Short Text Classification using Inductive Graph Attention Network

Anup Roy[1] **Arnab Bhattacharya**[2] **Rachna Saxena**[3]
Nrutyangana Mohapatra[4] **Abhijeet Kumar**[5] **Mridul Mishra**[6]
[1] IIT Kanpur, India [2] IIT Kanpur, India [3] Fidelity Investments, India
[4] Fidelity Investments, India [5] Fidelity Investments, India [6] Fidelity Investments, India
{anupdogrial}@gmail.com {arnabb}@iitk.ac.in {Rachna.Saxena}@fmr.com
{Nrutyangana.Mohapatra,Abhijeet.Kumar,mridul.mishra}@fmr.com

Abstract

Traditional methods and deep learning approaches have struggled to effectively generalize to short texts due to their limited structural characteristics. In this paper, we propose a novel approach to address the challenges of short text classification in natural language processing by leveraging the power of the Inductive Graph Attention Network (GAT). Building upon the foundation laid by the authors of "InducT-GCN" [26], we extend their work and introduce the use of GAT [25] for experimentation.

Our method utilizes a similar approach to create feature and adjacency matrices, but by incorporating GAT, we achieve remarkable results for short text classification tasks. Furthermore, our model demonstrates comparative performance on longer texts, showcasing its potential in addressing text classification challenges across various lengths. By presenting an effective and superior approach that leverages the Inductive Graph Attention Network, our research contributes to advancing the field of text classification. The promising outcomes of our proposed method underscore its potential to enhance text classification tasks significantly.[1][2]

1 Introduction

Short text classification plays a vital role in various domains, including finance and healthcare, where categorizing intents and notes is crucial for decision-making. In the finance domain, one of the challenges is classifying the intent of an utterance. Similarly, in healthcare applications, the categorization of short notes written by medical practitioners is crucial for planning subsequent actions. However, traditional natural language processing (NLP) techniques struggle to handle the unique challenges posed by short texts. Limited length and lack of structural characteristics make it difficult to extract meaningful information using conventional methods. Existing text classification models primarily designed for longer texts, such as convolutional neural networks (CNN) [9] and recurrent neural networks (RNN) [29], often overlook global word co-occurrence and long-distance semantic relationships found in a corpus.

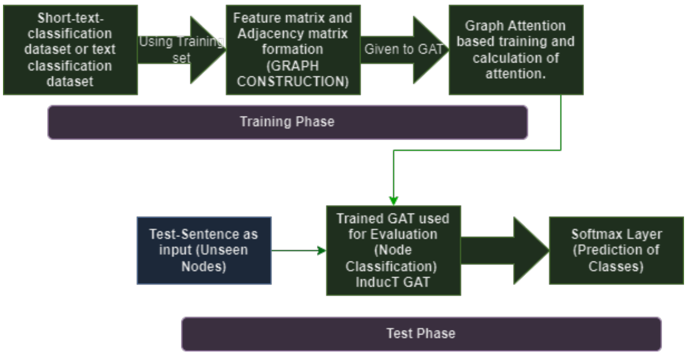
Short-text-classification dataset or text classification dataset
Using Training set
Feature matrix and Adjacency matrix formation (GRAPH CONSTRUCTION)
Given to GAT
Graph Attention based training and calculation of attention.
Training Phase
Test-Sentence as input (Unseen Nodes)
Trained GAT used for Evaluation (Node Classification) InducT GAT
Softmax Layer (Prediction of Classes)
Test Phase

Figure 1: Flow Diagram : InducT GAT based Model

Recent advancements in pre-trained models, such as BERT [2] and RoBERTa [13], have shown impressive results but come with computational demands and the need for external resources. To address these limitations, our proposed framework, InducT GAT, introduces an inductive graph-based approach for short text classification. Building upon the success of InducT GCN, our framework leverages the power of Inductive Graph Attention Networks (GAT) to effectively handle structured textual data. By constructing graphs and employing GAT learning, we capture global features and address the limitations of transductive models, overall flow of the system is depicted in Figure 1.

In this paper, we present our InducT GAT framework and its applications in short text classification across various domains. We evaluate its performance through extensive experiments and demonstrate its superiority compared to traditional methods and existing graph models. The promising results obtained validate the effectiveness of our approach in enhancing text classification tasks. Our research contributes to advancing the field of short text classification by introducing an inductive graph-based framework that overcomes the limitations of existing methods and improves performance in real-world scenarios.

2 Related Work

Text classification has evolved from Bag of Words (BoW) [4] based models to sequence and graph-based approaches. BoW-based models remain a solid foundation for text classification tasks like fastText [6]. Recurrent neural network (RNN) and long short-term memory (LSTM) [23] models are gaining popularity in natural language processing due to their ability to consider historical information and word order. Convolutional neural networks and transformer models like Bidirectional Encoder Representations (BERT) have emerged to enable parallel processing. The fusion of advanced models has transformed NLP, enabling

[1] Public Document

[2] Disclaimer: The views or opinions expressed in this paper are solely those of the author and do not necessarily represent those of Fidelity Investments. This research does not reflect in any way procedures, processes, or policies of operations within Fidelity Investments.

researchers and practitioners to tackle complex language tasks effectively.

