SUICIDAL IDEATION DETECTION USING TEXT GRAPH CONVOLUTIONAL NETWORK

SUNIL SHARANAPPA

THESIS REPORT

Liverpool John Moores University – Masters in AI & ML

DECEMBER-2021

DEDICATION

Successful individuals aren't necessarily born with it, they work hard and achieve their goals on purpose. This work is dedicated to everyone (family &friends) who has been so kind & understanding.

ACKNOWLEDGEMENT

This thesis was written as part of Liverpool John Moores University's Master of Science in Artificial Intelligence and Machine Learning programme. I'd like to express my gratitude to Mr. Bharath Kumar Bolla, my thesis supervisor, for his critical criticism, technical assistance, and unwavering encouragement during the study time.

ABSTRACT

Now a days lot of progress has been done in Natural language processing field, which gives close to human intelligence performance for different NLP tasks which are like text Summarisation, Unsupervised text classification, Entity recognition and so on. Using NLP for analysing of online social media platform data is a promising way to capture influences which are contributing to individual risk for suicidal ideations and behaviours. The main challenge in suicide prevention is knowing and detecting the intricate risk factors and warning indications that may give raise to the unexpected event. In our thesis, we like to introduce a novel approach that make uses of the social media platform Reddit to detect posts containing suicide related content. In our work, we will built our models with text GCN (Yao et al., 2018b).and we did comparative sturdy by varying the volume of train data and tuning parameters and we also compared results against traditional models.

The graph built can perform relations between word-word and word-document together. Unlike traditional methods, which need word embeddings. We also compared the results varying the windows size and embedding dimensions while constructing the graph. Overall, from our thesis we found that Text GCN can perform better even when we have less training data and carries more contextual data between nodes compared to traditional models. We have evaluated the outcome based on various evaluation methods likes accuracy, F1, precision, and recall against TGCN model.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATIONS

Glove	Global vectors for word representation
GCN	Graph convolution network
GAN	Generative adversarial network
DFS	Depth-first search
BFS	Breadth-first search
Bert	Bidirectional Encoder Representations Transformers
BOW	Bag of Words
TF-IDF	Term Frequency – Inverse Document Frequency
CoVe	Contextualized Word Vectors
PCA	Principal Component analysis
iNLTK	Natural Language Toolkit for Indic Languages
CBOW	Continuous Bag of Words
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
WHO	World Health Organization
PMI	Pointwise mutual information

CHAPTER 1

INTRODUCTION

1.1 Background of the study

Suicide is one of the most serious health issues of our day. According to the World Health Organization (WHO), suicide is the fourth leading cause of mortality among young people (15 to 29 years) (Source: Suicide, n.d.). Suicide rates have risen in this epidemic, owing to people's high levels of worry about their jobs and families. Personal issues, such as worry, despair, impulsivity, or circumstances, such as social seclusion and unpleasant life experiences, as well as previous suicide attempts, can all drive someone to adopt extreme measures.

Millions of peoples in the world are victims to suicide each year, becoming the prevention is an important world-wide community health goal. There are increasing proofs that the Cyberspace and social media can affect suicide connected behavior (social media and Suicide - Wikipedia, n.d.). Suicidal Detection indicates whether someone has suicidal ideation or thoughts by identifying the textual content which are created by the individual. Since the increase in social media platforms & anonymity, lot of people interact with other people on the Internet. Individuals may now use online media to express their pain, sentiments, and suicide intentions. As a result, online media have established themselves as a screening tool for suicidal ideation, as well as mining social platform material to improve suicide prevention(Lopez-Castroman et al., 2020) & the society's mental health.

Very bizarre phenomena are rising, with online communities coming to an accord on self-harm and suicide. For example, in 2015-2016, a social platform group named "Blue Whale Game" employed different self-wounding activities to encourage participants to take the extreme step of committing suicide at the end of the game. Suicide is a serious societal problem that claims thousands of lives each year. As a result, detecting suicidal thoughts and preventing suicide before victims commit themselves is critical. Victims with suicidal thoughts may express their ideas of their extreme measures in a brief suicide plan, thoughts, and also sort of roleplays, and early detection and therapy are considered as effective strategies to avoid suicide attempts.

Recognizing suicidal ideas entails looking for possible motives or circumstances before an unforeseen catastrophe occurs. Ideas as a screening tool have limits, according to a study (McHugh et al., 2019), but individual manifestation of suicidal thoughts indicates psychological distress. Early detection of suicidal thoughts can help identify people and create

a consultation portal where social workers or family members can help them overcome their mental health concerns.

Social platforms with its psychological health groups have become a new study area in Natural language processing. It gives a good research platform for the growth of new methods and improvements which may bring about a new dimension in suicide recognition and further suicide probability inhibition (Marks, 2019 & De Choudhury et al., 2016) researched the change from a mental health issue to suicide thoughts in Reddit platform. He created a propensity score based statistical methods to derive the distinctive markers of this shift.

Traditional Social media post classification techniques involve extracting features from the text corpus followed by is a classifier to generate predictions and they are all computationally intensive. The most critical issue with tradition models is with the volume of labeled data. Accuracy of the traditional models gets better with the huge volume of train data but that comes with huge computational overhead, Whereas Text GCN will give better accuracy with a smaller number of labeled data. Deep learning advances have piqued researchers' interest in detecting suicidal thoughts. For suicidal text depiction learning, more innovative approaches such as graph neural networks and attention mechanisms can be employed. There are a few different strategies that may be utilized, such as reinforcement, adversarial, and transfer. Knowledge of mental wellness detection, for example, may be applied to the detection of suicidal thoughts, and GAN can be used to produce samples for data augmentation.

The major objective of our study is to use a Deep learning model to communicate information about suicidal thoughts on Reddit and Twitter (microblogging) social media platforms. The main focus of our research is on the Graph Convolution Network (GCN) with various pre-trained embeddings. Techniques. Graph Convolution Neural Networks (GCNs), which are intended to function on irregularly structured graphs, have recently gained traction in a number of fields. Although Social media posts represent a sequence of words, a document contains an implicit graph structure in the form of syntactic and semantic relationships (Mihalcea & Tarau, 2004) GCN is not explored on social media posts. A text corpus can be arranged into graph by making use of both intra-document and inter-document relationships (Yao et al., 2019). Graph from the corpus (Huang et al., 2019) creates graph on per each document basis. We will build Text GCN model for suicidal ideation prediction. Which can be made as part of any online forums and blog for early detection of suicidal ideation.

