Reproducing Bag-of-Words vs. Graph vs. Sequence in Text Classification: Questioning the Necessity of Text-Graphs and the Surprising Strength of a Wide MLP (ACL 2022)

Mahak Pandia and Sahana Ramnath
University of Southern California
{pandia, sramnath}@usc.edu

1 Introduction

Text classification is the task of assigning topical categories to text units out of a finite number of predefined categories i.e. given a set of predefined classes, $\mathcal{C} = \{c_1, ..., c_K\}$ and text *t*, text classification is the assignment of *t* to class *c* where $c \in \mathcal{C}$.

Why is Text Classification relevant in today's world? In the past few decades, owing to advances in communications, storage systems and computing, enormous amounts of electronic data is being generated everyday, with 80-90% of the data being unstructured in nature. Text classification helps businesses extract meaning from this unstructured data and hence, increase their value.

Contemporary research in Text Classification: Text classification is a widely researched today in the field of natural language processing - using machine learning methods provides a faster, more efficient and more effective way to classify text automatically. There is a large amount of literature available on the different methods / models used for text classification (some recent surveys - Kadhim (2019); Kowsari et al. (2019); Bayer et al. (2021)). While these models help increase business value, it does not offset the computational costs of using these methods.

In this work, we aim to reproduce Bag-of-Words vs. Graph vs. Sequence in Text Classification: Questioning the Necessity of Text-Graphs and the Surprising Strength of a Wide MLP, a text-classification based research work published at ACL 2022 (Galke and Scherp, 2022). The original work focuses on three different families of text classification methods - it compares the performance and computational costs of 16 different models for text classification task on five different datasets in an inductive setting. The three families of methods are as follows:

- **Bag-of-Words (BoW) based models:** These are multi-token models that use the number of occurrences of all tokens in the input, while disregarding word order and position. These then rely on word embeddings and fully connected feedforward layer(s) (Kowsari et al., 2019; Kadhim, 2019) to produce the output.

- **Graph based models:** A Graph Neural Network (GNN) is a neural network that directly works with graph-structure data and hence, requires setting up a synthetic graph induced from input text. GNNs have gained popularity in recent years and have led to graph-based text classification methods becoming more common and prevalent (Yao et al., 2019; Ragesh et al., 2021; Liu et al., 2020).

- **Sequence based models:** Transformers such as BERT and recurrent networks such as LSTMs fall under the category of sequence based neural networks. In transformers, the input is a (fixed-length) sequence of tokens, which is then fed into multiple layers of self-attention. In recurrent networks, the input tokens are recursively fed to the network. In both cases, the final embedding produced is used for producing the text-classification output.

By making and discussing these model comparisons, the original work's authors show that BoW-based models provide performances that are competent, if not higher, than graph-based models on different text classification tasks, while being more computationally efficient in terms of space and time requirements. Hence, their experiments corroborate that simpler models can be strong competitors of more advanced models in industries, and contribute to the way these models are applied in large scale applications.

2 Scope of reproducibility

The original work compares the the *performance*, *setting*, and *speed* of three different kinds of neural networks (as described in Section 1) on text classification tasks (5 different datasets). They do this by comparing the test set accuracies and training/inference runtimes of multiple models on the chosen 5 datasets. We aim to reproduce all the model training/inferences that are done by the original work, as well as two additional experiments which were reported (from prior works) but not done in the original work.

The primary claim from the original work that we aim to reproduce is that simple Bag-of-Words based multi-layer perceptron (MLP) models perform effectively on text categorization tasks, and also provide *higher accuracy* than graph based models in the inductive setting. We do this by training and testing the corresponding models on all considered datasets.

The second claim that we aim to reproduce is that BERT-based models produce state-of-the-art performance on all tasks, but are significantly worse in terms of computational efficiency. We do this by training and testing the relevant models on the considered datasets, as well as measuring and comparing the runtimes.