Graph-based models, particularly graphical neural networks (GNNs) [21], are interesting because they can handle large-scale relational structures. Two main approaches for schematizing documents are HyperGAT [1] and Bert-GCN [11], which combine graph-based and BERT models. Special techniques have been developed to improve the accuracy of short text classification, such as heterogeneous graph attention networks (HGATs) [1], heterogeneous information networks (HINs), SHINE [16], and STGCN [24]. Recent improvements in graph-based classification (GCN) have focused on both transductive and inductive systems. Transformation-based GCNs deal with complexity and over-smoothing, while inductive-based models like DeepGL [19] and TGAT [30] are used for different graphs like transfer learning and topology learning. InducT-GAT is a text classification framework based on inductive graphs, which is a further extension of InducT GCN, utilizing the properties of inductive learning to facilitate generalization to unseen nodes.

3 Proposed Method

The inductive GAT-based text classification model signifies a significant leap in advancing transductive models toward an inductive learning framework. By exclusively utilizing training document information and excluding test set data, our model remains unbiased and impervious to the influence of unseen nodes. With a focused approach to the training set, we enhance its ability to generalize to unseen nodes encountered during testing, aligning newly observed subgraphs with optimized nodes from training. This strategic improvement empowers our model to effectively generalize to new and unexplored instances, leading to substantial advancements in text classification capabilities.

3.1 InducT-GAT Graph Construction

To establish a solid foundation for our InducT-GAT model, we address two key requirements akin to InducT-GCN. The graph is treated as a homogeneous structure during propagation, aligning input vectors for both word and document types. To avoid bias, one-hot vectors are avoided for document representation, and TF-IDF vectors are used for training document node vectors. While one-hot vectors are used to represent word nodes, TF-IDF vectors are harnessed to train document node vectors. The TF-IDF values in the TF-IDF vector represent values associated with the specific words in the respective documents.

Concerning the construction of graph edges, The methodology builds upon the principles of 'InducT GCN' and draws inspiration from the TextGCN method. We used a similar definition of edges within the InducT-GAT graph as InducT-GCN:

Word-Word Edges with PMI: These edges establish connections between word nodes based on the Pointwise Mutual Information (PMI) of word co-occurrences within a defined window. The PMI measure quantifies the statistical relationship between pairs of words, enabling the exploration of their contextual associations.

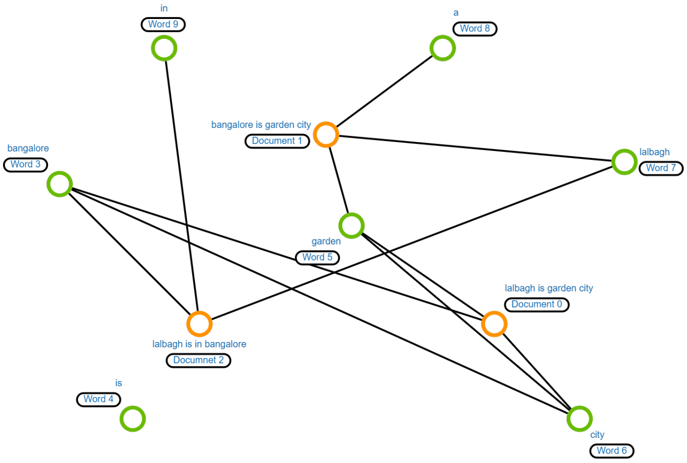

Figure 2: Graph having document and vocab/word as Node

Word-Document Edges with TF-IDF: These edges link word nodes to document nodes using TF-IDF values. TF-IDF, an abbreviation for Term Frequency-Inverse Document Frequency, evaluates the importance of a word within a specific document, providing a measure of its relevance for classification.

Moreover, we incorporate self-connections for each node in the graph, ensuring comprehensive coverage, and guaranteeing that all graph edges are symmetrical and undirected.

By leveraging these techniques, InducT-GAT model presents a robust framework for text classification in an inductive learning setting. The thoughtful construction of graph nodes and edges enhances the model's ability to capture essential features and relationships, facilitating the effective classification of text data. This research contributes to the development of advanced text classification methodologies with potential applications in various domains.

Graph construction for the dummy dataset, where
train-sentences = ['bangalore is garden city','lalbagh is garden city','lalbagh is in bangalore']
test-sentences = ['it hub']
word-list(vocab of dataset) = ['bangalore','is','garden','city','lalbagh','in','a']

Figure 2 shows as Graph having document and vocab/word as Node where 0,1,2 is the document id reference to a number of instances in the training dataset and 3-9 is word-id.Feature matrix size is $10*7$ where $3*7$ is document feature vector, $7*7$ is one-hot embedding of word-id and adjecency matrix size is $10*10$, In above example feature matrix looks like Table 1 and adjecency matrix looks like Table 2.