1.2 Problem Statement

One of the most important suicide warning signs is a sudden shift in behavior. If a person's conduct has altered fast, especially if it is linked to a painful incident, loss, or major life change, the likelihood of suicide is higher. Considering this in the context of social media platforms, where individuals routinely post messages and consciously convey their emotions. Despite the importance of this application field and the fact that suicide is one of the leading causes of death worldwide, there is a paucity of published research on text categorization for suicidal thoughts. Some of the factors leading to this underrepresentation include a lack of freely available labelled datasets and the difficulty of extracting someone's post or message due to data privacy policies established by social networking sites.

There are several approaches to solving the text classification problem. Traditional research focuses largely on feature engineering. They create well-designed characteristics via feature engineering, such as the bag-of-words, ngrams, and graph features (Luo et al., 2016). However, these techniques need manually creating text representations, making training more difficult. Deep learning approaches are also frequently employed in text classification problems, in addition to conventional methods.

1.3 Aim and Objective

We will use graph neural network (Yao et al., 2018a) to train models using suicidal datasets from social media (Reddit), where each word or document will be represented as a word vector for Text GCN input. The numerical representation of text data is called a word vector. We'll use a variety of pre-trained embeddings. On the built graph, we will additionally modify the dense layers and dropouts in Text GCN using both words (from vocabulary) and documents.

There are three goals in our thesis:

- To evaluate Text GCN on social media corpus.
- To assess effect of the size of the target data with different sizes of the training data against traditional models.
- To evaluate the results by varying window size, embedding dimensions, dense layers, drop out and learning rates and compare the results with traditional classification models.

1.4 Research Question

- What are the common emotional words in the people who has got suicidal ideations?
- Does Text GCN better than traditional classifications models on social media posts?
- How does Text GCN evaluate on test set compared to traditional models when the size of trained data varies from 80%, 60%, 50% , 30% and 20%?

1.5 Significance of the Study

Recognizing the important factor and it is the first step in preventing suicide. However, because of the cultural stigma associated with mental illness, some people may be hesitant to seek professional treatment (Corrigan, 2004). They may become less dependent on fewer formalized support services (Rickwood et al., 2005).

In recent years, online social media platforms have proven to be an informal research source. People who are at danger are turning to modern-day technology (social media, online forums, microblogging sites) to express their distress without having to confront someone one-on-one, according to research (Moreno et al., 2011). Risk factors and warning signs have been seen in these online places because of this we believe that large amounts of data on people's emotions and behaviors may be utilized to quickly identify behavioral differences in those at danger, and to further prevent tragedies.

The main contributions of our thesis are twofold. Firstly, empirically compare the Text Graph Convolutional Network (Text GCN) solution with other traditional models. Text GCN was initially developed for news corpus classifications. To our understanding this is the first illustration of a GCN being used for classification on any social media dataset. Secondly Focus on the semi-supervised situation where the number of labels is highly limited, and the compute is restricted.

1.6 Scope of the Study

The scope of study involves predicting Suicidal Ideation based on Reddit and Twitter posts which are already labeled. Apart data subtext, characters, habits, emotions etc., few individuals may not want to express on social platforms, Especially, when they are immersed in loneliness, misery and in suicidal thoughts, Individuals tend to be isolated and have very low motivations to create content or leave some info on social platform.

Due to rapid changes in emotional fluxes, individuals having suicidal motives tend to remove their old posts related to suicidal thoughts, attempting to hide the true innermost intentions.

However, the study only focuses on Suicidal Ideation but does not focus on other mental health issues which can also be part of corpus and wrongly labeled either suicidal or non-suicidal.

1.7 Structure of the Study

This section discusses the thesis report's overall structure. In Chapter 1, we introduced the topic and established the research's goal, purpose, and inspiration. The research questions that the research article attempts to address have been cited. We also highlighted the importance of the study, as well as the ramifications of this work on other GNN related tasks, before concluding this chapter with the study's scope.

In chapter 2, we have covered the literature review done for the thesis. We started with introduction to graph and discussing various types of graphs. We also covered the structure of graph and how data is saved on graph. Further, we discussed about using convolution in graph and the drawback of it and how we can overcome using the graph convolution network. We will also see an example of including Adjacency Matrix (A) in the forward pass equation of GCNs, the model is able to learn the characteristics of nearby nodes. This method may be thought of as a message relaying system for the graph's nodes. We also discussed about embedding and we discussed Deep walk in details. We also covered text GCN and how document embeddings are done and Text GCN implementation on popular datasets.

In our chapter 3 The Research methodology is covered in depth. We reviewed data selection, pre-processing stages in the first part. The technical specifics and design of the different embeddings that we are utilizing in the research were detailed later in this chapter. At last, in this section, we also see how evaluation metrics will be carried out.

CHAPTER 2

LITERATURE SURVEY

The background and associated research that was conducted as part of the study are presented in this chapter. The many forms of GNNs are discussed, as well as an example of GNN with sample data, we will also discuss GCN which is a variant of GNN. The literature evaluation and limitations of previous work are also discussed in this chapter, which serves as the foundation for the study being conducted in this paper.

2.1 Introduction

The chapter ‘Literature survey' discusses the background and associated articles of GNN & GCN. This chapter is broken into sub-sections that cover the various components of the research in order. First section talks about introduction to graph. Where we will see what graph structure and the usage of graph in real world is. In the next section, we discuss the how the data is stored in the graph and challenges in storing the data in the conventional graph.

Further, we learn about the convolution in graph and graph representation using convolution and generalizing the graph on convolution and challenges with it. Next sub section we study how traditional convolution can be converted into graph convolution to address the drawback of convolution on the graph data. We will also see an example on how feature importance is extracted by normalizing the feature nodes also we will see how the graph data can pass through 2 hidden dense layers with activation function set to Relu.We will further see how effective GCN with just one forward propagation of graph data through dense layer is can already cluster the nodes.

Then in the later part of subsections we will learn about how the message are being passed from one node to another node along the edges. We will also see how the deep walk embedding technique is used and how the similarity in the original and embedding space are being calculated. Then in the last section we will see how the text GCN is evolved from GCN and how the document and word embedding happens. Further we see popular graph datasets and their outcome when run using different graph techniques.

2.2 An Overview of Graphs

Real-world objects are frequently characterised in terms of their connections to other things therefore, graphs exist all around us. A graph is a natural representation of a collection of things and their connections. For more than a decade, researchers have been developing neural

networks that operate on graph data known as graph neural networks, or GNNs (Scarselli et al., 2008).

