A Short Text Classification Approach with Event Detection and Conceptual Information

Wei YIN
Department of Computer Science and Engineering
School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
Shanghai, China
yinwei@sjtu.edu.cn

Liping SHEN
Department of Computer Science and Engineering
School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
Shanghai, China
lpshen@cs.sjtu.edu.cn

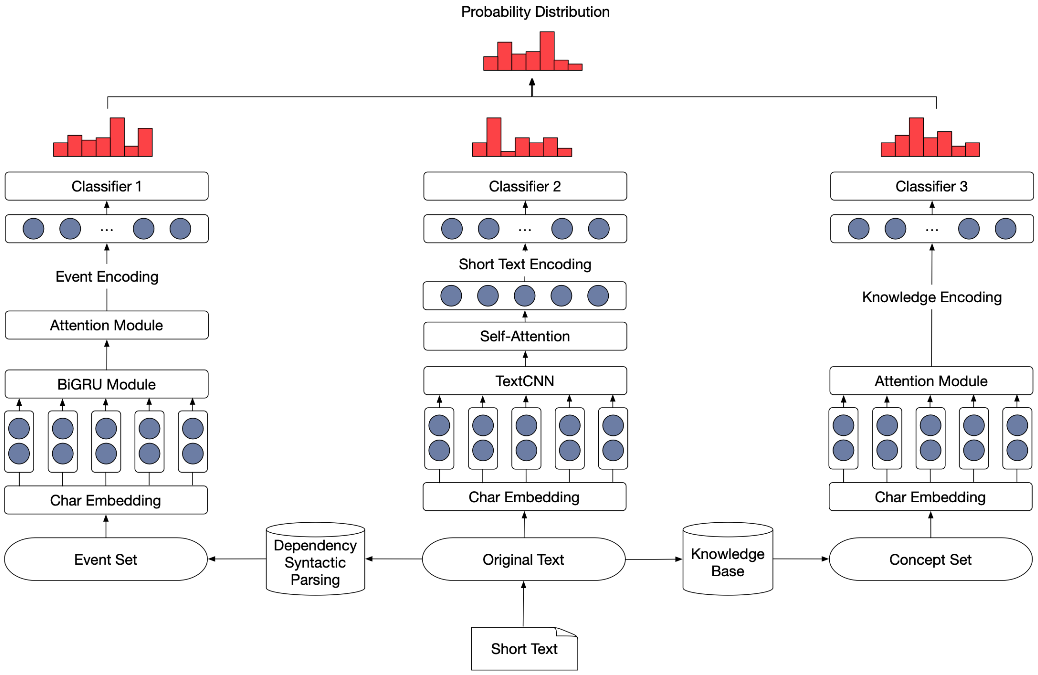

Figure 1: Model Architecture

ABSTRACT

Text classification is an elementary task in Natural Language Processing (NLP). Existing methods, such as Long Short-Term Memory Networks (LSTM) and Attention Mechanism have recently achieved strong performance on multiple NLP related tasks. However, in the field of text classification, their results are often limited by the quality of feature extraction. This phenomenon is particularly prominent in short text classification tasks, since short text does not have enough contextual information compared to paragraphs and documents. To address this challenge, in this article, we propose a method to enhance the semantic information of short text with two aspects: event-level information extracted from text and conceptual information retrieved from external knowledge base. We take event and conceptual information as a type of supplementary knowledge and incorporate it into deep neural networks. Attention mechanism is utilized to measure the importance of the supplementary knowledge. Meanwhile, we have discussed the granularity selection for Chinese word segmentation, and select char-based models. Finally, we classify a short text with the help of event and conceptual information. The experimental results show that the proposed method outperforms the state-of-the-art methods.

CCS CONCEPTS

• **Computing methodologies → Information extraction**.

ICMLT 2020, June 19-21, 2020, Beijing, China
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7764-5/20/06...$15.00
https://doi.org/10.1145/3409073.3409091

KEYWORDS

short text classification; attention mechanism; event extraction; concept knowledge

ACM Reference Format:

Wei YIN and Liping SHEN. 2020. A Short Text Classification Approach with Event Detection and Conceptual Information. In *Proceedings of ACM Conference (ICMLT 2020).* ACM, Beijing, China, 7 pages. https://doi.org/10.1145/3409073.3409091

1 INTRODUCTION

Short text has gradually become the most common text form. Short text is mainly generated in scenarios such as bullet screen interaction, product evaluation, and news headlines. Enterprises need to understand the actual needs of users based on this information in order to introduce better services and products. Relevant government departments can also monitor public opinion based on these data.

