijing, Beijing 100083, China*
[2]*Department of bioengineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA*
E-mail: minghao@xs.ustb.edu.cn
[*]*Corresponding Author*

Received 09 March 2021; Accepted 01 April 2021;
Publication 02 June 2021

Abstract

Short text classification is an important foundation for natural language processing (NLP) tasks. Though, the text classification based on deep language models (DLMs) has made a significant headway, in practical applications however, some texts are ambiguous and hard to classify in multi-class classification especially, for short texts whose context length is limited. The mainstream method improves the distinction of ambiguous text by adding context information. However, these methods rely only the text representation, and ignore that the categories overlap and are not completely independent of each other. In this paper, we establish a new general method to solve the problem of ambiguous text classification by introducing label embedding to represent each category, which makes measurable difference between the categories. Further, a new compositional loss function is proposed to train the model, which makes the text representation closer to the ground-truth label and farther away from others. Finally, a constraint is obtained by calculating the similarity between the text representation and label embedding. Errors caused by ambiguous text can be corrected by adding constraints to the output layer of the model. We apply the method to three classical models and conduct experiments on six public datasets. Experiments show that our method can effectively improve the classification accuracy of the ambiguous texts. In addition, combining our method with BERT, we obtain the state-of-the-art results on the CNT dataset.

Keywords: Ambiguous text, deep language models, label embedding, text classification, triplet loss.

1 Introduction

Text classification is a fundamental task for natural language processing (NLP), and plays an important role in many applications, such as sentiment analysis [1, 2], question answering [3] and semantic understanding [4]. One of the key steps in these is to represent the text as a feature vector, which we call "text representation". To this end, a mass of works represents the text by designing various neural networks, such as convolutional neural networks (CNNs) [5–8], recurrent neural networks (RNNs) [9, 10], and neural networks with attention mechanisms [11, 12] have achieved great success.

However, the performance of the above model in short text classification is not ideal, this is due to the short text length and the lack of context information. Existing methods for solving these questions are expanding text information [13] proposed a unified framework to expand short texts via word embedding cluster and convolutional neural network [14] suggested a framework that conceptualizes a short text as a set of relevant concepts and merged it with short text representation by convolutional neural network [15] introduced an attention mechanism to measure the importance of knowledge and integrated the knowledge and text representation embedded by BLSTM and Self-attention.

By enriching semantic information, the above methods try to solve the problem of inaccurate short text classification to a certain extent. However, for short texts with strong ambiguity, these methods fail to accomplish the desired results since they assume that categories are strictly independent, and use One-hot vector to represent each category. Since the One-hot vectors are orthogonal to each other, the product between the vectors is zero, so the similarity between categories is not measurable. However, in practical applications, the categories overlap with each other. For example, the phrase "Jay Chou married in Selby church" can easily be classified as "Entertainment" rather than "Travel", but it is hard to distinguish between "Entertainment" and "Celebrity".

In order to solve this problem, we first introduce label embedding to represent each category, so as to measure the similarity between the categories. We define the ground-truth labels as positive labels l^p, and others as negative labels l^n. The text representations can be trained closer to l^p and farther from l^n, thereby increasing the discrimination of similar categories. Then, we calculate the reciprocal of Euclidean distance between the text representation and the label embedding as a constraint, and add it to the output layer of the neural network to correct the error of the original model.

Our method is very effective, since the lower the similarity between the text representation and the label embeddings, the larger is their Euclidean distance, and closer is the reciprocal value to zero resulting in a lesser impact on the decision-making results of the model. On the contrary, the higher the similarity between the text representation and the label embedding, the smaller is their Euclidean distance and greater is the reciprocal value resulting in a greater impact on the decision-making results of the model. Therefore, when the model makes classification decision, the constraint term assists the decision-making on the categories with low discrimination; at the same time not affecting the categories with high discrimination. In our method, we propose a new compositional loss function comprising of cross-entropy loss and triple loss. At the same time, we optimize triplet loss to fit our requirements.

In order to verify our method, we have carried out experiments on six public short text datasets. We apply our method to three basic models (CNN, LSTM, BLSTM), and the experiments show that our method improves the classification accuracy of the three models, which proves that this method can be effectively used in the classification of ambiguous texts. At the same time, we apply our method to the BERT model and get the best results on the CNT dataset.

The main contributions of this paper are summarized as follows:

- We proposed a method which uses the similarity between the text representation and the label embedding as constraints to help the model classify ambiguous texts.