Recent advancements have expanded their expressiveness and capability. Graphs are a general language for describing and modelling complex things. Graph structure data usage is found everywhere around us like social network, antibacterial discoveries(Stokes et al., 2020), financial transactions, traffic predictions, recommendation systems, program graphs, scene graphs and Google's Knowledge Graph helps in Search Engine Optimization (SEO) as shown in the below figure.

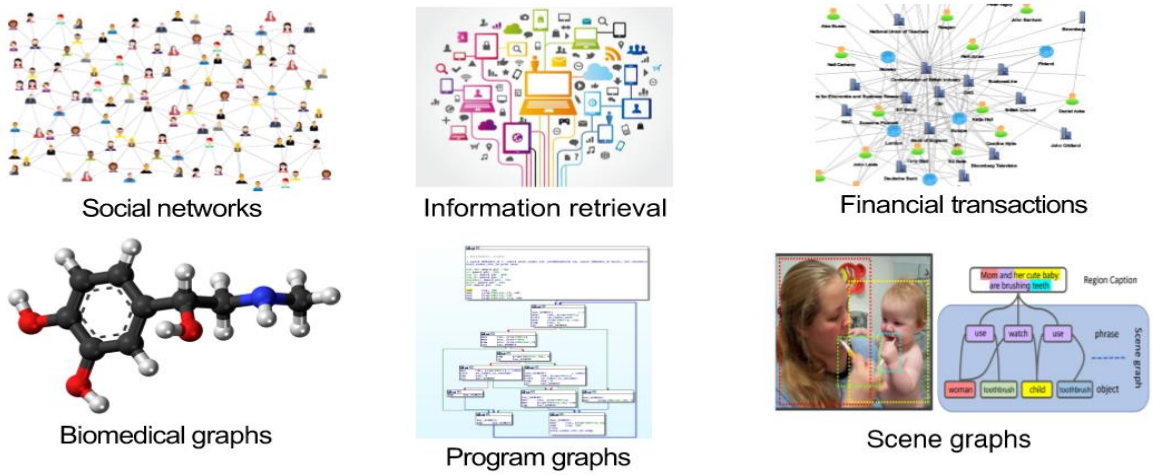


Figure 2.1 Graph usages in real world

2.2.1 Types of graphs

Graphs are broadly divided into below two categories. In our thesis we will mainly focus on spectral convolution network type.

- **Spatial Convolutional Network**: Spatial Convolution analyses a node's local neighbourhood and determines its characteristics based on its k local neighbours. GraphSage is one of the examples for Spatial Convolution (Graham, 2014)
- **Spectral Convolutional Network**: We conduct an Eigen decomposition of the graph's Laplacian Matrix in spectral graph convolution. This Eigen decomposition aids in the comprehension of the graph's underlying structure, allowing us to discover clusters and sub-groups. PCA. ChebNet, GCN are some of the examples that use Spectral Convolution (Rippel et al., 2015).

2.3 Storing Graph Information

Graph G represented as G=(V,E,u) is an unordered collection of vertices (a synonym for nodes) and an unordered collection of edges. A graph depicts the connections (edges) between various

items (nodes). Graph generally consists of set of nodes (representing objects), set of edges (representing their relations/interactions) and feature vector.

Figure 2.2 Graph Representation

2.3.1 Representation and Types of Graphs

An edge ***e*** has a source node v_{src} and a destination node v_{dst} in directed edge, data is passed from v_{src} to v_{dst}. They can also be undirected, meaning there are no source or destination nodes and data flows in both ways. In undirected graph, the relation between the nodes is symmetric.

Figure 2.3 Directed and Undirected edge

An example of directed Graph in the social media network is a Twitter where if the first person follows the second person, the second person does not need to follow back or follow the first person. On the other hand, an example of undirected graph is Facebook in Facebook, if two people are friends, they're friends and there's no notation of direction of friendship.

To mathematically represents graph data, we could use the notation G to describe all the information. Because graphs are made up of nodes and edges, we'll need to find out how to represent them as arrays. There is also a type of graph where each node is connected to all other nodes, this is known as complete graph as shown in the below figure.

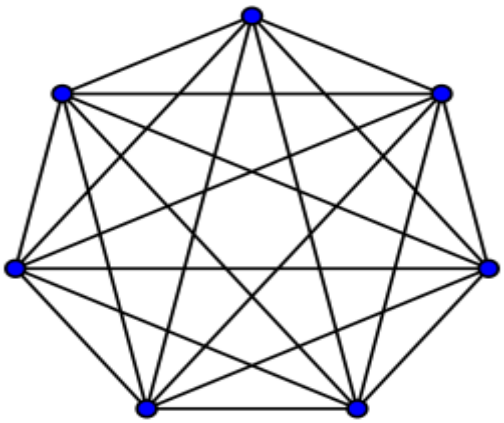
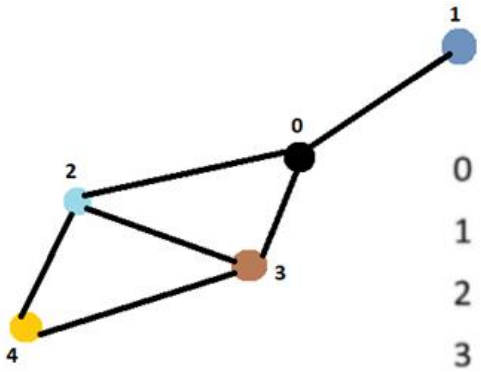


Figure 2.4 Complete Graph

2.3.2 Matrices Required to Create Graph

An adjacency matrix is a N x N matrix with either 0 or 1 as the row and column values, where N is the total number of nodes (Singh & Sharma, 2012). Adjacency matrices can express the existence of edges that connect node pairs by using the matrices' values. For example, if our network has 5 nodes, the matrix will have the structure [5, 5]. If an edge exists between nodes i and j, the matrix element Aij is 1. We can see in the adjacency matrix above in the below figure that the link between nodes 2 and 3 (A23) is represented as 1 in the matrix since they are connected, but A21 is not represented as 1 since they are not connected.



Adjacency Matrix: A =

	0	1	2	3	4
0	0	1	1	1	0
1	1	0	0	0	0
2	1	0	0	1	1
3	1	0	1	0	1
4	0	0	1	1	0

Figure 2.5 Illustration of Adjacency Matrix

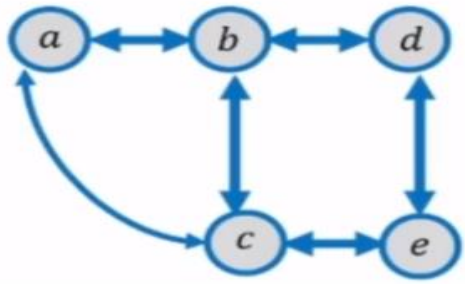
We could also have numbers(weight) instead of 1 and 0 which is called wight matrix. We could replace this binary representation with a different float number, instead of the values that are set as one, and in this case, we are defining the Matrix as such that all those zeros still show no connection between the nodes(Dehmamy et al., 2019).