2.1 Addressed claims from the original paper

Claims from the original work that we are testing in our reproduction study:

- Bag-of-words MLP > graph-based models in the inductive setting, in terms of model performance

- BERT-based models > all other models in terms of model performance, *but* are much worse in terms of training and inference speeds.

2.2 Extension to the original work

The original work extensively discusses the performance of graph-based models developed in previous works by directly using the test accuracy numbers reported in those works (i.e., the authors do not re-run the graph-based models). To provide a complete analysis of the original work's claims, we re-run one of the discussed graph-based models (TextGCN (Yao et al., 2019)) in the transductive and inductive settings, and report those results as well.

3 Methodology

In this section, we describe our methodology towards reproducing the original work. We first describe the models, datasets, and hyperparameters used by the authors. We then discuss our implementation, experimental setup and computational requirements.

3.1 Model descriptions

There are 16 different models that are used in this paper, divided into 3 different classes.

- **Graph-based models:** The authors of the paper do not run their own experiments but rely on the results obtained from previous works for the graph-based models. These models include TextGCN, HeteGCN, HyperGAT, DADGNN, TensorGCN and SGC (Ding et al., 2020; Ragesh et al., 2021; Yao et al., 2019; Liu et al., 2020). In our reproduction, we provide our own results for Text-GCN (transductive and inductive settings), to ensure a fair comparison.

- **Bag-of-Words based models:** The authors run experiments on wide-MLP models (with different types of input tokenizations and embeddings), and report scores for fastText, SWEM and logistic regression from a prior work (Ding et al., 2020). For the MLP models, the authors consider pure Bag-of-Words, weighted TF-IDF as well as averaged GloVe representations (Pennington et al., 2014).
  - WideMLP: This model has one hidden layer with 1,024 rectified linear units (one input-to-hidden and one hidden-to-output layer). Dropout is applied after each hidden and initial embedding layer, except for GloVe+WideMLP wherein dropout and ReLU are not applied to the pretrained embeddings.
  - WideMLP-2: This models has two ReLU activated hidden layers with 1,024 units in each layer.

- **Sequence based models:** The authors consider the following three sequence based models:
  - LSTM: The scores from a pretrained LSTM are reported by the authors (Ding et al., 2020).

- BERT models: The authors fine tune BERT (Devlin et al., 2018) and DistilBERT (Sanh et al., 2019), and run their own experiments on these. Hyperparameters for fine tuning:

  epochs = 10
  learning rate = $5 \cdot 10^{-5}$
  maximum input sequence length = 512

  The authors further run two more experiments with their BERT models. For the first experiment, they set all position embedding to zero in order to remove the effect of word ordering (BERT w/o pos emb) and for the second experiment, they shuffle each sequence to augment training data (BERT w/ shuf. Aug.).

3.2 Data descriptions

Experiments are run on five widely-used text classification datasets:

1. **20NG[1] (bydate version):** This dataset contains long posts/documents categorized into 20 classes. There are 11.3k documents in the train set, and 7.5k documents in the test set. On average, the documents are 550 words long, with a standard deviation of 2047.

2. **Ohsumed[2]:** This dataset consists of medical abstracts from the MEDLINE database, which are categorized into 23 disease classes. There are 3.3k documents in the train set, and 4k documents in the test set. On average, the documents have a length of 285 ± 123 words.

3. **R52 and R8 (all-terms version), two subsets of Reuters 21578[3]:** The Reuters dataset is a collection of documents that appeared in the Reuters newswire in 1987. R8 has 8 categories, with 5.4k documents in the train set, and 2.2k documents in the test set. On average, R8's documents are 119 ± 128 words long. R52 has 52 categories, with 6.5k documents in the train set, and 2.6k documents in the test set. On average, R52's documents are 126 ± 133 words long.

4. **Movie Review[4] (MR)** (Pang and Lee, 2005): This is a binary sentiment classification task on movie reviews. Each review contains just one sentence. There are 5.3k positive and negative reviews each, and the training/test dataset splits from Tang et al. (2015) are used. On average, the sentences are 25 ± 11 words long.

