Graph Attentive Leaping Connection Network for Chinese Short Text Semantic Classification

Jingdan Zhu(✉)

East China Normal University, Shanghai 200062, China

Abstract. Chinese short text semantic classification is a ubiquitous task that widely occurs in natural language processing. The existing methods are generally utilized in English, leading to multiple limitations: Unable to capture word-level abundant semantic when the input tokens are character sequence. It would be more vulnerable to data sparsity and the presence of out-of-vocabulary (abbr., OOV) words if utilizing word-based models, and thus more prone to overfitting. The very few approaches that consider both granularities are still limited. To tackle these problems, we propose a novel Graph Neural Network. Our model adopts the word lattice graph to keep multi-granularity information and utilizes the pre-trained model to obtain powerful semantics. Additionally, the attention mechanism and the layer leaping connection enable better structure-aware representation. Experimental results on three Chinese datasets demonstrate that our model achieves state-of-the-art performance in short text classification models.

Keywords: Semantic classification · Graph attention network · Leaping connection

1 Introduction

Text classification is pushed forward into many target applications [10], e.g., sentiment analysis, question answering, natural language inference, etc. It aims to process different kinds of texts and classify them into pre-defined labelled categories. Short text semantic classification serves the role analogous to sentence pair classification in Chinese context semantic environment. For sentence pair classification tasks, two text sequences will be considered the input for approaches and a label or a scalar value indicating their relation will be received. Numerous tasks, including natural paraphrase identification [14] and answer selection [17] can be seen as specific forms of text matching problems.

As a surge of interest and distinguished work [6,15] emerge in natural language processing (abbr., NLP) recently, choosing proper methods becomes more practical and challenging. Pre-trained learning models (e.g., BERT or GPT and

W. Zhang et al. (Eds.): WISE 2021, LNCS 13080, pp. 412–420, 2021.
https://doi.org/10.1007/978-3-030-90888-1_32

their variants) outperform more better than traditional machine learning methods in almost all scenarios.

Of note, deep graph neural networks are preferred utilized in text classification, efficiently capturing semantic connections between words, phrases or sentences, evolving into feasible representation methods. Nevertheless, most of the datasets used for text classification only provide English version. Not only Chinese datasets, but also how to migrate the methods for text classification needs to be measured. Early work utilizes Chinese characters as input to the model, or first segments each sentence into words and then takes these words as input tokens. Word-based models are more susceptible to sparse data and the presence of out-of-vocabulary words will also lead to performance degradation, and thus more prone to overfitting [7]. However, character-based models cannot fully utilize explicit word information, which is not negligible in Chinese semantic classification.

In this paper, we propose a Graph Attention Leaping Connection Network (abbr., GLCN) to consider both semantic information and multi-granularity information, achieving sufficient information aggregation while alleviating over smoothing. Our model needs to build a pair of word lattice graphs. In order to reduce noise and computation, only several segmentation paths are utilized to form the lattice graph during the construction process. Also, we get the initial word representation by aggregating features from the character-level interaction. For nodes updating, we use an attention mechanism to weigh "important" neighbors more. When getting the final representation of each node, we introduce a leaping connection policy for the first time, which considers information from all nodes in the graph and can be generalized to new graphs by Max-Pooling.

There are four main aspects of our contribution:

1) Our model makes full use of the multi-granularity information of characters and words.
2) Attention mechanism is introduced to better aggregate the information between words and characters.
3) Leaping connection constructed by adaptive Max-Pooling achieves node information aggregation without introducing additional learning parameters while avoiding over-smoothing.
4) Experiments on three datasets demonstrate that our model outperforms the state-of-the-art model.

2 Related Work

Deep Text Classification. Recently, pre-trained models (abbr., PTMs) like BERT [5] have shown their powerful ability in learning contextual word embeddings. For Chinese text classification, BERT takes a pair of short texts as input and each character is a separated input token. It has ignored word information. To tackle this problem, some Chinese variants of original BERT have been proposed, e.g. BERT-wwm [4], ERNIE [12] and its update ERNIE2.0 [11]. They take the word information into consideration based on the whole word masking mechanism during pre-training.