Weight Matrix: W =

	0	1	2	3	4
0	0	2	1.5	4	0
1	5	0	0	0	0
2	1.5	0	0	1	1
3	12	0	1	0	1
4	0	0	1	1	0

Figure 2.6 Illustration of Weight Matrix

Degree Matrix is a diagonal matrix which shows the number of neighbours of each node. Degree of Matrix shows influence of each node on the whole graph (Cao et al., 2015). From the below figure we see the node B and C have a higher influence because they're connected to more nodes.



$$D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

Figure 2.7 Illustration of Degree Matrix

Laplacian of graph is the difference between the degree matrix and the adjacent matrix or in the case that we use weight matrix instead of adjacency then Laplacian graph will be the difference between D and W, which is degree and the weight of the matrix (Shaw & Jebara, 2009).

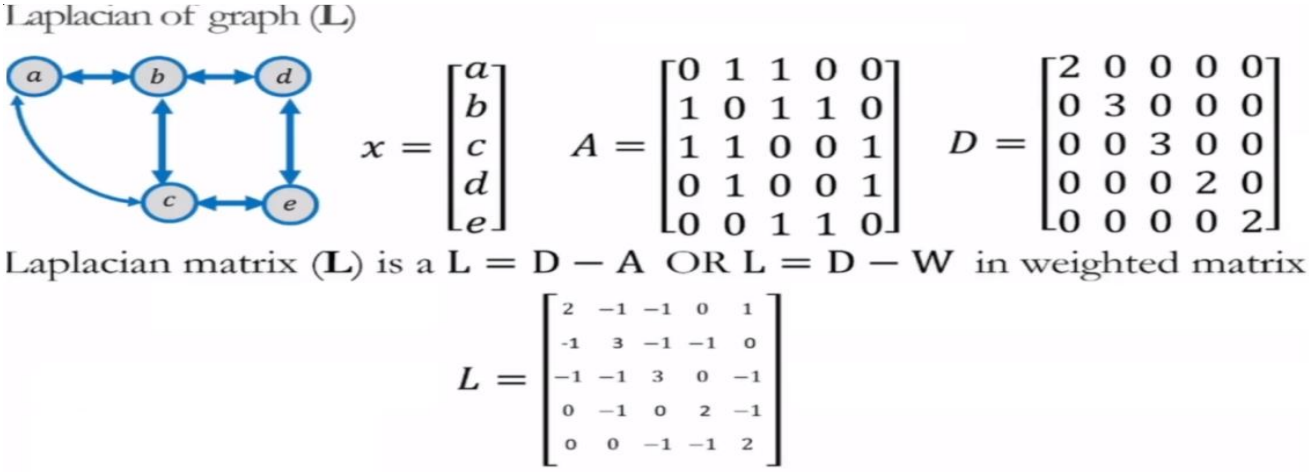


Figure 2.8 Illustration of laplacian Matrix

2.3.3 Graph Representation in Embedding Space

As we seen, graph is relatively complex representation. The benefit of this complex representation is that we could include as much as detail possible, which makes our understanding and analysis much better. However, on the other hand, this complex representation is very difficult to analyse or understand. Therefore, one common way that learning start to be introduced into the graph representation is to simplify the model. The goal is to preserve as much as information that is possible but have a lower dimension and less complex representation. In this case, most of the time, what you will see is that every node of a graph will be transformed into a new dimension. This is referred to as Latin representation or embedding space (Goyal & Ferrara, 2018).

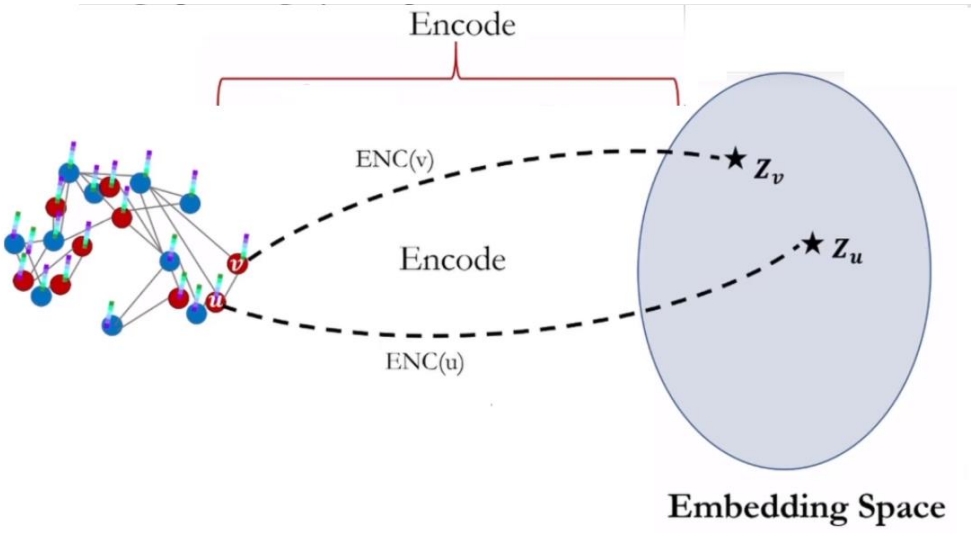


Figure 2.9 Mapping Graph nodes on Embedding space

We will embed the graph into a different space where u and v from the above figure will become Z_v & Z_u, this process is referred as encoding. We have to introduce an encoder that encodes each node into an embedding space. The goal of the encoder is to preserve the similarity between the nodes. This means that we want the similarity between U and V to be similar to the similarity between Z_u and Z_v. let's represent the similarity in the original space as S_G and the similarity in the embedding space as S_E so we want to S_G and S_E to be similar for the nodes in the original space and the nodes in the embedding space.

2.3.4 Graph Encoder and Decoder Technique

Taking a graph data from its original space to the embedding space is called encoder. Decoder is a function that describes the similarity between the Data nodes in an embedding space. The output of the decoder for every two nodes will be a number, which is a positive number, and it will show how similar to nodes are in an embedding space.

We want to see S_G in original space to be similar to the S_E in an embedding space. One common way to find S_G and S_E similarity is to obtain a Euclidean distance, a difference between the S_G and S_E values so the goal, therefore, will be to minimize this cost function.