- We suggested a novel compositional loss function, which can train the text representation to be closer to the l_p and farther away from the ln.
- Experiments on six public datasets show that our method is robust and improves the classification accuracy of multiple models. In addition, combining our method with BERT, we can get the state-of-the-art results on the CNT dataset.

The rest of this paper is organized as follows. Section 2 gives a brief review of related works in this field. The preliminaries for the text classification are elaborated in Section 3. A conceptual foundation framework of our research is presented in Section 4. An extensive experimental results and analysis is covered in Section 5. Finally, the conclusions are summarized in Section 6.

2 Related Works

For a long time, obtaining an excellent text representation is the goal of deep language model. Nowadays, language models with pre-training have exhibited good results. Pre-training can be seen as injecting basic knowledge into the model, e.g., word embeddings and fine-tuning models. Jeffrey et al. [16] proposed the global vectors for word representation (GloVe) trained by five corpuses. Peters et al. [17] suggested embeddings from language models (ELMo) pre-trained by Billion Word Benchmark [18].

For fine-tuning the models, Radford et al. [19] proposed the generative pre-training (GPT) trained by the BooksCorpus dataset [20], containing over 7,000 unique unpublished books from a variety of genres including Adventure, Fantasy and Romance. Jacob et al. [21] proposed bidirectional encoder representations from transformers (BERT) pre-trained on both – the BooksCorpus (800M words) and Wikipedia (2,500M words). However, especially in the task of short text classification, due to the short text length and insufficient context information, the above methods are not effective.

Label embedding has been proved to be powerful information and is effective in various domains and tasks, such as image classification [22], text recognition in images [23] and multi-task learning [24]. By visualizing the embeddings of both documents and labels on a 2D map by t-SNE [25] and finding each embedding of label falling into the internal region of the respective manifold, proved that the label embeddings are strongly correlated with the documents. However, the above method only takes label embedding as the supplementary semantics of features, but ignores its supervision function.

Triplet loss [26] aims to separate the positive pair from the negative and this approach is widely used in different areas: [27] proposed a deep binary embedding for the image retrieval task. It is particularly successful on the task of person re-identification task; [28] indicated that models can be trained using a variant of the triplet loss; [29] suggested an improved triplet loss function to solve the problem of non-overlapping fields of view between cameras; [30] design a quadruplet loss, a function which can lead the model output with a larger inter-class variation and a smaller intra-class variation compared to the triplet loss. However, as far as we know, there are few applications of triplet loss in the field of text classification.

Inspired by the above methods, we utilized the pre-training model to generate text representation, and employed label embedding to represent categories. Triplet loss is used to train text representation to make it closer to the real label. It can solve the problem of misclassification of ambiguous texts.

3 Preliminaries

In this section, we briefly introduce classical text classification methods: *Cross-Entropy Based Model*. Throughout this paper, we denote vectors as lower-case letters, amd matrices as upper-case letters. Meanwhile, we denote activation function as bold lower-case letters. Given a set of pair-wise data $\{(x^n, y^n), n = 1, 2, \ldots, N\}$, where $x \in \mathcal{X}$ is the text sequence composed of tokens, and $y \in \mathcal{Y}$ is the corresponding label of x , n is the amount of data. Our goal is to learn a classification model based on deep neural network namely, a complex mapping function $\boldsymbol{f} : \mathcal{X} \mapsto \mathcal{Y}$. As shown in Figure 1, $\boldsymbol{f}$ can be briefly divided into three steps:

- E: Represents the text sequence x into a fixed-size feature vector $e_x \in \mathbb{R}^d$, namely text representation.
- $Dense$: Adjusts the dimension of e_x through a dense layer.

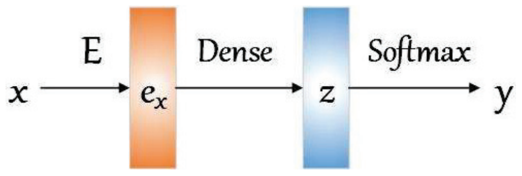

Figure 1 Process of traditional text classification model.

- $Softmax$: Generates probabilities of the adjusted representation e_x, in each category via the $Softmax$ function and selects the category which has the highest probability of prediction $\hat{y}$, as shown in Equations (1) and (2).

