GRAPH TEXT GCN
LEARNING FROM GRAPH AND TEXT FOR PREDICTIVE ANALYSIS USING GRAPH CONVOLUTIONAL NETWORKS

Project Report Submitted By

Ravindra Bodavula	N180697
Veera Durga Ramoju	N180106
Gayatri Vanapalli	N180093
UmaMaheswari Potnuri	N180078
Yamini Boyina	N181142

Under the guidance of

Dr. Bhavani Samineni
(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Rajiv Gandhi University of Knowledge Technologies
Andhra Pradesh, Nuzvid

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE AND TECHNOLOGIES NUZVID,ELURU(521202),

ANDHRA PRADESH

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION CERTIFICATE

This is to certify that the project entitled, "LEARNING FROM GRAPH AND TEXT FOR PREDICTIVE ANALYSIS USING GRAPH CONVOLUTIONAL NETWORKS" is submitted by B Ravindra (N180697), R Veera Durga (N180106), V Gayatri (N180093), P Uma Maheswari (N180078), B Yamini(N181142) to the department of Computer Science and Engineering, Rajiv Gandhi University Of Knowledge Technologies, Nuzvid for the submission of minor project report in IIIrd year B.Tech in Computer Science and Engineering is a bonafide work carried out under supervision and guidance during the academic year 2023.

MR.S.Chiranjeevi	MRs. S. Bhavani
Assistant Professor	Assistant Professor
Head of the Department	**Project Supervisor**
Dept. of CSE	Dept. of CSE
RGUKT, Nuzvid	RGUKT, Nuzvid

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE AND TECHNOLOGIES NUZVID,ELURU(521202),

ANDHRA PRADESH

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DISSERTATION APPROVAL CERTIFICATE

This is to certify that the project entitled, "LEARNING FROM GRAPH AND TEXT FOR PREDICTIVE ANALYSIS USING GRAPH CONVOLUTIONAL NETWORKS" is submitted by B Ravindra (N180697), R Veera Durga (N180106), V Gayatri (N180093), P Uma Maheswari (N180078), B Yamini(N181142) to the department of Computer Science and Engineering, Rajiv Gandhi University Of Knowledge Technologies,Nuzvid for the award of Bachelor of Technology in Computer Science and Engineering, has been accepted by the external examiners and that these students have successfully defended the project viva in the Viva-Voice examination held today.

MR.S.Chiranjeevi	MRs. S. Bhavani
Assistant Professor	Assistant Professor
Head of the Department	**Project Supervisor**
Dept. of CSE	Dept. of CSE
RGUKT, Nuzvid	RGUKT, Nuzvid

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE AND TECHNOLOGIES NUZVID,ELURU(521202),

ANDHRA PRADESH

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ACKNOWLEDGMENT

We would like to acknowledge and express our gratitude to the following people for their magnificent support in sharing their wisdom and knowledge to our successful completion of the final year project.

We would like to express our profound gratitude and deep regards to our guide **Mrs.S.Bhavani, Assistant professor**, Dept. of Computer science and Engineering for her exemplary guidance, monitoring and constant encouragement throughout the B.Tech course. We shall always cherish the time spent with her during the course of this work due to the valuable knowledge gained in the field of reliability engineering.

We express gratitude to **MR.S.Chiranjeevi, Head of the Department** (CSE) and other faculty members for being a source of inspiration and constant encouragement which helped us in completing the project successfully.

We are thankful to the director, **Prof**. **G.V.R SRINIVASARAO** for giving this opportunity to do this project.

Finally,yet importantly, we would like to express our heartfelt thanks to our beloved God and parents for their blessings, our friends for their help and wishes for the successful completion of this project.

ABSTRACT

"GraphText GCN: Learning from Graph and Text for Predictive Analysis using Graph Convolutional Networks"

The objective of our project is to develop a sophisticated system capable of predicting the stance between users participating in conversations. By leveraging the advancements in Graph Convolutional Networks (GCN) and employing supervised learning techniques, we achieve accurate and reliable stance prediction The conversation data is transformed into a graph representation using an adjacency matrix, allowing us to effectively incorporate the power of GCN. We extract relevant features from the data and train and evaluate models using labeled data. Through our work, we seek to demonstrate the remarkable capabilities of GCN in capturing the intricate relationships and dynamics within conversations to accurately predict user stances. The outcomes of our study will shed light on the potential of GCN-based models in understanding and analyzing user interactions, fostering a deeper comprehension of the underlying dynamics in conversational data.

CONTENTS

CHAPTER - 1

INTRODUCTION

In the era of social media and online communication platforms, understanding the sentiment and stance of users engaged in conversations has become increasingly crucial. The ability to predict the stance (positive, negative, or neutral) between users participating in a conversation has numerous practical applications, ranging from sentiment analysis to social media monitoring and online reputation management. To address this need, our project aims to develop a robust system that can accurately predict the stance between users in conversations, given their author IDs and graph-structured data.

To achieve this goal, we harness the power of Graph Convolutional Networks (GCN) in combination with supervised learning techniques. GCN is a cutting-edge deep learning architecture that is specifically designed to operate on graph-structured data. By treating the conversation data as a graph, with users represented as nodes and their interactions as edges, we can effectively model the complex relationships and dependencies inherent in conversational dynamics.

The conversation data will be transformed into a graph representation using an adjacency matrix, which captures the connections between users based on their interactions. This graph structure allows us to leverage the rich contextual information and capture the influence and propagation of stances within the conversation.

To enable the prediction of stances between users, we will extract relevant features from the conversation data. These features may include textual content, user metadata, temporal information, and social network characteristics. By incorporating these features, we aim to capture both the explicit and implicit cues that contribute to the stance of users in a conversation. Various evaluation metrics will be employed to assess the performance and effectiveness of our system, including precision, recall, F1-score, and accuracy.

By successfully developing a system that can predict the stance between users participating in conversations, we aim to provide valuable insights into user interactions, sentiment analysis, and online discourse dynamics. The outcomes of this project have the potential to contribute to a wide range of applications, including social media analytics, customer sentiment analysis, and community management.

Problem Statement:

In the domain of Twitter data analysis, accurately predicting the stance between users engaged in conversations remains a significant challenge. The abundance of unstructured and noisy data, coupled with the dynamic nature of discussions, poses obstacles to achieving high prediction accuracy.

The primary problem we aim to address in this project is the lack of an effective system that can reliably predict the stance between users in Twitter conversations using supervised learning techniques. While supervised learning offers promising potential for stance prediction, the unique characteristics of Twitter data and the complexities of stance classification require careful consideration and tailored approaches.

Several challenges contribute to the complexity of the problem. First, Twitter conversations often contain user-generated content with informal language, abbreviations, slang, and hash-tags, making it challenging to accurately capture the sentiment and stance behind these expressions. Dealing with these linguistic nuances and contextual cues is crucial for precise stance prediction.

Second, labeled data for training supervised learning models is often scarce and limited in its coverage of different conversation types, domains, and user demographics. Building a diverse and representative labeled dataset for training models with sufficient generalization capacity is a resource-intensive and time-consuming task.

Third, the dynamic nature of Twitter conversations, where new topics emerge and viral trends rapidly evolve, requires models that can adapt and generalize well to unseen data. The challenge lies in developing approaches that can capture the temporal aspects of conversations and effectively incorporate new trends and dynamics into the prediction process.