Graph Neural Networks. Graph neural networks derive from network embedding, which effectively maps nodes to low-dimensional representations and records the structure of the network. As a typical kind of non-Euclidean data, graph-structure data is playing a crucial role in the field of deep neural networks [16,18]. These deep neural network architectures are known as Graph Neural Networks (abbr., GNNs), which have been proposed to learn meaningful representations for graph-structure data.

3 Graph Attentive Leaping Connection Model

3.1 Problem Definition

For ease of presentation, we define the notations and key data structures used in this paper.

Definition 1 (Chinese Test Classification). Given two Chinese short text sequences $S^{^a} = \left\{s_1^a, s_2^a, \cdots, s_{T_a}^a\right\}$ and $S^{^b} = \left\{s_1^b, s_2^b, \cdots, s_{T_b}^b\right\}$, the goal of our text classification model $f\left(S^a, S^b\right)$ is to predict weather S^a and S^b have the same semantics. Where s_i^a and s_j^b represent the i-th and j-th Chinese character in two texts respectively, and T_a and T_b denote the number of characters.

Definition 2 (Chinese Lattice Graph). A Lattice Graph consists of the result of Chinese word segmentation and the original character sequence. Since keeping all possible segmentation paths will lead to excessive computation and noise, we stay several paths by random selection like Fig. 2 to form a word lattice graph $G = (\mathcal{V}, \mathcal{E})$. Each word and character represents a node. $\mathcal{V}$ is the set of nodes. $\mathcal{N}(v_i)$ denotes the set of all neighbor nodes of node v_i except itself.

3.2 Model Description

As shown in Fig. 1, our model consists of four components: a lattice embedding module, a neighborhood interaction-based attention module, a leaping connection module and a final semantic classifier.

Lattice Embedding Module. For each node v_i in graph lattice, the initial representation of word w_i is the aggregation of contextual character representations. We first recombine the two original character-level text sequences to a new one and then feed them to the BERT pre-train model to obtain the contextual representations for each character $C = \left\{c^{\text{CLS}}, c_1^a, \cdots, c_{T_a}^a, c^{\text{SEP}}, c_1^b, \cdots, c_{T_b}^b, c^{\text{SEP}}\right\}$.

Next, we define the characters contained in each word w_i in each graph as $\{s_i, s_{i+1}, \cdots, s_{i+n_i-1}\}$, which means the node v_i has n_i consecutive character tokens and s_i denotes the index of the first character of v_i in the text S^a and S^b. Then, we calculate a feature-wised score vector u_k, with a two layers feed forward network(abbr., FFN) for each character c_{i+k} $(0 \leq k \leq n_i)$ in w_i like [2] and then normalized with a feature-wised softmax as Fig. 2.

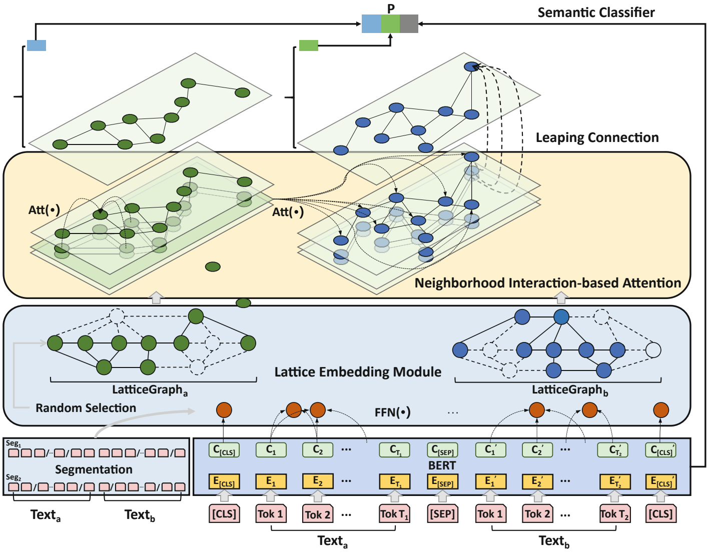

Fig. 1. The framework of GLCN-BERT model.

$$\mathrm{u}_{i+k} = \operatorname{softmax}\left(\operatorname{FFN}\left(c_{i+k}\right)\right) \tag{1}$$

The corresponding character embedding c_{i+k} is weighted with the normalised scores u_{i+k} to obtain the initial node embedding $\mathrm{v}_i = \sum_{k=0}^{n_i-1} \mathrm{u}_{i+k} \odot \mathrm{c}_{i+k}$. where $\odot$ represents element-wise product of two vectors.

