Machine Learning Project

- Team Name : True Positives
- Team Members: 1.Bhavya Jain(12240420)
  2.Kriti Arora(12240880)
  3.Mallarapu Hema Varshini(12240950)
  4.Kaki Venkata Vaneesha(12240740)
- Title: Classifying Research Documents by Combining Text Embeddings and Citation Networks for Enhanced Accuracy

Novelty

The novelty of our project lies in the combination of structural embeddings from citation networks and text embeddings from documents for classification. While previous works have explored text-based classification or citation-based classification independently, we combine both sources of information, which allows us to capture both semantic and structural patterns..

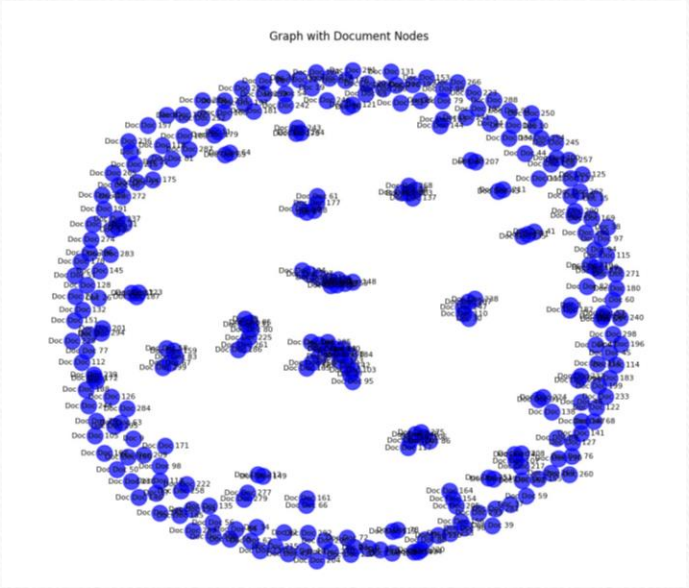
Previous Work

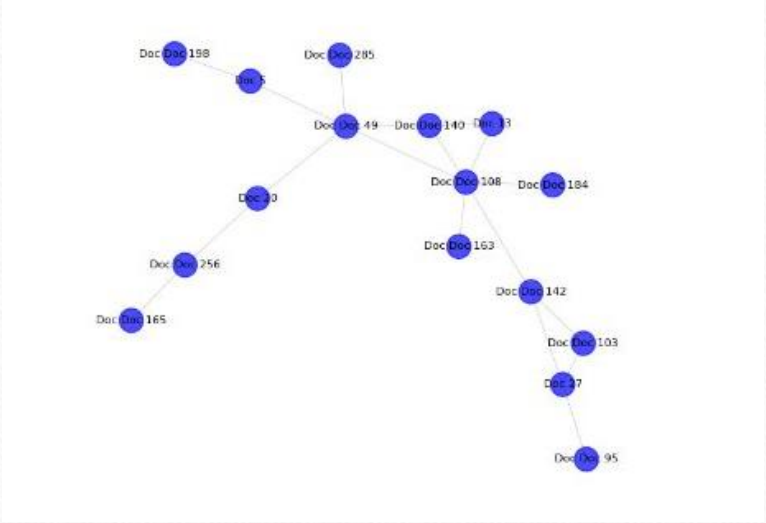
TextGCN Model: TextGCN (Liang Yao et al., 2019) utilizes a graph of words and documents as nodes, to generate text embeddings. This method captures semantic relationships based on word co-occurrences within documents

GNN on Citation Graphs: Studies such as those by (Weihua Hu et al.) applied Graph Neural Networks (GNNs) to citation graphs, where documents are linked based on citation relationships. Graph models are trained to reflect structural patterns in the citation network .

Combining Text and Graph Embeddings: Some research (e.g., Paheli et al., 2022) has explored combining text embeddings (like Doc2Vec) and graph embeddings from citation networks.

Dataset Description





- For Gnn the dataset is ogbn-arxiv dataset is a graph representing the citation network among Computer Science (CS) ARXIV papers indexed by MAG. Here's a concise overview of the dataset.
- **Nodes:** Represent ARXIV CS papers.
- **Edges:** Edges indicate that one paper cites another.
- **Node Features:**128-dimensional feature vectors for each paper from Doc2Vec embeddings of the abstract
- **Objective:** Predict the primary subject area (from 40 categories, e.g., cs.AI, cs.LG) for each paper.

This dataset is particularly useful for benchmarking graph-based machine learning models.

Methodology

1.Preprocessing:

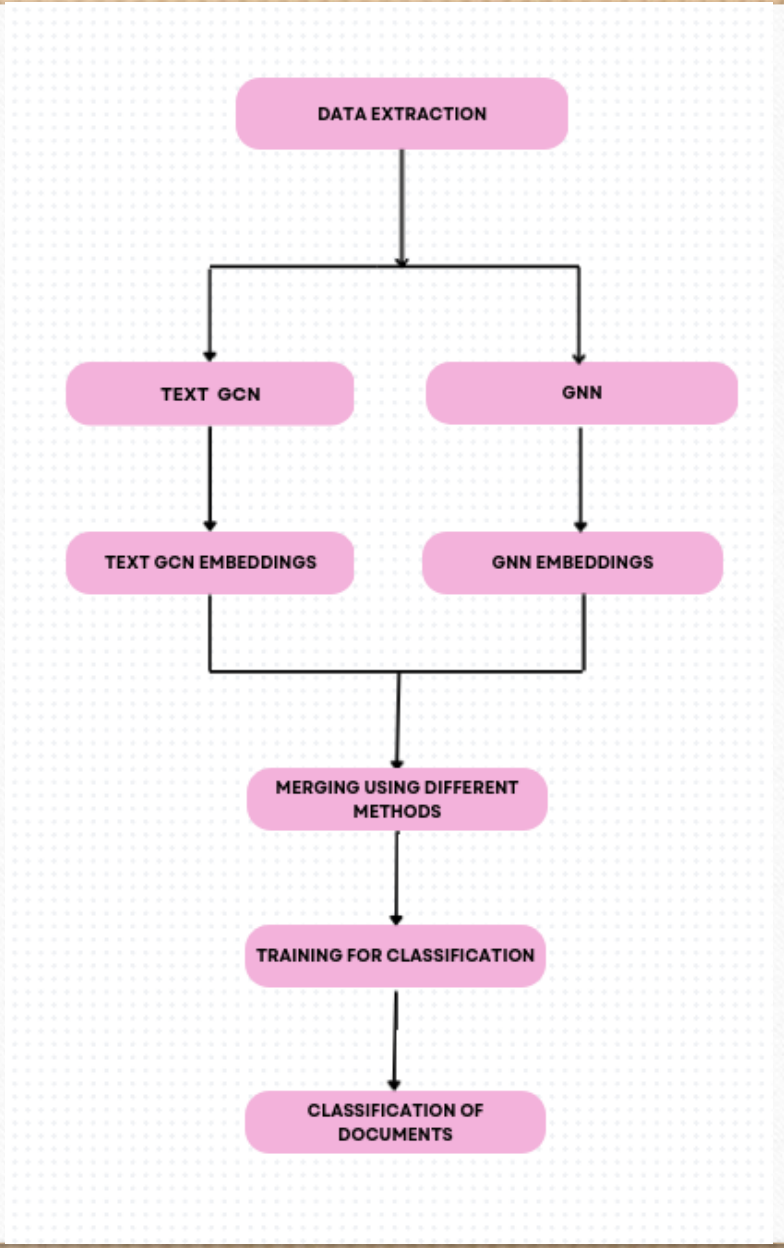
- Clean the text by removing stopwords, punctuation, and irrelevant terms.
- Extract Vocab to make doc-word,word-word graph.

