Classification of Chinese Hotel Reviews Based on GCN

Zhiyu Lin\textsuperscript{1,*}, Jingxuan Zhang\textsuperscript{1}, Haoxi Zhang\textsuperscript{2}

\textsuperscript{1} School of Computer Science, Chengdu University of Information Technology, Chengdu 610225, China
\textsuperscript{2} School of Cyberspace Security, Chengdu University of Information Technology, Chengdu 610225, China

*linzhiyu08@163.com

Abstract. Text classification is a fundamental part of natural language processing and can help with many downstream tasks, such as emotion analysis, question and answer systems, and recommendation systems. The graph convolution neural network has the natural superiority in the non-Euclidean space data. For Chinese text data, there is a lot of correlation between the data, using the graph convolutional neural network for text classification can achieve good results. In our experiment, we use a simple one-hot encoding of the word vector to process our words and use of the graph convolutional neural network can achieve 94.24% accuracy in our data set.

Keywords: Text Classification; Non-Euclidean Space Data; Graph Convolution Neural Network; Word Vector

1. Introduction
In recent years, with the rapid development of information technology, a large amount of data, such as text, images and so on, has been generated on the Internet. Micro-blogs, T-Witters, commodity reviews, headlines, etc. generate a large number of short text data, filling the entire Internet. There is too much information available in these short texts, including potential user needs, specific user points of interest, user intent needs, and so on. How to extract the information we need from short texts has become an urgent problem.

How to obtain the data we need from these short texts is more challenging than normal text data. We can use one-hot encoding and word2vector to process text data. However, when using this fixed vector space model alone to vectorize the text, the text data will become sparser. Extracting relevant features during network training will become more complicated. Therefore, using traditional text classification methods to process short text classification cannot achieve the desired effect.

The emergence of graph neural network \cite{1-2} solves the problem of short text classification difficulty from another aspect. No longer focus on the distribution of words in the vector space, but explore the existing connections between words. We know that the data that exists on the Internet does not exist alone. And other related data are closely related. Using graph convolutional networks to extract relevant connections between data can better help us in text classification.

2. Related work
In recent years, there are two research directions in text classification. The first is to develop an efficient pre-training model that uses a better word vector to represent a word, and then fine-tune us according to specific tasks in downstream tasks. Model. The second is to develop an efficient classification model. In the efficient classification model, the representation of words is not very important. The classification effect of the entire experiment is more reflected in the construction of the classification model.

2.1. Pretraining model
In the early days we used One-hot encoding. In One-hot encoding, only one position is 1, and the other positions are all 0. For example, the word "apple" ranks 250th in our corpus (the size of the corpus is 10,000). If the corpus is particularly large, using one-hot will cause the problem of dimensional explosion and the relationship between words will not be reflected.

In Mikolov's paper [3], Word2vec was proposed, a vector that uses a fixed length to represent different words. Word2vec will not have the same latitude explosion problem as one-hot. But Word2vec also has a considerable disadvantage, that is, it cannot solve the problem of polysemous words.

The next Embedding from Language Models (ELMO) [4] has alleviated the problems caused by polysemous words to a certain extent. ELMO uses a two-way Long short-term memory (LSTM) model for pre-training tasks, and then in downstream tasks, the word vectors of each layer of the network corresponding to the words are extracted from the pre-training network as new features to supplement the downstream tasks.

In Devlin J's paper [5], a new language representation model, BERT, is introduced. The BERT pre-training model uses a bidirectional Transformer, which has better physical sign extraction capabilities than LSTM. BERT refreshed the current best performance record of 11 natural language processing tasks at the beginning of its birth.

2.2. Deep learning model
In the last century, Yann Lecun of New York University proposed a convolutional neural network. Convolutional neural networks (CNN) include one-dimensional convolutional neural networks, two-dimensional neural networks and three-dimensional neural networks. Convolutional neural networks are mainly used to identify displacement and scaling invariant data, and can fully extract features between texts when processing text data.