At the end of this module, we get two lattice graph embedding sets G^a and G^b, which consist of both character-level and word-level representations.

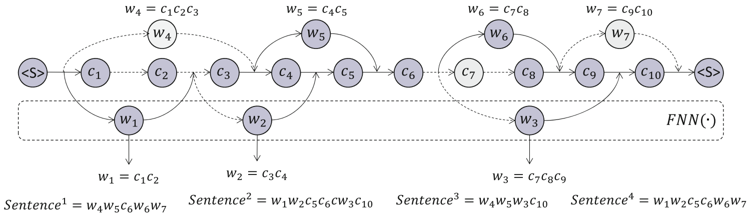

Fig. 2. Contextual word embedding.

Neighborhood Interaction-Based Attention Module. Since the utilization of the attention mechanisms allows the learning process to focus on parts of the graph that are more relevant to a specific task. As Fig. 2 shows, the graph attention classification module takes the contextual node embedding u_i as the initial representation h_i^0 for each node v_i, then updates its representation from one layer to the next. We simplify the update strategy into two steps:

(1)Message Propagation. At l-th step, each node v_i in G^a (the same with G^b) will first aggregates messages from its own neighbor nodes and then combine the result with the node representation from the last iteration,

$$\mathbf{h}_i^{self} = \mathrm{GRU}\left(\mathbf{h}_i^{l-1}, \sigma\left(\sum_{v_q \in \mathcal{N}(v_i)} \alpha_{ij}\left(\mathbf{W}^{self}\mathbf{h}_j^{l-1}\right)\right)\right) \tag{2}$$

In order to make full use of the information of G^b, we also aggregate messages from all nodes in graph G^b,

$$\mathbf{h}_i^b = \sigma\left(\sum_{v_q \in \mathcal{V}(v_b)} \alpha_{iq}\left(\mathbf{W}^b\mathbf{h}_q^{l-1}\right)\right) \tag{3}$$

Here , the σ is a non-linear activation function, e.g. a ReLU. And α_{ij} and α_{iq} are attention coefficients [13].

(2)Representation Updating. After message propagation, each node v_i will update its representation from $\mathbf{h}_i^b$ to $\mathbf{h}_i^l = \mathrm{GRU}\left(\mathbf{h}_i^{\mathrm{self}}, \mathbf{h}_i^{\mathrm{b}}\right)$ with a gate recurrent unit (abbr., GRU) [3].

After updating node feature L steps, we will obtain the graph-aware representation $\mathbf{h}_i^L$ for each node v_i.

Leaping Connection Module. Without introducing any additional parameters, we selectively adopt max-pooling as the core of the LC module like Fig. 3, which can balance the contradiction between training consumption and over-smoothing. We can get the final representation $\mathbf{h}_v^{final} = \mathrm{MaxPooling}\left(\mathbf{h}_v^1, \mathbf{h}_v^2, \cdots, \mathbf{h}_v^L\right)$ through this module. Where $\left\{\mathbf{h}_v^1, \mathbf{h}_v^2, \cdots, \mathbf{h}_v^L\right\}$ means the representation of each node at each layer.

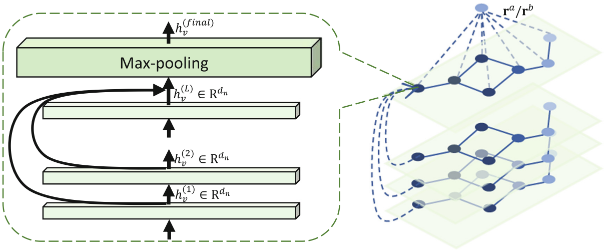

Fig. 3. Leaping connection module.

For each text S^a or S^b, the text representation vector $\mathbf{r}^a$ or $\mathbf{r}^b$ is obtained by attentive-pooling which can compute the representations of all nodes in each graph.

Semantic Classifier. With two text vectors $\mathbf{r}^a$, $\mathbf{r}^b$, and the vector $\mathbf{c}^{\mathrm{CLS}}$ obtained by BERT, our model will predict the similarity of two texts, the training object is to minimize the binary cross-entropy loss.