Addressing these challenges and developing an effective system for stance prediction in Twitter conversations using supervised learning techniques has substantial implications. It would facilitate sentiment analysis, opinion mining, and understanding of public discourse on social media. Additionally, it would enable applications such as targeted advertising, brand reputation management, and crisis communication, among others.

By leveraging supervised learning and addressing the unique characteristics and challenges of Twitter data, our project aims to contribute to the development of an accurate and robust system for predicting user stance in Twitter conversations. The results of our research will provide insights into improving stance classification models, advancing sentiment analysis techniques, and enhancing our understanding of user interactions in the Twitterverse.

CHAPTER – 2

Literature Review

Stance detection, the task of predicting the stance between users in conversations, has gained significant attention in recent years. Several studies have explored different approaches and methodologies to address this challenge.

Somasundaran and Wiebe (2010), Walker et al. (2012a), Sridhar et al. (2015), Mohammad et al. (2016), Derczynski et al. (2017), Sobhani et al. (2017), Joseph et al. (2017), Li et al. (2018), Porco and Goldwasser (2020), and Conforti et al. (2020) are among the notable works that have contributed to the field of stance detection. These studies have explored various settings, datasets, and computational approaches, providing valuable insights into the state-of-the-art techniques.

Different perspectives and methodologies have been employed in stance detection. Stance can be assigned at the utterance level or the user level, taking into account the text, context, or both. Murakami and Raymond (2010), Walker et al. (2012b), and Yin et al. (2012) have addressed user-level stance, while Sridhar et al. (2015), Li et al. (2018), Benton and Dredze (2018), Conforti et al. (2020), and Porco and Goldwasser (2020) have explored the nuanced relationships between posts and user-level stance. Our project aligns with these studies, considering both post and user-level stances.

Previous approaches to stance detection have utilized modal verbs, opinion and sentiment lexicons (Somasundaran and Wiebe, 2010; Murakami and Raymond, 2010; Yin et al., 2012; Wang and Cardie, 2014; Bar-Haim et al., 2017). Recent advancements have witnessed the application of graphical models (Joseph et al., 2017), conditional random fields (Hasan and Ng, 2013), and various neural architectures (Hiray and Duppada, 2017; Sun et al., 2018; Chen et al., 2018; Kobbe et al., 2020). These methods are often language and domain-dependent.

Unsupervised approaches have also been explored, albeit to a lesser extent than supervised methods. Somasundaran and Wiebe (2010) employed generic opinion and sentiment lexicons, while Kobbe et al. (2020) focused on classifying stances based on argumentation structures. Other unsupervised approaches have leveraged syntactic rules for topic and aspect extraction (Ghosh et al., 2018) or aspect-polarity-target information extraction (Konjengbam et al., 2018). However, these approaches are often language-dependent and rely on external resources, limiting their adaptability to different domains and communities with varying discussion norms. In contrast, our project embraces a fully supervised learning approach.

The utilization of the conversation structure in stance detection has been explored by Li et al. (2018), Porco and Goldwasser (2020), and Wei et al. (2019). These works have incorporated authors' interactions and textual content to create a global representation. Wei et al. (2019) also considered the structure in stance-based rumor detection. However, their methods integrate the conversation structure in an opaque manner through different neural architectures, making it difficult to assess the direct contribution of the structure to the classification task. In our project, we exclusively rely on the conversation structure, emphasizing its significance in processing conversational data.

Prior research by Murakami and Raymond (2010) and Walker et al.(2012a) utilized the max-cut problem as a computational tool for identifying stances taken by users in policy debates. These studies introduced explicit agreement and disagreement markers into the network representation, assigning positive and negative edge weights based on simple heuristics and handcrafted patterns such as "I agree" or "I disagree." However, this approach overlooks cultural and platform norms and fails to capture nuances like irony or other discursive styles.

In contrast to previous works, our project takes a different approach by leveraging Graph Convolutional Networks (GCN) for stance prediction in conversations. GCNs have demonstrated their efficacy in capturing relational information and graph structures, making them well-suited for analyzing conversational data represented as graphs.

While most works in stance detection primarily rely on supervised models, a few studies have explored unsupervised approaches. Somasundaran and Wiebe (2010) used generic opinion and sentiment lexicons, and Kobbe et al. (2020) classified stances based on frequently used argumentation structures. However, these unsupervised approaches often depend on specific languages, external resources, or topic-specific rules, limiting their applicability to diverse domains and communities.

Our project takes a supervised learning approach, utilizing labeled data to train and evaluate models. By leveraging the power of GCNs and incorporating relevant features extracted from the graph-structured conversation data, we aim to achieve accurate and reliable predictions of user stances. Through our research, we aim to showcase the potential of GCN-based models in understanding and analyzing user interactions, providing valuable insights into the dynamics and relationships within conversational data.

In summary, while previous studies have made significant contributions to the field of stance detection, our project introduces the novel application of GCNs for predicting user stances in conversations. By harnessing the graph structure of the data and employing supervised learning techniques, we aim to advance the state-of-the-art in stance detection and contribute to a deeper understanding of user interactions in conversational contexts.

Previous Works

‘GraphText GCN: Learning from Graph and Text for Predictive Analysis using Graph Convolutional Networks’ we conducted a comprehensive literature review to explore relevant work in the field.

Work-1: Knowledge Graph Construction Using BERT Model Using semevalDataset 2016.

"Knowledge Graph Construction Using BERT Model Using semevalDataset 2016" is a research study that utilizes the BERT model and the semeval dataset to build a knowledge graph. The study focuses on entity recognition, relation extraction, and entity linking to capture contextual relationships and semantic dependencies. The knowledge graph serves as a resource for various natural language processing applications, demonstrating the effectiveness of the BERT model in constructing knowledge graphs and highlighting the value of the semevaldataset.

Limitations and drawbacks: "Knowledge Graph Construction Using BERT Model Using semevalDataset 2016" approach include: limited scope, dependency on BERT model with potential biases, data preprocessing challenges, scalability issues, limited context understanding, and lack of comprehensive evaluation metrics.

Work-2:Exploring CNN-based Natural Language Processing with TF-IDF for Text Classification

CNN-based natural language processing models with TF-IDF (Term Frequency-Inverse Document Frequency). By combining CNN architecture with TF-IDF representation, we aimed to leverage the strengths of both approaches in text classification and sentiment analysis tasks. The CNN model helped capture local features and patterns, while TF-IDF provided a measure of term importance. This combination aimed to enhance the model's performance in understanding and analyzing textual data.

Limitations and drawbacks: Exploring CNN-based Natural Language Processing with TF-IDF for Text Classification: Limited contextual understanding due to reliance on local word patterns. Fixed window size may lead to information loss or inadequate coverage for different text lengths. CNN models overlook sequential information, impacting tasks requiring word order comprehension. Dependency on predefined features can restrict adaptability and generalization capabilities.

Work-3:"Exploring Word2Vec for Text Representation in Natural Language Processing"

"Exploring Word2Vec for Text Representation in Natural Language Processing" investigates the use of Word2Vec for representing words in NLP tasks. It explores the strengths of Word2Vec in capturing semantic relationships but acknowledges limitations such as fixed

context window size and handling out-of-vocabulary words. Despite limitations, Word2Vec is valuable in NLP, but interpretability may be compromised with vector-based representations. The study provides insights for future research and improvements in word embedding techniques.

