DEIM Forum 2019 P1-145

Exploring Multi-label Classification Using Text Graph Convolutional Networks on the NTCIR-13 MedWeb Dataset

Sijie TAO† and Tetsuya SAKAI†

† Department of Computer Science and Engineering, Waseda University
3–4–1 Okubo, Shinjuku-ku, Tokyo 169–8555, Japan
E-mail: †tsjmailbox@ruri.waseda.jp, †tetsuyasakai@acm.org

Abstract The NTCIR-13 Medical Natural Language Processing for Web Document (MedWeb) task requires participant systems to perform multi-label text classification, where labels representing eight different diseases or symptoms are assigned to each pseudo-tweet. While a recent study showed that a method based on Text Graph Convolutional Networks outperforms previous approaches in text classification, its performance on multi-label text classification remains unclear. Hence, in this study, we construct a model based on Text Graph Convolutional Networks (Text GCN) and evaluate its performance on multi-label classification. Our experimental results show that Text GCN could not outperform the baseline system on the NTCIR-13 MedWeb task.

Key words multi-label classification, graph convolutional network, medical natural language processing

1 Introduction

As a fundamental problem in natural language processing (NLP), text classification has been applied to a number of tasks such as document organization, news filtering and spam detection [1]. On the other hand, there are other research applying text classification to detect disease epidemics such as influenza, via analyzing text data on socical network services [2].

The NTCIR-13 Medical Natural Language Processing for Web Document (MedWeb) task is another similar example. The task is designed to require participant systems to perform multi-label classification, where eight labels representing different diseases or symptoms are assigned to each tweet-like message [10]. The task also covers multiple languages: Chinese, English, and Japanese. In order to build parallel corpora, the original Japanese pseudo-tweets are translated to Chinese and English.

In previous work [4] [11], deep learning models like Character-level Convolutional Networks (CharCNN) [14], and Hierarchical Attention Network (HAN) [12] and Long Short-Term Memory (LSTM) [3] have been used to challenge the NTCIR-13 MedWeb task. Recently, Yao et al. proposed a new graph neural network based method named Text Graph Convolutional Networks, which shows outstanding performance in text classification and outperforms previous approaches on several datasets [13]. However, as Yao et al. focused on single-label text classification, the performance of Text Graph Convolutional Networks on multi-label text classification remains unclear.

In this paper, we tackle the NTCIR-13 MedWeb task with a model based on Text Graph Convolutional Networks to test its performance on multi-label classification. Although Text GCN shows its potential in multi-label classification, our experimental results show that Text GCN could not outperform the baseline system due to limitations of its own graph structrue.

The remainder of this paper is organized as follows. In Section 2, related works are introduced. Section 3 provides the detail of the proposed model. Section 4 discusses the method of experiments. Section 5 shows and analyzes the experiment results. Finally, Section 6 gives the conclusion of this work.

2 Related Work

2.1 The NTCIR-13 MedWeb task participant systems

Among the participant systems at NTCIR-13, Iso et al. and Wang et al. challenge the MedWeb task with deep learning based methods [10].

In the work of Iso et al. [4], two neural network models based on both Hierarchical Attention Network (HAN) [12] and Character-level Convolutional Networks (CharCNN) [14] are utilized to establish a system to tackle the task. Their modeling strategy involves creating bootstrap dataset samples to feed to the models, which makes their classifier more robust. Moreover, with respect to two neural networks and three loss functions, six methods are generated for each boot-

strap sample. Finally, the result is obtained via taking the ensembles of the model combinations.

As the corpora of different languages are built parallel, which means that every original Japanese pseudo-tweet shares the same label set with its Chinese and English translation, it is possible to perform multi-language learning by feeding the model with a concatenation of the vectors of the pseudo-tweets in all three languages. The combination of the three languages enables a multi-language input which can carry richer information.

The results in the work of Iso et al. show that a system taking an ensemble of two neural network models (HAN and CharCNN) and two loss functions (negative likelihood and hinge) with a multi-language input achieves the best performance among all the participant systems at NTCIR-13 [10].

On the other hand, in the work of Wang et al. [11], a model based on Long Short-Term Memory (LSTM) is built to tackle the task. Rather than performing multi-language learning, Wang et al. challenge the English subtask only. Furthermore, instead of classifying all the eight labels at once, Wang et al. build one LSTM model for each label. Therefore, their system consists of eight LSTM classifiers.

2.2 Text Graph Convolutional Networks

Recently, research about graph neural networks has drawn wide attention. Kipf et al. presented a graph neural network model called Graph Convolutional Network (GCN) [5], which achieved stated-of-the-art performance on a number of graph datasets. Applying GCN to NLP, Yao et al. proposed a novel graph neural network-based method for text classification named Text Graph Convolutional Networks (Text GCN) [13]. In Text GCN, a singe large heterogeneous graph is built from an entire corpus. Then the graph is fed into GCN and it learns word and document embeddings jointly. In their experiments, Text GCN clearly outperforms several previous approaches on single-label text classification. On the other hand, to the best of our knowledge, the performance of Text GCN on multi-label text classification remains unclear yet.