In Adjacency Matrix we added 1 diagonally to use that node feature also while learning or training basically we added self-loop to that node.

3.2 Model Architecture

In this study, we employed Graph Attention Networks (GAT) under the inductive learning setting, where the node classification model was trained on the original graph con-

Table 1: Input Vectors Representations when three input documents are given(dummy train-sentences)

Document & word-id	vocab1	vocab2	vocab3	vocab4	vocab5	vocab6	vocab7
0	0.1014	0.0000	0.1014	0.2747	0.0000	0.0000	0.0000
1	0.0000	0.0000	0.1014	0.0000	0.1014	0.2747	0.0000
2	0.1014	0.0000	0.0000	0.0000	0.1014	0.0000	0.2747
3	1	0	0	0	0	0	0
4	0	1	0	0	0	0	0
5	0	0	1	0	0	0	0
6	0	0	0	1	0	0	0
7	0	0	0	0	1	0	0
8	0	0	0	0	0	1	0
9	0	0	0	0	0	0	1

Table 2: Input Adjacency Matrix Representations when three input documents are given(dummy train-sentences)

Document & word-id	0	1	2	3	4	5	6	7	8	9
0	1.	0.	0.	0.40546511	0.	0.40546511	1.09861229	0.	0.	0.
1	0.	1.	0.	0.	0.	0.40546511	0.	0.40546511	1.09861229	0.
2	0.	0.	1.	0.40546511	0.	0.	0.	0.40546511	0.	1.09861229
3	0.40546511	0.	0.40546511	1.	0.	0.	0.40546511	0.	0.	0.40546511
4	0	0	0	0	1	0	0	0	0	0
5	0.40546511	0.40546511	0.	0.	0.	1.	0.40546511	0.	0.40546511	0.
6	1.09861229	0.	0.	0.40546511	0.	0.40546511	1.	0.	0.	0.
7	0.	0.40546511	0.40546511	0.	0.	0.	0.	1.	0.40546511	0.40546511
8	0.	1.09861229	0.	0.	0.	0.40546511	0.	0.40546511	1.	0.
9	0.	0.	1.09861229	0.40546511	0.	0.	0.	0.40546511	0.	1.

taining labeled document nodes. During the training phase, training documents/sentences underwent two layers of InducT GAT, resulting in node embeddings, either for short texts or longer texts. These text embeddings, representing documents or sentences, were then fed into a softmax layer for classification.

To evaluate the model's performance on unseen nodes, we adopted a methodology similar to InducT GCN. Specifically, new documents, referred to as Test Nodes, were required to be integrated into the training graph. This integration was achieved by utilizing the Term Frequency-Inverse Document Frequency (TF-IDF) technique, which employed the document frequency of the training set.

Figure 3 provides a visual representation of the proposed model architecture, illustrating the flow of information and the process of incorporating new and unseen nodes into the existing graph. During the testing phase, the prediction process involves combining or aggregating the representations of the first-order and second-order neighbors for each test document. It is crucial to emphasize that the test documents are solely used for testing purposes and are not employed to update all the nodes in the graph during propagation. Instead, a one-directional propagation approach is adopted, where only the test document nodes are updated.

By employing this approach, the testing process is streamlined without the requirement to update the entire graph. This selective updating of the test document nodes allows for efficient and focused inference, as the model leverages the information captured from the neighboring nodes to make predictions without altering the existing graph structure.

In essence, the one-directional propagation approach ensures that the graph remains unchanged during the testing phase, except for the updates specifically made to the test document nodes. This strategy allows for a more targeted and resource-efficient prediction process, facilitating faster inference and reducing computational overhead.

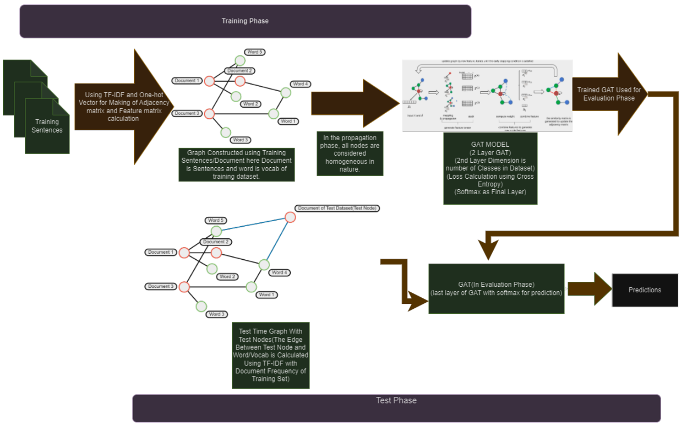
Training Phase
Training Sentences
Using TF-IDF and One-hot Vector for Making of Adjacency matrix and Feature matrix calculation
Word 5
Document 2
Document 1
Word 4
Word 2
Document 3
Word 1
Word 3
Graph Constructed using Training Sentences/Document here Document is Sentences and word is vocab of training dataset.
In the propagation phase, all nodes are considered homogeneous in nature.
GAT MODEL (2 Layer GAT) (2nd Layer Dimension is number of Classes in Dataset) (Loss Calculation using Cross Entropy) (Softmax as Final Layer)
Trained GAT Used for Evaluation Phase
Document of Test Dataset(Test Node)
Test Time Graph With Test Nodes(The Edge Between Test Node and Word/Vocab is Calculated Using TF-IDF with Document Frequency of Training Set)
GAT(In Evaluation Phase) (last layer of GAT with softmax for prediction)
Predictions
Test Phase