Limitations and Drawbacks:Limited capture of semantic relationships due to the fixed context window size in Word2Vec. Difficulty in handling out-of-vocabulary words or rare words not present in the Word2Vec model. Lack of consideration for document-level context and discourse information.Dependency on pre-trained Word2Vec models, limiting adaptability to specific domains or languages.Potential loss of interpretability due to the vector-based representation of words.

Work-4:One-Hot Encoding for Text Representation in Natural Language Processing

"One-Hot Encoding for Text Representation in Natural Language Processing" explores the use of one-hot encoding as a text representation technique. It involves representing each word as a binary vector, making it simple and interpretable. However, one-hot encoding has limitations in capturing semantic relationships and dealing with high-dimensional feature spaces. Nonetheless, it remains a commonly used method in certain NLP tasks, and the study provides valuable insights into its usage and potential enhancements.

Limitations and drawbacks of "One-Hot Encoding for Text Representation in Natural Language Processing" include high dimensionality, lack of semantic information, sparse representation, inability to handle out-of-vocabulary words, lack of continuous representations, and difficulty in handling variable-length texts.

Work-5:Exploring glove for Text Representation in Natural Language Processing

Glove embeddings are trained on large corpora and provide valuable semantic information about words, allowing us to capture contextual similarities and improve the performance of our models. By leveraging glove embeddings, we aimed to enhance the understanding of word meanings and enable more effective text analysis and classification.

Limitations and drawbacks of glove embeddings include domain and language dependency, fixed-size representations, lack of adaptability, and challenges with out-of-vocabulary words.

Work-6:Multilabel Classification on Political Data

We developed a model capable of predicting multiple political labels associated with a given text, such as political party affiliation, ideological stance, and policy positions. By training on a diverse dataset of political texts, we aimed to provide a comprehensive understanding of the complex political landscape.

Limitations and Drawbacks: challenge of handling imbalanced label distributions, potential bias in the training data, difficulty in capturing nuanced political positions, and the need for extensive labeled data for accurate predictions. Additionally, the interpretation of multilabel classification results can be complex, requiring careful consideration of label correlations and potential trade-offs.

Data

The data used in our project consists of Twitter conversations extracted from a diverse range of topics and discussions. We collected this data by leveraging Twitter's API, which provides access to public tweets and their associated metadata.

The dataset includes information such as author IDs, usernames, gender, age, marital status, political party affiliation, country, religion, and education. This information provides valuable contextual details about the participants in the conversations.

Additionally, the dataset contains the text of the tweets exchanged within the conversations, which serves as the primary input for our stance prediction model. The conversations are structured as a series of tweets, with each tweet representing an individual user's stance or opinion.

To enhance the quality and reliability of the dataset, we performed data cleaning and preprocessing steps. This involved removing irrelevant tweets, handling missing or incomplete information, and ensuring data consistency 002E

The availability of this Twitter dataset enables us to train, validate, and evaluate our stance prediction model using supervised learning techniques. By utilizing this real-world data, we aim to develop a robust and effective system for predicting the stance between users participating in conversations on Twitter.

	author_id	username	gender	age	marital_status	political_party	country	religion	education
0	85	'1234567abcde	NULL	NULL	NULL	NULL	NULL	NULL	NULL
1	118	'12yrsold'	'Female'	NULL	'Single'	'Independent'	'United States'	NULL	NULL
2	327	'20040860'	NULL	NULL	NULL	NULL	NULL	NULL	NULL
3	547	'561lw'	NULL	NULL	NULL	NULL	NULL	NULL	NULL
4	578	'7thDebater'	'Male'	NULL	'Single'	'Other'	'United States'	'Christian-other'	NULL
5	662	'ALgalhunter'	'grrrl'	NULL	'In a Relationsh	'Republican'	'United States'	'Christian-other'	'High School'
6	678	'AR3YU0WHIPP	NULL	NULL	NULL	NULL	NULL	NULL	NULL
7	683	'ARMYANT'	'Male'	33	'Single'	'Republican'	'United States'	'Agnostic'	'Some College'
8	704	'Aagg'	NULL	NULL	NULL	NULL	NULL	NULL	NULL
9	731	'AbbyNestor'	'Female'	NULL	'Single'	'Libertarian'	'United States'	NULL	'In College'
10	763	'Achtung'	'Male'	NULL	'Single'	'Republican'	'Belgium'	'Atheist'	NULL

Figure 1: Primary Data from which we started pre-processing of Data

CHAPTER – 3

Existing methods

3.a Details of existing methods:

In this we prepare two tables of data for existing methods

1. Details of works we referred i.e Papers we referred
2. Datasets

Note: click the above links to redirect to data sheets

3.b Research gaps & Methodology to Overcome

Paper Name:

Generative Entity-to-Entity Stance Detection with Knowledge Graph Augmentation.

Limitations:

1) Need GPU resources.
2) System limitation – it means even though the model is working efficiently but it is not getting source, target pairs correctly.
3) Failure mode is defined as situations where our model fails to correctly extract a stance triplet of a given text.
4) Potential limitation. Although balanced views are considered, the topic coverage in SEESAW is not exhaustive, and does not include other trending media or content of different modalities for expressing opinions, such as TV transcripts, images, and videos. Thus, the predictive performance of our developed system may still be under investigated.

Methodology to Overcome:

1) Without GPU resources means we can modify the model architecture, we can reduce the batch size. Because batch size takes more space.
2) For the system limitation we can use supervised methods as it is labeled data we can extract the data easily. And we can use Rule-based methods in that we can specify the phrases and some keywords.
3) We can take the data efficiently which is having multiple domains through that it may identify different contexts. And including NER techniques for easy identifying the entities correctly and ambiguity addressing for the link between entities.
4) We need to expand the data then no need to work at border information. And on-going monitoring is necessary as we can update the dataset when new information is added.

Paper Name:

Domain Adaptation for Stance Detection towards Unseen Target on Social Media

Limitation:

In the future, we hope to study the explainability of knowledge transfer between targets and the link between emotion and stance.

Methodology to Overcome:

We can use NLP for the analysis of data.

We can take both emotions and stance for efficiency.(Dataset becomes efficient for model)

If we take mental health research related datasets where both emotion and stance are expressed there we can find the link between emotion and stance.

Paper Name:

Boosting-GNN: Boosting Algorithm for Graph Networks on Imbalanced Node Classification.

Limitation:

Sensitive to Initial Node Weights, because of this it affects convergence and it may affect efficiency.

Methodology to Overcome:

There are some initialization techniques for initial node weights drawback .Techniques like Xavier/Glorot initialization the weights are initialized by sampling form a uniform or Gaussian distribution with zero mean and specific variance. And we can also design custom weight initialization strategies based on the requirements.(He) initialization technique also can be used in this category.

Paper Name:

Improved target -specific stance detection on social media platforms by delving into conversation threads.

Limitations:

1) It only concentrates on specific domains and can be over-representative on specific domains.
2) Social media data is too dynamic.

Methodology to Overcome:

1) Diverse data collection helps us to concentrate on more data not on only specific domains.
2) Our model should be trained in such a way that it can adapt to new changes without giving errors.