3 Proposed Method

In this section, we describe the detail of the proposed Text Graph Convolutional Networks based model for multi-label text classification.

At the first step of constructing a Text Graph Convolutional Network, we build a large and heterogeneous text graph to model global word co-occurence in the whole corpus and word occurence in documents [13]. The graph contains word nodes and document nodes. The number of the nodes in the graph is the sum of the vocabulary size (the number of unique words) and the corpus size (the number of documents). Edges are built among word nodes and document nodes. The weights of the edges are defined to represent word co-occurence and word occurence in documents. Following the strategy of Yao et al., we compute the value of point-wise mutual information (PMI) for every word pair, and when the value is positive, we set it as the weight of the edge between the two word nodes, to represent word assciations. Moreover, in order to model word occurence in documents, the weight of the edge between a word node and a document node is defined as the term frequency-inverse document frequency (TF-IDF).

Second, we feed the text graph into a GCN. As previous work [5] [7] [13] report that GCN performs well when the number of layers is set up to two, we also construct a two-layer GCN to implement the graph convolution.

Finally, with the purpose of performing multi-label classification, we feed the second layer embeddings into a sigmoid classifier and the loss function is defined as the binary cross entropy. The formula of the loss function for each label is given as:

$$loss = -\sum_{n=1}^{N}[y_n \log(p_n) + (1 - y_n) \log(1 - p_n)] \qquad (1)$$

where N stands for the size of training set, p_n is the predicted probability and y_n is the true label.

4 Experiments

This section describes the experiments conducted in our study. The environment of experiments is shown in the table below.

Table 1 Environment of experiments

CPU	Intel Core i7-6700K
Memory size	32GB
GPU	Nvidia GeForce RTX 2080Ti
GPU memory size	11GB

4.1 Baseline

As the method of Iso et al. shows the best performance among all the participant systems at NTCIR-13 [10], we select the experiment results shown in the work of Iso et al. [4] as a baseline to compare with the performance of our model. The detail of the baseline method is shown in Section 2.1.

4.2 Dataset

The performance of multi-label text classification of the proposed model is tested on the NTCIR-13 MedWeb task dataset. As there are three subtasks for different languages (Chinese, English and Japanese), the dataset contains three parallel corpora. The Japanese corpus is the original, the Chinese and English corpora are constructed by translating the Japanese pseudo-tweets. For each corpus, the training

set contains 1,920 pseudo-tweets (75% of the corpus) and the test set has 640 (25%).