For the short text classification problem, there are two main features, one is that the average text word count is short, and the other is that the text contains less information.Compared with other forms of text, short text, as the name implies, there doesn't exist sufficient contextual information. As a consequence, its semantics are ambiguous and difficult to classify. In order to overcome this problem, the current mainstream idea is to add more information, which can be divided into two types of methods: explicit information and implicit information.

For explicit representation, a short text is represented as a sparse vector where each dimension is an explicit feature, corresponding to syntactic information of the short text including n-gram, POS tagging and syntactic parsing [1] Researchers develop effective features from many different aspects such as knowledge base and the results of dependency parsing. The explicit model is interpretable and easy to understand for human beings. However, the explicit representation usually ignores the context of short text and cannot capture deep semantic information.

In terms of implicit representation, a short text is usually mapped to an implicit space and represented as a dense vector [2]. The implicit model is good at capturing syntax and semantic information in short text based on deep neural networks. However, it ignores important semantic relations such as is_A and is_Property_Of that exist in Knowledge Bases (KBs). Such information is helpful for the understanding of short texts, especially when dealing with previously unseen words. However, this method also has certain limitations, because the process of feature learning requires a large amount of data.

A typical original input text consists of characters and numbers, which cannot be processed directly by a computer. As a consequence, at first, text data need to be encoded into the corresponding vector. For Chinese Language Processing, whether word segmentation is necessary and the granularity of word segmentation have become a key issue.

In this paper, we propose a novel model for short text classification based on event-level information extracted from text and conceptual information retrieved from external knowledge base.We take event and conceptual information as a type of supplementary knowledge and incorporate it into deep neural networks. We utilize attention mechanism to measure the importance of the supplementary knowledge. Meanwhile, we have discussed the granularity selection for Chinese word segmentation, and select char-based models. Finally, we classify a short text with the help of event and conceptual information. The experimental results show that the proposed method outperforms the state-of-the-art methods.

The remaining of this paper is structured as follows. In Section II, we review related work about short text classification and neural network. Section III presents our model in detail. In Section IV, we explain details about the experiments. Finally, we conclude the paper in Section V.

2 RELATED WORK

Over the years, numerous approaches have been proposed for short text classification and Chinese word segmentation. In traditional text classification approaches, explicit and statistical information is widely used in features engineering. In addition, some derivative features have been designed, such as part-of-speech tags [3], noun phrases and tree kernels [4].

Recently, deep neural networks can learn under-lying features automatically and have been used in the language processing. Most representative progress was made by Zeng et al. [5], who utilized convolutional neural networks (CNN) for relation classification. While CNN is not suitable for learning long-distance semantic information, thus apart from CNN, our approach also contains Recurrent Neural Network (RNN). LSTM model is an improved RNN model and it can learn long-term dependencies. Zhou et al. [6] applied Bidirectional Long-Short Term Memory (BiLSTM) to the semantic analysis of the text. By merging the state of the forward and backward transfer layers and making full use of the contextual information, more semantic features were achieved.

A modified version of LSTM, gated recurrent unit (GRU), was proposed to make each recurrent unit to adaptively capture dependencies of different time scales. Similarly to the LSTM unit, the GRU has gating units that modulate the flow of information inside the unit, however, without having separate memory cells.

Recent years have witnessed tremendous success of word embedding in various NLP tasks [7][8]. The basic idea behind is to learn the distributed representation of a word using its context. Among existing approaches, the continuous bag-of-words model (CBOW) and Skip-Gram model are simple and effective, capable of learning word embeddings efficiently from large-scale text corpora [9][10][11].

Besides its success in English, word embedding has also been demonstrated to be extremely useful for Chinese language processing [12][13]. The work on Chinese generally follows the same idea as on English, i.e., to learn the embedding of a word on the basis of its context. However, in contrast to English where words are usually taken as basic semantic units, Chinese words may have a complicated composition structure of their semantic meanings. Under this circumstance, Chen et al. (2015) introduced a character-enhanced word embedding model (CWE), which learns embeddings jointly for words and characters but ignores radicals. Sun et al.[14] and Li et al.[15] utilized radical information to learn better character embeddings. Similarly, Shi et al.[16] split characters into small components based on the Wubi method and took into account those components during the learning process. We will discuss Chinese text segmentation granularity problem in this paper.