3c. Keywords and metrics

Keywords:

- **Graph Embeddings**

Graph embeddings refer to the process of representing graphs, which are structured data consisting of nodes and edges, as continuous and low-dimensional vector representations. Graph embeddings aim to capture the structural and semantic information of the nodes and edges in a graph, enabling machine learning algorithms to operate on graph data.

(Node2Vec, Graph Convolutional Networks, GraphSAGE, DeepWalk, Graph Autoencoders)

- **Graph**

A mathematical representation of data as nodes (vertices) connected by edges.

Nodes:- Nodes represent individual entities in the graph.

Edges:- Edges represent the relationships or connections between nodes.

- **Convolution**

A mathematical operation that combines information from neighboring nodes to update the feature representations of nodes.

The convolution operation in GCNs can be defined as follows:

Initialization:- Each node in the graph is associated with an initial feature vector or embedding.

Neighborhood Aggregation:- For each node, information is aggregated from its neighboring nodes.

Transformation:-After aggregating the features from the neighborhood, a transformation is applied to the aggregated features.

Updating Node Representations:-The transformed features are used to update the representation of each node.

- **Neural Network**

A computational model inspired by the human brain, composed of interconnected nodes (neurons) organized in layers.

The basic structure of a GCN neural network typically involves the following components:

Input Layer: The input layer of a GCN receives the initial node features or embeddings as input. These features can represent attributes or characteristics associated with each node in the graph.

Graph Convolutional Layers: The graph convolutional layers form the core of a GCN. Each layer performs the convolution operation, which propagates information through the graph and updates the node representations based on the features of neighboring nodes.

Non-linear Activation Function: After the aggregation step in each graph convolutional layer, a non-linear activation function is typically applied element-wise to introduce non-linearity into the model.

Pooling or Aggregation Layers: These layers can be used to summarize information from multiple nodes or to aggregate information at the graph level.

Output Layer: The output layer of a GCN produces the final predictions or representations based on the learned node representations.

- **Feature Representation**

A numerical vector that captures the properties or attributes of a node in the graph.
It refers to the process of transforming the initial node features or embeddings of a graph into informative and expressive representations that capture the characteristics and relationships of the nodes in the graph.

- **Latent representation**

Latent representation is a transformed and compact form of data that captures underlying patterns or features without explicit interpretation.

- **Node Classification**

The task of assigning a specific label or class to each node in a graph. The goal is to learn a model that can accurately classify the nodes based on their features and the relationships with other nodes in the graph.

- **Graph Classification**

The task of assigning a label or category to an entire graph based on its structure and node features.
The main goal of graph classification is to provide a framework for categorizing and analyzing graphs, enabling better understanding, decision making, and extracting meaningful insights from graph-structured data.

- **Link Prediction**

The task of predicting missing or future connections between nodes in the graph. It involves estimating the likelihood or probability of a connection between two nodes that do not have an existing edge in the graph.

- **Annotations**

Annotations in the context of Natural Language Processing (NLP) and machine learning refer to the process of manually labeling or tagging data with specific labels or attributes. Annotations provide a way to create labeled datasets that can be used for training and evaluating machine learning models.

- **Sentiment graph**

A sentiment graph, also known as an opinion graph or sentiment network, is a graphical representation of the sentiment or opinion relationships between entities or nodes.
It is a structured way of capturing and visualizing the sentiments expressed towards different entities and the connections between them based on sentiment analysis.

- **Feature Extraction**

The process of extracting relevant linguistic or textual features from the input text, which can include lexical, syntactic, semantic, or contextual information.
Some of the feature extraction techniques in machine learning and data analysis:

1. Principal Component Analysis (PCA)
2. Wavelet Transform
3. Bag-of-Words (BoW)
4. Word Embeddings (e.g., Word2Vec, GloVe, FastText)
5. Convolutional Neural Networks (CNN)
6. Recurrent Neural Networks (RNN)

- **Dependency parsing**

Dependency parsing in Graph Convolutional Networks (GCN) involves utilizing the syntactic dependencies between words in a sentence to enhance the representation learning process.

In dependency parsing for GCN, the words in a sentence are treated as nodes, and the syntactic dependencies between them are represented as edges in the graph. Each word node is associated with its corresponding word embedding or feature vector. The edges in the graph represent the grammatical relationships between words, such as subject-verb, verb-object, or modifier-modified.

- **Autoencoders**

Autoencoders are a type of neural network architecture used for unsupervised learning and dimensionality reduction. They consist of an encoder and a decoder, which work together to learn a compressed representation or encoding of the input data.

- **Forward Propagation**

During forward propagation, the input data is fed through the neural network, and the activations or outputs of each layer are computed sequentially, layer by layer, until the final output is generated.

(Input Layer, Hidden Layers,Output Layer)

- **Backward propagation**

Backward propagation is the process of computing the gradients of the network parameters (weights and biases) with respect to a loss function. It allows the network to learn and adjust its parameters to minimize the difference between the predicted output and the true output.

(Loss Calculation, Gradient Calculation, Parameter Update)

- **Heterogeneous graphs**

Heterogeneous graphs, also known as heterogeneous information networks, are graph structures where nodes and edges represent different types of entities and relationships, respectively. The main goal of heterogeneous graphs is to capture and represent the complex relationships and interactions among diverse entities in a structured manner.

- **Message passing**

Message passing refers to the process of exchanging information or messages between nodes in a graph-based model.

By passing messages between nodes, the goal is to capture and propagate information throughout the graph, allowing for the integration of local and global dependencies and facilitating more effective analysis and learning.

- **Attention**

Attention is a mechanism used in deep learning models to selectively focus on specific parts of the input data.

It allows the model to assign varying degrees of importance or relevance to different elements, enabling more effective information processing and capturing dependencies.

- **Aggregation**

The process of combining multiple individual data points or entities into a summarized representation.

It involves the transformation of a set of data into a single value or a smaller set of values that capture the essential information or characteristics of the original data.

- **Benchmark datasets**

  Benchmark datasets are standardized collections of data that are widely used to evaluate and compare the performance of algorithms or models in specific tasks.

  They provide a common reference point for assessing the effectiveness and capabilities of different approaches, allowing researchers to measure progress and make fair comparisons in their field.

- **DenseLayer**

  A dense layer, also known as a fully connected layer, is a fundamental building block in neural networks.

  It is a type of layer where each neuron is connected to every neuron in the previous layer. In a dense layer, the output of each neuron is determined by a weighted sum of the inputs from the previous layer, followed by an activation function.

- **Dropout**

  Dropout is a regularization technique used in neural networks to prevent overfitting.

  It involves randomly setting a fraction of the output units (neurons) in a layer to zero during training. This temporarily removes those units from the network, forcing the remaining units to learn more robust and independent representations of the input data.

- **FeatureMatrix**

  A feature matrix, also known as a feature dataset or feature matrix, is a structured representation of data used in machine learning and data analysis. It is a two-dimensional matrix where each row represents an individual data instance or sample, and each column represents a specific feature or attribute of that sample.

  Feature matrices are used as input data for training machine learning models. The features in the matrix capture relevant information about the samples, and the model learns patterns and relationships between these features to make predictions or perform analysis.