Figure 3: InducT GAT Architecture[3]

Overall, our approach leverages the power of GAT and inductive learning to classify nodes, particularly focusing on document nodes, in a graph structure. By extending the model's capabilities to handle previously unseen nodes, we demonstrate the effectiveness of our method in addressing the challenges posed by inductive learning scenarios.

4 Experimental Setup

Within our experimentation, we delve into a diverse array of domains, encompassing the analysis of sentiments across a multitude of subjects, such as evaluations of movies, posts on social media platforms, and Twitter data. Furthermore, we focus on the classification of various question types, news articles, and proprietary datasets associated with the financial sector, with the aim of achieving comparable results on other datasets used for classifying concise text.

4.1 Datasets

We utilized a range of benchmark datasets in our experimentation, allowing us to evaluate text classification methods across various domains along with proprietary datasets.

The text classification datasets used in this study include R8[4], MR[5], TagMyNews [12], Twitter[6], TREC[7], SST-2[8], NICE[9], and STOPS[10]. R8 is a subset of the Reuters news dataset. MR is a movie-review dataset with an average text length of 20.39 tokens, while TagMyNews is a news title dataset with seven classes. Twitter has tweets categorized as negative or positive based on sentiment. TREC is a question-type classification dataset with the shortest texts.

[4] https://www.daviddlewis.com/resources/testcollections/reuters21578/

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

[6] https://www.nltk.org/howto/twitter.html#Using-a-Tweet-Corpus

[7] https://cogcomp.seas.upenn.edu/Data/QA/QC/

[8] https://nlp.stanford.edu/sentiment/

[9] https://www.wipo.int/classifications/nice/en/

[10] https://www.wipo.int/nice/its4nice/ITSupport_and_download_area/20220101/MasterFiles/index.html

SST-2 is a subset of the Stanford Sentiment Treebank, containing reviews labeled as positive or negative. The datasets also include NICE, a classification system for goods and services, and STOPS, a dataset derived from Amazon descriptions and Yelp business entries. These datasets provide a realistic representation of products and services. Furthermore, we incorporated a proprietary dataset for evaluation, specifically focusing on intent classification within the finance domain. This dataset includes a total of 193 distinct classes.Each datasets overall description is in Table 5,Table 4 and Table 3.

By utilizing this diverse range of datasets, we aim to comprehensively evaluate and benchmark text classification methods across various domains and text lengths, enabling a robust assessment of their performance and applicability.

Dataset	#Doc	#Train	#Test	Avg. length	Classes
R8	7,674	5,485	2,189	65.72	8
MR	10,662	7,108	3,554	20.39	2
TagMyNews	32,549	29,294	3254	5.1	7
Twitter	10,000	7,000	3,000	11.64	2
TREC	5,952	5,452	500	10.06	6
SST-2	9,613	7,792	1,821	20.32	2

Table 3: Characteristics of benchmark datasets.

Dataset	#Doc	#Train	#Test	Avg. length	Classes
NICE-45	9,593	6,715	2,878	3.75	45
NICE-2	9,593	6,715	2,878	3.75	2
STOPS-41(1%)	20,341	14,238	6,103	5.64	41
STOPS-2(1%)	20,341	14,238	6,103	5.64	2

Table 4: Characteristics of goods and services datasets.

We preprocess all the datasets as follows. We remove non-English characters, Didn't remove stopwords in Proprietary datasets but for the remaining datasets, we removed them and applied lowercase to all the datasets.

4.2 Models for Text Classification Comparison

In the realm of short text classification, various models have been developed and experimented with to achieve accurate results. One such model is the SECNN [10], which leverages a CNN-based approach.

In addition to SECNN, Researchers also explored the use of graph neural network (GNN)-based models for short text classification or for text classification. Some notable examples include SGNN [17], ESGNN [31], and C-BERT [27].