Recurrent neural network is a kind of neural network specially used to process sequence data. Compared with convolutional neural network, it can process data with sequence changes. For example, the meaning of a word will have different meanings because of the different content mentioned above, and the cyclic neural network can solve this kind of problem well.

In recent years, the Attention [6] mechanism has been widely used in various fields of deep learning. The attention mechanism can be divided into spatial attention and temporal attention. The attention mechanism is often combined with CNN and LSTM, and there will be unexpected results in processing text data.

3. Proposed scheme
In this part, I mainly introduce the reference model graph convolutional neural network for text classification in our experiment.

3.1. Graph Convolutional Neural Network (GCN)
In recent years, deep learning has achieved great success in image processing and natural language processing, such as machine translation, image processing and speech recognition. CNN can extract image features well in two-dimensional space, and LSTM can perform well on sequence tasks. But both CNN and LSTM can only process data in Euclidean space. They are powerless for non-Euclidean space data. Data in non-European spaces is very common in our lives, such as public transportation
networks and social networks. This type of data is not translation-invariant like Euclidean space data. Therefore, we cannot use commonly used deep learning networks for processing.

For a graph convolutional neural network [6-7], a graph \( G = (V,E) \) is first established, where \( V \) represents the set of all nodes in the graph. That is, all the words that appear in the text, each word representing a node. We can be sure to consider only the first 50,000 words, which can reduce the number of invalid nodes in the graph and make the graph look not too big. \( E \) is the set of all the edges in the graph. So let's say that each node has an edge relation to itself, which is \((v,v) \in E\). X \( \in \mathbb{R}^{n \times m} \) Represents all vertices, \( n \) represents the number of vertices; \( m \) is the characteristic dimension of each node. A is used to represent the adjacency matrix of graph \( G \), and D is the degree matrix, where \( D_{ii} = \sum_j A_{ij} \), is a diagonal matrix.

In the graph convolutional neural network, only one layer of convolution is needed to get the information of adjacent nodes. When we use multiple convolutional layers, we can integrate a lot of relevant information. For single-layer GCN, calculate the new k-dimensional node feature matrix

\[
L^{(1)} = \rho(\tilde{A}XW_{0})
\]

\( \tilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \) is a normalized symmetric matrix, \( W_{0} \in \mathbb{R}^{m \times k} \) is a weight matrix, \( \rho \) is an activation function and in our experiment we used Relu activation function. When multiple volume bases are used, the feature extraction function is:

\[
L^{j+1} = \rho(\tilde{A}L^{j}W_{j})
\]

Where \( j \) represents the number of layers, \( L^{0} = X \).

3.2. Network Architectures

In our experiment, we use a 2-layer graph convolutional neural network. Each node in the graph represents all the words that appear in the text, and each word is represented by one-hot. The activation function we use is the Relu activation function.

Fig. 1 shows the network structure of the modified GCN. In our experiment, we first encode the data one-hot, and then each word after the encoding is used to construct our experiment model.

![Fig. 1 The network structure of GCN](image)

4. Experiments

4.1. Dataset

The data in my experiment are real data obtained from Ctrip.com, which is the comment data of hotels in popular scenic spots. These include Dali, Dujiangyan, Leshan, Lijiang and Xiqiao Mountain. Each
data set is a two-category data set, and the classification granularity in our data set is good and bad reviews.

In our experiment, we first carried out data cleaning. Since we are the original data set obtained from the Internet, there is data information in the text that is not related to the experiment, so we must clean the data to filter out what we don’t need. information. Then we split the data. In our experiment, we used 70% of the data as the training set, 15% of the data as the validation set, and 15% of the data as the test set.

| Data Set     | DaLi  | DuJiangYan | LiJiang | LeShan | XiQiaoShan |
|--------------|-------|------------|---------|--------|------------|
| Total Set    | 33767 | 34390      | 31115   | 29723  | 8000       |
| Training Set | 23636 | 33903      | 21780   | 20806  | 5600       |
| Test Set     | 5065  | 5158       | 4667    | 4458   | 1200       |
| Validation Set| 5065 | 5158       | 4667    | 4458   | 1200       |

**Tab. 1** The Data set processing

4.2. **Baseline Method and Evaluation Metrics**

In our experiment, the following benchmark models were used as comparative experiments. In the word embedding part, one-hot encoding was selected. The comparison model used is as follows.