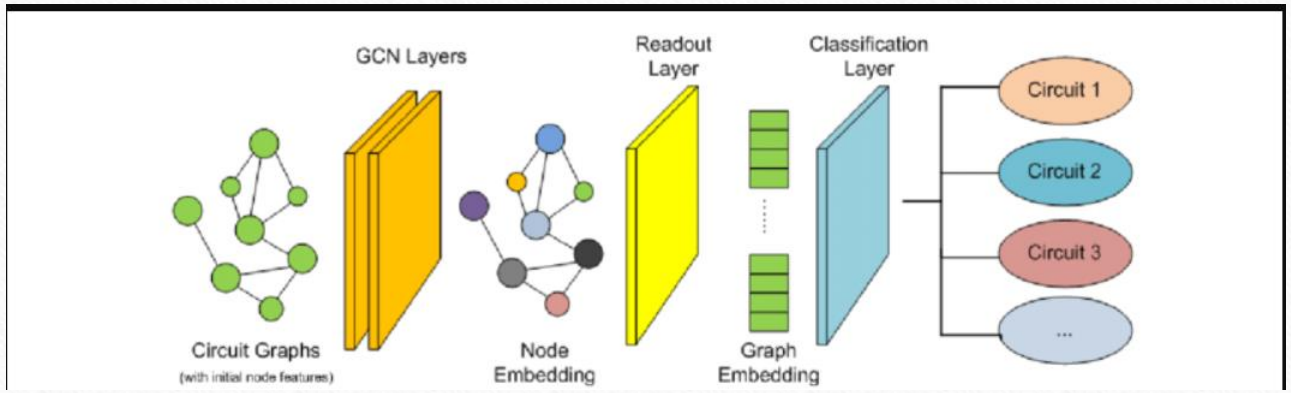
2. TextGCN Graph Processing

- Generate document embeddings based on the word-document co-occurrence graph.
- Capture semantic relationships within the document text.

3.GNN:

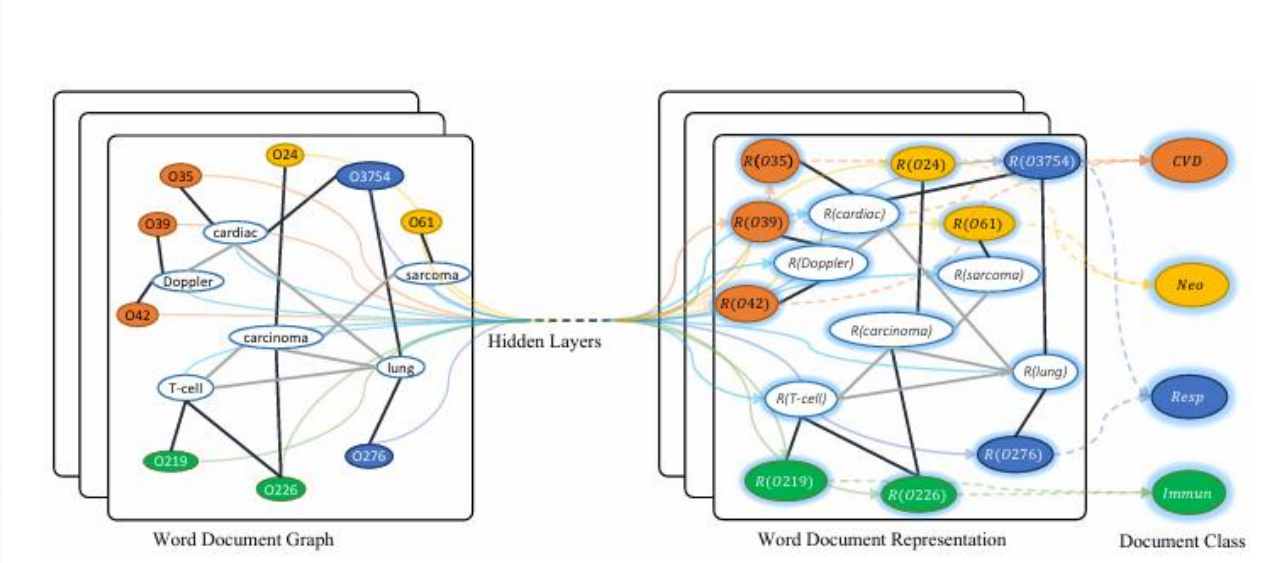
Use citation graphs to generate embeddings, focusing on structural relationships.





Architecture for GNN

Architecture for TextGCN



4.Combined Model:

- Train each model separately on the respective graph representations
- Fuse embeddings from TextGCN (semantic) and GNN (structural).

5. Training and Evaluation

•**Training**:

- Trained the combined model to integrate both semantic and structural features.
- Use a neural network to integrate and process combined embeddings.

•**Evaluation Metrics**:

- Use metrics such as accuracy, F1-score, precision, recall,precision for classification tasks.

Comparison with SOTA:

- Our results are on par with other state-of-the-art models that focus on either text or citation embeddings but lag in fully integrating both approaches in a unified model.

Key Insights:

- Using Node2Vec and MetaPath2Vec, we can generate only structural embeddings, as they do not require initial texual features, which will allow a more broader information capture.
- Using a domain-specific pre-trained (here, research papers) will render better initial textual features for GNN

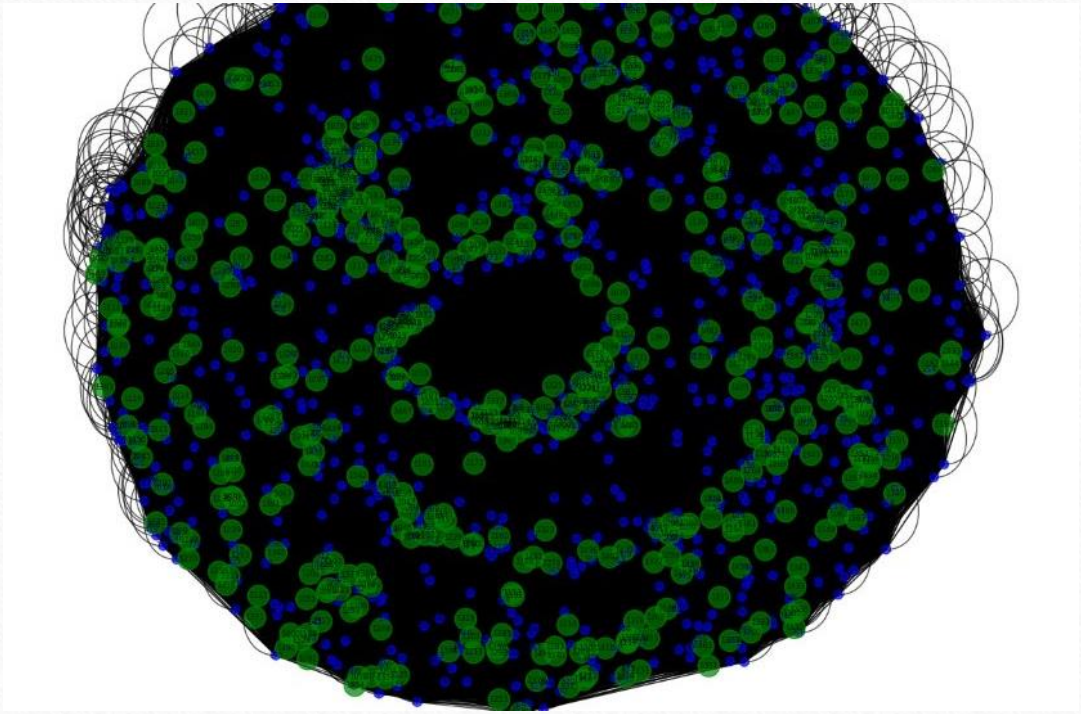
Results:

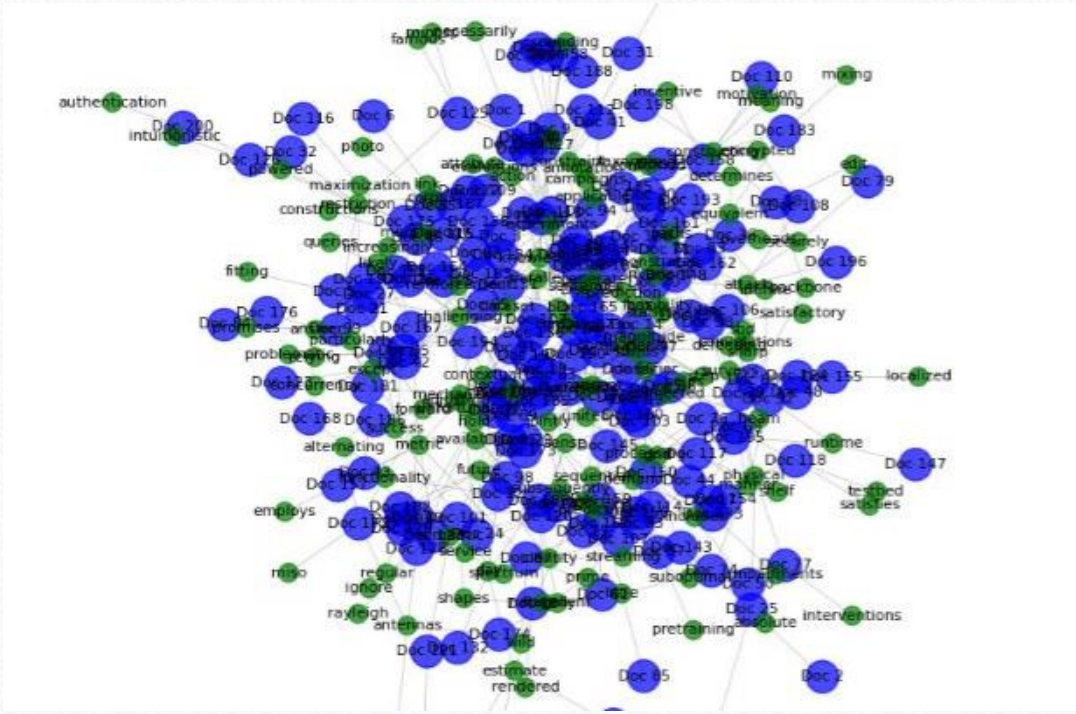
Methods of Merging	Validation Accuracy	Test Accuracy
Addition	55.50%	48.22%
Multiplication	46.33%	42.69%
concatenation	58.26%	52.17%

Method	Accuracy
TextGCN	12.9%(for 2 layers)
GNN	58.34%
Combined Model	52.17%

Individual Contributions:

- **Kriti Arora:**
  - Training TextGCN, and GNN
  - CometML integration
  - Preparing GCN Embeddings
- **Bhavya Jain** :
  - Cleaned and preprocessed the raw textual data for TextGCN
  - Analysis of GCN Outputs
  - Conceived the methodology for our approach
  - Training TextGCN
  - Final Fused model
- **Mallarapu Hema Varshini:**
  - Preparing GNN Embeddings
  - Concatenation of Embeddings
  - Experimented on MLP
- **Kaki Venkata Vaneesha**:
  - Enhanced Improved GNN and GCN visualization.
  - Enabled clearer graph model analysis.
  - Experimented on Node2Vec





Graph for TextGCN

Challenges in Our Work

1. **Framework Compatibility**: Resolving issues between TensorFlow and Torch compotibility
2. **Data Preprocessing**: Extensive cleaning to handle labels, vocab, and sentence structures, addressing inconsistencies between presplit and non-presplit labels.
3. **High-Class Complexity**: Achieving meaningful accuracy with 40 classes despite experimenting with layers, epochs, and model tuning.
4. **Graph Embedding Extraction**: Managing 5,000+ nodes, extracting embeddings for 6,241 nodes, and isolating document-specific IDs in the graph.
5. **Memory limitations:** The model crashed when scaling to 5000+ nodes.

Future Work:

• Exploring scalable models to handle larger graphs without memory bottlenecks.

• Fine-tuning preprocessing steps and feature engineering for better class distinction in imbalanced datasets.