3 OUR MODEL

In this section, we propose our model in detail. Figure 1 illustrates the logic structure of our model. The current mainstream idea for short text classification is to add more information beyond the short text itself. Based on this idea, we have constructed a model structure with the following characteristics:

- For Chinese text classification tasks in the context of deep learning, character granularity is selected in instead of word granularity in the pre-processing stage
- Added event-level information extracted from text as supplementary information
- Added concepts information acquired from external knowledge graphs as supplementary information

In the following paragraphs, we introduce the model structure in detail.

3.1 Chinese Short Text Preprocessing

Given a piece of Chinese short, at first, we apply basic data preprocessing procedures. In the context of deep learning, the segmented words are usually treated as the basic units for operations (word-based models). Each segmented word is associated with a fixed-length vector representation, which will be processed by deep learning models in the same way as how English words are processed. Word-based models come with several fundamental disadvantages, as will be discussed below, such as data sparsity and out-of-vocabulary problem. As a consequence, after apples-to-apples comparison we decide to utilize character based (char-based) models in preprocessing stage. Relevant experiment detail will be discussed in section 4.

3.2 Input Embedding

The input consists of two parts: short text s of length n and concept set C of size m. According to the granularity selection for Chinese language processing task proposed by Yuxuan et al.[17], we choose character as the granularity unit in data pre-processing stage. Character embedding layer is responsible for mapping each word to a high-dimensional vector space.

3.3 Event Extraction

As is shown in figure 2, we conduct an event extraction method based on the triplet of $\langle subject, verb, object\rangle$ [18] since the event-level information contains more explanatory knowledge. In our model Stanford's dependency analysis tool is used to conduct a dependency analysis for each sentence in a document in order to obtain the dependency structure. Based on the dependency structure of the document, we iterate the whole structure and extract the combination of $\langle subject, verb\rangle$ and $\langle verb, object\rangle$ for each sentence. Then if two combinations in the same sentence share a common verb, we merge them into a $\langle subject, verb, object\rangle$ triplet. Otherwise, a semantic-natural word is used to replace the blank.

For each triplet in the form of $\langle w_{subj}, w_{verb}, w_{obj}\rangle$, we pass it into a pre-trained Word2Vec [19] model which is based on Chinese Wikipedia and with 128 as its embedding dimension. In this way, we can obtain three 128-dimensional vectors correspond to subject verb and object, respectively. Then, as is shown in figure 2, we concatenate three corresponding vectors into one 384-dimensional vector as the representation of the event and the input of the attention-based BiGRU network.

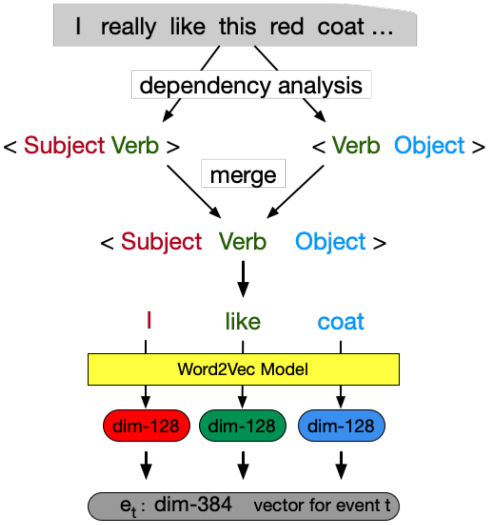

Figure 2: Event Extraction and Event Encoding Procedure

3.4 Event-level attention

Long Short-Term Memory (LSTM) units are firstly proposed by Hochreiter and Schmidhuber [20] to overcome the gradient vanishing problem. The main idea is to introduce an adaptive gating mechanism, which decides the degree to which LSTM units keep the previous state and memorize the extracted features of the current data input. A Gated Recurrent Unit (GRU) was first proposed by Cho et al. [21] Similarly to the LSTM unit, the GRU has gating units that modulate the flow of information inside the unit, however, without having separate memory cells. According to the experiment conducted by Chung et al. LSTM unit and GRU have similar performance, while GRU contains fewer parameters thus reducing the computational cost. Therefore, we choose GRU instead of LSTM as the basic module.

Bidirectional GRU network: Standard LSTM networks and GRU networks are widely used and have proved their efficiency for the sequential data. However, the text classification task is not a sequential problem. Instead, more emphasis should be put on the context and semantic information. Thus, we apply a Bidirectional GRU (BiGRU) network in our model so as to address this challenge. The BiGRU networks extend the unidirectional GRU networks by introducing an additional layer in the opposite direction. In this way, the GRU model can be used to exploit more context and semantic information. In this paper, our BiGRU model contains two sub-networks which represent forward and backward word-order respectively.