We use the same datasets and train-test splits used in the original work Galke and Scherp (2022) (they in turn, use the same datasets/splits and preprocessing steps done in Yao et al. (2019)).

Preprocessing details (taken from Yao et al. (2019)): The datasets are first cleaned and tokenized as done in Kim (2014). Then, stop words (detected using NLTK, as well as low frequency words ($<$ 5k occurrences in the dataset)) are removed. Words were not removed for the Movie Reviews dataset since the documents were already very short.

3.3 Implementation

The authors of the original work have made their code public on Github (code). For reproducing the experiments, we write our own code closely following their code and its structure. Our code can be found here. Apart from the code, we also provide instructions/commands to download the data, to create a virtual environment with required libraries, and to run all experiments.

To run TextGCN (Yao et al., 2019) (extension to original work), we directly use the code released by the corresponding authors (here is the code for transductive setting, code for inductive setting).

3.4 Hyperparameters

We use the same set of hyperparameters used by the original work, except for cases where we had to change them to fit into our computational resources. We get the exact hyperparameter configurations the authors used from the run scripts in their Github repository.

Bag-of-Words MLP models:

- **Batch-size:** 16
- **Learning rate:** 0.001
- **Epochs:** 100

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

[2] http://disi.unitn.it/moschitti/corpora.htm

[3] https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection

[4] https://www.cs.cornell.edu/people/pabo/movie-review-data/

- **Number of hidden layers:** 1, 2 and 3 layers
- **Hidden units:** 1024
- **Dropout:** 0.0 for the embedding layer, 0.5 otherwise
- **Tokenization and embeddings:** Either GloVe representations (Pennington et al., 2014) or BERT-base uncased tokenization

BERT models:

- **Batch-size:** 8 (the original work used 16, but we had to reduce it because of limited computational resources)
- **Learning rate:** 0.00005
- **Epochs:** 10
- **BERT variant used:** Base and uncased
- **Tokenization:** BERT/DistilBERT-base uncased tokenization

Text-GCN:

- **Batch-size:** 256
- **Learning rate:** 0.001
- **Epochs:** 50 or 200 (depending on dataset)
- **Hidden units:** 200

3.5 Experimental setup

Our code to reproduce the original work can be found in this Github repository. We run all experiments on a single 1080ti GPU. We provide commands to run all experiments in our README.

3.6 Computational requirements

Estimated requirements: To reproduce the results of the original work, we are required to run experiments with training MLPs from scratch (5 models x 5 datasets) and fine-tuning BERT and DistilBERT (4 models x 5 datasets). On a CPU-only machine, an MLP training experiment takes less than 10 minutes to run, so we estimate at max, 4 hours to run these experiments. We estimate a similar training time for the BERT-based experiments as well. Further, experiment runtimes provided in the original work also indicate similar durations.

Actual requirements: We were able to run all experiments on a single 1080ti GPU within reasonable time. We provide the actual runtime (in seconds) for all our experiments in Table 2. To summarize, every MLP experiment took between 3-12 minutes to run depending on the size of the dataset *and* the average size of the input sentences, and every BERT experiment took between 8 minutes and 1.4 hours. The TextGCN experiments took between half a minute to 2.4 hours.

4 Results

Overall, we found that the claims made by the original work were fully reproducible, in terms of actual test accuracy numbers and in terms of observed trends. All of our reproduced test accuracies can be found in Table 1. The results are split into Bag-of-Words (BoW) based models, Sequence-based models and Graph-based models. The table has both results reproduced by us, as well as results directly taken from other works (the latter rows all have corresponding citations mentioned).

4.1 Result 1

The first claim made by the original work is that bag-of-words based MLP models are better than graph-based models (in the *inductive* setting), in terms of test accuracy. As we see in Table 1, for all 5 datasets, the BoW models' test accuracies are better than, or competent with the test accuracies produced by the graph-based models in the inductive setting. We verify this result using our reproduced MLP and Text-GCN (transductive and inductive) models, as well as other BoW models and graph-based models from literature.