To address the over-smoothing problem commonly encountered in GNNs and enable deeper network stacking, the DADGNN [15] model has been developed. DADGNN leverages attention diffusion and decoupling techniques, which help alleviate over-smoothing and allow for the stacking of deeper networks.

Table 5: Characteristics of Proprietary Datasets.(%)

Dataset	Doc	Train	Test	Avg.Length	Classes
Proprietary Datasets	27400	19,180	8220	7.8	193

Another model that has gained popularity in text classification tasks is the Bidirectional Long Short-Term Memory (Bi-LSTM). Bi-LSTM is a variant of LSTM and is widely used in various natural language processing tasks.

Moving beyond short texts, there are leading models for text classification that excel across texts of all lengths, employing either Transformer architecture or graph neural network architecture. Some notable examples in this domain include BERT, RoBERTa [14], DeBERTa [5], ERNIE 2.0 [22], DistilBERT [20], ALBERTv2 [8], WideMLP [3], and InducT-GCN. Each of these models introduces unique advancements and techniques to enhance performance and efficiency in text classification tasks, catering to texts of varying lengths.

Moreover, it is crucial to consider transductive models when discussing text classification. Transductive models focus on leveraging both labeled and unlabeled data during the training process. Notable transductive models in this context include SHINE, BertGCN [11], RoBERTa GCN [18], and ST-GCN.

4.3 Parameters and Hardware setting

In order to conduct our experiment within the constraints of a limited environment, we made use of different hyperparameters for different datasets while training the InducT GAT model as shown in Table 8 and Table 9. Our approach involved implementing two layers of the Graph Attention Network and varying the hidden dimension. We employed the Adam optimizer with a specific learning rate same as Induct GCN and dropout same as Induct GCN to enhance generalization. We randomly selected 10% of the training set to create a validation set as it helped to assess performance on new data, while carefully chosen hyperparameters and proportions ensured meaningful results within our limitations as shown in Table 8 and Table 9.

For our research experiments, we utilized an Amazon SageMaker Notebook instance, carefully chosen to meet our computational needs. The ec2 instance provided ample resources, including multiple vCPUs, a substantial amount of RAM, and sufficient attached storage for efficient experimentation.

4.4 Metrics

Accuracy is the metric used in the experiments to measure short text or text classification. The formula is as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP is the number of positive classes classified correctly, TN is the number of negative classes classified correctly, FP is the number of positive classes wrongly classified, and FN is the number of negative classes wrongly classified. For multi-class cases, the subset accuracy is calculated.

5 Results and Analysis

The comprehensive experiment was undertaken in a carefully controlled and limited environment, exploring the depths of knowledge encompassed within the realm of eight benchmark datasets and proprietary datasets. These datasets, meticulously chosen to represent a diverse range of domains and complexities, served as the fertile ground for evaluating and comparing the performance of various state-of-the-art models as documented in different scholarly research papers. Through this endeavor, a treasure trove of insights and revelations was unearthed, shedding light on the intricate nuances and intricacies of the data and uncovering each comparative model's unique strengths and weaknesses. Table 6 and 7 basically give a result about all models that have been basically experimented on nine different datasets [7]. Table 8 and 9 gives the comparison of InducT GAT with different best models on that particular dataset.

In comparison to SOTA on different datasets, InductT GAT outperforms SOTA on three datasets, NICE-45, TagMyNews, and Proprietary Dataset.

In comparison to the graph-based model InducT GAT, outperforms on NICE-45, NICE-2, TagMyNews, STOPS-41, STOPS-2, and Proprietary Dataset.

In Table 9 the best graph model results shown comprised of the inductive and transductive models.

For STOPS-41 and STOPS-2 datasets, the accuracy score mentioned in Table 6 and 7 is on Full datasets, while for our experiments we used only 1% of it so we took only best graph performing model on this dataset as it was InducT GCN we experimented only using this to compare our score, while we didn't evaluate Transformer models on 1 percent dataset of STOPS-41 and STOPS-2. While on this dataset we surpass the score of the best graph-performing model(InducT GCN), so we said in the Graph model category we surpassed the score.

Now Comparing our results with InducT GCN of which our paper is an extension, on datasets like NICE-45, NICE-2, STOPS-41, and STOPS-2 which are specially designed datasets for short text we surpass InducT GCN by a significant margin, the possible reason is using an attention mechanism, by which GAT knows which node to give more attention and which do not.

While coming to datasets like R8, MR, TagMyNews, Twitter, TREC, SST2, and Proprietary datasets, InducT GAT performed significantly well on all the datasets beating InducT GCN accuracy score in all datasets, this result shows how significant is attention in graphs, as not only for short text but for longer text also it performed equally well.