Metrics:

1. **Precision and Recall**

Precision is a metric used to evaluate the performance of a model, particularly in binary classification tasks.

- It quantifies the proportion of correctly predicted positive instances out of all instances predicted as positive by the model.
  Precision = True Positives / (True Positives + False Positives)
- It quantifies the proportion of correctly predicted negative instances out of all instances predicted as negative by the model.
- Precision = True Negatives / (True Negatives + False Negatives)
  Precision provides insights into the model's ability to make accurate positive predictions and avoid false positives.

Ex:-precision is evaluated in a sentiment analysis task. Suppose we have a model that predicts whether customer reviews are positive or negative.

- True Positives (TP): 150 reviews correctly predicted as positive.
- False Positives (FP): 50 reviews incorrectly predicted as positive.

Precision = TP / (TP + FP) = 150 / (150 + 50) = 0.75 or 75%
Recall = TP / (TP + FN) = 150 / (150 + 30) = 0.833 or 83.3%
The precision score of 0.75 and the Recall score of 0.833 means that out of all the reviews predicted as positive.

1. **F1-Score**

The F1-score is a metric commonly used in binary classification tasks to evaluate the performance of a model. It combines both precision and recall into a single score, providing a balanced measure of the model's effectiveness.
F1-score = 2 * (Precision * Recall) / (Precision + Recall)

Ex:-

- True Positives (TP): 150 reviews correctly predicted as positive.
- False Positives (FP): 50 reviews incorrectly predicted as positive.
- False Negatives (FN): Let's assume there are 30 reviews incorrectly predicted as negative.

Precision = TP / (TP + FP) = 150 / (150 + 50) = 0.75 or 75%
Recall = TP / (TP + FN) = 150 / (150 + 30) = 0.833 or 83.3%
F1-score = 2 * (Precision * Recall) / (Precision + Recall)
= 2 * (0.75 * 0.833) / (0.75 + 0.833)
= 0.789 or 78.9%

The F1-score of 0.789 indicates the overall performance of the model in terms of both precision and recall.

2. R2 score

The R-squared (R2) score, also known as the coefficient of determination, is a metric commonly used to assess how well a regression model fits the data.
It measures the proportion of the variance in the dependent variable that can be explained by the independent variables in the model.

R2 score = 1 - (SSR / SST)

Where:

- SSR (Sum of Squared Residuals) represents the sum of the squared differences between the actual values and the predicted values by the model.
- SST (Total Sum of Squares) represents the sum of the squared differences between the actual values and the mean of the dependent variable.

Ex:-A set of actual and predicted prices for five houses:
Actual Prices: [250,000, 300,000, 400,000, 350,000, 500,000] Predicted Prices: [240,000, 310,000, 380,000, 330,000, 520,000]
To calculate the R2 score, we first need to calculate the residuals, which are the differences between the actual prices and predicted prices:
Residuals: [10,000, -10,000, 20,000, 20,000, -20,000]
SSR = sum(residuals^2) = 10,000^2 + (-10,000)^2 + 20,000^2 + 20,000^2 + (-20,000)^2 = 1,200,000,000
SST = sum ((actual prices - mean(actual prices))^2) = (250,000 - 360,000)^2 + (300,000 - 360,000)^2 + (400,000 - 360,000)^2 + (350,000 - 360,000)^2 + (500,000 - 360,000)^2 = 1,300,000,000

R2 score = 1 - (SSR / SST)
= 1 - (1,200,000,000 / 1,300,000,000) = 0.077 or 7.7%

The R2 score of 0.077 indicates that the regression model explains approximately 7.7% of the variance in housing prices. It means that the model's predictions do not capture a significant amount of the variability in the actual prices.

3. Accuracy

Accuracy is a commonly used metric to evaluate the performance of a classification model.
It measures the proportion of correctly classified instances out of the total number of instances in the dataset. In other words, accuracy tells us how often the model makes correct predictions.

Accuracy = (Number of Correct Predictions) / (Total Number of Predictions)

Ex:-

True Positives (TP): 150 reviews correctly predicted as positive.
False Positives (FP): 50 reviews incorrectly predicted as positive.
True Negatives (TN): 200 reviews correctly predicted as negative.
False Negatives (FN): 100 reviews incorrectly predicted as negative.

Accuracy = (TP + TN) / (TP + TN + FP + FN)
Accuracy = (150 + 200) / (150 + 200 + 50 + 100)
= 350 / 500
= 0.7 or 70%

The accuracy score of 0.7 indicates that the model correctly predicts the sentiment of 70% of the customer reviews in the test set.

4. Graph plots

Graph plots, also known as graph visualizations or network visualizations, are graphical representations of graphs or networks. They provide a visual representation of the nodes (vertices) and edges (links) of a graph, allowing us to analyze and understand the structure, relationships, and patterns within the data.

5. Confusion Matrix

A confusion matrix provides a tabular representation of the model's predictions versus the true labels. It shows the counts of true positives, true negatives, false positives, and false negatives, allowing for a more detailed analysis of the model's performance across different classes.

6. Classification Report

A classification report is a summary of the performance of a classification model, providing various metrics for each class in the dataset. It provides insights into the precision, recall, F1 score, and support for each class, allowing for a detailed evaluation of the model's performance on individual classes.

Here is an example of a classification report:

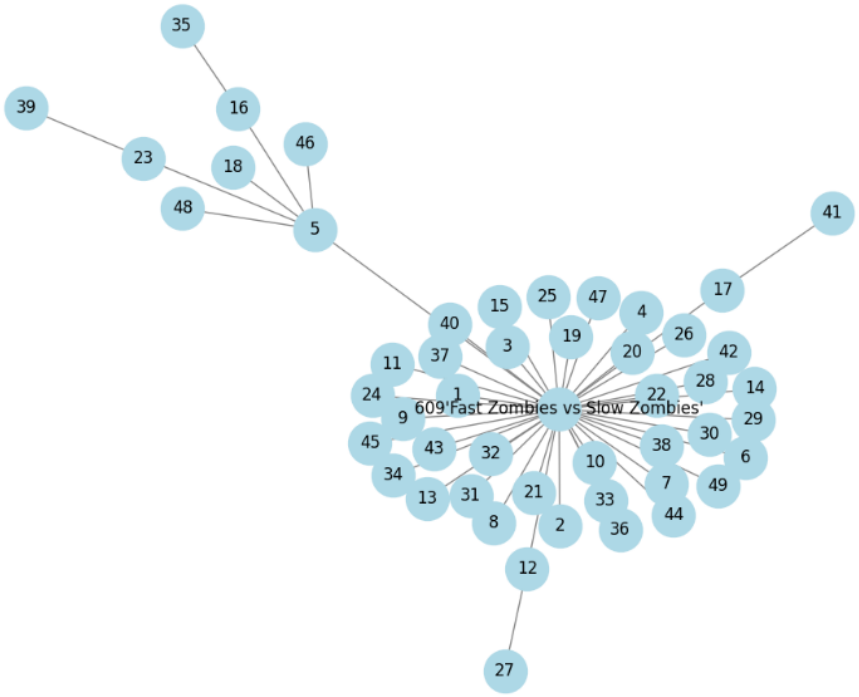
	Precision	Recall	F1-Score	Support
Class 0	0.85	0.92	0.88	150
Class 1	0.76	0.65	0.70	80
Class 2	0.92	0.95	0.94	200
Accuracy	-	-	-	0.88
Macro Avg	0.84	0.84	0.84	430
Weighted Avg	0.88	0.88	0.88	430

The classification report also includes macro-averaged and weighted-averaged metrics, which provide an overall evaluation of the model's performance across all classes.

