# TGCN: A Novel Deep Learning Model for Text Classification – Plan

Yuan Li
New York University
Brooklyn, NY
yl6606@nyu.edu

Dongzi Qu
New York University
Brooklyn, NY
dq394@nyu.edu

## 1. Problem Statement

Text classification is an essential and classical problem in natural language processing. There are various applications related to text classification, such as: news filtering, spam detection and document integration/organization. In order to solve this task, most traditional methods take use of features engineering by constructing the embeddings (representations) for documents. Recently, deep models are widely appied to learn the representations for documents. Under the help of neural networks, the semantic and syntactic information inside the document can be captured. However, both traditional methods and deep models may ignore the word co-occurrence in some long-term or non-consecutive sequence analysis. In this work, we propose to use graph convolutional networks (GCN) for text classification. The implementation of GCN will be covered in the fourth part.

## 2. Related Work

### 2.1. Traditional Deep Learning

Traditional deep learning methods for text classification can be divided into two ways. One way of studies give more attentions on training data. These studies proved that reasonable use of word embeddings can achieve great success in deep learning methods for text classification. Wang[5] further introduced an attention framework that combined text and text label embeddings.

Besides training data, other studies focused on architecture improvement. Two common used deep learning models are CNN and RNN. In order to give more flexibility of representing sentence in these deep learning models, attention mechanism[4] was used as an internal sub-module[6].

However, these methods mainly focus on sentence in short distance, but ignore the global word relationships in a whole document.

### 2.2. Graph Neural Network

Graph Neural Network (GNN) is a novel model that has received a lot of attention in the field of deep learning recently. Some studies present new approaches of GNN model like CNN to work on fixed structured graphs, which makes it possible for us to use GNN to solve text classification. Kipf and Welling[2] provided the graph convolutional network (GCN), which got state-of-art results on a number of NLP related graph datasets. Based on GCN, Yao[7] then presented the Text Graph Convolutional Network (Text GCN) for a whole document classification. It can jointly generate the embeddings of text and text label together. Meanwhile, text GCN can catch information in the whole document, and request less data than traditional deep learning model for text classification. Our model is mainly constructed based on text GCN and we will modify it to support Chinese dataset.

## 3. Dataset

According to our plans, we plan to run our experiment on two different widely used benchmark datasets with English documents, including 20-Newsgroups (20NG), Ohsumed and R52 and R8 of Reuters 21578. Meanwhile we are interested in how this model will perform on the Chinese documents. So we choose two Chinese documents dataset, which are THUCTC and fastTextData, to measure our model's robustness.

- The 20NG dataset[1] (bydate version) contains 18846 documents evenly categorized into 20 different categories.
- The Ohsumed corpus[2] is from the MEDLINE database, which is a bibliographic database of important medical literature maintained by the National Library of Medicine
- R52 and R8[3](all-terms version) are two subsets of the Reuters 21578 dataset.

[1] http://qwone.com/ jason/20Newsgroups/

[2] https://www.mat.unical.it/OlexSuite/Datasets/SampleDataSets-about.htm

[3] https://www.cs.umb.edu/˜smimarog/textmining/datasets/

- The THUCTC[4] (THU Chinese Text Classification) contains more than 740000 news from 2005 to 2011 collected by Sina News. All these news documents are evenly distributed into 14 categories, including financial, education, real estate, sports and so on.
- The fastTextData[5] is relatively small only containing around 24000 news data within 6 categories compared with THUCTC. While, documents in this dataset are tokenized word by word manually. Consequently, it's much easier to implement the experiment on this dataset.

## 4. Method

### 4.1. Graph Convolutional Network

A GCN has the similar structure as the normal convolutional networks. It contains multiple layers, where each layer is constructed in terms of the graph data structure. Formally, assume of graph $G = (V, E)$, where $V$ $(|V| = n)$ and $E$ denote the vertices set and edges set. Also, we introduce an adjacency matrix $A \in \mathbb{R}^{n \times n}$ and a diagonal matrix $D \in \mathbb{R}^{n \times n}$, where $D_{ii} = \sum_j A_{ij}$. Moreover, let $X \in \mathbb{R}^{n \times k}$ be a feature matrix (map) for all the vertices in $V$ of one layer. To obtain the new feature matrix using this single layer, the new feature matrix $(L^{(i+1)})$ with the different dimension can be computed as

$$L^{(i+1)} = \rho(\tilde{A} L^{(i)} W_i) \tag{1}$$

where $\tilde{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$, which is same among all layers and $W_i$ is a weight matrix and its shape depends on the second entry of $L^{(i)}$ and the first entry of $L^{(i+1)}$. $\rho$ is a activation function, which we can use ReLu or tanh. Also, $L^{(0)} = X$, which is the initialized feature matrix.

### 4.2. Text GCN

Given the structure of the basic GCN, we build a large and heterogeneous text graph which contains word nodes and document nodes. In this graph, edges exist between word to word or word to document. The edge between node and document can be calculated by Tf-idf of the word in the document. As for the word to word edge, the weight can be computed using point-wise mutual information (PMI), a popular measure for word associations. Also, for each node itself, we need to add a self-loop on it. After building the text graph, assume we have $j$ layers, then the output $Z \in \mathbb{R}^{n \times F}$ of $j^{th}$ layer can be computed as

$$Z = softmax(\tilde{A} L^{(j-1)} W_{j-1}) \tag{2}$$

where $F$ is the number of categories and we use softmax function to make predictions.

## 5. Anticipated Outcomes

We plan to reimplement the experiment from Yao[7], modifiing the base text GCN model and adding two Chinese documents datasets for evaluation. Four baselines we consider to use are:

- TF-IDF + LR: TF-IDF vectors with logistic regression model as classifier.
- fastText: an efficient text classification model provided by Joulin[1].
- SWEM: a simple word-embedding-based models for text classification baseline[3].
- LEAM: jointly label-embedding attentive models[5].

Our model's test accuracy supposed to be higher than other baselines. However, the normal Chinese documents are constructed sentence by sentence using the single Chinese characters without any space but some punctuation. This may cause some uncertainty of the final accuracy results for two Chinese datasets we want to use. Some modifications may also need to be made on the Chinese datasets. For instance, if we take use of the Chinese documents without any space, we must use a different tokenization method instead of the same one for English documents.

## References

[1] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[2] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[3] D. Shen, G. Wang, W. Wang, M. R. Min, Q. Su, Y. Zhang, C. Li, R. Henao, and L. Carin. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. *arXiv preprint arXiv:1805.09843*, 2018.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. 2017.

[5] G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, R. Henao, and L. Carin. Joint embedding of words and labels for text classification. *arXiv preprint arXiv:1805.04174*, 2018.

[6] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.

[7] L. Yao, C. Mao, and Y. Luo. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377, 2019.

[4] http://thuctc.thunlp.org/

[5] https://github.com/CementMaker/cnn_lstm_for_text_classify/tree/master/data/fastTextData