4.2 Result 2

The second claim by the original work is that BERT-based models are the best in performance for text classification (in comparison to both BoW models and graph-based models), but are much worse in terms of training and inference speeds. We see from Table 1 that the BERT models are indeed the best in performance for all 5 datasets, and we see from Table 2, that the BERT models are much worse in time complexity as compared to MLP models, and most graph-based models.

Note: It is to be noted that the reported runtimes for the MLP and BERT models include data creation, training, *and* inference times, whereas the reported runtimes for Text-GCN (transductive and inductive) include only the training and inference times (since data/graph creation is done as a separate preprocessing step).

Model	**20ng**	**R8**	**R52**	**ohsumed**	**MR**
BoW models					
Logistic Regression (Ragesh et al., 2021)	83.70	93.33	90.65	61.14	76.28
SWEM (Ding et al., 2020)	85.16	95.32	92.94	63.12	76.65
fastText (Ding et al., 2020)	79.38	96.13	92.81	57.70	75.14
TF-IDF + MLP	80.18	96.44	93.81	59.58	75.63
MLP-1	78.03	96.94	94.04	57.66	76.36
MLP-2	74.02	97.26	91.9	52.19	76.2
GloVe + MLP-2	75.41	96.53	93.15	60.13	75.48
GloVe + MLP-3	70.51	96.57	92.33	57.11	76.62
Seq.-based models					
LSTM (pretrain) (Ding et al., 2020)	75.43	96.09	90.48	51.10	77.33
DistilBERT	86.76	98.13	96.65	72.79	85.14
BERT	87.64	98.17	97.00	74.3	86.47
BERT w/o pos. emb.	82.57	97.53	96.03	68.59	80.81
BERT w. shuf. Aug. (0.2)	87.24	97.99	96.57	74.33	86.1
Graph-based models					
Text-GCN (transductive)	86.23	96.94	93.61	68.32	76.56
Text-GCN (inductive)	81.43	89.77	85.59	55.11	75.46
More transductive graph models					
HeteGCN (Ragesh et al., 2021)	87.15	97.24	94.35	68.11	76.71
SGC (Wu et al., 2019)	88.5	97.2	94.0	68.5	75.9
TensorGCN (Liu et al., 2020)	87.74	98.04	95.05	70.11	77.91
More inductive graph models					
HeteGCN (Ragesh et al., 2021)	84.59	97.17	93.89	63.79	75.62
HyperGAT (Ragesh et al., 2021)	86.62	97.07	94.98	69.90	78.32
DADGCNN (Liu et al., 2021)	–	98.15	95.16	–	78.64

Table 1: Reproduced test accuracies of all models on the 5 datasets. All results that have been taken from previous works have been provided with citations. All rows without a cited work have been run by us.

5 Discussion

One of the broad aims of the original work was to encourage researchers to *not* disregard simple MLP models while developing models for the task of text classification. Recently, with the advent of strong transformer models, it has become the norm to directly jump to BERT-based models (or even graph-based models as the original work's authors discuss), without giving a single consideration to MLP models. With these results, the authors hoped to prove that MLP models are equally comptetent, and must not be disregarded.

That being said, the authors also discuss that for tasks where graph-based structures are actually needed to solve the task (such as when working with social graphs or network-based datasets), graph-based models are the state-of-the-art. They also discuss that in tasks where word-order information is relevant (such as question answering, or natural language inference), positional information is essential for good performance. In fact, even in the standard topical text classification task (such as the ones investigated by this work) where word order information isn't highly relevant, the performance does decrease when positional embeddings are removed (they compare BERT and BERT without positional embeddings to make this conclusion, refer Table 1 for the accuracy numbers).

Overall, given all these task-specific constraints and discussions, we consider the main takeaway of this paper to be that MLPs should *not* be disregarded when it comes to NLP tasks. Time and memory-complexity wise, they are always the best, and there are many cases where they are also the best (or competent) in task performance.