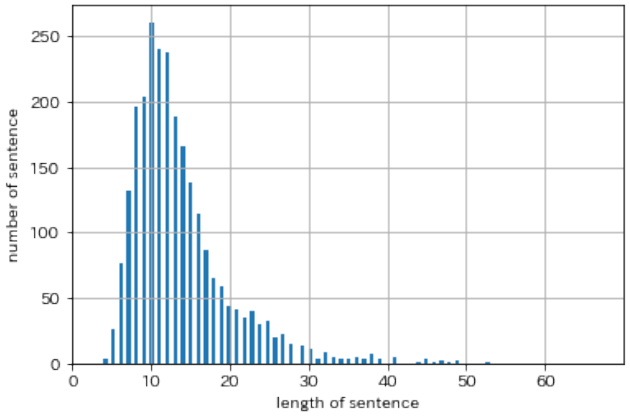

Figure 1 The Length of Sentence in the Japanese corpus

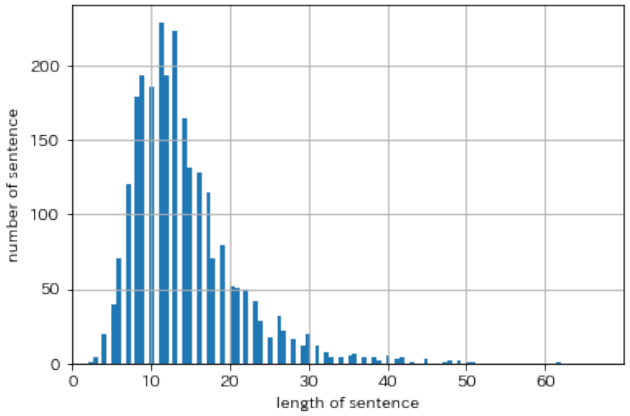

Figure 2 The Length of Sentence in the English corpus

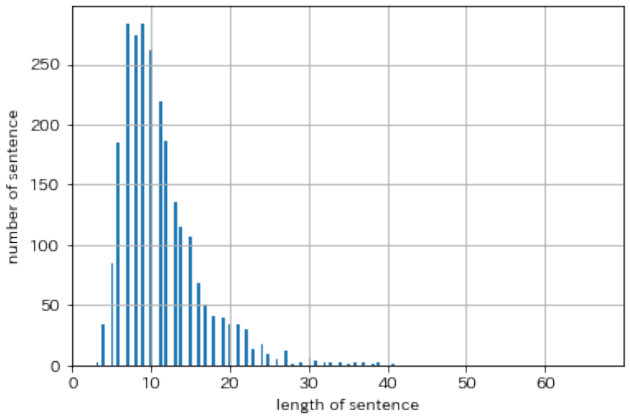

Figure 3 The Length of Sentence in the Chinese corpus

In the task dataset, eight diseases or symptoms, including Cold, Cough, Diarrhea, Fever, Hay fever, Headache, Flu and Runny nose, are set up as labels for classification. Every pseudo-tweet is labeled as 'p' for positive and 'n' for negative for each disease or symptom. In every subtask corpus, the ID of the pseudo-tweet corresponds to the corpora of other language subtasks. For example, the pseudo-tweet with ID '1en' (English) corresponds to the pseudo-tweets with ID '1zh' (Chinese) and '1ja' (Japanese). As the corpora are built parallel, the labellings of the corresponded pseudo-tweets are identical.

Before building a text graph, there is a necessity of tokenizing the tweets, especially for the Chinese and the Japanese corpora. We used jieba [9], one of the famous Chinese word segmentation tool, to tokenize the Chinese tweets. Moreover, the Japanese tweets are broken down into tokens with MeCab [6], a widely used Japanese morphological analysis tool. The average token-level length of the corpora are 13.7 (Japanese), 14.1 (English) and 11.2 (Chinese). Figure 1, 2 and 3 show the distribution of the token-level length of the pseudo-tweets in each corpus.

4.3 Parameters

Inspired by the work of Yao et al. [13], the parameters are initially set up as showing in the table below.

Table 2 Parameters

Parameter	Value
The first embedding size	200
Window size	20
Dropout rate	0.5
Learning rate	0.02
Number of epochs	200

4.4 Evaluation Metrics

In order to evaluate our model and make a clear comparison with the baseline, we compute the same evaluation metrics described in the NTCIR-13 MedWeb task overview paper [10]. The performance of our model in the subtasks is assessed by computing the exact match accuracy, F1-micro and macro based on precision and recall, and Hamming loss.

5 Results and Discussions

The results obtained in the experiments are given in this section. Following this, the analysis of the results are presented.

5.1 Results

Table 3 shows the performance of the proposed system on the three subtask test datasets and the results of the baseline system, where P stands for precision and R stands for recall.

Moreover, we present the experimental results of every label for each subtask. The detail of the results are given in Tables 4, 5 and 6.

5.2 Discussions

Looking into the results given in Table 3, it is clear that our method could not outperform the baseline system. As one of

Table 3 Experiment results

	ja	en	zh	baseline
Exact match	0.728	0.681	0.716	0.880
P (macro)	0.801	0.758	0.770	0.887
R (macro)	0.785	0.757	0.781	0.925
F1 (macro)	0.791	0.755	0.775	0.906
P (micro)	0.810	0.775	0.784	0.899
R (micro)	0.794	0.754	0.791	0.941
F1 (micro)	0.802	0.764	0.787	0.920
Hamming loss	0.046	0.054	0.050	0.019

Table 4 Classification report of the Japanese subtask

	P	R	F1	support
Influenza	0.571	0.667	0.615	24
Diarrhea	0.879	0.797	0.836	64
Hayfever	0.867	0.848	0.857	46
Cough	0.885	0.863	0.873	80
Headache	0.877	0.740	0.803	77
Fever	0.673	0.774	0.720	93
Runnynose	0.855	0.813	0.833	123
Cold	0.805	0.778	0.791	90
avg / total	0.818	0.794	0.804	597

Table 5 Classification report of the English subtask

	P	R	F1	support
Influenza	0.515	0.708	0.597	24
Diarrhea	0.845	0.766	0.803	64
Hayfever	0.800	0.783	0.791	46
Cough	0.869	0.913	0.890	80
Headache	0.741	0.779	0.760	77
Fever	0.705	0.667	0.685	93
Runnynose	0.806	0.707	0.753	123
Cold	0.786	0.733	0.759	90
avg / total	0.779	0.754	0.765	597

Table 6 Classification report of the Chinese subtask

	P	R	F1	support
Influenza	0.615	0.667	0.640	24
Diarrhea	0.849	0.875	0.862	64
Hayfever	0.729	0.761	0.745	46
Cough	0.870	0.838	0.854	80
Headache	0.822	0.779	0.800	77
Fever	0.731	0.817	0.772	93
Runnynose	0.803	0.797	0.800	123
Cold	0.744	0.711	0.727	90
avg / total	0.786	0.791	0.788	597

the latest graph neural network method for NLP, the advantage of Text GCN is its ability of jointly learning both word and document embeddings with an initialization of one-hot representation. Furthermore, experimental results suggest the robustness of Text GCN as it still performs well with a lower percentage of training data [13]. However, modeling word co-occurence and word importance in document by building a text graph also brings limitations. Unlike attention network or LSTM, the graph structrue in Text GCN ignores the word orders in documents, which might be very useful in text classification especially when the documents are relatively short [13]. In the work of Yao et al., experimental results also show that Text GCN did not overperform the baseline system on a short text corpus.

Furthermore, the limited size of vocabulary and corpus leads to a relatively small text graph, which could be another limitation for Text GCN. A small text graph directly results in a lack of edges. With a small number of edges in the graph, the calculated PMI and TF-IDF values could not well model the word co-occurence and term weight in corpus because Text GCN only utilizes the information from the input corpus.

Moreover, the values given in the support column of Tables 4, 5 and 6 suggest the existence of a class imbalance problem. For example, among 640 tweets in the test dataset, only 24 tweets are labeled positive for Influenza. This problem can also be found in the training set, which means the whole dataset suffers from a class imbalance problem [11]. For each label, there are many more tweets labeled negative than those labeled positive. This problem leads to a low classification accuracy of our model.

In conclusion, in our experiments, Text GCN could not overcome the problems above to show a better performace on the NTCIR-13 MedWeb Dataset than the baseline system.

6 Conclusion

In this work, we explore multi-label classification using Text Graph Convolutional Networks on the NTCIR-13 MedWeb Dataset. The perfomance of Text GCN on the dataset is evaluated. Although our method could not outperform the baseline system, Text GCN still shows its potential in multi-label classification with limited labeled data. For future work, we plan to apply this novel neural network method to other classification tasks related to medical language processing.

Acknowledgements

This research was undertaken with the support from The Real Sakai Laboratory [8], Waseda University. I would like to express my thanks of gratitude to members in The Real Sakai Laboratory, who helped me and gave me many useful pieces of advice for continuing the research.

Last but not least, I would like to express my specail thanks to my parents, my cousin and my lovely nephew, who motivated me to move on and be a better person.

References

[1] Aggarwal, C. C. and Zhai, C.: A Survey of Text Classification Algorithms, *Mining Text Data* (2012).

[2] Aramaki, E., Maskawa, S. and Morita, M.: Twitter Catches The Flu: Detecting Influenza Epidemics using Twitter, *EMNLP 2011* (2011).

[3] Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, *Neural Computation*, Vol. 9, pp. 1735–1780 (1997).

[4] Iso, H., Ruiz, C. A., Murayama, T., Taguchi, K., Takeuchi, R., Yamamoto, H., Wakamiya, S. and Aramaki, E.: MedWeb Task : Multi-label Classification of Tweets using an Ensemble of Neural Networks, *In Proceedings of the 13th NTCIR Conference on Evaluation of Information Access Technologies (NTCIR-13)* (2017).

[5] Kipf, T. N. and Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks, *CoRR*, Vol. abs/1609.02907 (2016).

[6] Kudo, T., Yamamoto, K. and Matsumoto, Y.: Applying Conditional Random Fields to Japanese Morphological Analysis, *EMNLP* (2004).

[7] Li, Q., Han, Z. and Wu, X.-M.: Deeper Insights Into Graph Convolutional Networks for Semi-Supervised Learning, *AAAI* (2018).

[8] RSL: The Sakai Laboratory (2018). `http://sakailab.com`.

[9] Sun, J.: jieba (2017). `https://github.com/fxsjy/jieba`.

[10] Wakamiya, S., Morita, M., Kano, Y., Ohkuma, T. and Aramaki, E.: Overview of the NTCIR-13 : MedWeb Task, *In Proceedings of the 13th NTCIR Conference on Evaluation of Information Access Technologies (NTCIR-13)*, pp. 40–49 (2017).

[11] Wang, C.-K., Singh, O., Tang, Z.-L. and Dai, H.-J.: Using a Recurrent Neural Network Model for Classification of Tweets Conveyed Influenza-related Information, *DDDSM@IJCNLP* (2017).

[12] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. J. and Hovy, E. H.: Hierarchical Attention Networks for Document Classification, *HLT-NAACL* (2016).

[13] Yao, L., Mao, C. and Luo, Y.: Graph Convolutional Networks for Text Classification, *AAAI* (2019).

[14] Zhang, X., Zhao, J. J. and LeCun, Y.: Character-level Convolutional Networks for Text Classification, *NIPS* (2015).