Integrating Graph-Based Representations with Deep Contextual Models for Text Classification

Sumit Mamtani

New York University

Abstract. Accurate text classification requires both deep contextual understanding and structural representation of language. This study explores a hybrid approach that integrates transformer-based embeddings with graph-based neural architectures to enhance text classification performance. By leveraging pre-trained language models for feature extraction and applying graph convolution techniques for relational modeling, the proposed method captures both semantic meaning and structural dependencies in text. Experimental results demonstrate improved classification accuracy over traditional approaches, highlighting the effectiveness of combining deep contextual learning with graph-based representations in NLP tasks.

Keywords: Text Classification, Graph Neural Networks, BERT, Hybrid Embeddings, Document Modeling

1 Introduction

Text classification is a foundational task in Natural Language Processing (NLP), involving the assignment of labels to text units such as documents, sentences, or phrases. It supports various applications including sentiment analysis, spam detection, and topic categorization. Traditional approaches relied on manual feature engineering and shallow models, but the advent of deep learning has shifted focus toward automatic representation learning.

Recent contextual language models like BERT (Bidirectional Encoder Representations from Transformers) have transformed NLP by leveraging self-attention and large-scale pretraining to learn rich semantic and syntactic patterns. While effective for capturing sequential dependencies, such models lack mechanisms to explicitly encode structural relationships that can influence document-level classification.

Graph Neural Networks (GNNs) are designed to operate on structured data, enabling the modeling of both local and global dependencies through graph-based message passing. When applied to text, GNNs represent words and documents as nodes, with edges denoting syntactic, semantic, or statistical co-occurrences. Graph Convolutional Networks (GCNs), a prominent GNN variant, have been particularly effective in capturing global word-document relations.

David C. Wyld et al. (Eds): CSIT, AMLA, IPDCA, NLPA, AIS, IPPR, SPTM – 2025
pp. 123 -133, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.151409

This work proposes a hybrid model combining semantic embeddings from a fine-tuned BERT with structural representations derived from GCNs built on document-word graphs. Unlike prior methods treating these aspects independently, the proposed framework unifies them to leverage their complementary strengths.

We begin with a literature review on text classification and graph-based approaches. Then, we detail our methodology, including preprocessing, graph construction, embedding generation, and fusion techniques. Experimental results demonstrate the benefits of this integration over individual models. Finally, we discuss observed limitations and suggest future improvements involving attention-based GNNs and extended evaluation on diverse datasets.

2 Related Work

The study of human-AI interaction has gained attention, especially regarding user prompt formulation for Large Language Models (LLMs). Desai [4] explores prompting strategies such as Zero-Shot and Few-Shot learning to enhance model responses on platforms like ShareGPT and Midjourney.

Desai [6] presents a market analysis of India's electric vehicle (EV) landscape, examining policy, technology trends, and adoption challenges. Pricing models using machine learning are addressed in [7], which applies Random Forest regression to predict product prices based on key features.

Active learning in text classification is studied in [3], showing how minimal annotation effort can yield strong performance. Desai [5] also investigates Progressive Web Applications (PWAs) for scalable inventory systems as efficient alternatives to native apps.

In immersive technologies, Ganji [9] discusses the role of Augmented Reality (AR) in real-world enhancement for sectors like healthcare and education. Srivastava and Singh [19] focus on intrinsic motivation in reinforcement learning agents.

Patel [15] studies neural network robustness under rotated semantic segmentation datasets. Patel also explores Knowledge Graph Embeddings (KGEs) for question answering [17], and evaluates blockchain security in PoS systems [14]. Further contributions include improving video streaming QoS [16] and decentralized computing models [13].

These studies highlight interdisciplinary advancements across AI, machine learning, NLP, and blockchain systems.

3 Methods

Section 3 describes the proposed hybrid framework, including dataset preparation, graph construction, contextual and structural embedding generation, fusion strategies, and classification.

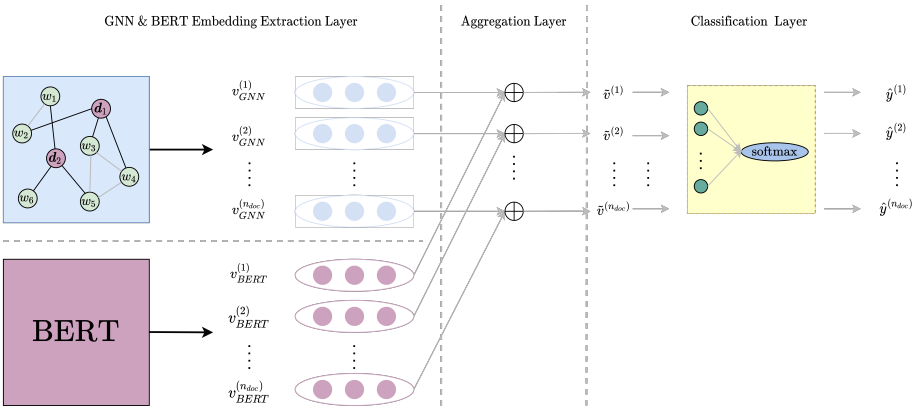

Fig. 1: Model Architecture

3.1 Model Overview

The model integrates semantic and structural insights by combining Distil-BERT and GCN embeddings. The GCN captures global structural dependencies, while Distil-BERT encodes contextual features. These representations are fused and passed to a classification layer. The architecture is shown in Figure 1.

3.2 Dataset Description

We use the *20 Newsgroups* dataset [**huggingface20news**, 1], which includes 18,846 articles across 20 categories. It is split into 11,314 training and 7,532 test documents. The dataset's diversity in content and structure makes it suitable for benchmarking text classification.

3.3 Document-Word Graph Construction

The dataset is modeled as a heterogeneous graph with words and documents as nodes. Edges capture word-word co-occurrence and word-document associations. The graph's structure is represented by an adjacency matrix A, while the feature matrix $X \in \mathbb{R}^{n \times n}$ is initialized with one-hot vectors, where $n = n_{\text{doc}} + |V|$. This ensures unbiased node representations prior to training.

The adjacency matrix is defined as:

$$A_{ij} = \begin{cases} \text{PMI}(i,j) & \text{if } i,j \text{ are words and PMI} > 0 \\ \text{TF-IDF}_{ij} & \text{if } i \text{ is a document and } j \text{ a word} \\ 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

3.4 Adaptive Graph Learning via Learnable Adjacency

To improve flexibility, we introduce a learnable adjacency matrix $\widetilde{A}$ that combines the fixed structure with a trainable component $\widehat{A}$. This approach allows the model to discover latent relationships during training.

$$\widetilde{A} = \alpha\widehat{A} + (1-\alpha)A \tag{2}$$

The parameter $\alpha \in (0,1)$ determines the contribution of the learnable structure, and the gradient updates apply only to $\widehat{A}$, allowing the model to preserve essential fixed relationships while discovering new ones.

3.5 Graph Neural Network Encoding

For structural embedding, we use a two-layer Graph Convolutional Network implemented via the PyTorch Geometric library [**pyg**]. The model propagates information across connected nodes using the learnable adjacency matrix.

The propagation follows this recursive formulation:

$$L^{(j+1)} = \rho\left(\widetilde{A}L^{(j)}W_j\right) \tag{3}$$

For our two-layer GCN, the final output is:

$$Z = \text{softmax}\left(\widetilde{A}\cdot\text{ReLU}(\widetilde{A}XW_0)W_1\right) \tag{4}$$

The loss function used is the cross-entropy over labeled documents:

$$\mathcal{L} = -\sum_{d\in\mathcal{Y}_D}\sum_{f=1}^{F} Y_{df}\log Z_{df} \tag{5}$$

Here, $\mathcal{Y}_D$ denotes the indices of labeled documents, and F is the number of target classes. The intermediate document embeddings are extracted from the following equations:

$$E_1 = \widetilde{A}XW_0 \tag{6}$$

$$E_2 = \widetilde{A}\cdot\text{ReLU}(\widetilde{A}XW_0)W_1 \tag{7}$$

We use E_1 as the document representation, giving each node a 200-dimensional vector, denoted as .

3.6 Contextual Embeddings via Distil-BERT

For semantic representation, we fine-tune Distil-BERT on the 20NG dataset using the Hugging Face Transformers library [**huggingface**]. Each document is tokenized and processed through 13 transformer layers, producing contextual vectors of shape $(13, 768)$.

To create a unified document embedding, we extract the output of the final layer (13th) as a 768-dimensional vector, denoted as . Alternative pooling strategies (e.g., mean or max pooling) may also be used but are not explored here.