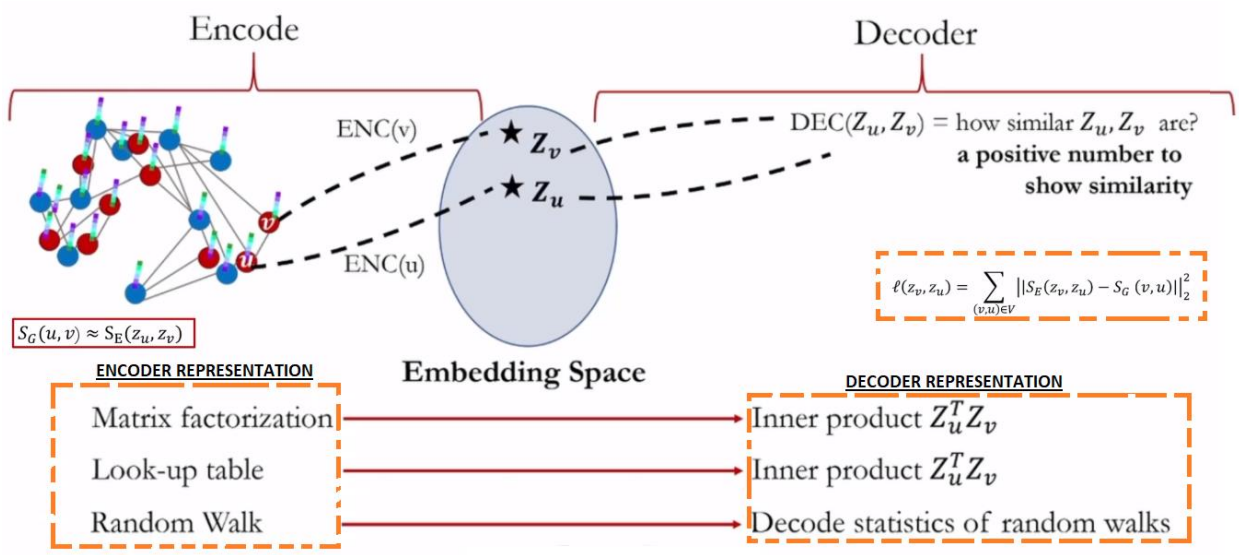


Figure 2.10 Representation of encoder and decoder on graph data

We are looking for an encoder that could minimize the loss function. The Matrix Factorization uses the inner product, and it tries to minimize the inner product between the two nodes in the embedding space, a lookup table, which is another encoding methods, does the same thing by finding the inner product between the two nodes in the embedding space, random walk use a decode statistic of random walks in order to find or obtain the loss function. The representation

of the encoder is shown on the left side, which are the matrix factorization lookup table and random walk and the representation of the decoder, which is a similarity of the two embedding space nodes, is an inner product for matrix factorization and lookup table and decode statistic for random walk.

We introduce an encoder and decoder and their definitions we mentioned that our goal is to minimize the differences within the similarity of the original space and an embedding space but there are three main drawbacks with the methods that was discussed in the previous section.

- First, there's no parameter sharing means that the process will be computationally expensive and because most of the times Graphs are used for huge data representation.
- Secondly, there's no semantic information consider this means that integrating features on the nodes will be difficult in the encoder.
- Lastly, the existing encoder decoder definitions are not inductive. This means that we cannot predict the outcome for an unseen data.

To overcome above drawbacks, we use convolution in a graph representation, which is known as Graph Neural Network.

2.4 Introduction to Convolution

A convolutional neural network (CNN) is a form of artificial neural network that is especially intended to analyse pixel input and is used in image recognition and processing(Kim, 2017). Convolutional neural networks (CNNs) are neural networks with one or more convolutional layers that are primarily utilised for image processing, classification, segmentation, and other auto-correlated data. Convolution is the process of sliding a filter across an input signal.

2.4.1 Overview of Convolutional Neural Networks (CNN)

Before venturing into graph convolution network let’s understand workings of convolution network. In the above Let's consider grey scale image, which includes binary values of zero or one for different pixel values. To apply the convolution, let's define a kernel which is the size of 3X3.Our goal is to apply the process of convolution using the kernel on this image and get the output. Let’s recall that in the process of learning convolution, we are learning the parameters in the kernel. When we applied the process of convolution as a result, we will get another image. We will apply this kernel to the starting of the image from the top left side, every element of our image will be multiplied by every element of our learning kernel, and then there will be a summation of the values, one will be multiplied by the zero, which is seen at the top left side of our learnable kernel. Then zero will be multiplied by one, then zero will

be multiplied by one again and all the way through all the six pixels for the results. We will add each of the resultant products and we will put it as the value of our image on the right side. Next, we will shift our data with one stride or shift of one to the right and again, the same learnable kernel. Once we do this for all of the data, we will get a resultant image and this image will represent a fusion of our pixels with the learnable kernel.