For event t in a certain input document, the GRU calculates the event state h_t by linearly interpolating the previous event state h_{t-1} and the candidate state $\tilde{h_t}$, as:

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h_t} \tag{1}$$

The candidate state $\tilde{h_t}$ is calculated by non-linearly combination of input event vector e_t and the previous state, as:

$$\tilde{h_t} = tanh(W_h e_t + r_t * U_h h_{t-1} + b_h) \tag{2}$$

The candidate state $\tilde{h_t}$ is calculated by non-linearly combination of input event vector e_t and the previous state, as: where r_t signifies the reset gate which aims to control how much the past state information should be used for updating the new state and z_t means the update gate, deciding how much past information should be kept and how much new information should be added. The two gates mentioned above are calculating by the following formulas respectively:

$$r_t = \sigma(W_r e_t + U_r h_{t-1} + b_r) \tag{3}$$

$$z_t = \sigma(W_z e_t + U_z h_{t-1} + b_z) \tag{4}$$

Consequently, we can obtain the latent vectors for each event through GRU. In order to get more context and semantic information, we concatenate the latent vectors from both directions to build a comprehensive vector h_i for each event vector e_i as: (Suppose that the input document contains T events)

$$\overrightarrow{h_i} = \overrightarrow{GRU}(e_i), i \in [1, T] \tag{5}$$

$$\overleftarrow{h_i} = \overleftarrow{GRU}(e_i), i \in [T, 1] \tag{6}$$

$$h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}] \tag{7}$$

Therefore, the comprehensive vector h_t contains both the information of its surrounding context and itself. Further, event vector h_t is served as the input for the attention layer.

Event-level Attention: Attention neural networks have recently achieved good performance in a wide range of tasks ranging from question answering, machine translations, speech recognition, to image captioning. Considering the fact that not all events contribute equally in terms of the document type prediction. We apply an attention layer to reevaluate the weight of each kind of event by assigning new attention value to each event.

$$u_i = sigmoid(W_h h_i + b_h) \tag{8}$$

$$\alpha_i = \frac{exp(u_i)}{\sum_j exp(u_j)} \tag{9}$$

$$V_E = \sum_i \alpha_i u_i \tag{10}$$

We first pass the event vector h_i through an one-layer network to get the event-level attention value u_i. Then we calculate a normalized attention weight α_i through a softmax function. Finally, we obtain the reorganized overall document vector V_E as a weighted sum of each event vector respectively. Thus, we obtain the event-level feature through an attention-based BiGRU network.

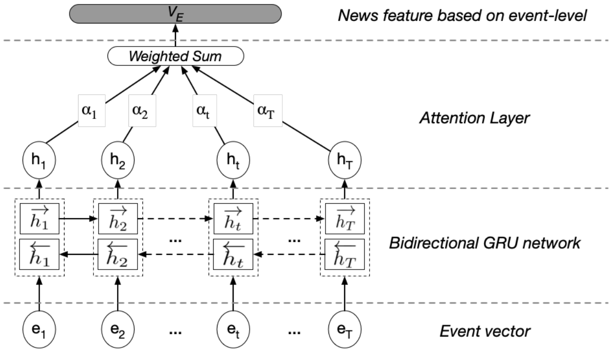
V_E
News feature based on event-level
Weighted Sum
α_1
α_2
α_t
α_T
Attention Layer
h_1
h_2
h_t
h_T
$\overrightarrow{h_1}$ $\overrightarrow{h_2}$ $\overrightarrow{h_t}$ $\overrightarrow{h_T}$
$\overleftarrow{h_1}$ $\overleftarrow{h_2}$ $\overleftarrow{h_t}$ $\overleftarrow{h_T}$
...
Bidirectional GRU network
e_1
e_2
e_t
e_T
Event vector

Figure 3: Structure of Attention-based BiGRU network for event-level information

3.5 Knowledge Retrieval

The goal of this module is to retrieve relevant knowledge from knowledge bases. There exist various type of relationship in knowledge bases, such as *has_instance*,*isA* and *is_related_to*. In this paper, we take *isA* relation as an example. Chen et al.[22] proposed a typical method for knowledge retrieval. Specifically, given a short text *s*, we hope to find a concept set *C* relevant to it. We achieve this goal by two major steps: entity linking and conceptualization.