4 Experiment

4.1 Experimental Setup

Dataset. We conduct experiments on three Chinese datasets for the Chinese short text semantic classification task: LCQMC [8], BQ [1] and ATEC. ATEC is the semantic similarity learning contest data set provided by Ant Financial Services Group. The sample in all datasets contains a pair of texts and a binary label indicating whether the two texts have the same meaning or share the same intention. The statistics of the datasets is shown in Table 1.

Table 1. Features of three datasets

Dataset	Size	Pos:Neg	Domain
LCQMC	260068	1.3:1	Open-domain
BQ	120000	1:1	Bank
ATEC	100000	4:1	Finance

Hyper-parameters. The number of neighborhood interaction graph updating layers L is 3 on both datasets. The dimensions of both word representation and hidden size are 128. The model is trained by AdamW with an initial learning rate of 0.0002 and a warmup rate of 0.1. The learning rate of the BERT layer is multiplied by an additional factor of 0.1. As for batch size, we use 32 for all datasets. The dropout was applied after the word and character embedding layers with a keep rate of 0.3. It was also applied before the fully connected layers with a keep rate of 0.5. Moreover, the patience number is 4.

Environment Settings. Our model is constructed by python3.7, with the help of the PyTorch framework. All the following experiments are conducted on one CentOS server with two Intel Xeon 2.2 GHz CPUs, 128 G RAM, and one RTX 2080Ti GPU. The input word lattice graphs are produced by the combination of three segmentation tools: jieba[1] and HanNLP[2].

4.2 Evaluation Metrics and Baseline

Evaluation Metrics. For each dataset, the accuracy (abbr., ACC.) and F1 score are used as the evaluation metrics. ACC. is the percentage of correctly classified examples. F1 score of matching is the harmonic mean of the precision and recall.

Baseline. We compare our model with several BERT-based models pre-trained on large-scale corpora. **Bert-base** [5] is the official Chinese BERT model released

[1] https://pypi.org/project/jieba/.

[2] https://pypi.org/project/hanlp/.

by Google. It discards the traditional RNN and CNN, and converts the distance of two words at any position to 1 through the attention mechanism. **ERNIE** [12] is designed to learn language representation enhanced by knowledge masking strategies, which include entity-level masking and phrase-level masking. **BERT-wwm** [4] is a Chinese BERT, which was trained on the latest Chinese Wikipedia dump and adapt whole word masking in Chinese text. **BERT-wwm-ext** [4] is a variant of BERT-wwm with more training data and training steps. **ERNIE2.0** [11] is an upgraded version of ERNIE, proposing a mechanism for continual learning. **Roberta** [9] is an enhanced version of BERT that modifies key hyper-parameters, eliminates the pre-training target for the next sentence, and trains with larger mini-batches and learning rates.

4.3 Result and Analysis

From Table 2, we find that BERT variants all outperform the original one, which indicates that using word-level information in pre-training is crucial for Chinese text classification. Our model GLCN-BERT performs better than almost all these BERT-based models. It demonstrates that using word-level information and different fusion methods in the fine-tuning stage effectively boosts performance. It can even rival larger models with larger corpus and training time.

Table 2. Performance of various models on LCQMC, BQ and ATEC test datasets.

Models	LCQMC		BQ		ATEC	
	ACC.	F1	ACC.	F1	ACC.	F1
BERT [5]	85.7	86.8	84.5	84.0	88.1	88.7
BERT-wwm [4]	86.8	87.8	84.9	84.3	88.3	88.6
BERT-wwm-ext [4]	86.7	87.7	83.9	84.7	88.2	88.5
ERNIE [12]	87.0	87.9	84.7	84.2	88.5	88.9
ERNIE2.0 [11]	**87.9**	-	85.0	-	89.0	-
Roberta [9]	87.2	-	84.7	-	88.8	-
GLCN-BERT(Our)	**87.9**	**88.7**	**85.3**	**85.1**	**89.2**	**89.4**

In addition, as shown in Fig. 4, using the leaping connection method significantly improves the performance of the model for all three datasets. It indicates that our model can aggregate the information of the node itself and neighbor nodes well. This may be since the short text contains a short sequence of contexts. As the depth increases, the expansion of the node aggregation range leads to each node containing too much global information, which can easily lead to overfitting. Our model takes into account the problem and avoids it effectively.