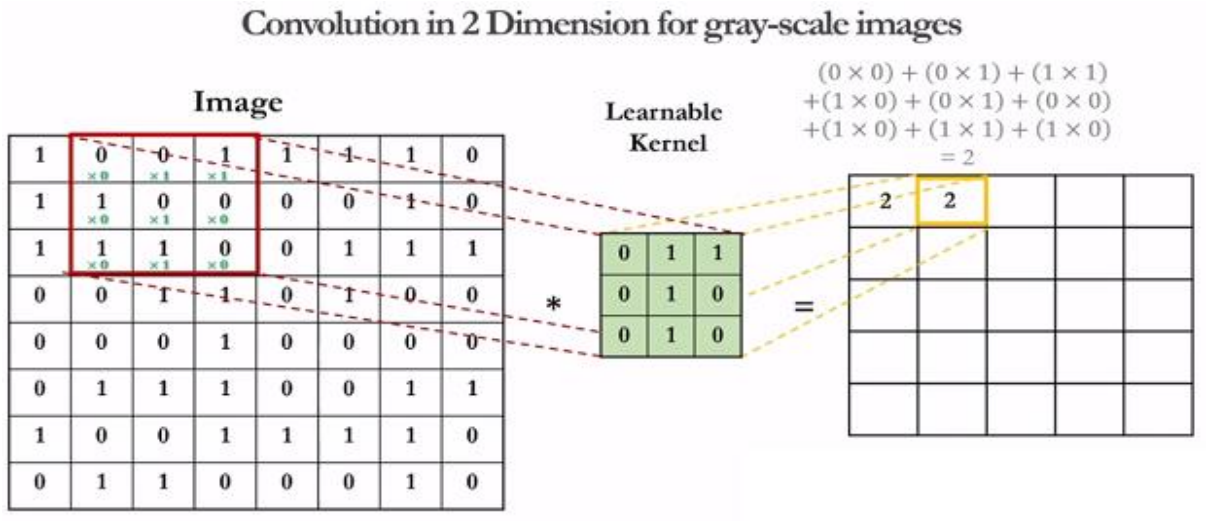


Figure 2.11 Convolution in 2 Dimension for gray-scale image

2.4.2 Benefits of Using Convolutional Neural Networks (CNN)

The benefits of this process of convolution is that we are applying a parameter sharing. As you can see, after applying one stride to the right, the same parameters is being applied to a different pixels and the process is to learn these values of the learnable kernel. This means that for this image, we do not need to find the same size parameters. We just need to find 3X3 parameters in order to obtain a result and image and this is where the parameters are being shared, which means there is less parameters to work with Secondly, the process is translation invariant, which means that rotating an image will not cause harm to our learnable parameters and lastly, the number of the parameters in the learnable kernel is independent of the input. This means that we can have a bigger image by applying padding to get the same results.

2.5 Challenges in Generalizing Convolution for Graph

CNN only works on data with regular structure (Euclidean data), such as pictures (2-dimensional) and text (1-dimensional). In traditional Convolution neural network, we usually deals with images, Images are inherently networks of pixels linked to other pixels, but their structure is always fixed. Convolutional neural network shares weights among neighbour cells depending on few assumptions, such as that a 3X3 pixel area may be considered a

"neighbourhood." Our convolutional neural networks are based on 2-dimensional, regular data also known as Euclidean data.

Our social media networks, chemical structure representations, and map locations are all three-dimensional. They also lack the required size and structure. When trying to pack non-Euclidean or arbitrarily structured data into CNNs, we run into problems since that's when they hit their limit In CNN and cease being helpful.

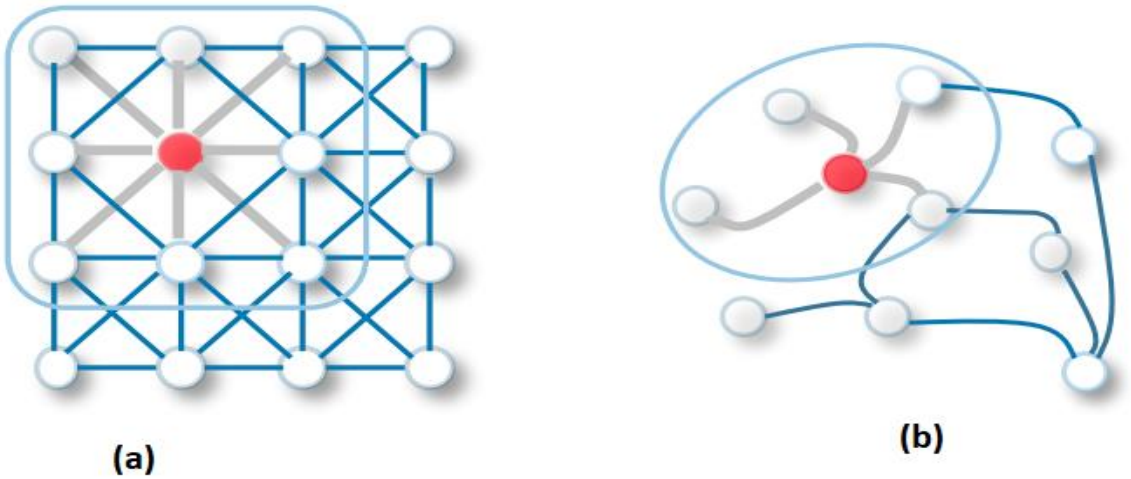


Figure 2.12 Convolution vs. Graph Convolution (Kipf & Welling, 2016)

(a) 2D Convolution. Analogous to a graph, each pixel in an image is taken as a node where neighbours are determined by the filter size. The 2D convolution takes the weighted average of pixel values of the red node along with its neighbours. The neighbours of a node are ordered and have a fixed size.
(b) Graph Convolution. To get a hidden representation of the red node, one simple solution of the graph convolutional operation is to take the average value of the node features of the red node along with its neighbours. Different from image data, the neighbours of a node are unordered and variable in size.

The number of the attributes can vary. This means that in addition to having to pixels, having the edges on the pixels that connect them together or nodes, we can have an additional set of features and this feature could be different for different nodes and different graphs. Another challenge is that, as we mentioned earlier, we may have heterogeneous graph, which means different nodes may have different meanings and they may also carry-on different attributes. Lastly, the node ordering can change, which is known as a Homorphism problem.(Liquiere, n.d.).

2.6 Convolution Neural Network to Graph Convolution Neural Network

In the figure we have adjacency matrix S indicating which is node is connected which node and node feature matrix y indicating all the nodes.

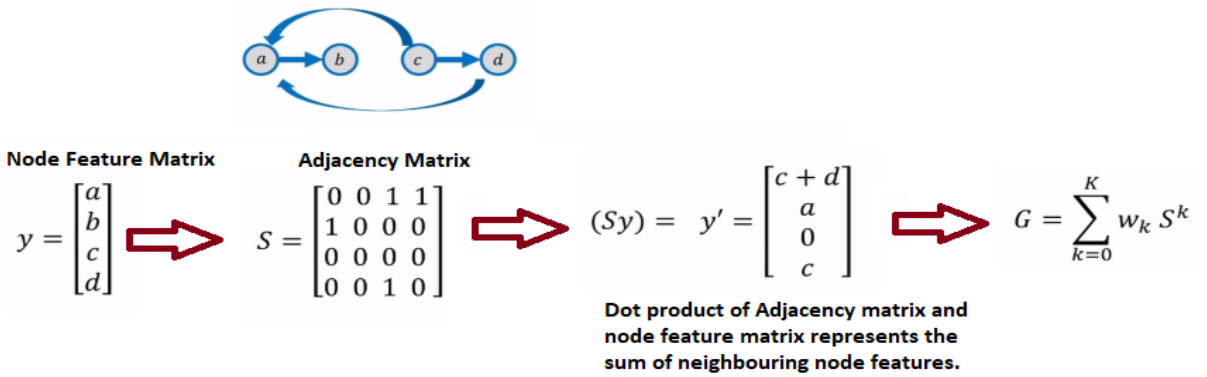


Figure 2.13 Graph Convolution Kernel

By multiplying the adjacency matrix with node feature matrix, we get shifted version of graph So it means what happens if we shift or move the graph based on its existing connections once. The convolution in convolution neural network with the kernels is weighted shift. Therefore, the convolution in graph is multiplying S (adjacency matrix) by different weights. Our graph convolution will be w_s. and we could have a different set of weights for different shifting order. So for the first shift we could have one weight. If we are shifting it twice, we could have a different weight and we can go for a number of shifts. So the W illustrates the weighted and the S illustrate the shift. In the convolution neural network, we were using kernel with the size of nxn and then we were shifting it over our data. In the case of graph, we use adjacency and weight matrix, and we multiply that by our graph to apply this shift process. for the points wise nonlinearity, we can apply to shift and then similar to the convolution neural network, put that results and shift into a function that is nonlinear, such as a sigmoid in to get to the next layer.

The results from the below figure show that AX reflects the total of the characteristics of adjacent nodes. The first row of AX, for example, refers to the total of node characteristics related to node 0, which are nodes 1, 2, and 3. This helps to understand how GCNs propagate hidden features and how node connectivity affects the hidden features representation viewed by GCNs.

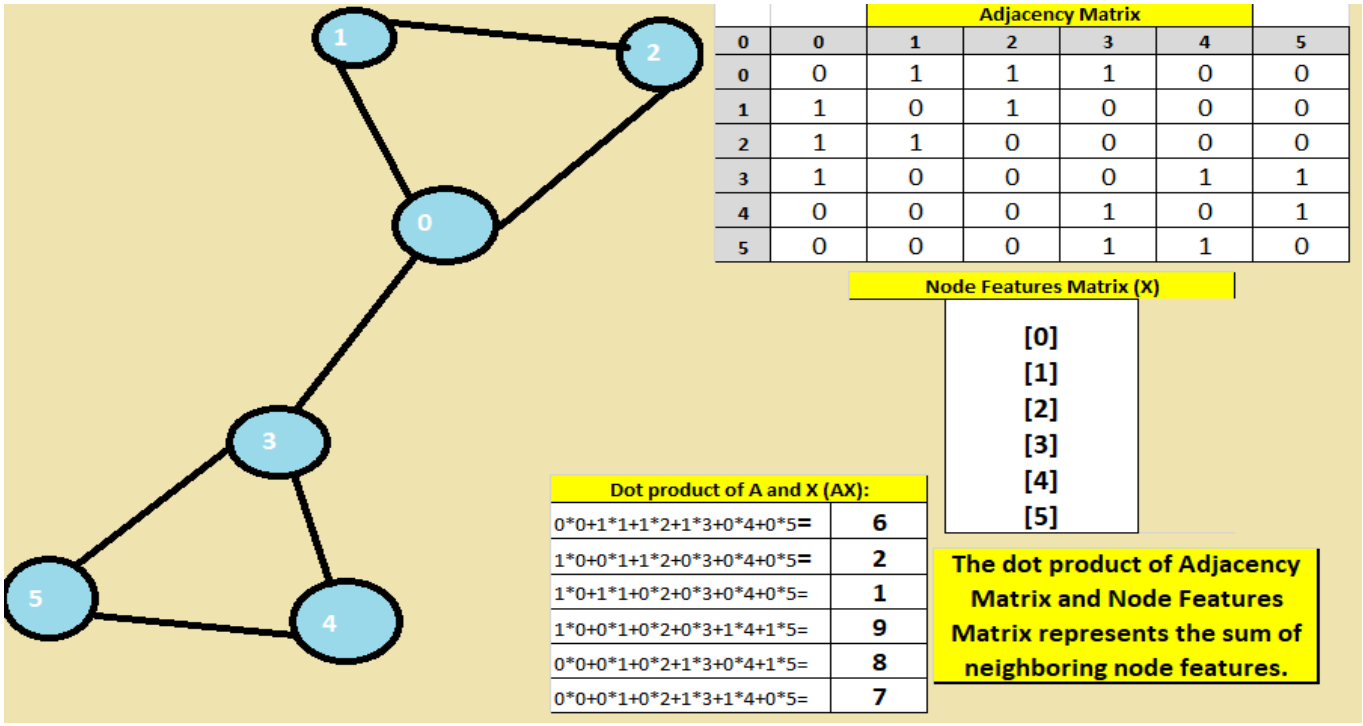


Figure 2.14 Example Showing Significance of Neighbouring node

2.6.1 Adding Self-loops on Each Node

The total of nearby node features is represented by the dot product of Adjacency Matrix and Node Features Matrix. However, if we look about it further, we'll see that while AX adds up the features of nearby nodes, it ignores the node's own features. To solve this problem, **we've** added self-loops to each node of A's. Adding self-loops is essentially a way for a node to link to itself. Because each node is linked to itself, all the diagonal elements of Adjacency Matrix A will now become 1. Let's rename A_hat to reflect the addition of self-loops, and recalculate AX, which is now the dot product of A_hat and X.

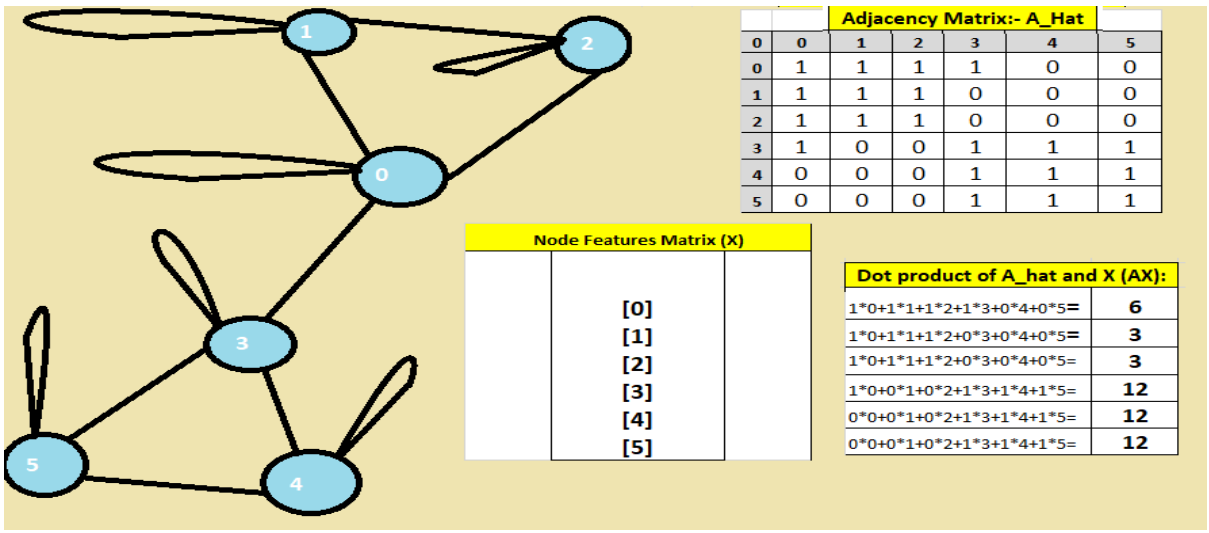


Figure 2.15 Showing Significance of Neighbouring node After Adding Self-Loop

2.6.2 Normalizing the Feature Nodes

You might be able to see another issue now. AX's components are not normalised. In order for the model to converge, we must normalise the features, just as we would for any other Neural Networks operation. This prevents numerical instabilities and vanishing/exploding gradients. We normalise our data in GCNs by computing the Degree Matrix (D) and executing a dot product operation with AX on the inverse of D normalized features $= D^{-1}AX$

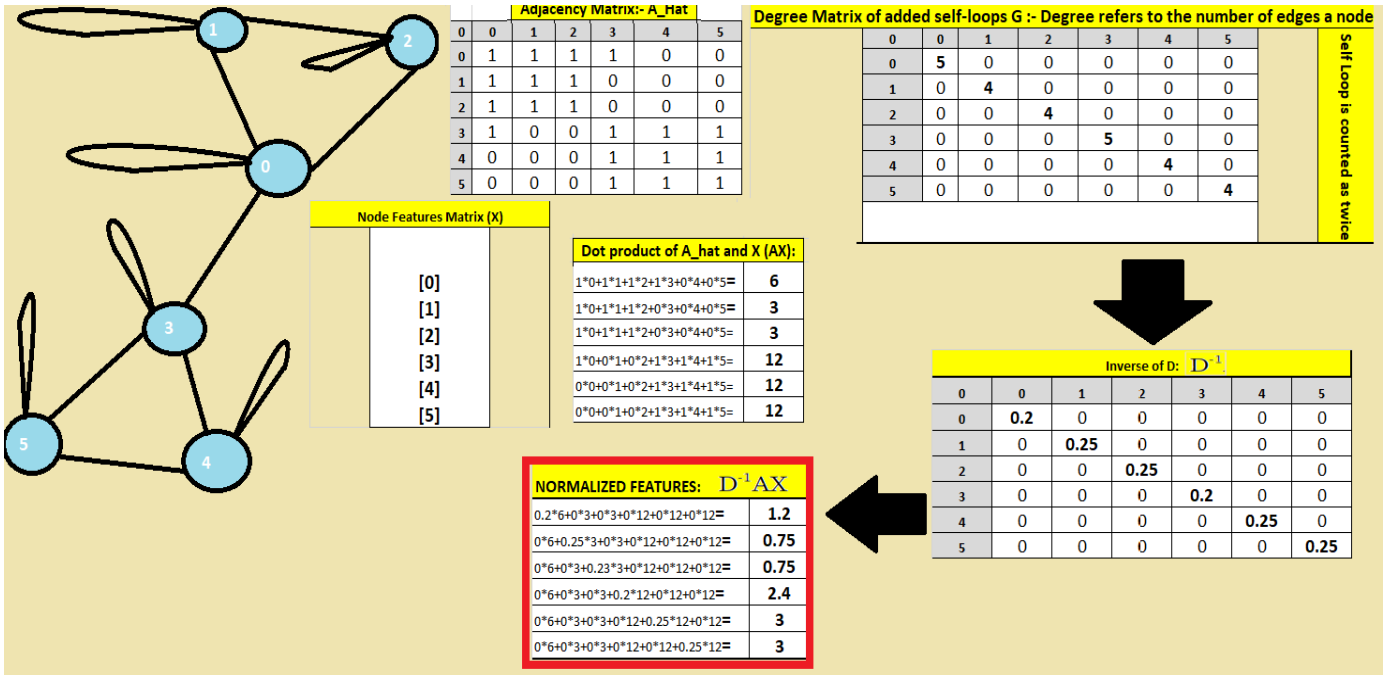


Figure 2.16 Normalizing the Feature Nodes

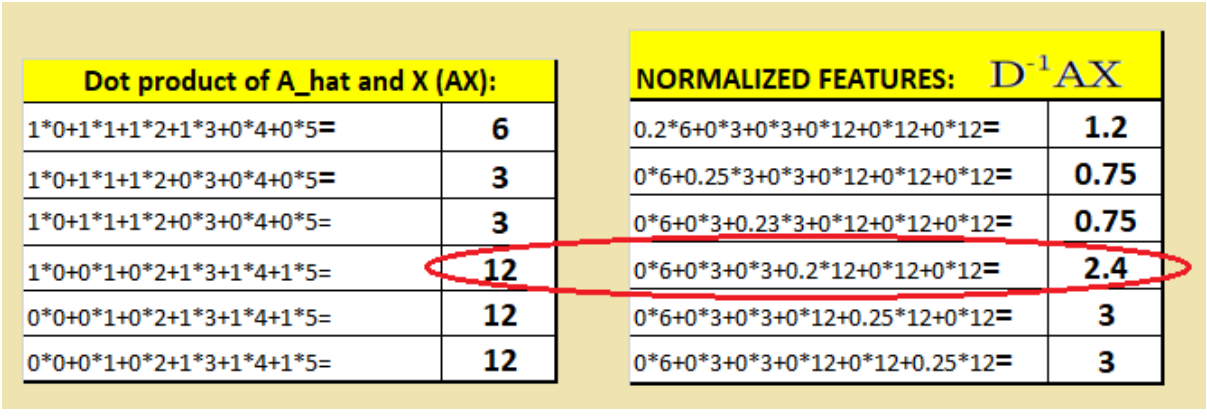


Figure 2.17 Comparing Normalized and Not Normalized Feature Nodes

The influence of normalisation on DAX can be seen in the element that corresponds to node 3, which has lower values than nodes 4 and 5. But, if node 3 has the same starting value as nodes 4 and 5, why would it have different values after normalisation?

Let's take a look at our graph again. Node 3 has three incident edges, whereas nodes 4 and 5 only have two. Because node 3 has a greater degree than nodes 4 and 5, its characteristics are given less weight in DAX. In other words, the lower a node's degree, the more likely it is that it belongs to a certain group or cluster. Symmetric normalisation makes dynamics more interesting, thus the normalisation equation is changed (Kipf & