Graph Convolutional Networks for Text Classification

COURSE CODE: TOPICS IN DEEP LEARNING - CS7.602 (2021-2022)

Advisor: Prof. Charu Sharma Prof. Makrand Tapaswi; **Project Mentor:** Tushar Abhishek
Project Members: Prateek Jaiswal (2021701009) , Pawan Baswani (2021701035) , Madan Nandiwada (2018900075)

Abstract:
This paper made an attempt to introduce the GCNs for text classification task and proved that GCNs are effective for NLP. Text classification is an important issue in natural language processing (NLP). Text classification has several uses, including document organisation, news filtering, spam detection, opinion mining, and computational phenotyping.
We create a single text graph for a corpus based on word co-occurrence and document word relations, then train a Text Graph Convolutional Network (Text GCN) for the corpus. Our Text GCN is trained using a one-hot representation for each word and document, and it then jointly learns the embeddings for both words and documents, as supervised by the known class labels for documents.

Introduction:
Text classification is the activity of labeling natural language texts with relevant categories from a predefined set. It can be useful in understanding the customer behavior by categorizing the conversation on different social networks, feedback and other web sources. Text classification is being used in many domains, ranging from document filtering to automated metadata generation, word sense disambiguation, a populace of hierarchical catalogs of Web resources, and in general any application that requires document organization or selective and adaptive document send off. Text classification starts from feature extraction where first we try to get the important features/words from the text in order to distinguish them. So its important to get useful features rather than all the features from the corpus.

Dataset:
We are using a set of 5 widely benchmark corpora for this task as we want to classify several domain and check the performance on any kind of domain. Following (refer Table 1) are the datasets 20-Newsgroups (20NG), Ohsumed, R52 and R8 of Reuters 21578 and Movie Review (MR). Apart from these datasets, all the indic languages are crawled from different news websites and extracted the raw text from html format along with the news article category for training. Table 1, detailed the dataset statistics.

Datasets	Docs	Training	Test	Words	Nodes	Classes	Average length
20NG	18846	11314	7532	42757	61603	20	221.26
R8	7674	5485	2189	7688	15362	8	65.72
R52	9100	6532	2568	8892	17992	52	69.82
Ohsumed	7400	3357	4043	14157	21557	23	135.82
MR	10662	7108	3554	18764	29426	2	20.39
Hindi	38121	36214	1907	17383	42661	5	471.18
Marathi	15884	15099	785	26125	23306	7	742.10
Bengali	20541	19500	1041	43725	30876	8	561.05
Telugu	38526	36599	1927	20674	43231	9	310.98
Tamil	35591	33811	1780	34694	42461	12	543.46
Malayalam	38104	36201	1903	31793	45975	5	708.96
Kannada	36647	34814	1833	35006	44660	9	469.17
Gujarathi	36922	35079	1843	50028	47871	11	545.40

Table 1: *Data Statistics*

Results and Discussions:
For training the model we have used 100 epochs with early stopping. GNN model contains 2-layers with ReLU activation, learning rate=0.02, input_dim=300, output_dim=200 with loss as Cross entropy. We have

used 2 GPUs, Maximum ram-size=78GM, disk space 10GB for performing experiments. Table 2, shows the experimental results with best model performance on each language. Figure **??**, shows the visualisation of different dataset data points with TSNE (for GCN Model).

Dataset	LSTM	Bi-LSTM	GCN	GAT	SAGE	BERT
20ng	0.62	0.74	**0.85**	0.42	0.74	0.16
oshumed	-	0.44	0.68	0.87	**0.97**	0.21
R8	0.89	0.95	**0.96**	0.90	0.94	0.93
R52	-	-	0.93	0.87	**0.96**	-
MR	0.72	0.72	**0.77**	0.76	0.76	0.62
Hindi	0.92	0.94	**0.89**	**0.84**	0.54	**0.94**
Telugu	0.85	**0.86**	0.63	**0.73**	0.69	0.54
Kannada	**0.85**	0.84	0.78	0.76	**0.78**	0.76
Bengali	0.53	0.54	0.68	0.73	0.67	**0.76**
Marathi	0.53	0.42	**0.699**	0.695	0.68	0.68
Tamil	0.85	**0.86**	0.81	0.68	0.43	0.82
Malyalam	0.61	0.62	-	**0.6269**	-	0.62
Gujrathi	0.92	0.92	-	-	-	**0.93**

Table 2: *Model Results*

Observations:

1. Most of the indic languages (that we have used) don't have the packages to preprocess. Due to unavailability of standard packages, the data contain some noisy text like emojis, urls (in indic script), unnecessary punctuations (which will not match with string.punctuation). This impact on graph construction (as the graph purely constructed on word-word and doc-word relations)
2. Indic-scripts are morphologically rich languages. This result in out of vocabulary words which slight change in the syntax of any word. Also, this impact on graph construction by considering the sub-words instead of words (increasing the number of nodes). This can be resolved with the language specific tokenizer.
3. For some of the indic-languages (hindi, telugu, kannada, and gujarati), we have used almost 1/4th of original data size due to "Out of memory" error for all GNN models. Whereas for NLP models we have used entire data size for all languages.
4. Compared to the NLP models, GNN models performing better even though we perform the experiments with very less data.
   [1] [17] [2] [10] [3] [4] [7] [9] [8] [9] [6] [11] [12] [5] [13] [15] [14] [3] [16]

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[4] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE international joint conference on neural networks*, volume 2, pages 729–734, 2005.

[5] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[6] Hochreiter and Schmidhuber 1997. Long short-term memory. *Neural computation*, 1997.

[7] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[8] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura E. Barnes, and Donald E. Brown. Text classification algorithms: A survey. *Information*, 10(4), 2019.

[9] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

[10] Yuan Luo Liang Yao, Chengsheng Mao. Graph convolutional networks for text classification. *AAAI Conference on Artificial Intelligence (AAAI 2019)*, 2019.

[11] M. E. Maron and J. L. Kuhns. On relevance, probabilistic indexing and information retrieval. *J. ACM*, 7(3):216–244, jul 1960.

[12] Alessio Micheli. Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks*, 20(3):498–511, 2009.

[13] Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 world wide web conference*, pages 1063–1072, 2018.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[15] Sancheza V.David. Advanced support vector machines and kernel methods. *Neurocomputing*, 55(1):5–20, 2003. Support Vector Machines.

[16] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *stat*, 1050:20, 2017.

[17] Kim Y. Convolutional neural networks for sentence classification. *EMNLP*, 2014.