Finally, the model is optimized by minimizing the loss function of cross-entropy shown in Equation (3).

$$(y|e_x) = Softmax(e_x) \tag{1}$$

$$\hat{y} = argmax(p(y|e_x)) \tag{2}$$

$$\boldsymbol{loss} = -y_n \sum_{N} \log \hat{y}_n \tag{3}$$

More specifically, E is divided into two steps: E_1 employs a pre-trained word vector dictionary $D \in \mathbb{R}^{d \times |V|}$ to embed x into a feature matrix sized $t \times d$, where, $|V|$ is the size of the vocabulary V, d is the dimension of the word vector, and t is the length of the text sequence. E_2 utilizes a variety of neural networks to represent the feature matrix into a fixed-size feature vector e_x, which can be used for classification.

Substantial work [31, 32] is devoted to devise proper functions E_1 and E_2, such as pre-training word vector and designing the task specific architecture. Meanwhile, model pre-training is also proved to be effective in improving the performance of the text representation whereas, some researchers pre-trained the model composed of E_1 and E_2, fine-tuned the model in specific-task to achieve better performance, and few even outperformed humans in specific-tasks.

4 Proposed Conceptual Framework

By reviewing the three steps in the traditional pipeline of text classification, we understand that most of the methods strive for E_1 and E_2, namely, to enhance the representation ability of the text. However, they ignore an important premise that the categories are independent of each other. In practical applications, however, we face two dilemmas: 1. Unclear categories: two or more categories are similar but have lots of intersections. 2. Ambiguity of the text: the text can either be classified into category A or it can also be classified into category B.

In response to these two problems, we propose a method of combining the traditional DML method with the triplet loss method, because the triplet loss

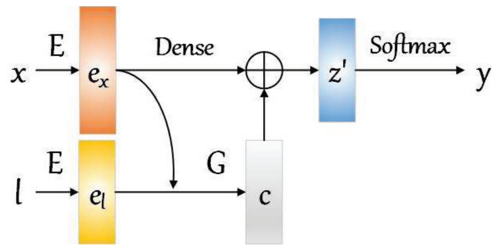

Figure 2 A brief introduction of our proposed model.

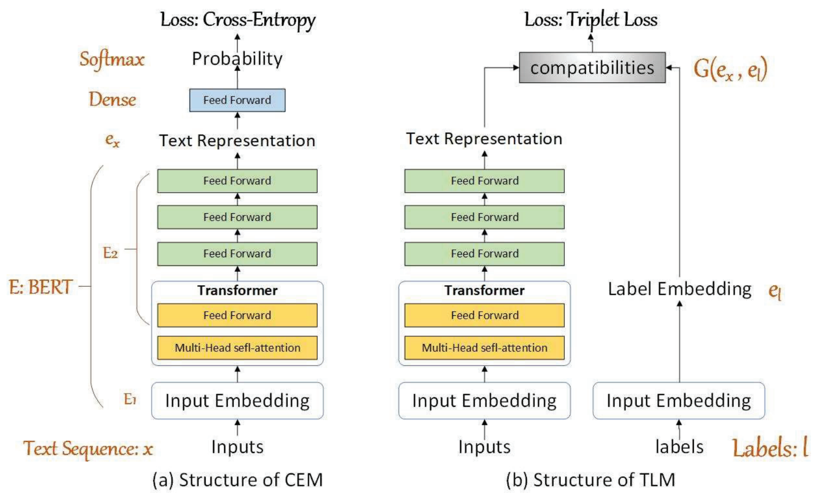

Figure 3 Structures of two models. The left is CEM and the right is TLM.

itself has a measure of otherness, which helps to distinguish the details. To measure the otherness of each category, we introduce the label embeddings e_l. The compatibility between e_x and e_l is calculated as a constraint term to improve the model's discrimination of ambiguous texts, as shown in Figure 2. In this section, we explain in detail form 3 aspects: *Cross-Entropy based model* (CEM), *Triplet Loss Based Model* (TLM) and *Joint Trained Model* (JTM).

4.1 Cross-Entropy Based Model

In this paper, we aim to improve the performance of CEM. So we build a CEM as a basic model. More specifically, we choose Bidirectional Encoder Representations from Transformers (BERT) as E.

As shown in Figure 3(a), E_1 is the “Input embedding” which is the sum of token embeddings (or word embedding), segment embedding and position embedding. E_2 is the transformer composed of self-attention and the feed forward network.

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4}$$

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \tag{5}$$

E is pre-trained by Masked LM and Next Sentence Prediction. Besides, in order to achieve the better performance, we add three feed forward networks into E_2 after transformer. e_x can be classified by $Softmax$ and the whole model is optimized by Cross-Entropy loss function.