3.7 Fusion Strategies for Embedding Aggregation

The two embeddings— and —are combined using one of the following strategies:

Concatenation In this approach, we directly concatenate the two vectors:

$$= [||], \quad \in \mathbb{R}^{968\times 1} \tag{8}$$

Element-wise Summation To perform element-wise addition, we first reduce the BERT embedding to match the GCN's dimensionality using PCA:

$$= \mathrm{PCA}_{200}() \tag{9}$$

$$= +, \quad \in \mathbb{R}^{200\times 1} \tag{10}$$

Trainable Trade-off This method introduces a learnable scalar λ to balance the contributions:

$$= [\lambda||(1-\lambda)], \quad \in \mathbb{R}^{968\times 1} \tag{11}$$

$$= \lambda + (1-\lambda), \quad \in \mathbb{R}^{200\times 1} \tag{12}$$

3.8 Classification Layer

The final representation is passed through a fully connected layer to generate class probabilities. The classifier is defined as:

$$z_{doc} = \mathrm{softmax}(W), \quad W \in \mathbb{R}^{n_{doc}\times d_{doc}} \tag{13}$$

The prediction output z_{doc} corresponds to the probability distribution over class labels for each input document.

4 Results

This section presents a detailed analysis of the experimental outcomes observed across multiple configurations of our model. We evaluate the effectiveness of the GCN and Distil-BERT modules independently and examine the performance when their representations are integrated using our proposed aggregation strategies (refer to aggregation). Additionally, we compare the final outcomes with well-established state-of-the-art (SOTA) models on the 20 Newsgroups dataset to understand the competitiveness of our approach.

4.1 Performance of GCN Models

To begin with, we assess the performance of our graph-based model in isolation. Here, document classification is performed solely based on structural embeddings generated by the Graph Convolutional Network (GCN). We explore both static and learnable graph connectivity matrices, as described in sec:methods. The GCN is evaluated across two architectural setups: a shallow two-layer model and a deeper three-layer variant. The nomenclature used is as follows: $\text{GCN}_{i,j,k}$ refers to a GCN with hidden layer sizes i, j, k, and $\text{L-GCN}_{i,j,k}$ incorporates the learnable adjacency mechanism.

Table 1: Accuracy scores of standalone GCN models using different architectures and adjacency matrix variants.

Model Configuration	**Training Accuracy (%)**	**Test Accuracy (%)**
$\text{GCN}_{200,20}$	100.00	66.50
$\text{L-GCN}_{200,20}$	100.00	67.50
$\text{GCN}_{2000,200,20}$	100.00	60.80
$\text{L-GCN}_{2000,200,20}$	100.00	60.70

From tab:results:gnn, we observe that incorporating a learnable adjacency matrix slightly improves test accuracy in shallower configurations. However, deeper architectures ($2000 \rightarrow 200 \rightarrow 20$) experience performance degradation, suggesting possible overfitting or gradient vanishing.

4.2 Distil-BERT Results

Next, we evaluate the semantic classification power of Distil-BERT, fine-tuned on the same 20 Newsgroups dataset. The model was trained over 30 epochs using the Hugging Face Transformers library. The accuracy plateaued around the 29th epoch, indicating convergence. The training and validation losses along with final test accuracy are provided in table:$\text{bert}_r esults$.

Distil-BERT alone achieves a test accuracy of 70.05%, outperforming all GCN-only configurations. This underscores the strength of transformer-based embeddings for contextual understanding.

4.3 Performance of Hybrid Model (GCN + BERT)

This section evaluates the combined framework, where embeddings from the GCN and Distil-BERT are aggregated before classification. While several fusion techniques were implemented, the concatenation strategy yielded the highest accuracy across all model variants, and thus only these results are reported in tab:results:combined.

Table 2: Distil-BERT training performance over epochs with validation accuracy.

Epoch	Training Loss	Validation Loss	Accuracy (%)
1	1.1764	1.217	65.64
2	.7607	1.174	68.12
3	.5118	1.440	67.90
⋮	⋮	⋮	⋮
28	.0684	3.372	69.98
29	.0817	3.397	**70.05**
30	.0799	3.404	69.99

Table 3: Document classification performance of hybrid models using concatenated GCN and Distil-BERT embeddings.