Table 6: Performance of Inductive and Transformer Models

Model	NICE-45	NICE-2	STOPS-41	STOPS-2
BERT	**72.79**	99.72	89.4	99.87
RoBERTa	66.09	**99.76**	89.56	99.86
DeBERTa	59.42	99.72	**89.73**	99.85
ERNIE 2.0	45.55	99.69	89.39	99.85
ERNIE 2.0 (optimized)	67.65	99.72	89.65	**99.88**
DistilBERT	69.28	99.75	89.32	99.85
ALBERTv2	59.24	99.51	88.58	99.83
WideMLP	58.99	96.76	88.2	97.05
DADGNN	28.51	91.15	26.75	97.48
InducT-GCN	47.31	94.97	86.11	97.71
LSTM (BERT)	47.81	96.63	86.27	96.05
Bi-LSTM (BERT)	52.39	96.63	85.93	98.54
LSTM (GloVe)	52.64	96.17	87.4	99.46
Bi-LSTM (GloVe)	55.35	95.93	87.38	99.43

Table 7: Performance of Inductive and Transductive Models

Inductive Models

Model	R8	MR	TagMyNews	Twitter	TREC	SST-2
BERT	98.171	86.94	-	99.96	99.4	91.37
Roberta	98.171	89.42	-	**99.9**	98.6	94.01
DeBERTa	98.451	**90.21**	-	99.93	98.8	94.78
ERNIE 2.0	98.041	88.97	-	99.97	98.8	93.36
ERNIE 2.0 (optimized)	98.171	89.53	-	99.97	**99**	**94.07**
DistilBERT	97.981	85.31	-	99.96	99	90.49
ALBERTv2	97.62	86.02	-	99.97	98.6	91.54
WideMLP	96.98	76.48	-	99.86	97	82.26
fastText	96.13	75.14	-	—	—	—
DADGNN	98.15	78.64	—	—	97.99	84.32
HyperGAT	97.97	78.32	—	—	—	—
HGAT	62.75	61.72	63.21	—	—	—
InducT-GCN	96.67	75.25	**66.68**	88.53	92.42	79.98
LSTM (BERT)	94.28	75.10	-	99.83	97	81.38
Bi-LSTM (BERT)	95.52	75.30	-	99.76	97.2	80.83
LSTM (GloVe)	96.34	74.99	25.52	95.23	97.4	79.95
Bi-LSTM (GloVe)	96.84	75.32	-	95.53	97.2	80.17

Transductive Models

Model	R8	MR	TagMyNews	Twitter	TREC	SST-2
SHINE	86.48	63.21	62.50	71.49	79.90	62.56
STGCN	97.2	78.2	34.74	—	—	—
STGCN+BiLSTM	—	78.5	—	—	—	—
STGCN(bert,biLstm)	**98.5**	82.5	—	—	—	—
TextGCN	97.07	76.74	54.28	—	—	—
BertGCN	98.1	86.0	—	—	—	—
RoBERTaGCN	98.2	89.7	—	—	—	—
TextGCN-BERT-serial-SB	97.78	86.69	—	—	—	—
TextGCN-CNN-serial-SB	**98.53**	87.59	—	—	—	—

Table 8: Performance of InducT GAT and Best Models for particular datasets. (Test dataset)(%)

Dataset	InducT GAT Score	Best Model Score	Best Model	Parameter used by InducT GAT
NICE-45	**77.19**	72.79	**InducT GAT**	Epochs: 80, Attention Heads: 10, Hidden dim: 256
NICE-2	95.15	99.76	Roberta	Epochs: 30, Attention heads: 10, Hidden dim: 256
TagMyNews	**68.04**	62.7	**InducT GAT**	Attention heads: 2, Hidden dim: 256, Epoch: 100
Proprietary Dataset	**83.26**	81.06	**InducT GAT**	Attention Heads: 12, Hidden dim: 256, Epochs: 500
MR-DATASET	85.12	90.21	BERT (80)	Attention Heads: 2, Hidden dim: 256, Epochs: 100
STOPS-41 DATASET(1%)	82.54	-	-	Attention Heads: 10, Hidden dim: 256, Epochs: 80
STOPS-2 DATASET(1%)	84.83	-	-	Attention Heads: 10, Hidden dim: 256, Epochs: 80
TREC	95.84	99.4	ALBERT	Attention Head: 7, Hidden dim: 256, Epochs: 100
R8	97.57	98.53	TextGCN-CNN	Attention Head: 2, Hidden dim: 256, Epochs: 300
Twitter	97.14	99.97	ALBERT	Attention Head: 7, Hidden dim: 256, Epochs: 100
SST2	81.37	94.78	DeBERTa	Attention Head: 8, Hidden dim: 256, Epochs: 200

Table 9: Performance of InducT GAT and Best Graph Models for particular datasets.(Test dataset)(%)