4.2 Triplet Loss Based Model

To solve the problem of unclear category, we propose Triplet Loss Based Model as shown in Figure 3(b). This method can be roughly divided into three parts: Label embeddings, Constraint calculation and Training.

4.2.1 Label embedding

Most traditional DMLs represent labels as a one-hot vector to guarantee that categories are orthogonal, i.e., each category is independent of each other. However, this representation method has two downsides: 1. It ignores the labels’ information, which results in the classification mainly depending on the text-level representation; 2. Categories are not strictly independent in practical situations, and run counter to the assumption of “each category is independent of each other”.

Therefore, we employ the real-value vectors to represent the labels. This not only utilizes the information of labels, but also gives a measurement for each category. We select the most representative word in each category and embed it into a matrix sized $c \times d$ by D, where c is the number of classes. We refer the label embedding in each category to e_{l^c}.

4.2.2 Constraint calculation

Similar to CEM, we obtain e_x by BERT and adjust its dimension to d by $Dense$ layer. Then, we calculate the Euclidean distance between e_x and label embeddings as their compatibilities by Equation (6). Finally, we select the minimum $G(e_x, e_l)$ as our prediction as Equation (7). We hope that e_x is

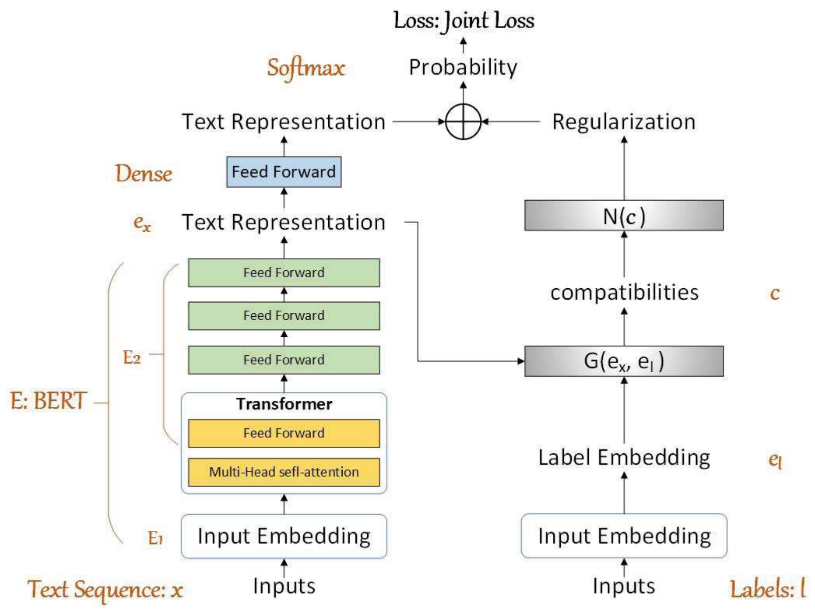

Figure 4 Structures of joint trained model.

closer to l_p than l_n, so we borrow the idea of triplet loss and design a new loss function as Equation (8) to train the model.

$$G(e_x, e_{l_c}) = \sqrt{\sum_{1}^{d} (e_{x_i} - e_{l_i^c})^2} \tag{6}$$

$$\hat{y} = argmin(G(e_x, e_l)) \tag{7}$$

$$loss = \frac{c \cdot G(e_x, e_{l^p})}{\sum G(e_x, e_{l^n})} \tag{8}$$

4.2.3 Joint trained model

In traditional Cross-Entropy based methods, models will select the category with the highest probability as predication. However, models will be confused between one or two categories, when the text is ambiguous since their probabilities are similar. To solve this problem, we add the constraint computed by e_x and e_l to the original feature (namely, e_x itself) to enhance the distinction between the ambiguous text as shown in Figure 4.

More specifically, we take the reciprocal of G. It will have little influence on irrelevant categories excluded by CEMs, but it can increase the distinction of similar categories and guide the model to make the right classify. To align the distributions of e_x and the constraint, we normalize them before Equation (9). Finally, the whole model is optimized by new joint training loss as Equation (10), where α and β are impact factors between CEM and TLM. We control the influence of constraints by setting the value of α and β. Generally, the values of α and β are between 0.2 and 0.8.

$$z' = Softmax(e_x) + \alpha \frac{1}{\log(G(e_x, e_l))} \tag{9}$$

$$loss = -y_n \sum_{N} \log \hat{y}_n + \beta \frac{c \cdot G(e_x, e_{l^p})}{\sum G(e_x, e_{l^n})} \tag{10}$$

5 Experimental Results and Analysis

In this section, we evaluate the performance of the proposed method based on six benchmark datasets for text classification. We make a brief introduction to the datasets in 5.1. The details of the experiment settings are description in 5.2. The performance of the proposed model is evaluated by comparing it with classical model based on Cross-Entropy in 5.3.