Model	20ng	R8	R52	ohsumed	MR
BoW models					
TF-IDF + MLP	685	280	332	201	344
MLP-1	578	258	312	180	334
MLP-2	599	273	324	187	358
GloVe + MLP-2	323	155	200	138	203
GloVe + MLP-3	364	178	204	143	205
BERT models					
DistilBERT	2217	679	802	588	206
BERT	4253	1322	1660	1192	357
BERT w/o pos. emb.	4304	1332	1635	1136	454
BERT w. shuf. Aug. (0.2)	5137	1604	1930	1511	417
Graph-based models					
Text-GCN (transductive)	2048	213	517	391	35
Text-GCN (inductive)	8500	732	3448	3594	2181

Table 2: Time taken (in seconds) to reproduce test accuracies of models on the 5 datasets

5.1 What was easy

Since the original work released a working and well-documented code, explicit instructions on how to download the data and setup the virtual environment, as well as run scripts with exact hyperparameter configurations, we found it easy to reproduce all their experiments. Apart from a few minor Pythor errors which we had to fix, the code ran smoothly.

5.2 What was difficult

The only difficulty we faced was in replicating the Text-GCN experiments which was our proposed extension. The data and run-scripts were provided, but the exact virtual environment to be setup was not specified - we faced mild difficulties in installing and using the correct tensorflow version for the same, since tensorflow has since been heavily updated. However, we would like to note here that this is a deficiency of the Text-GCN work (Yao et al., 2019) and not the original work (Galke and Scherp, 2022) we were reproducing.

5.3 Recommendations for reproducibility

Given that the data, the virtual environment and the code are all readily available, it is fairly easy to reproduce all experiments done by the original work. We recommend that future researchers who attempt to reproduce this work take the time to go over the code thoroughly (especially the hyperparameter specifications in the code) and try a variety of configurations in order to get a complete set of results.

6 Communication with original authors

We did not contact the original work's authors for any help regarding the reproduction of the experiments. However, we mailed the final report and code to them to gain feedback about our attempt at reproducing their work.

7 Conclusion

In this work, we successfully reproduced all experiments and claims made by Galke and Scherp (2022). Specifically, we found that for text-classification tasks, simple MLP models are highly competitive in terms of both task performance and computational complexity.

Ethical Considerations

The original work provides a discussion on ethical considerations of their work, which we reiterate here. Since the work is primarily about text-classification, any potential risk that exists for automated text classification (such as biases etc.) also exists for the methods used in this work. *However*, since the original work advocates for the use of MLPs that don't necessarily use pre-trained models (except for cases where they use pre-trained embeddings such as GloVe), the original work's authors claim that any work that uses their proposal should be able to handle biases by managing thier own training data.

References

Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2021. A survey on data augmentation for text classification. *ACM Computing Surveys*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. 2020. Be more with less: Hypergraph attention networks for inductive text classification. *arXiv preprint arXiv:2011.00387*.

Lukas Galke and Ansgar Scherp. 2022. Bag-of-words vs. graph vs. sequence in text classification: Questioning the necessity of text-graphs and the surprising strength of a wide mlp. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4038–4051.

Ammar Ismael Kadhim. 2019. Survey on supervised machine learning techniques for automatic text classification. *Artificial Intelligence Review*, 52(1):273–292.

Yoon Kim. 2014. Convolutional neural networks for sentence classification in: Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp), 1746–1751.. acl. *Association for Computational Linguistics, Doha, Qatar*.

Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. *Information*, 10(4):150.

Xien Liu, Xinxin You, Xiao Zhang, Ji Wu, and Ping Lv. 2020. Tensor graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8409–8416.

Yonghao Liu, Renchu Guan, Fausto Giunchiglia, Yanchun Liang, and Xiaoyue Feng. 2021. Deep attention diffusion graph neural networks for text classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8142–8152.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Rahul Ragesh, Sundararajan Sellamanickam, Arun Iyer, Ramakrishna Bairi, and Vijay Lingam. 2021. Hetegcn: Heterogeneous graph convolutional networks for text classification. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 860–868.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1165–1174.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377.