Dataset	InducT GAT Score	Best Graph Model Score	Best Model	Parameter used by InducT GAT
NICE-45	**77.19**	47.31	**InducT GAT**	Epochs: 80, Attention Heads: 10, Hidden dim: 256
NICE-2	**95.15**	94.97	**InducT GAT**	Epochs: 30, Attention heads: 10, Hidden dim: 256
TagMyNews	**68.04**	62.7	**InducT GAT**	Attention heads: 2, Hidden dim: 256, Epoch: 100
Proprietary Dataset	**83.26**	81.06	**InducT GAT**	Attention Heads: 12, Hidden dim: 256, Epochs: 500
MR-DATASET	85.12	89.43	ConTextING-RoBERTa	Attention Heads: 2, Hidden dim: 256, Epochs: 100
STOPS-41 DATASET(1%)	**82.54**	81.21	**InducT GAT**	Attention Heads: 10, Hidden dim: 256, Epochs: 80
STOPS-2 DATASET(1%)	**84.83**	82.87	**InducT GAT**	Attention Heads: 10, Hidden dim: 256, Epochs: 80
TREC	95.84	97.99	DADGNN	Attention Head: 7, Hidden dim: 256, Epochs: 100
R8	97.57	98.53	TextGCN-CNN	Attention Head: 2, Hidden dim: 256, Epochs: 300
Twitter	97.14	98.16	DADGAN	Attention Head: 7, Hidden dim: 256, Epochs: 100
SST2	81.37	84.32	DADGAN	Attention Head: 8, Hidden dim: 256, Epochs: 200

5.1 Limitations

While we explored different datasets for our experiments and performed our comparison, we faced difficulties due to the size of the dataset. For instance, In STOPS-41 and STOPS-2 Datasets, we used only 1 percent of their total size, as it has a total instance size of more than 200K while building the training graph for that many instances our AWS notebook instance was getting crashed(even instance with 32 GB of RAM, and 50 GB of attached storage were not sufficient)

The second issue that we faced, was as different datasets have different sizes, including all datasets having different average lengths of sentences to preprocess and build graphs, also with limited capabilities in terms of hardware, because of this we needed to adjust the Hyper-parameter used by InducT GAT, as shown in Table 8 and 9 because if we were increasing the Hyper-parameter like number of attention heads or Hidden dimensions AWS notebook instance was getting crashed.

These are the few limitations that our work has because of that we needed to experiment with taking lesser dataset samples like the case of STOPS and utilizing different parameters in order to evaluate our model.

While coming to the overall comparison of our model with all different NLP models, we are still behind in performance if the longer text is taken into account as one other limitation of our work.

5.2 Generalisation

In our experiments, we investigate a diverse range of domains, encompassing sentiment analysis across different themes like movie evaluations, social media posts, and Twitter content. We also delve into the classification of various question types (e.g., TREC dataset), news articles (e.g., R8 and TagMyNews datasets), and Proprietary datasets centered around the financial sector. Our objective is to attain comparable outcomes on alternative datasets utilized for classifying concise text.

Furthermore, we employ novel datasets named NICE and STOPS, specifically designed for the categorization of products and services. These datasets incorporate supplementary attributes not present in conventional benchmark datasets.

Throughout our research, we extensively explore a multitude of models within each framework, with a particular emphasis on the most prevalent and high-performing ones.

6 Conclusion and Future Work

This research introduces a pioneering framework, InducT-GAT, which revolutionizes inductive graph-based text classification, with a particular emphasis on short text classification. Building upon the foundation of InducT GCN, our proposed framework extends the capabilities of existing transductive GCN-based models by enabling them to operate in an inductive manner.

By constructing a graph solely based on training set statistics and harnessing the power of InducT GAT, our approach effectively captures global information using fewer parameters and achieves a smaller space complexity. Notably, the InducT-GAT model surpasses a few graph-based text classification baselines or performs comparable to them and even outperforms models reliant on pre-trained embeddings. Moreover, we showcase the generalization capability of inductive graph construction and learning framework by applying it to different domains.

Further extension of the work can be done using pre-trained embedding instead of using a one-hot vector for word/vocab embedding in feature matrix creation.

This paper serves as a beacon, illuminating the potential future integration of lightweight and efficient inductive graph neural networks across various natural language processing (NLP) tasks. The findings presented here not only advance the field of text classification but also pave the way for exploring faster and more agile approaches within the broader realm of NLP. Researchers can forge new paths toward enhanced efficiency and effectiveness in NLP applications by leveraging the power of inductive graph neural networks.

7 Acknowledgements

We express our sincere gratitude to Emerging Analytics Team from "Fidelity Investments, India" for providing the internship opportunity to me and for their valuable contribution to the insightful discussion surrounding the problem statement that served as the impetus for this research endeavor. Their input and engagement significantly enriched our understanding and propelled us toward the advanced formulation of this study. Also, we want to express our gratitude towards the author of InducT GCN [26] and Adversarially Regularized Graph Attention Networks for Inductive Learning on Partially Labeled Graphs [28] of which we adapted for implementing the proposed solution. We thank the owners and contributors of the same for their magnificent work. Also, we express gratitude towards the author of "Transformers are Short Text Classifiers: A Study of Inductive Short Text Classifiers on Benchmarks and Real-world Datasets" [7] for their amazing comparison-based study of models for classification task.