5.1 Datasets

In this section, we introduce six datasets used in our experiments.

SST: This is the Stanford Sentiment Treebank dataset, an extension of a movie review dataset but with train/dev/test splits provided as fine-grained labels (very positive, positive, neutral, negative, very negative). The class distribution is 1837/3118/2237/3147/1516. In this paper, we only use the training and testing sets for experiment.

AG: We obtained the AG's News corpus from [33], since they choose the four largest categories from this corpus to construct the dataset, using only the title and description fields. In this paper, we view the title and the description field of the text.

TREC: This is a question classification dataset [34]. The task involves dividing a question into 6 question types (abbreviation, description, entity, human, location, numeric value).

Table 1 Characteristics of datasets

Dataset	Train Size	Test Size	Num_class	Max-length	Avg-length
SST	8,544	2,210	5	53	19
AG	120,000	7,600	4	249	43
TREC	5,452	500	6	37	9
CNT	47,850	15,590	32	72	18
Sogou	50,000	10,000	10	46	27
THUC	41,000	11,200	14	46	19

CNT: This dataset is obtained from [35]. The dataset contains 47,952 titles comprising of 32 classes for training data and 15,986 titles for testing. After deleting the titles containing special characters that cannot be processed, we retain 47,850 training titles and 15,950 testing titles.

Sogou: Sogou news consists of ten categories of social media news. There are 50,000 datasets for training and 10,000 for testing. Each news item contains a title and its context, but we used only the title to construct the dataset.

THUC: This dataset was established by a laboratory in Tsinghua University. It includes 14 types of social hot topics, such as sport, education, technology, etc., including 41,000 datasets for training and 11,200 datasets for testing. Each sample has a title and its context but we used only the title.

Summary statistics of the datasets used are displayed in Table 1. Three of them are Chinese datasets (CNT, Sogou, THUC) and others are English datasets (SST, AG, TREC). Meanwhile, AG can be seen as the long text and others as short text.

5.2 Experiment Setting

We divide the data into a tuple *(label, text)*. For label, we use provided tokens and their synonyms if the datasets have (CNT, THUC, Sogou, TREC). If dataset use numbers as the labels, we design the label tokens for each class of dataset (for example, we employ “Terrible”, “Negative”, “Neuter”, “Positive”, “Wonderful” for SST and “Politics and Governments”, “Sports and Competitions”, “Finances and Economics”, “Information and Technologies” for AG).

We embed the label and the text tokens in two ways: word embedding and BERT based sentence embedding. We view the token of the label as a sentence to embed. For English embedding, we segment the texts by NLTK [36] and use 300 dimensional GloVes [16] to embed them. For Chinese embedding, to compare with BERT, we use 300dimensional character-level embedding proposed by [37]. Meanwhile, we use BERT to embed each word, but the performance is worse than word embedding hence, we do not report these results in Table 2. For BERT based sentence embedding, we employ the pre-trained BERT model in different corpus with same structure (12 layers, 768 hidden units and 12 attention heads) to embed Chinese and English text. We choose the pre-trained model which is not sensitive to English capitalization. Besides, we only need to fine-tune the dense layer in the BERT based model. It is worth mentioning that in our model, label embeddings are not trained with the whole model. At the same time, in Bert based embedding, we only train three full connection layers, not the Bert itself.

Python version 3.5 and Tensorflow version 1.16 [38] are employed to build the neural networks. The models using our method are all optimized by Adam [39], with an initial learning rate of 0.001. For regularization, we selected the dropout rate [40] in the range of [0.1, 0.99], and we set 0.4 for the word embedding layer, 0.8 for the hidden layer and 0.8 for the penultimate layer by a grid search. We also use the l2 penalty with a coefficient 0.005 over the parameters and the Intel 8700k CPU, 16g memory, two Nvidia GeForce 1080Ti as the GPU cluster hardware.

5.3 Contrast Experiment

5.3.1 Classification Tasks

We apply our method to three classical classification models: Convolutional Neural Network (CNN), Long Short Term Memory (LSTM) and Bidirectional Long Short Term Memory (BLSTM), and experiment on aforesaid six public datasets, to verify its versatility and robustness. This can be used in many models. In this section, we compare our method with three classical methods: That is, we employ these three classical models as E to generate text representations. Besides, in order to verify the performance of triplet loss based model, we implement the TLM model using text representation generated by CNN, LSTM and BLSTM.