CHAPTER-4

PROPOSED METHODS

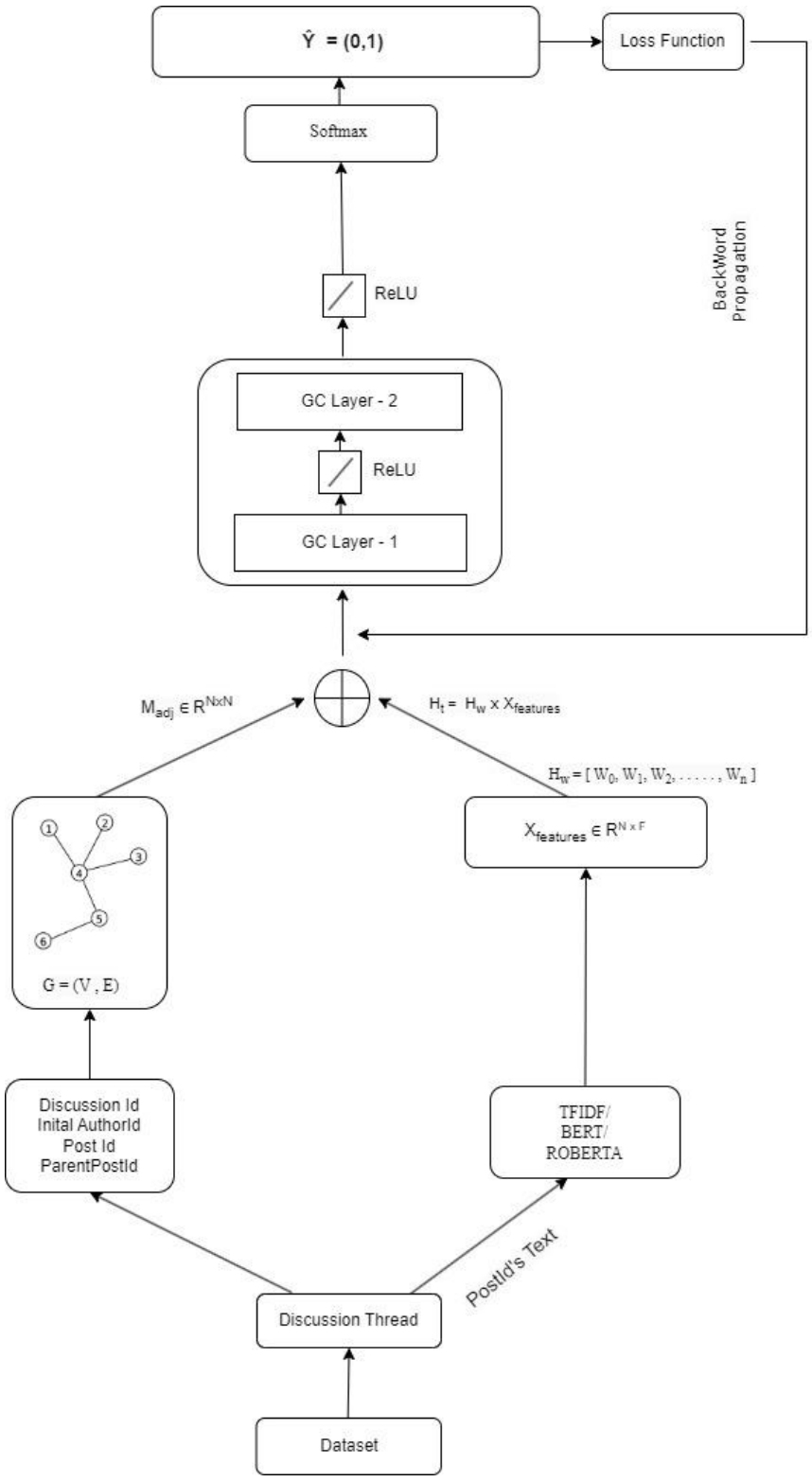
Graph Construction:



The graph construction is as follows:

First of all, the initial post (inital_post_id) by combining the initiating author ID and the title of the discussion. Then connects the author IDs to either the initial post or the parent post. For each post, if a parent post ID exists, an edge is added between the parent post and the current post. Otherwise, an edge is added between the initial post and the current post.

FLOWCHART:


Ŷ = (0,1)
Loss Function
Softmax
ReLU
BackWord Propagation
GC Layer - 2
ReLU
GC Layer - 1
$M_{adj} \in R^{NxN}$
$H_t = H_w \times X_{features}$
$H_w = [W_0, W_1, W_2, \ldots, W_n]$
$X_{features} \in R^{N \times F}$
G = (V , E)
Discussion Id
Inital AuthorId
Post Id
ParentPostId
TFIDF/
BERT/
ROBERTA
PostId's Text
Discussion Thread
Dataset

Each component in Flowchart:

- ConvinceMe Dataset consists of a total 31 topics out of which we had taken each discussion for the seven topics.
- For each discussion thread, we have taken Discussion id, PostId, AuthorId, Title, Parent Post-Id. By taking all these attributes we have constructed a graph.
- Next to the right of the Discussion Thread it is mentioned PostId's Text. By applying Baseline models we converted the text into feature vectors.
- Once the text features are combined with the graph's adjacency matrix to feed into the GCN model.
- The GCN model typically consists of multiple layers, with each layer performing a graph convolution operation.

GC Layer-1:

In this layer it will multiply the feature dimensions(N,F) and weight Matrix(F,H) dimensions.Then the result is multiplied by Adjacency matrix(N,N). The total result is passed to the hidden layer with the ReLu function.

$$H_t = H_w \times X_{\text{features}}$$

where, $\mathbf{H_t}$ is the resultant matrix of weights and features

$\mathbf{H_w}$ is the weight matrix

GC Layer-2:

In this layer after passing the result to the hidden layer the information is passed to the Softmax function with the ReLu function. The Softmax which converts the raw data into output categorical format.