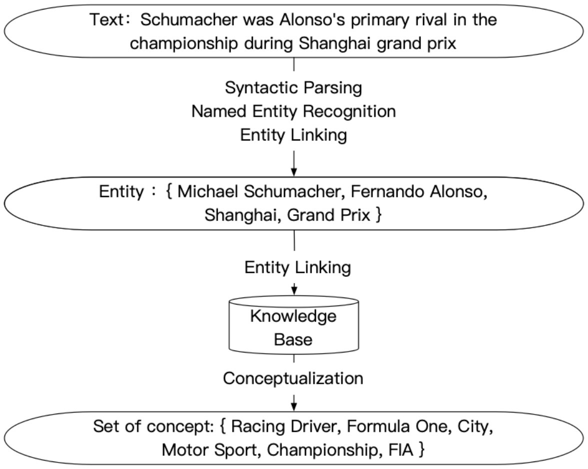
Text：Schumacher was Alonso's primary rival in the championship during Shanghai grand prix
Syntactic Parsing
Named Entity Recognition
Entity Linking
Entity：{ Michael Schumacher, Fernando Alonso, Shanghai, Grand Prix }
Entity Linking
Knowledge Base
Conceptualization
Set of concept: { Racing Driver, Formula One, City, Motor Sport, Championship, FIA }

Figure 4: Example of Conceptual Knowledge Retrieval

Entity linking is an important task in NLP and is used to identify the entities mentioned in the short text [23]. We acquire an entity set E of a short text by leveraging the existing entity linking solutions [24]. Then, for each entity $e \in E$, we acquire its conceptual information from an existing KB, such as YAGO [25], Probase [26] and CN-Probase [27] by conceptualization. For instance, given a short text "Kobe and Lebron are NBA players", we obtain the entity set $E = \{KobeBryant, LebronJames\}$by entity linking. Then, we conceptualize the entity Kobe Bryant and acquire its concept set C = {person, athlete, actor, basketball player} from a certain Chinese database. In this paper we choose CN-Probase as our knowledge base. After extracting relevant concept words from a sentence, we will assign weights to these concept words through the attention module.

3.6 TextCNN model for short text precessing

The goal of this module is to produce the short text representation q for a given short text of length n which is represented as the sequence of d-dimensional word vectors $(x_1, x_2, ..., x_n)$.

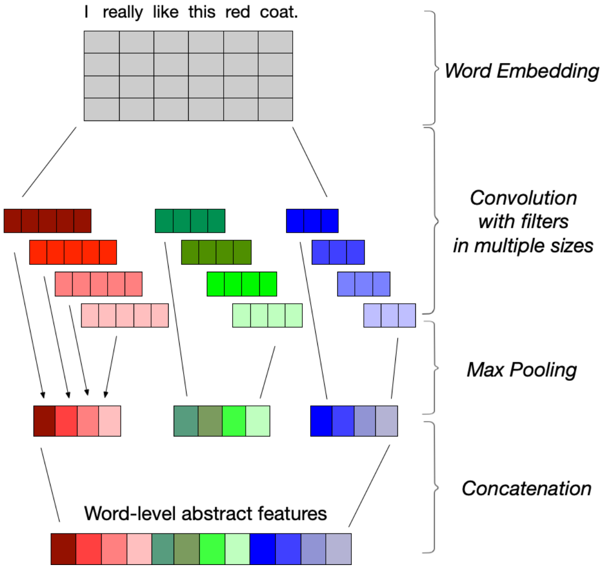
I really like this red coat.
Word Embedding
Convolution with filters in multiple sizes
Max Pooling
Concatenation
Word-level abstract features

Figure 5: Structure of TextCNN model

Recently, with the great success of deep and neural learning in computer vision [28] and speech processing tasks [29], deep learning has been applied to many tasks of NLP. We apply a CNN-based text classification approach inspired the model proposed by Kim [30].

Figure 5 illustrates the basic structure of CNN for document classification. After the basic data pre-processing, such as word segmentation and stop-word removal, we built a vocabulary dictionary and use the word embedding method to encode word to vector. In this way, these vectors can servers as the input of a neural network. Then, we design three kinds of convolutional kernels of different filter region sizes 3, 4 and 5 so as to obtain more features. 1-max pooling is performed over each feature map, i.e., the largest number from each feature map is recorded. Finally, we concatenate three vectors generated by three kinds of filters and we obtain a comprehensive vector which represents abstract feature from word-level.

3.7 Model Fusion

With the help of previous module, given a input short text, we obtain the original text and its related event information and concept set. Based on these three type of information, we develop three classifiers respectively. Further, inspired by the idea of ensemble learning, we apply wighted average method to learn the parameter and generate the final classification result.

4 EXPERIMENTS

4.1 Dataset and Experimental Setup

We evaluate our model on two datasets. As shown in Table 1, the first one is a Chinese Weibo emotion analysis dataset from NLPCC2013[31]. There are 7 kinds of emotions in these weibos, such as anger, disgust, fear and etc. The second one is product review dataset from NLPCC2014 [32]. The polarity of each review is binary, either positive or negative. The average word length of the above-mentioned datasets is over 12.

The experiment is based on the Keras deep-learning framework. The dependency analysis tool is Stanford's NLP Library[1]. The Chinese word segmentation tool is HANLP[2]. The pretrained Word2Vec model is based on Chinese Wikipedia.

4.2 Baseline Models

We compare our method with several baseline methods as follows:

CNN: This model utilizes the idea of CNN on the pre-trained word embedding.

RCNN[33]: This method uses a recurrent convolutional neural network for text classification. It applies RNN to capture contextual information and CNN to capture the key components in texts.

CharCNN [34]. This method uses CNN with only character level features as the input.

BiLSTM-MP [35]: This model is proposed for sequential short text classification. It uses a LSTM in each direction, and use max-pooling across all LSTM hidden states to get the sentence representation, then use a multi-layer perceptron to output the classification result.

BiLSTM-SA [36]: This method uses BiLSTM and source2token self-attention to encode a sentence into a fixed size representation which is used for classification.

BiGRU-MP We apply a bidirectional GRU network to replace the LSTM unit in BiLSTM-MP method to compare the performance of LSTM and GRU. The input of BiGRU-MP is the same as that in BiLSTM-MP.

BiGRU-SA We apply a bidirectional GRU network to replace the LSTM unit in BiLSTM-SA to evaluate the performance of LSTM and GRU. The input of BiGRU-SA is the same as that in BiLSTM-SA.

KPCNN [37]: It utilizes CNN to perform classification based on word and character level information of short text and concepts.

TextCNN We apply a TextCNN model based on method proposed by Kim et al.[38] to evaluate the performance of only using short text itself as the input. We use the word embedding method to transform a word into a 256-dim vector.

STCKA-word [22] STCKA stands for Short Text Classification with Knowledge Powered Attention. It utilizes knowledge information along with short text itself and applies attention mechanism to allocate weight parameter for each knowledge concept in a piece of text. Since this model utilizes word-level granularity in data pre-processing part, we mark it as STACK-word in the comparison table.

[1] https://nlp.stanford.edu/software
[2] https://github.com/hankcs/HanLP

Table 1: Details of the experimental datasets

Dataset	Number of class	Avg. Chars	Avg. Words	Training/Validation/Test set
Weibo Emotion	7	26.51	17.23	3766/660/500
Product Review	2	64.71	40.31	7698/1300/1000

STCKA-char This model is the state- of-the-art method for short text classification. In this model, we modify the text segmentation method and apply character granularity on the STCKA model.

4.3 Result and Analysis

F_{avg} is applied to evaluate the performance of models on Weibo and product review dataset, where F_{avg} is the average of the F_{score} for each category.

Table 2: Comparison of Experimental Results

Model	Weibo Emotion	Product Review
CNN	0.390	0.729
RCNN	0.404	0.728
CharCNN	0.410	0.701
BiLSTM-MP	0.416	0.729
BiLSTM-SA	0.412	0.731
BiGRU-MP	0.423	0.735
BiGRU-SA	0.427	0.732
KPCNN	0.424	0.734
TextCNN	0.476	0.741
STCKA-word	0.432	0.743
STCKA-char	0.512	0.736
Proposed	**0.552**	**0.775**

Table II shows the results of our experiments. We discuss several key points in the results in the following paragraphs.

Role of the attention modules: Here we take KPCNN and STCKA as a control group. Both of their ideas are to apply conceptual information and short text itself to the text classification process at the same time, but STACK adds an additional attention module to identify more important concepts. The results of the experiment confirmed the effectiveness of the attention module, which means an additional attention layer can enhance the classification accuracy.

GRU vs. LSTM: GRU is network achieves a similar performance with LSTM with fewer parameters. Thus, GRU shows high efficiency in practical text classification tasks.

Word-level granularity vs. Char-level granularity: In this paper, our discussion of granularity is in the context of deep learning and in the context of Chinese text. For the same version of input, we applied word-level granularity and char-level granularity in segmentation process. Results show that for deep learning-based Chinese Natural Language Processing task, word-level granularity has several disadvantages. Firstly, word-level granularity tends to have data sparsity problem, which inevitably leads to overfitting and the ubiquity of (out-of-vocabulary) OOV words limits the model's learning capacity. Meanwhile, by checking the word frequency dictionary, the distribution of Chinese words is more uneven than that of Latin words. As a result, it is less possible for the model to fully learn semantics information from the text. Secondly, due to the peculiarity of the Chinese language, the boundaries of words are not as clear as those of Latin language. Thus, the way of word segmentation is mainly based on experience and corpus training. As a result, the accuracy of word segmentation is greatly affected by the quality of words and word segmentation tools, which greatly increases the uncertainty. Thirdly, in short text classification task, we observe that char-based models consistently outperform word-based model and we also observe that char-based models perform better or at least as good as the hybrid model, indicating that char-based models already encode sufficient semantic information. As is shown in Table 3, we select three widely used deep learning models and conduct apples to apples experiments. The experiment result shows that char-based models perform better or at least as good as the word-based model.

Table 3: Results of models on short text classification task with different granularity on Weibo dataset

Input Form	Model	Granularity	
		char-level	**word-level**
Short Text	BiLSTM-MP	0.434	0.416
	BiGRU-MP	0.441	0.423
	TextCNN	0.492	0.478
Event	BiLSTM-MP	0.332	0.316
	BiGRU-MP	0.320	0.354
	TextCNN	0.356	0.322
Concept	BiLSTM-MP	0.233	0.225
	BiGRU-MP	0.231	0.265
	TextCNN	0.263	0.251

The role of supplementary information

According the experiment results in Table 3, for a specific model, different input will lead to different upper limit. This is also an important issue in the short text classification task. Compared with paragraphs or documents, short texts are more ambiguous since they have not enough contextual information. As a consequence, as is shown in Table 4 introducing more information helps increase the classification accuracy of short text.

Embedding Tunning: We totally use three embeddings in our model. Concept embeddings are randomly initialized and fine-tuned in the training stage. As for character and word embedding, we try three embedding tuning strategies:

- Rand: The embedding is randomly initialized and then modified in the training stage.

Table 4: Results of models with different type of input information on Weibo dataset

Input Source		Model	F_avg
Single Source	Short Text	TextCNN	0.476
Bi-Source	Short Text + Concept	STCKA	0.512
Tri-Source	Short Text + Event + Concept	Proposed	0.552

- Static: Using pre-trained embedding which is kept static in the training.
- Non-static: Using pre-trained embedding initially, and tuning it in the training stage.

5 CONCLUSION AND FUTURE WORK

In this paper, we study the possible optimization aspects of the text classification problem, including source text, segmentation granularity, embedding method, model and model fusion. We propose a novel short text classification method which combines event information and conceptual knowledge from external knowledge base together as supplementary information in order to obtain more contextual and semantic information. The application of TextCNN and attention mechanism makes full use of the advantages of feature extraction capability of deep learning. We also discuss the granularity selection problem for Chinese text under deep learning context and utilize char-based model in our proposed method.

Our future work will be: 1) Continue to follow the idea of introducing more information to aid classification, test new information for input 2) for Chinese word segmentation problem, test more granularity such as radicals 3) optimize the ensemble method for classifiers.

REFERENCES

[1] Pang B, Lee L, Vaithyanathan S. Thumbs up?: sentiment classification using machine learning techniques[C]//Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics, 2002: 79-86.

[2] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.

[3] Li C, Cheng Y, Wang H. A Novel Document Classification Algorithm Based on Statistical Features and Attention Mechanism[C]. 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, 2018: 1-6.

[4] Post M, Bergsma S. Explicit and implicit syntactic features for text classification[C].Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2013, 2: 866-872.

[5] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]. Advances in neural information processing systems. 2013: 3111-3119.

[6] Zhou P, Shi W, Tian J, et al. Attention-based bidirectional long short-term memory networks for relation classification[C]. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2016, 2: 207-212.

[7] Holmes D E, Jain L C. Innovations in machine learning[M]. Springer-Verlag Berlin Heidelberg, 2006.

[8] Wu H, Wang H. Pivot language approach for phrase-based statistical machine translation[J]. Machine Translation, 2007, 21(3): 165-181.