5.3.2 Results and analysis

The experimental results are shown in Table 2. It is observed that the classification accuracy of the three classical models has improved to a certain extent after application our method. Among them, CNN has the largest improvement. This indicates that our method has strong robustness. At the same time, we find that our method is not effective on Ag dataset, because

Table 2 Result of comparison experiment. ‘Model’ refers to the embedding model for text; ‘Loss’ refers to the loss function used for training model.

Model	Loss	SST	AG	TREC	CNT	Sogou	THUC
CNN	CEM	45.52	**93.36**	93.60	75.86	92.46	88.94
	TLM	45.29	92.30	93.20	73.81	90.25	87.90
	JLM	**45.78**	93.29	**94.40**	**77.33**	**92.53**	**89.70**
LSTM	CEM	45.39	**93.20**	91.60	76.37	89.23	89.27
	TLM	**45.48**	92.28	93.20	**77.33**	89.01	87.70
	JLM	**45.48**	92.79	**93.60**	77.24	**90.93**	**89.42**
BLSTM	CEM	**46.56**	92.63	**93.00**	76.28	89.76	89.11
	TLM	43.67	92.37	92.20	73.29	88.09	85.75
	JLM	44.80	**92.84**	92.80	**77.46**	**90.08**	**90.08**

Table 3 Test Accuracy on CNT classification tasks, in percentage

Model	BoWs	CNN	LSTM	BLSTM	LEMA	BERT	Ours
Accuracy	76.3	75.86	76.37	76.28	79.55	80.25	**84.13**

Ag dataset is a long text dataset, and there are few ambiguous texts. This shows that our method can mainly be used in short text classification task.

By comparing with CEM and TML, we find that TML proposed in this paper indeed achieves comparable performance to that of the classical methods. In some cases, however, TML improves pure classical models. This may be attributed to the fact that the label information is added to the e_x and at the same time, these datasets are sensitive to the label.

5.4 Classification on Benchmark Datasets

To demonstrate the practical value of our method, we combine our method with BERT, and test this model on Chinese News Titles dataset. In addition to the above three models, we also compare the three classification models: (1) the bag-of-words in, (2) Label-Embedding Attentive Model (LEAM), (3) BERT. To quantify the prediction performance, we utilize the classification accuracy as the evaluation standard. The results are shown in table.

5.4.1 Results and analysis

As show in Table 3, our method with BERT yields the best score. In particular, our model is nearly 4% better than BERT and 8.27% higher than CNN. This

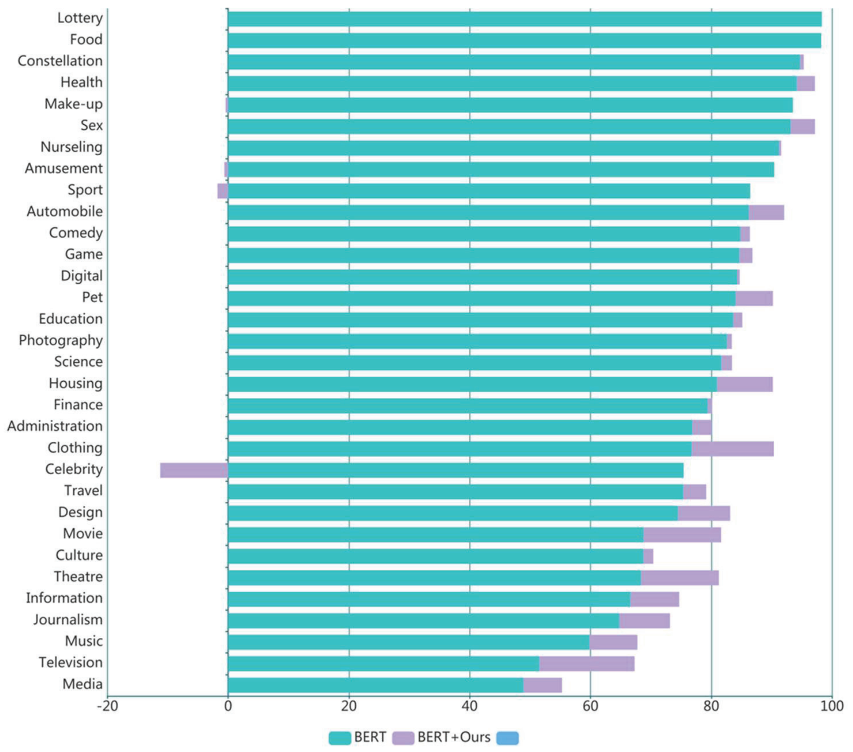

Figure 5 Compare the classification accuracy of BERT and our method. For the convenience of comparison, the purple part is the difference between our method and the BER classification accuracy.

shows that our method combined with the latest model gives better results. In the LEAM, label embedding is also used to enhance the ability of text representation giving better results than BERT but higher than the baseline, which proves that LEAM can effectively improve the accuracy of short text classification. Compared with LEAM, our method is more general and can be combined with more advanced models to achieve better results.

