

# EEE 586: Survey for the Term Project

## Text Classification with Graph Neural Networks

Tuna Alikasıfoğlu

*Dept. of Electrical and Electronics Engineering*

*Bilkent University*

[t.alikasifoglu@bilkent.edu.tr](mailto:t.alikasifoglu@bilkent.edu.tr)

Arda Can Aras

*Dept. of Electrical and Electronics Engineering*

*Bilkent University*

[can.aras@bilkent.edu.tr](mailto:can.aras@bilkent.edu.tr)

*Abstract*—Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

*Index Terms*—Text classification, graph neural networks, GNN

### I. INTRODUCTION

- Introduction to Text Classification task, overview of the previous approaches, in [Section II-A](#).
- Introduction to Graph Neural Networks (GNNs), what is GNN, why they are being utilized, in [Section II-B](#).
- GNN in NLP, more specifically in Text Classification in [Section III](#).

## II. PREVIOUS WORK

### A. Text Classification

*Text classification* (text categorization) is the “procedure of designating predefined labels for the text”. The task is to assign labels or tags to the text based knowledge, i.e., textual units such as sentences, paragraphs and documents, where the labels are usually defined by humans, but can also be defined by the machine. This task is a fundamental part of Natural Language Processing (NLP), and it is significant to its applications such as sentiment analysis, question answering, text summarization, etc. [1]. Text classification task can be partitioned into five phases as preprocessing, feature extraction, dimensionality reduction (optional), classifier selection and evaluation:

1) *Preprocessing*: Text preprocessing is a crucial prerequisite for a successful feature extraction. The input of the text classification frameworks consists of raw text data, which are in the form of a sequence of sentences. In this step, “cleaning” of the text datasets is performed to transform the data into a form that is suitable for feature extraction. The cleaning process is usually performed by tokenization, capitalization, slang and abbreviation handling, noise removal, spelling correction, stemming and lemmatization [2].

2) *Feature Extraction*: After preprocessing step, another crucial step, feature extraction step is necessary. Two common methods of text based feature extraction are weighted word and word embedding techniques. In the weighted word aspect, we have old techniques like bag-of-words and term frequency-inverse document frequency (TF-IDF). In the relatively recent aspect, we have the word embedding techniques like *word2vec*, *GloVe*, *FastText*, etc. [2].

3) *Dimensionality Reduction*: The dimensionality reduction is an optional step of a text classification task, but based on the size of the dataset, it may be a must to have a computable result. In this aspect of the task, we try to reduce the dimensionality of the feature space while preserving the information of the original features space. Possible dimensionality reduction techniques include (principal / independent) component analysis, linear discriminant analysis, non-negative matrix factoriza-

tion, random projection, autoencoder and stochastic neighbor embedding [2].

4) *Classifier Selection*: Selecting the optimal classifier is the most important aspect of a text classification task. Currently we have both traditional and deep learning oriented classifiers. The traditional classifiers are based on the statistical analysis of the training data, and the deep learning classifiers are based on the neural networks. The main distinction between the traditional and deep learning based approaches can be described as follows: Good feature extraction methodology is crucial for the traditional classifiers. They obtain sample features by artificial methods and then make classifications based on these features. Hence, the performance of the traditional classifiers are mainly restricted by feature extraction. On the other hand, by making feature mapping via nonlinear transformations a part of the learning process, deep learning based classifier selection can integrate feature extraction aspect into the model fitting process [1].

Some traditional classifiers are logistic regression, (kernel) support vector machine, Naive Bayes,  $k$ -nearest neighbors, decision tree, random forest, etc. [1], [2]. On the other hand, the deep learning classifiers are usually based on the neural networks, such as deep feed forward neural networks, convolutional neural networks (CNNs), recurrent neural networks (RNNs), lately attention and transformer based models such as BERT [3] variations and fine tuning pre-trained language models [4], and finally what we will focus on, graph neural network (GNN) based models [1], [2], [5].