Finally, Fig. 5 shows the results when we set the early stop value to 3 (training will stop when the best result is not exceeded three times in a row). Thus, we could know that for short sequences of text pairs, a small number of epochs already tend to achieve a good result.

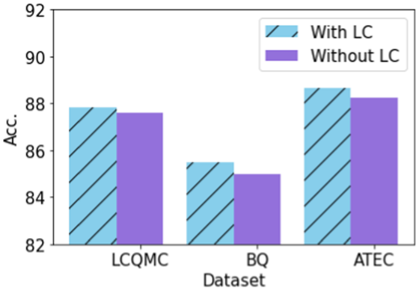

Fig. 4. Test accuracy.

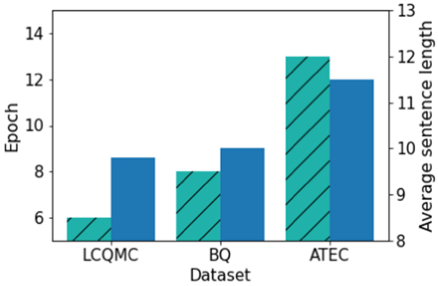

Fig. 5. Early stopping epochs and average text length.

5 Conclusion and Future Work

In this work, we propose a Graph Attentive Leaping Connection Network(GLCN-BERT) for Chinese short text classification. Our model takes two word lattice graphs and utilizes a graph attention network structure to obtain information from each layer. Then the leaping connection method is used to aggregate the information flexibly while avoiding overfitting. The proposed approach is evaluated on three Chinese benchmark datasets and achieves the best performance. Extensive experiments also demonstrate that both semantic information and multi-granularity information are essential for text classification modeling.

In the future, we will further investigate the effect of the network depth on text classification and introduce external knowledge, such as paraphrase database, to help learn more accurate and robust text representation.

References

1. Chen, J.J., et al.: The BQ corpus: a large-scale domain-specific chinese corpus for sentence semantic equivalence identification. In: EMNLP (2018)
2. Chen, L., et al.: Neural graph matching networks for chinese short text matching. In: ACL (2020)
3. Chung, J., et al.: Empirical evaluation of gated recurrent neural networks on sequence modeling. In: ArXiv abs/1412.3555 (2014)
4. Cui, Y., et al.: Pre-training with whole word masking for chinese BERT. In: ArXiv abs/1906.08101 (2019)
5. Devlin, J., et al.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2019)
6. Jiang, J.-Y., et al.: semantic text matching for long-form documents. In: The World Wide Web Conference (2019)
7. Li, Y., et al.: Enhancing pre-trained chinese character representation with word-aligned attention. In: ArXiv abs/1911.02821 (2020)
8. Liu, X., et al.: LCQMC: a large-scale chinese question matching corpus. In: COLING (2018)

9. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. In: ArXiv abs/1907.11692 (2019)
10. Minaee, S., et al.: Deep learning-based text classification: a comprehensive review. ACM Comput. Surv. **54**(3), 62:1-62:40 (2021). https://doi.org/10.1145/3439726
11. Sun, Y., et al.: ERNIE 2.0: a continual pre-training framework for language understanding. In: ArXiv abs/1907.12412 (2020)
12. Sun, Y., et al.: ERNIE: enhanced representation through knowledge integration. In: ArXiv abs/1904.09223 (2019)
13. Velickovic, P., et al.: Graph attention networks. In: ArXiv abs/1710.10903 (2018)
14. Wang, Z., Hamza, W., Florian, R.: Bilateral multi-perspective matching for natural language sentences. In: ArXiv abs/1702.03814 (2017)
15. Wang, Z., et al.: Match$^{\mathbf{2}}$: a matching over matching model for similar question identification. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (2020)
16. Xu, K., et al.: How powerful are graph neural networks?. In: ArXivabs/1810.00826 (2019)
17. Yang, Y., Yih, W.-t., Meek, C.: WikiQA: a challenge dataset for open-domain question answering. In: EMNLP (2015)
18. Ying, R., et al.: Hierarchical graph representation learning with differentiable pooling. In: NeurIPS (2018)