GCN Architecture	Aggregation Method	Test Accuracy (%)
$GCN_{200,20}$	Concatenation	67.20
L-$GCN_{200,20}$	Concatenation	**69.70**
$GCN_{2000,200,20}$	Concatenation	64.90
L-$GCN_{2000,200,20}$	Concatenation	63.20

These results confirm that integrating structural and semantic information benefits classification. The model L-$GCN_{200,20}$ + Distil-BERT comes close to the standalone performance of Distil-BERT, achieving 69.70% accuracy.

4.4 Comparison with State-of-the-Art Models

To contextualize our findings, we benchmark the performance of our models against popular state-of-the-art methods evaluated on the same dataset. The comparison is illustrated in tab:results:sota. Notably, while our models do not reach the upper echelons of transformer-based ensembles, they still outperform traditional and early deep learning approaches.

Our best-performing hybrid model delivers accuracy competitive with early BERT-based solutions and provides a balance between computational efficiency and interpretability. While transformer-GCN hybrids such as BertGCN or RoBERTaGCN yield superior results, they often require extensive resources and fine-tuning strategies that are outside the scope of this study.

5 Discussion

This section discusses the implications of the experimental results and addresses limitations encountered during model design and implementation.

Table 4: Comparison of our models with existing baselines and SOTA methods on the 20NG dataset.

Model	**Test Accuracy (%)**
PV-DM [**le14paragraph**]	51.10
LSTM [**hochreiter97lstm**]	65.70
L-GCN (Ours)	67.50
L-GCN + Distil-BERT (Ours)	69.70
Distil-BERT Fine-Tuned (Ours)	**70.05**
RoBERTa [**liu2019roberta**]	83.80
BERT [8]	85.30
TextGCN [22]	86.30
RoBERTaGAT [23]	86.50
BertGAT [18]	87.40
SGC [21]	88.50
BertGCN [12]	89.30
RoBERTaGCN [23]	**89.50**

5.1 Reflection on Experimental Outcomes

The experimental findings provide valuable insights into the strengths and limitations of our proposed hybrid framework. While our model did not outperform the highest-ranking state-of-the-art (SOTA) models on the 20 Newsgroups dataset, it showed promising performance when compared to conventional and early deep learning approaches. Specifically, the integration of graph-based and transformer-based representations through aggregation techniques led to consistent improvements over standalone GCNs and approached the performance of fine-tuned Distil-BERT.

The best results were achieved when combining the learnable GCN ($\text{L-GCN}_{200,20}$) with Distil-BERT embeddings using simple concatenation. This combination resulted in a test accuracy of 69.70%, which, although slightly below the standalone Distil-BERT accuracy (70.05%), demonstrated that structural information still contributes meaningfully. These findings validate the hypothesis that semantic and relational features can complement each other when effectively integrated, particularly for complex classification tasks involving inter-document or inter-word dependencies.

5.2 Implementation Challenges

Throughout the project, we encountered several implementation-level challenges that affected both the design choices and the resulting performance.

Dataset Variability and Labeling Inconsistencies One of the foremost difficulties was related to the nature of the 20 Newsgroups (20NG) dataset. While the original benchmark has been widely used in semi-supervised contexts—particularly in works like [11, 22]—our version of the dataset was sourced from the Hugging Face

Datasets Library [**huggingface20news**], where all documents are fully labeled. This discrepancy posed complications in aligning our experimental protocol with prior works.

Most notably, the Graph Convolutional Network (GCN) model in PyTorch Geometric is designed primarily for semi-supervised learning, where only a subset of labeled nodes contributes to the training loss. To adapt this to our fully supervised setup, we configured the entire training split as the labeled subset. While this workaround allowed for training, it may have limited the GCN's performance due to architectural misalignment with the dataset's labeling structure.

Architectural Rigidity in PyTorch Geometric The second challenge stemmed from the constraints imposed by the PyTorch Geometric framework. Its GCN implementation assumes that the graph structure and node indices remain static during training. As a result, adopting an inductive setting—where new, unseen graphs or nodes are expected at test time—was non-trivial. This limitation restricted our ability to experiment with fully inductive learning strategies or dynamic graph updates during training.

Overfitting in Deep GCNs When exploring deeper GCN architectures such as $\text{GCN}_{2000,200,20}$, we observed diminishing returns in test performance, despite perfect training accuracy. This suggests the model was overfitting to structural patterns in the training data. The lack of regularization or attention mechanisms in basic GCN layers may have contributed to poor generalization.

5.3 Potential Improvements and Future Work

Building on the lessons learned from the current implementation, several promising directions can be pursued to enhance both accuracy and robustness of our framework:

Adopting Inductive GCN Architectures One major improvement would be to transition the framework to a fully inductive learning pipeline. Models such as FastGCN [2] provide the flexibility to work with unseen nodes at inference time and significantly reduce computational overhead by introducing importance-based sampling. This adaptation would better align with our fully labeled dataset and allow for broader generalization beyond the training graph.

Incorporating Graph Attention Mechanisms Another enhancement could involve integrating attention-based GNN architectures such as Graph Attention Networks (GATs) [20] or Attention-GCN [10]. These models dynamically assign varying importance to neighboring nodes, thereby capturing more expressive structural features than fixed aggregation methods. Replacing or augmenting basic GCN layers with attention-enhanced components may help address the over-smoothing and uniformity problems commonly observed in deeper GCNs.

Dynamic Graph Construction and Updating Future iterations could explore the use of adaptive or task-specific graph generation strategies. Instead of using static graphs constructed via pointwise mutual information (PMI) and TF-IDF, one could use learnable edge weights or employ reinforcement learning to optimize the graph structure jointly with the classification task.

Multimodal Fusion Strategies While our current model employs simple concatenation or weighted sums to combine embeddings, more advanced fusion techniques—such as bilinear pooling, cross-modal transformers, or co-attention mechanisms—may yield richer representations. These could better capture the interaction between semantic and structural modalities.

Exploring Broader Benchmarks While the 20NG dataset provides a controlled environment for experimentation, validating the model on larger and more diverse corpora (e.g., AG News, Reuters, or multi-label datasets) would help establish its generalizability and practical utility.

6 Conclusion

In summary, this study proposed a hybrid framework that leverages both graph-based and transformer-based models for document classification. Through extensive experimentation, we demonstrated that the fusion of GCN-derived structural embeddings and Distil-BERT contextual embeddings yields competitive performance, even if not surpassing the latest SOTA benchmarks. Our findings support the complementary nature of semantic and relational features in text classification tasks.

While the project encountered several implementation bottlenecks, especially related to dataset structure and GCN training protocols, the insights gained lay a strong foundation for future enhancements. With adjustments toward inductive graph learning, attention mechanisms, and more sophisticated fusion strategies, the hybrid modeling approach holds considerable promise for scalable and accurate text classification across diverse NLP tasks.

References

1. M. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. E. Barnes, and D. E. Brown, "Text Classification Algorithms: A Survey," *Information*, vol. 10, no. 4, p. 150, 2019.
2. Q. Li, W. Song, J. Ma, and X. Guo, "A Survey on Text Classification: From Shallow to Deep Learning," *arXiv preprint arXiv:2008.00364*, 2020.
3. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018.
4. J. Howard and S. Ruder, "Fine-tuned Language Models for Text Classification," *arXiv preprint arXiv:1801.06146*, 2018.

5. S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep Learning Based Text Classification: A Comprehensive Review," *arXiv preprint arXiv:2004.03705*, 2020.
6. J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.
7. 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, 2021.
8. Z. Zhang, P. Cui, and W. Zhu, "Deep Learning on Graphs: A Survey," *arXiv preprint arXiv:1812.04202*, 2018.
9. L. Sun, J. Wang, P. S. Yu, and B. Li, "Adversarial Attack and Defense on Graph Data: A Survey," *arXiv preprint arXiv:1812.10528*, 2018.
10. H. Peng, J. Gao, S. Yao, and X. Sun, "Hierarchical Taxonomy-Aware and Attentional Graph Capsule RCNNs for Large-Scale Multi-Label Text Classification," *arXiv preprint arXiv:1906.04898*, 2019.
11. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph Attention Networks," *arXiv preprint arXiv:1710.10903*, 2018.
12. 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 (ACL)*, 2019, pp. 241–251.
13. T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," *arXiv preprint arXiv:1609.02907*, 2017.
14. L. Yao, C. Mao, and Y. Luo, "Graph Convolutional Networks for Text Classification," *arXiv preprint arXiv:1809.05679*, 2018.
15. P. Veličković, "Theoretical Foundations of Graph Neural Networks," DeepMind, 2021. [Online]. Available: `https://petar-v.com/talks/GNN-Wednesday.pdf`
16. B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online Learning of Social Representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2014, pp. 701–710.