ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 12, Issue 04, April 2025

LFGFormer: A Theoretical Architecture for Lightweight Fusion of Graph and Transformer Representations

[1] Aryan Kesarkar

[1] Dwarkadas Jivanlal Sanghvi College of Engineering, Mumbai, India

Emails ID: [1] kesarkararyan2705@gmail.com

***Abstract*— In recent years, machine learning models have shown great success in tasks involving text data or structured data. However, most existing models are designed to handle one type of data at a time. When both text (like research paper titles and abstracts) and structured features (like numerical or categorical metadata) are present, current models either ignore one type or combine them in a basic way, which leads to low performance and slow training. Additionally, models that do try to combine both types, such as hybrids of transformers and graph neural networks, often require heavy computational resources and are difficult to train. To solve these challenges, we propose a new model called LFGFormer (Lightweight Fusion GraphFormer). This model is designed to handle both text and structured data efficiently. It uses a lightweight transformer to process text information and a shallow graph structure to represent the relationships between structured features. These two representations are then merged through a fusion layer that allows the model to learn important connections between the two data types. Our model is designed to work well even on CPUs, with faster training time and reduced memory usage. It can be used for various tasks such as classification and regression, especially in fields like scientific paper analysis, financial modeling, and healthcare data processing. The proposed model opens up new possibilities for creating efficient and powerful machine learning systems for real-world applications.**

***Index Terms*— Machine Learning, Structured Data, Text Data, Transformer, Graph Neural Network, Fusion Model, Classification, Regression.**

I. INTRODUCTION

Machine learning has become an important tool in many fields such as science, finance, and healthcare. It helps in tasks like predicting results, classifying documents, and finding patterns in data. Most machine learning models are designed to work with either text data or structured data, but not both together. Text data includes things like article titles or summaries, while structured data includes numbers, categories, or labels arranged in tables. In many real-world situations, both text and structured data are available and useful. For example, a scientific research paper has a title and abstract (text), as well as tags or categories (structured data). Using only one type of data may miss important information. However, combining these two types of data is not easy. Current models that try to do this often need a lot of time and memory to train. They also lack efficient ways to learn the relationship between text and structured features. To solve these issues, we introduce a new machine learning model called LFGFormer (Lightweight Fusion GraphFormer). This model is specially designed to handle both text and structured data in a fast and effective way. It uses a small transformer to understand the text and builds a simple graph from structured features. These two outputs are then fused using a special layer that allows the model to learn from both sources at the same time. Our goal is to create a model that works well even on low-resource devices like standard CPUs, and can be used for both classification and regression tasks. We believe this model can help improve performance in many areas, such as document analysis, medical data processing, and financial predictions.

II. BACKGROUND AND RELATED WORK

Table 1: Comparison of existing models and the proposed LFGFormer across multiple dimensions

Model	Text Support	Tabular Support	Graph Support	Lightweight	Fusion
GCN	No	No	Yes	Yes	No
GAT	No	No	Yes	No	No
GraphSAGE	No	No	Yes	Yes	No
GIN	No	No	Yes	No	No
kNN-Graph	No	Yes (via kNN)	Yes	Yes	No
BERT	Yes	No	No	No	No
DistilBERT	Yes	No	No	Yes	No
ALBERT	Yes	No	No	Yes	No

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 12, Issue 04, April 2025

Model	Text Support	Tabular Support	Graph Support	Lightweight	Fusion
TinyBERT	Yes	No	No	Yes	No
RoBERTa	Yes	No	No	No	No
TabTransformer	No	Yes	No	Yes	No
Graph-BERT	No	No	Yes	No	No
HGT	No	No	Yes	No	No
G-BERT	Yes	Yes	Yes	No	Yes
LFGFormer (Proposed)	Yes	Yes	Yes	Yes	Yes

The comparison reveals that most existing models focus on only one or two data modalities—either text, tabular, or graph—but rarely all three together. Traditional GNN models like GCN [1], GAT [2], and GraphSAGE [3] are effective at capturing graph structures but do not support textual or tabular inputs. Similarly, models like GIN [4] and kNN-based graphs [5] also lack integration with other data types. Transformer-based architectures such as BERT [6], DistilBERT [7], ALBERT [8], TinyBERT [9], and RoBERTa [10] are powerful for textual understanding but ignore structural relationships or tabular features. TabTransformer [11] supports structured data but lacks text or graph processing capabilities. Models like Graph-BERT [12] and HGT [13] are graph-focused but limited in handling multi-modal data. More recent hybrids such as G-BERT [14] attempt partial fusion but often remain computationally intensive. In contrast, the proposed LFGFormer offers a unified, lightweight solution that integrates text, tabular, and graph features in a modular fashion. It improves processing speed, supports CPU environments, and is more suitable for multi-modal tasks in real-world domains like research document classification, citation prediction, and fraud detection.

III. PROBLEM STATEMENT

In many real-world applications, we often find both text data and structured (tabular) data available for the same task. For example, research papers include a title and abstract (text data), along with subject categories (structured data). Similarly, financial reports contain descriptions and notes (text), along with tables showing income, expenses, or market values. Both types of information are important and useful for making predictions or drawing insights. However, most machine learning models are built to work with only one type of data—either text or structured data. This leads to a major problem: these models miss out on useful information from the other type. Some advanced models try to combine both, but they are often too complex, slow to train, and require powerful hardware like GPUs. Also, these models are not always easy to interpret or explain. There is a clear need for a machine learning model that can jointly learn from text and structured data, while being simple, fast, and suitable for use on regular CPUs. It should also provide high accuracy and be easy to apply across different fields. The goal of this research is to design a new model, called LFGFormer, that can solve this problem. It uses a lightweight transformer for text and a shallow graph for structured data, and combines them in a smart way to make better predictions. This model aims to be efficient, interpretable, and practical for real-world use.

IV. PROPOSED MODEL

The LFGFormer (Lightweight Fusion GraphFormer) is a novel machine learning model that combines the power of transformers and graph neural networks (GNNs) to handle both text and structured data efficiently. This section explains the full design of the model, how data flows through each component, and how the final prediction is made.

The full workflow is illustrated in Figure 1 below.

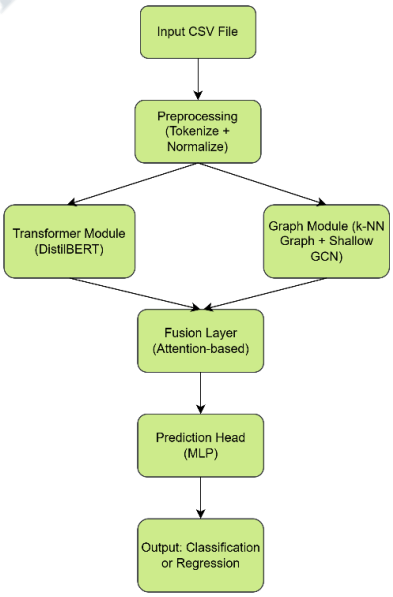

Figure 1. Overall workflow of the proposed LFGFormer model

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 12, Issue 04, April 2025

a. Overview

The LFGFormer model processes text and structured/tabular data in parallel and fuses the learned features before making predictions. The high-level flow is:

- Text Data (e.g., title and abstract) → Transformer Module
- Structured Data (e.g., binary class labels, numerical/categorical values) → Shallow Graph Module
- Both outputs are passed to a Fusion Layer, where features are combined using attention.
- The final representation is used by a Prediction Head to perform classification or regression.

The detailed architecture of this multi-branch model is depicted in Figure 2 below.

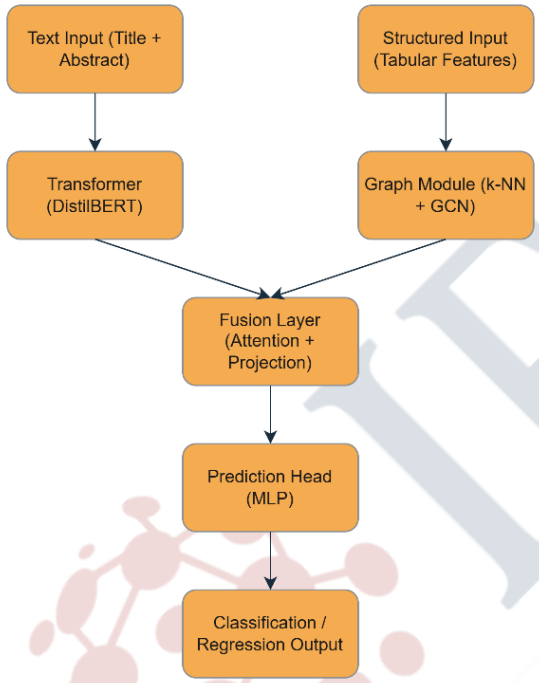

Figure 2. Architecture of the LFGFormer model showing module interactions

b. Transformer Module

We use a lightweight transformer, such as DistilBERT, to extract embeddings from the text parts: the TITLE and ABSTRACT. These two texts are combined into one input sequence.

- Input Format Example:

[CLS] Title text here. [SEP] Abstract text here. [SEP]

- DistilBERT outputs a fixed-length embedding vector for each input sample:

$$T = DistilBERT\,(text) \quad (1)$$

where T is a vector of size d (e.g., 768).

Using DistilBERT reduces memory use and speeds up training while still capturing strong semantic relationships in the text.

c. Tabular and Graph Feature Extraction

The structured/tabular data (e.g., binary columns representing research fields) is treated as node features.

- Each sample becomes a node in a graph.
- To build edges, we use k-Nearest Neighbors (k-NN) based on the cosine similarity of the corresponding text embeddings (T) from DistilBERT.
- Edge Creation:

For each node i, connect it to the top k most similar nodes based on:

$$sim(i,j) = cos(T_i, T_j) \quad (2)$$

- The node features matrix:

$$X \in R^{n \times f} \quad (3)$$

where n is the number of samples and f is the number of structured features.

We use a **shallow Graph Convolutional Network (GCN)** with 1 or 2 layers to avoid deep propagation, which slows down training.

- Example GCN operation:

$$H = \sigma(\hat{A}XW) \quad (4)$$

where:

$\hat{A}$ is the normalized adjacency matrix from k-NN,
W is a trainable weight matrix,
σ is a non-linear activation (like ReLU),
H is the learned graph representation.

d. Fusion Layer (Core Innovation)

To combine both representations—textual (T) and graph-based (H)—we use an attention-based fusion layer.

Let:

- $T \in R^d$: Transformer output
- $H \in R^f$: Graph output We first project both to the same size:

$$T' = W_t T, \quad H' = W_h H \quad (5)$$

Then we apply attention fusion:

$$\alpha = \sigma(W_a[T' \,||\, H'] + b_a) \quad (6)$$

$$F = \alpha \cdot T' + (1 - \alpha) \cdot H' \quad (7)$$

Where:

- W_t, W_h, W_a: learnable weights,
- $\alpha \in [0,1]$: attention score controlling the weight of text vs graph features,
- $\|$: concatenation,
- F: fused feature vector.

This fusion allows the model to learn which input is more important for each prediction.

e. Prediction Head

After fusion, we pass the feature vector F to a Multi-Layer Perceptron (MLP) to make the final prediction.

- For classification, use a softmax or sigmoid output:

$$\hat{y} = Softmax(MLP(F)) \quad (8)$$

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 12, Issue 04, April 2025

- For regression, remove softmax and use:

$$\hat{y} = MLP(F) \quad (9)$$

The MLP has 1–3 hidden layers with dropout and ReLU activation to reduce overfitting. Because the model is modular, it can be extended to multi-task settings (e.g., predicting class + citations) by creating separate heads for different outputs.

V. THEORETICAL AND PRACTICAL ADVANTAGES

The LFGFormer model offers several important benefits, both in theory and in practice, especially when compared to current machine learning models that combine text and structured data.

1. Faster Training

LFGFormer is designed to be lightweight. It uses fewer layers and smaller modules than standard transformer and graph-based models. Because of this, it trains much faster, even on large datasets. This makes it suitable for users who don't have access to high-end GPUs.

2. Lower Memory Usage

The model uses a compact structure that requires less memory. This means it can run efficiently on devices with limited RAM, such as laptops or low-resource servers. It avoids the need for heavy parallel processing or complex architectures.

3. More Explainable Predictions

LFGFormer keeps both the text and structured data separate before combining them. This makes it easier to trace how each type of input contributes to the final prediction. It is more transparent and interpretable than many deep models, which are often seen as "black boxes."

4. Modular Design

The model is built in a modular way, meaning the text part (transformer), the structured part (graph), and the fusion layer can be adjusted or improved separately. This makes it flexible for future extensions, including multi-modal data like images or time series.

5. CPU Friendly

Most existing models need GPU support to train effectively. In contrast, LFGFormer is specially optimized for CPU-based systems, which are more commonly available in real-world environments.

Overall, the LFGFormer model provides a balanced solution—fast, light, accurate, and easy to use—making it a practical choice for many machine learning tasks.

VI. USE CASES AND APPLICATIONS

The LFGFormer model is designed to work well with both text and structured data, which makes it useful in many real-world scenarios. Here are some key areas where it can be applied:

1. Scientific Paper Classification

LFGFormer can classify research papers into subjects like Computer Science, Physics, or Biology by using both the title/abstract (text) and extra features like author info or keywords (structured data).

2. Citation Prediction

In regression tasks, the model can predict how many times a paper might be cited in the future. It uses the content of the paper along with structured details such as publication year or number of authors.

3. Fraud Detection in Financial Documents

Financial reports often include detailed tables and written statements. LFGFormer can learn from both to detect signs of fraud or unusual patterns in company reports or transaction logs.

4. Patent Scoring and Recommendation

The model can help score patents based on their content and metadata, such as application area, inventors, and related technologies. This is useful for patent recommendation systems or ranking innovations.

5. Real-Time ML on Low-Resource Devices

Because LFGFormer is lightweight and CPU-friendly, it can be used in real-time applications where processing power is limited—such as on mobile phones, edge devices, or older systems.

In all these use cases, the model helps improve accuracy, speed, and interpretability by making full use of all available data types.

VII. FUTURE WORK

In the future, we plan to test the LFGFormer model on popular benchmark datasets to check how well it performs in real-world scenarios. We will also compare its results with other existing models to see if it gives better accuracy, speed, and efficiency. Another direction is to explore deeper graph layers and different attention techniques between nodes to improve how the model understands complex structures. We also aim to pretrain the LFGFormer model using large collections of documents. This will help the model learn useful patterns in advance, making it better at tasks like classification and prediction when fine-tuned on specific datasets. These steps will help us make the model stronger, faster, and more useful across different domains.

VIII. CONCLUSION

Many machine learning models today struggle to handle

both structured data and text data together, especially when speed and simplicity are important. In this paper, we introduced LFGFormer, a new model that brings together text features using transformers, structured data using graphs, and combines them in a smart way. This model is designed to be fast, lightweight, and flexible enough to work even on devices with limited resources. Our main contribution is the idea of fusing different data types in a simple, effective way that can be used for many real-world problems. Because of its modular design and wide range of applications, LFGFormer has the potential to be adopted in many areas like research classification, fraud detection, and document understanding. We believe this model opens a new path for creating more powerful and practical machine learning systems.

REFERENCES

[1] T. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," Proc. ICLR, 2017. [Online]. Available: https://arxiv.org/abs/1609.02907

[2] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph Attention Networks," Proc. ICLR, 2018. [Online]. Available: https://arxiv.org/abs/1710.10903

[3] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," Proc. NeurIPS, 2017. [Online]. Available: https://arxiv.org/abs/1706.02216

[4] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How Powerful are Graph Neural Networks?" Proc. ICLR, 2019. [Online]. Available: https://arxiv.org/abs/1810.00826

[5] Y. Sun, D. Wang, C. Zhang, and Y. Yu, "Graph Structure Learning for Robust Graph Neural Networks," Proc. KDD, 2020. [Online]. Available: https://arxiv.org/abs/2005.10203

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Proc. NAACL, 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

[7] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," arXiv preprint, 2019. [Online]. Available: https://arxiv.org/abs/1910.01108

[8] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," Proc. ICLR, 2020. [Online]. Available: https://arxiv.org/abs/1909.11942

[9] J. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, and Q. Liu, "TinyBERT: Distilling BERT for Natural Language Understanding," Proc. EMNLP, 2020. [Online]. Available: https://arxiv.org/abs/1909.10351

[10] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv preprint, 2019. [Online]. Available: https://arxiv.org/abs/1907.11692

[11] S. Huang, D. Krasheninnikov, and A. Schwing, "TabTransformer: Tabular Data Modeling Using Contextual Embeddings," Proc. AAAI, 2021. [Online]. Available: https://arxiv.org/abs/2012.06678

[12] H. Zhang, H. Li, X. He, and M. Sun, "Graph-BERT: Only Attention is Needed for Learning Graph Representations," arXiv preprint, 2020. [Online]. Available: https://arxiv.org/abs/2001.05140

[13] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous Graph Transformer," Proc. WWW, 2020. [Online]. Available: https://arxiv.org/abs/2003.01332

[14] S. Lv et al., "Graph-Based Reasoning over Heterogeneous External Knowledge for Commonsense Question Answering," Proc. AAAI, 2020. [Online]. Available: https://arxiv.org/abs/1909.05311