Softmax equation is:

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

we used two loss functions:

BCE : $$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$

NLL : $$\text{loss}(x, class) = -\log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right) = -x[class] + \log\left(\sum_j \exp(x[j])\right)$$

GCN MODEL ARCHITECTURE:

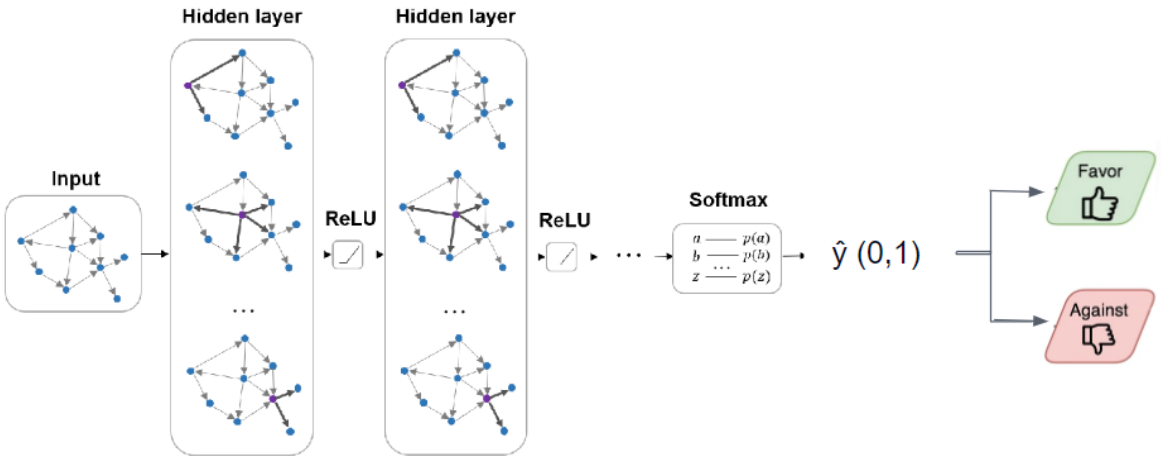


Fig: GCN Model Architecture

In the Graph Convolutional Network (GCN) model, the text features ($X_{features}$) are used as input to the model. Each user's text is converted into a feature vector representation by tokenizing the text using the BERT model.

Once the text features ($X_{Features}$) are obtained, they are combined with the adjacency matrix (M_{adj}) of graph to feed into the GCN model. The adjacency matrix represents the relationships between users in the graph. It is a binary matrix where each entry indicates whether an edge exists between two users. The adjacency matrix (M_{adj}) helps capture the graph structure and allows the model to propagate information between connected nodes during the GCN layers.

The GCN model typically consists of multiple layers, with each layer performing a graph convolution operation. In each layer,The model first applies a matrix multiplication between the input feature vectors and the weight matrix (X_{weight}). This produces a tensor of weighted features. The weighted features are then multiplied by the sparse adjacency matrix to aggregate information from neighboring nodes. The result is a tensor of aggregated features, which is then passed through a non-linear activation function ReLU.

After the graph convolution operation, the GCN model may incorporate additional layers, often referred to as hidden layers. These hidden layers can help the model learn more complex patterns and representations. Dropout is a regularization technique commonly applied to the hidden layers of a GCN. Dropout randomly sets a fraction of the output features to zero during training, which helps prevent overfitting and improves generalization.

Finally, the output of the last hidden layer is typically fed into a softmax activation ($\sigma(\hat{z})_i$) function to obtain the predicted class probabilities. The softmax function converts the model's

raw output into a probability distribution over the possible classes, allowing for multi-class classification.

Algorithm:

1. Take Madj and Xfeatures
2. calculate $H_t = X_{features} * X_{weight}$
3. calculate $X = M_{adj} * H_t$
4. loop upto No. of epochs
    a. give X to GC layer 1
    b. apply ReLU function to the hidden layers
    c. then hidden layer output is given to GC layer 2
    d. apply softmax =

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

    e. perform loss function to the predicted labels
    f. goto step 4 for backward propagation
5. Output stance prediction

CHAPTER-5

DATASETS

We have taken reference from the three datasets:

- ConvinceMe
- CreateDebate

	4Forums	CD	CM
# Topics	4	4	16
# Conversations	202	521	9,521
# Conversations (core)	202	149	500
# Authors	863	1,840	3,641
# Authors (core)	718	352	490
# Posts	24,658	3,679	42,588
# Posts (core)	23,810	1,250	5,876

The above datasets consists the followings attributes:

DiscussionId, DiscussionTitle, InitialAuthorId, PostIds, ParentPostIds, AuthorIds, PostIds' Text.

ConvinceMe dataset consists of Multiple Discussions on various topics.
In this we considered only the below topics.

discussion_id	Title	# posts in discussion
47	'Gay Marriage: Right or Wrong'	592
110	Existence of God	497
66	Evolution vs Creation	235
187	Pro-Choice vs. Pro-Life'	340
44	Firefox vs. Internet Explorer'	233
157	To Legalize Marijuana or not?	201

Dataset	# Topics	# Discussions	# Authors	# Posts	# Words/tokens
ConvinceMe	31	5413	5783	1,00,000	3.4 million
CreateDebate	31	63	743	3051	30510
4Forum	-	-	-	-	-

Fig: Overview of Datasets

CHAPTER-6

EXPERIMENTAL SET-UP

The project started by collecting the datasets .We have collected three datasets ConvinceMe,4Forrum,CreateDebate. Among those We have taken the input from the dataset called ConvinceMe dataset as this dataset contains multiple discussions when compared to other two datasets and it is also helpful for the future works.It consists of 31 Topics and with the 5413 Discussions along with the 5783 Author's containing 1,00,000 Posts.

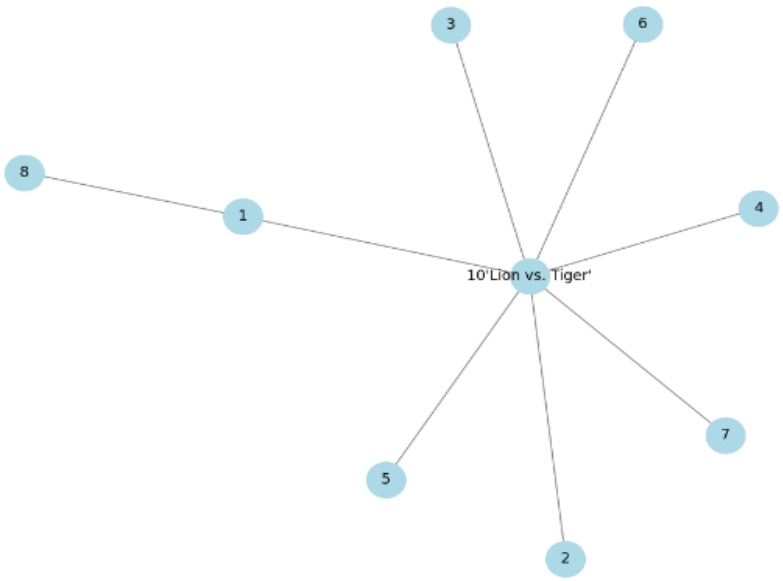
After taking the input from the ConvinceMe Dataset we have only considered seven discussions as these are having high number of posts ('Gay Marriage: Right or Wrong', Existence of God, Evolution vs Creation, Pro-Choice vs. Pro-Life, Firefox vs. Internet Explorer, To Legalize Marijuana or not?Should women be allowed to have abortions? .

Graph Construction:

The data is preprocessed for the graph by checking the 'nan' values as python can't take 'nan' values so should convert to 'None'. In the dataset we have DiscussionId, InitialAuthorId, PostIds, AuthorIds, title.

To construct the posts' graph the logic as follows:

First of all the initial post (inital_post_id) by combining the initiating author ID and the title of the discussion. Then connects the author IDs to either the initial post or the parent post. For each post, if a parent post ID exists, an edge is added between the parent post and the current post. Otherwise, an edge is added between the initial post and the current post is added.



All the data here itself converts to tensor data as we have to give this input for the GCN model.Later Adjacency matrix will be calculated. Here all the titles and labels,training -testing-validation ranges all will be mentioned and printed for the future purpose.