5) *Evaluation*: Evaluation is step that we understand how the our model performs under the given text classification task. There are several evaluation metrics that can be used to evaluate the performance of a supervised technique. The most common metrics are accuracy,  $F_\beta$ -score, micro/macro-averaging. Although we also have metrics like Matthews correlation coefficient and receiver operating characteristics (ROC) [1], [5]. In order to evaluate the performance of our model, based on the provided techniques, we need to use labeled data which are preferably specific to our text classification task, i.e., we need benchmark datasets like GLUE [6], TweetEval [7], among others.

### *B. Graph Neural Networks*

[8]–[11].

## III. RELATED WORK

[12]–[16]

## REFERENCES

- [1] Q. Li, H. Peng, J. Li, *et al.*, “A survey on text classification: From shallow to deep learning,” *CoRR*, vol. abs/2008.00364, 2020. arXiv: 2008 . 00364. [Online]. Available: <https://arxiv.org/abs/2008.00364>.
- [2] Kowsari, J. Meimandi, Heidarysafa, Mendu, Barnes, and Brown, “Text classification algorithms: A survey,” *Information*, vol. 10, no. 4, p. 150, Apr. 2019, ISSN: 2078-2489. DOI: 10.3390/info10040150. [Online]. Available: <http://dx.doi.org/10.3390/info10040150>.
- [3] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” *CoRR*, vol. abs/1810.04805, 2018. arXiv: 1810.04805. [Online]. Available: <http://arxiv.org/abs/1810.04805>.
- [4] J. Howard and S. Ruder, “Fine-tuned language models for text classification,” *CoRR*, vol. abs/1801.06146, 2018. arXiv: 1801 . 06146. [Online]. Available: <http://arxiv.org/abs/1801.06146>.
- [5] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, “Deep learning based text classification: A comprehensive review,” *CoRR*, vol. abs/2004.03705, 2020. arXiv: 2004 . 03705. [Online]. Available: <https://arxiv.org/abs/2004.03705>.
- [6] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, “GLUE: A multi-task benchmark and analysis platform for natural language understanding,” arXiv preprint 1804.07461, 2018. [Online]. Available: <https://gluebenchmark.com/>.
- [7] F. Barbieri, J. Camacho-Collados, L. Espinosa-Anke, and L. Neves, “TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification,” in *Proceedings of Findings of EMNLP*, 2020.
- [8] J. Zhou, G. Cui, S. Hu, *et al.*, “Graph neural networks: A review of methods and applications,” *AI Open*, vol. 1, pp. 57–81, 2020, ISSN: 2666-6510. DOI: <https://doi.org/10.1016/j.aiopen.2021.01.001>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2666651021000012>.
- [9] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A comprehensive survey on graph neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, Jan. 2021, ISSN: 2162-2388. DOI: 10 . 1109 / tnnls . 2020 . 2978386. [Online]. Available: <http://dx.doi.org/10.1109/TNNLS.2020.2978386>.
- [10] Z. Zhang, P. Cui, and W. Zhu, “Deep learning on graphs: A survey,” *CoRR*, vol. abs/1812.04202, 2018. arXiv: 1812 . 04202. [Online]. Available: <http://arxiv.org/abs/1812.04202>.
- [11] L. Sun, J. Wang, P. S. Yu, and B. Li, “Adversarial attack and defense on graph data: A survey,” *CoRR*, vol. abs/1812.10528, 2018. arXiv: 1812.10528. [Online]. Available: <http://arxiv.org/abs/1812.10528>.
- [12] L. Yao, C. Mao, and Y. Luo, *Graph convolutional networks for text classification*, 2018. arXiv: 1809.05679 [cs.CL].
- [13] T. N. Kipf and M. Welling, *Semi-supervised classification with graph convolutional networks*, 2017. arXiv: 1609.02907 [cs.LG].
- [14] H. Peng, J. Li, Q. Gong, *et al.*, *Hierarchical taxonomy-aware and attentional graph capsule rcnns for large-scale multi-label text classification*, 2019. arXiv: 1906.04898 [cs.IR].
- [15] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, *Graph attention networks*, 2018. arXiv: 1710.10903 [stat.ML].
- [16] Z. Guo, Y. Zhang, and W. Lu, “Attention guided graph convolutional networks for relation extraction,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 241–251. DOI: 10 . 18653 / v1 / P19 - 1024. [Online]. Available: <https://aclanthology.org/P19-1024>.