As shown in Figure 5, in addition to the "Celebrity" class, the addition of our method has little negative impact on the original model. On the contrary, the classification accuracy of "Media", "Television", "Music" and "Movie" has improved significantly, and these categories closely overlap each other. These results further support that our method is effective.

Of course, we also adjust the parameters for α and β. We find that the best effect is when α and β are between 0.6 and 0.8. Therefore, we can conclude that our method does affect the classification decision of the model to a certain extent.

6 Conclusion and Future work

Traditional text classification methods are always plagued by ambiguous texts since they focus only on the text itself. In this paper, a robust and effective method is proposed. Different from the traditional method wherein, we use label embeddings as a constraint to help these models accurately classify the ambiguous texts. This constraint is the normalized Euclidean distance between the text representation and label embeddings. What's more, a new loss function composed of cross-entropy and adjusted triplet loss is proposed for joint training of text representation and label embedding, and its performance is better than that of simply cross-entropy loss. Through this method, the classification ability of the model can be improved. Experiments on six open short text classification datasets show that the proposed method can effectively improve the classification accuracy of the model. Meanwhile, we combine our method with BERT to obtain the state-of-the-art results on the CNT dataset.

Although our method is very effective, there is still a lot of work to be explored, such as the relationship between label's token selection and ambiguity elimination, whether the semantic space of label embeddings can be different from text representation, etc.

Acknowledgement

This work is supported by National Key R&D Program of China (2019YFB1405900), National Natural Science Foundation of China (62006018), Fundamental Research Funds for the Central Universities (FRF-TP-18-060A1), and the Fundamental Research Funds for the Central Universities and USTB-NTUT Joint Research Program.

References

[1] Hu, Q., et al. SNNN: Promoting Word Sentiment and Negation in Neural Sentiment Classification. 2018. {AAAI} Press.

[2] Wu, Z., et al. Improving Review Representations With User Attention and Product Attention for Sentiment Classification. 2018. {AAAI} Press.

[3] Anderson, P., et al. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. 2018. {IEEE} Computer Society.
[4] Dong, X. and G. de Melo. A Helping Hand: Transfer Learning for Deep Sentiment Analysis. 2018. Association for Computational Linguistics.
[5] Kalchbrenner, N., E. Grefenstette, and P. Blunsom, A Convolutional Neural Network for Modelling Sentences. 2014. 1 %6: p. 655–665 %&.
[6] Kim, Y. Convolutional Neural Networks for Sentence Classification. 2014. {ACL}.
[7] Zeng, D., et al. Relation Classification via Convolutional Deep Neural Network. 2014. {ACL}.
[8] Conneau, A., et al. Very Deep Convolutional Networks for Text Classification. 2017. Association for Computational Linguistics.
[9] Zhou, P., et al. Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling. 2016. {ACL}.
[10] Lee, J.Y. and F. Dernoncourt. Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks. 2016. The Association for Computational Linguistics.
[11] Zhou, P., et al. Attention-based bidirectional long short-term memory networks for relation classification. 2016.
[12] Ma, F., et al. Dipole: Diagnosis Prediction in Healthcare via Attention-based Bidirectional Recurrent Neural Networks. 2017.
[13] Wang, P., et al., Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. Neurocomputing, 2016. 174 %6: p. 806–814 %&.
[14] Wang, J., et al. Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification. 2017. ijcai.org.
[15] Chen, J., et al. Deep Short Text Classification with Knowledge Powered Attention. 2019. {AAAI} Press.
[16] Pennington, J., R. Socher, and C.D. Manning. Glove: Global Vectors for Word Representation. 2014. {ACL}.
[17] Peters, M.E., et al. Deep Contextualized Word Representations. 2018. Association for Computational Linguistics.
[18] Chelba, C., et al. One billion word benchmark for measuring progress in statistical language modeling. 2014. {ISCA}.
[19] Radford, A., et al., Improving language understanding by generative pre-training. Proceedings of Technical report, OpenAI, 2018. %6: p. %&.

[20] Zhu, Y., et al. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. 2015. {IEEE} Computer Society.

[21] Devlin, J., et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019. Association for Computational Linguistics.

[22] Akata, Z., et al., Label-Embedding for Image Classification. CoRR, 2015. abs/1503.08677 %6: p. %&.

[23] Rodr'i, g.-S.J.e.A. and F. Perronnin. Label embedding for text recognition. 2013. {BMVA} Press.

[24] Frome, A., et al. DeViSE: A Deep Visual-Semantic Embedding Model. 2013.

[25] Maaten, L.v.d. and G. Hinton, Visualizing data using t-SNE. Journal of machine learning research, 2008. 9 %6(Nov): p. 2579–2605 %&.

[26] Schroff, F., D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. 2015. {IEEE} Computer Society.

[27] Zhuang, B., et al. Fast Training of Triplet-Based Deep Binary Embedding Networks. 2016. {IEEE} Computer Society.

[28] Hermans, A., L. Beyer, and B. Leibe, In Defense of the Triplet Loss for Person Re-Identification. CoRR, 2017. abs/1703.07737 %6: p. %&.

[29] Cheng, D., et al. Person Re-identification by Multi-Channel Parts-Based CNN with Improved Triplet Loss Function. 2016. {IEEE} Computer Society.

[30] Chen, W., et al. Beyond Triplet Loss: A Deep Quadruplet Network for Person Re-identification. 2017. {IEEE} Computer Society.

[31] Wang, S., J. Zhang, and C. Zong, Empirical Exploring Word-Character Relationship for Chinese Sentence Representation. {ACM} Trans. Asian Low Resour. Lang. Inf. Process., 2018. 17 %6(3): p. 14:1–14:18 %&.

[32] Johnson, R. and T. Zhang. Deep pyramid convolutional neural networks for text categorization. 2017.

[33] Zhang, X., J.J. Zhao, and Y. LeCun. Character-level Convolutional Networks for Text Classification. 2015.

[34] Li, X. and D. Roth. Learning Question Classifiers. 2002.

[35] Zhou, Y., et al. Compositional Recurrent Neural Networks for Chinese Short Text Classification. 2016. {IEEE} Computer Society.

[36] Loper, E. and S. Bird, NLTK: The Natural Language Toolkit. CoRR, 2002. cs.CL/0205028 %6: p. %&.

[37] Li, S., et al. Analogical Reasoning on Chinese Morphological and Semantic Relations. 2018. Melbourne, Australia: Association for Computational Linguistics.
[38] Abadi, M.i.n., et al. TensorFlow: A System for Large-Scale Machine Learning. 2016. {USENIX} Association.
[39] Kingma, D.P. and J. Ba. Adam: A Method for Stochastic Optimization. 2015.
[40] Hinton, G.E., et al., Improving neural networks by preventing co-adaptation of feature detectors. CoRR, 2012. abs/1207.0580 %6: p. %&.

Biographies

Ming Hao is a Ph.D. student at the University of Science and Technology Beijing, China. He attended the Taiyuan University of Technology, China where he received his B.Sc. in Software Engineering in 2013 and began to study for an M.Sc. in Software Engineering from the University of Science and Technology in Beijing, China in 2014. From 2015 to 2019, Ming worked for the Institute of Automation, Chinese Academy of Science, engaged in algorithm research in the field of natural language processing, and from 2019 to 2020, he was a visiting scholar with the Department of Computer Science, the University of Illinois at Urbana-Champaign, USA. His research interests include machine learning and natural language processing and reinforcement learning.

Weijing Wang is a Ph.D. student at the University of Illinois at Urbana Champaign since fall 2019. She attended the Hubei University of Chinese medicine and received her B.S there in 2019. Weijing Wang is currently completing a doctorate in Bioengineering at the University of Illinois at Urbana Champaign. Her Ph.D. work centers on smartphone-based point of care devices and micro and nano based research, she also received her master of engineering degree from this university.

Fang Zhou received the B.Sc, M.Sc and Ph.D degree in computer science from the University of Science and Technology Beijing, China, in 1995, 2002 and 2012. From 2015 to 2016, she was a Visiting Researcher with the Department of Computer and Information Sciences, Temple University, USA. She is currently an Associate Professor with the Department of Computer Science and Technology, University of Science and Technology Beijing. Her research interests include machine learning, information retrieval and computer vision.