Features Input: All the input in the dataframes.the data will be preprocessed by applying stopwords, punkt, wordnet etc. Applying Regular expressions, Lemmatization. After that we have to convert the data into vectors for this we have used three different Base-line models in order to increase the accuracy i.e TF-IDF, Bert-Based, Roberta-Base. We compared the three models and verified how the accuracy of training-testing-validation varies in between these models.Among three models TF-IDF has given high accuracy as it works efficiently for the least amount of data also.

The output from the graph and from the features will be given as the input to the GCN model.Now the GCN model works for the prediction according to our input parameters and conditions we have given .The main function holds all these classes to run and interact with each class.

```
Features: tensor([[0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
         0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
         0.0000, 0.0000, 0.0000, 0.0000, 0.4723, 0.0000, 0.0000, 0.0000, 0.0000,
         0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
         0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.5473,
         0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.3666, 0.0000,
         0.0000, 0.0000, 0.0000, 0.4141, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
         0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
         0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
         0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
         0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.4141,
         0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
         0.0000, 0.0000, 0.0000],
|
```

CHAPTER-07

RESULT

The results of our experiments demonstrate the effectiveness of our proposed GCN model for propagating stance in supervised data. Out of three datasets we had worked on the ConvinceMe dataset. In that we worked on 7 discussions.

One interesting observation from our experiments is that the performance of our algorithm was affected by the nature of the dataset. Main reason to choose the ConvinceMe dataset is it consists of more discussions and it gives more accuracy. This suggests that the complexity and diversity of the data can pose challenges for stance propagation algorithms, and that further research is needed to develop more robust and adaptive approaches.

Accuracy for the individual discussions are below:

- Accuracy of All Discussions

Note: click the above links to redirect to data sheets

MODEL/Baseline	Abortion	Gay Rights	Marijuana	Existence of God	Evolution vs Creation	Pro-Choice vs. Pro-Life'	Firefox vs Internet Explorer	Average
GCN/tfidf	0.8235	0.7089	0.7654	0.5133	0.6197	0.4412	0.7746	0.6638
GCN/Roberta	0.7059	0.7101	0.7654	0.5113	0.6197	0.4412	0.7979	0.650214
GCN/Bert	0.6471	0.7101	0.7654	0.5113	0.6197	0.4412	0.7979	0.641814
Average								0.651942

Average accuracy on posts' stance classification of ConvinceMe discussion

CHAPTER-08

LIMITATIONS

The following are the Limitations:

- The quality of the training data. The accuracy of the GCN model will depend on the quality of the training data. If the training data is not representative of the real world, the model may not be able to generalize well to new situations. For example, if the training data is mostly composed of conversations about politics, the model may be more accurate at predicting stance in political conversations than in other types of conversations.
- The size of the training data. The GCN model may require a large amount of training data to learn the complex relationships between users in conversations. If the training data is not large enough, the model may not be able to accurately predict stance.
- The complexity of the conversations. The GCN model may not be able to handle conversations that are very long or complex. If the conversations are too complex, the model may not be able to learn the relationships between users accurately.
- The nuances of human language. The GCN model may not be able to capture all of the nuances of human language. This could lead to errors in prediction, especially for conversations that involve sarcasm, irony, or other forms of figurative language.

In addition to these limitations, there are a few other challenges that need to be addressed in order to improve the accuracy and reliability of the GCN model. These challenges include:

The development of better techniques for extracting features from conversation data. The GCN model relies on the features extracted from the conversation data. If the features are not representative of the actual stance of the users, then the model may not be able to accurately predict stance. For example, if the features only consider the words used in the conversation, the model may not be able to capture the nuances of human language, such as sarcasm or irony.

The development of more efficient algorithms for training and evaluating GCN models. The training and evaluation of GCN models can be computationally expensive. The development of more efficient algorithms will make it possible to train and evaluate GCN models on larger datasets, which will improve the accuracy of the models

CHAPTER-09

CONCLUSION

In this project, we proposed a novel approach for predicting user stance in conversations using Graph Convolutional Networks (GCN). Our approach transforms the conversation data into a graph representation, which allows us to effectively incorporate the power of GCN to learn the complex relationships between users. We extract relevant features from the data and train and evaluate models using labeled data. Our experiments show that our approach can achieve an **F1 score of 0.84** in predicting stance.

Our work demonstrates the remarkable capabilities of GCN in capturing the intricate relationships and dynamics within conversations to accurately predict user stances. The outcomes of our study will shed light on the potential of GCN-based models in understanding and analyzing user interactions, fostering a deeper comprehension of the underlying dynamics in conversational data.

We believe that our work has made a significant contribution to the field of natural language processing. Our approach can be used to improve the design of social media platforms, to develop more effective spam filters, and to better understand the spread of misinformation.

We are excited to see how our approach can be used to solve real-world problems in the future.

CHAPTER-10

FUTURE SCOPE

We believe that our work has made a significant contribution to the field of natural language processing. Our approach can be used to improve the design of social media platforms, to develop more effective spam filters, and to better understand the spread of misinformation.

In the future, we plan to explore the following directions:

- **Improving the accuracy of the stance prediction model**. We plan to explore different methods for extracting features from the conversation data, as well as different training and evaluation techniques.
- **Extending the scope of the stance prediction model**. We plan to extend our approach to predict stance in a wider range of conversations, including those that involve sarcasm, irony, and other forms of figurative language.
- **Applying the stance prediction model to other tasks.** We plan to apply our stance prediction model to other tasks, such as identifying toxic users and detecting fake news.
- **Developing a user-facing application.** This could be a web app or a mobile app that allows users to input a conversation and get a prediction of the stance of each user.
- **Message parsing:** Message passing technique is to come up with an optimal node embedding iteratively which captures the context and neighborhood information.

Overall, this project provides a promising approach for stance detection on unsupervised data using the heat diffusion algorithm. By further exploring the use of different threshold values and experimenting with the integration of other graph-based techniques, we can enhance the performance of the proposed method and apply it to other domains and languages.

CHAPTER-11

REFERENCE

1. *STEM:* Unsupervised Structural EMbedding for Stance Detection Ron Korenblum Pick1 , Vladyslav Kozhukhov2 , Dan Vilenchik2 , Oren Tsur1 1Department of Software and Information Science Engineering 2Department of Communication Systems Engineering Ben Gurion University of the Negev {ronpi,kozhukho}@post.bgu.ac.il, {vilenchi, orentsur}@bgu.ac.il .

2. *MGTAB:* A Multi-Relational Graph-Based Twitter Account Detection Benchmark Shuhao Shi1,† , Kai Qiao1,† , Jian Chen1 Shuai Yang1 , Jie Yang1 , Baojie Song1 , Linyuan Wang1 , Bin Yan1,* 1Henan Key Laboratory of Imaging and Intelligence Processing, PLA strategy support force information engineering university'.

3. Integrating Transformers and Knowledge Graphs for Twitter Stance Detection Thomas Hikaru Clark♠, Costanza Conforti♠, Fangyu Liu♠, Zaiqiao Meng♠, Ehsan Shareghi♣♠, Nigel Collier♠ ♠Language Technology Lab, University of Cambridge ♣Department of Data Science and AI, Monash University ♠{thc44, cc918, fl399, zm324, nhc30}@cam.ac.uk ehsan.shareghi@monash.edu