FastTex is a fast text classification algorithm, a word vector calculation and text classification tool open sourced by Facebook in 2016. FastText is a shallow network, but it can achieve accuracy comparable to common deep networks.

CNN has a very wide range of applications in the field of deep learning, so there are quite a few variants. Since Kim used CNN for text classification tasks for the first time in 2014, many people have used its excellent feature extraction capabilities to achieve good results in text classification.

LSTM is a relatively classic network model in the field of natural language processing, which can take into account the interdependence of context words, and has natural advantages for sequence data. Bi-LSTM is a two-way LSTM. Compared with a one-way LSTM, Bi-LSTM can better capture the context information in a sentence.

The attention mechanism has attracted a lot of attention when it was proposed. Just as we humans value some important information more, Attention can assign weights to information, and finally perform weighted summation. Therefore, the Attention method has strong interpretability and better results.

4.3. **Experimental Results**

In our experiment, we used the average accuracy of the model test as our judgment identification. As can be seen from Table 2, when processing text data, LSTM and BI-LSTM have better effects than CNN and FastTex. The results of Attention are similar to those of LSTM. Our experimental model GCN is significantly better than the current common deep learning model, and the average result is 3%-4% higher than that of the current deep learning model. It indicates that GCN can well extract the features between texts and obtain the relevant information between texts.

| model      | DaLi  | DuJiangYan | LiJiang | LeShan | XiQiaoShan |
|------------|-------|------------|---------|--------|------------|
| CNN        | 89.04 | 89.76      | 90.21   | 89.82  | 88.77      |
| LSTM       | 91.34 | 91.44      | 90.89   | 90.14  | 89.54      |
| Bi-LSTM    | 92.03 | 92         | 91.12   | 90.89  | 90.77      |
| FastTex    | 89.89 | 89.4       | 89.45   | 89.21  | 88.46      |
| Attention  | 90.24 | 89.77      | 90.11   | 90.32  | 89.30      |
| GCN        | 94.98 | 94.35      | 94.31   | 94.09  | 93.45      |

**Tab. 2** Accuracy of experimental results
| GCN  | layer number |
|------|--------------|
|      | 1 | 2 | 3 | 4 |
| Accuracy | 89.45 | 94.24 | 94.11 | 93.94 |

**Tab. 3** Number of GCN layers and accuracy

As can be seen from Table 3, in the experiment, the more layers of GCN is not the better. When the number of layers of GCN is 1, the degree of fitting of the model is not enough; when the number of layers is 2, the model has the best effect.

5. **Conclusion**

In our experiment, we demonstrate the effectiveness of using the graph convolutional neural network for text classification. Most of the data in our experiment are short texts, which makes the effective edges in the graph become less when the graph neural network constructs the network model, which may have a great impact on the classification effect. In the following work, we can find ways to expand the edges in the graph to improve the classification effect.

**References**

[1] Wu Z, Pan S and Chen F 2020 A comprehensive survey on graph neural networks *J. IEEE. T. Neur., Net., Lear*

[2] Zhang Z, Cui P and Zhu W 2020 Deep learning on graphs: A survey *J. IEEE. T. Knowl., Data. En*

[3] Mikolov T, Chen K and Corrado G 2013 et al. *J. Efficient estimation of word representations in vector space*

[4] Liu L, Ren X and Shang J 2018 *J. Efficient contextualized representation: Language model pruning for sequence labeling*

[5] Devlin J, Chang M W and Lee K Bert: Pre-training of deep bidirectional transformers for language

[6] Yao L, Mao C and Luo Y 2019 Graph convolutional networks for text classification *AAAI., Conf., on, Art., Int*

[7] Huang L, Ma D and Li S 2019 *J. Text level graph neural network for text classification*