[9] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]//Advances in neural information processing systems. 2013: 3111-3119.

[10] Mikolov T, Yih W, Zweig G. Linguistic regularities in continuous space word representations[C]//Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies. 2013: 746-751.

[11] Chen X, Xu L, Liu Z, et al. Joint learning of character and word embeddings[C]//Twenty-Fourth International Joint Conference on Artificial Intelligence. 2015.

[12] Xu R, Chen T, Xia Y, et al. Word embedding composition for data imbalances in sentiment and emotion classification[J]. Cognitive Computation, 2015, 7(2): 226-240.

[13] Zhou G, He T, Zhao J, et al. Learning continuous word embedding with metadata for question retrieval in community question answering[C]//Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2015: 250-259.

[14] Sun Y, Lin L, Yang N, et al. Radical-enhanced chinese character embedding[C]//International Conference on Neural Information Processing. Springer, Cham, 2014: 279-286.

[15] Li Y, Li W, Sun F, et al. Component-enhanced chinese character embeddings[J]. arXiv preprint arXiv:1508.06669, 2015.

[16] Shi X, Zhai J, Yang X, et al. Radical embedding: Delving deeper to chinese radicals[C]//Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). 2015: 594-598.

[17] Li X, Meng Y, Sun X, et al. Is word segmentation necessary for deep learning of Chinese representations?[J]. arXiv preprint arXiv:1905.05526, 2019.

[18] Zeng D, Liu K, Lai S, et al. Relation classification via convolutional deep neural network[C]. Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. 2014: 2335-2344.

[19] Sun R, Guo S, Ji DH. Topic Representation Integrated with Event Knowledge[J]. Chinese Journal of Computer, 2017, 40(4):791-804.

[20] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[21] Cho K, Van Merrinboer B, Bahdanau D, et al. On the properties of neural machine translation: Encoder-decoder approaches[J]. arXiv preprint arXiv:1409.1259, 2014.

[22] Chen J, Hu Y, Liu J, et al. Deep short text classification with knowledge powered attention[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33: 6252-6259.

[23] Moro A, Raganato A, Navigli R. Entity linking meets word sense disambiguation: a unified approach[J]. Transactions of the Association for Computational Linguistics, 2014, 2: 231-244.

[24] Chen L, Liang J, Xie C, et al. Short text entity linking with fine-grained topics[C]//Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 2018: 457-466.

[25] Suchanek F M, Kasneci G, Weikum G. Yago: A large ontology from wikipedia and wordnet[J]. Journal of Web Semantics, 2008, 6(3): 203-217.

[26] Wu W, Li H, Wang H, et al. Probase: A probabilistic taxonomy for text understanding[C]//Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. 2012: 481-492.

[27] Shuyan, T. 2018. Cn-probase concept api. Accessed May 22, 2018. http://shuyantech.com/api/cnprobase/concept.

[28] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.

[29] Hinton G, Deng L, Yu D, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups[J]. IEEE Signal processing magazine, 2012, 29(6): 82-97.

[30] Kim Y. Convolutional neural networks for sentence classification[J]. arXiv preprint arXiv:1408.5882, 2014.

[31] Zhou H, Huang M, Zhang T, et al. Emotional chatting machine: Emotional conversation generation with internal and external memory[C]//Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

[32] Zhou Y, Xu R, Gui L. A sequence level latent topic modeling method for sentiment analysis via cnn based diversified restrict boltzmann machine[C]//2016 International Conference on Machine Learning and Cybernetics (ICMLC). IEEE, 2016, 1: 356-361.

[33] Lai S, Xu L, Liu K, et al. Recurrent convolutional neural networks for text classification[C]//Twenty-ninth AAAI conference on artificial intelligence. 2015.

[34] Zhang X, Zhao J, LeCun Y. Character-level convolutional networks for text classification[C]//Advances in neural information processing systems. 2015: 649-657.

[35] Lee J Y, Dernoncourt F. Sequential short-text classification with recurrent and convolutional neural networks[J]. arXiv preprint arXiv:1603.03827, 2016.

[36] Lin Z, Feng M, Santos C N, et al. A structured self-attentive sentence embedding[J]. arXiv preprint arXiv:1703.03130, 2017.

[37] Wang J, Wang Z, Zhang D, et al. Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification[C]//IJCAI. 2017: 2915-2921.

[38] Kim Y. Convolutional neural networks for sentence classification[J]. arXiv preprint arXiv:1408.5882, 2014.