This paper was prepared for informational purposes during the internship and is not worked with any fidelity business. Fidelity makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy, or reliability of the information contained herein.

References

[1] Chaofan Chen, Zelei Cheng, Zuotian Li, and Manyi Wang. Hypergraph attention networks. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1560–1565, 2020. doi: 10.1109/TrustCom50675.2020.00215.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[3] Lukas Galke and Ansgar Scherp. Bag-of-words vs. graph vs. sequence in text classification: Questioning the necessity of text-graphs and the surprising strength of a wide mlp. *arXiv preprint arXiv:2109.03777*, 2021.

[4] Lukas Galke and Ansgar Scherp. Bag-of-words vs. graph vs. sequence in text classification: Questioning the necessity of text-graphs and the surprising strength of a wide mlp, 2022.

[5] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

[6] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hervé Jégou, and Tomás Mikolov. Fasttext.zip: Compressing text classification models. *CoRR*, abs/1612.03651, 2016. URL http://arxiv.org/abs/1612.03651.

[7] Fabian Karl and Ansgar Scherp. Transformers are short text classifiers: A study of inductive short text classifiers on benchmarks and real-world datasets. *arXiv preprint arXiv:2211.16878*, 2022.

[8] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[9] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12):6999–7019, 2022. doi: 10.1109/TNNLS.2021.3084827.

[10] Zheng Li, Yonghao Wu, Bin Peng, Xiang Chen, Zeyu Sun, Yong Liu, and Deli Yu. Secnn: A semantic cnn parser for code comment generation. *Journal of Systems and Software*, 181: 111036, 2021.

[11] Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu. Bertgcn: Transductive text classification by combining gcn and bert. *arXiv preprint arXiv:2105.05727*, 2021.

[12] Hu Linmei, Tianchi Yang, Chuan Shi, Houye Ji, and Xiaoli Li. Heterogeneous graph attention networks for semi-supervised short text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4821–4830, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1488. URL https://aclanthology.org/D19-1488.

[13] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

[14] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[15] Yonghao Liu, Renchu Guan, Fausto Giunchiglia, Yanchun Liang, and Xiaoyue Feng. 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, 2021.

[16] Yuan Luo. Shine: Subhypergraph inductive neural network. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 18779–18792. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/7721f1fea280e9ffae528dc78c732576-Paper-Conference.pdf.

[17] Guangxu Mei, Ziyu Guo, Shijun Liu, and Li Pan. Sgnn: A graph neural network based federated learning approach by hiding structure. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2560–2568. IEEE, 2019.

[18] Hiromu Nakajima and Minoru Sasaki. Text classification using a graph based on relationships between documents. In *Proceedings of the 36th Pacific Asia Conference on Language, Information and Computation*, pages 119–125, 2022.

[19] Ryan A Rossi, Rong Zhou, and Nesreen K Ahmed. Deep inductive network representation learning. In *Companion proceedings of the the web conference 2018*, pages 953–960, 2018.

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

[21] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1): 61–80, 2008.

[22] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8968–8975, 2020.

[23] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*, 2012.

[24] Kshitij Tayal, Rao Nikhil, Saurabh Agarwal, and Karthik Subbian. Short text classification using graph convolutional network. In *NIPS workshop on Graph Representation Learning*, 2019.

[25] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.

[26] Kunze Wang, Soyeon Caren Han, and Josiah Poon. 1. induct-gcn: Inductive graph convolutional networks for text classification. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 1243–1249, Aug 2022. doi: 10.1109/ICPR56361.2022.9956075.

[27] Thomas Weissmann, Yixing Huang, Stefan Fischer, Johannes Roesch, Sina Mansoorian, Horacio Ayala Gaona, Antoniu-Oreste Gostian, Markus Hecht, Sebastian Lettmaier, Lisa Deloch, et al. Deep learning for automatic head and neck lymph node level delineation provides expert-level accuracy. *Frontiers in Oncology*, 13:1115258, 2023.

[28] Jiaren Xiao, Quanyu Dai, Xiaochen Xie, James Lam, and Ka-Wai Kwok. Adversarially regularized graph attention networks for inductive learning on partially labeled graphs. *CoRR*, abs/2106.03393, 2021. URL https://arxiv.org/abs/2106.03393.

[29] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative study of cnn and rnn for natural language processing, 2017.

[30] Xiaosong Yuan, Ke Chen, Wanli Zuo, and Yijia Zhang. Tcgat: Graph attention network for temporal causality discovery, 2023.

[31] Ke Zhao, Lan Huang, Rui Song, Qiang Shen, and Hao Xu. A sequential graph neural network for short text classification. *Algorithms*, 14(12):352, 2021.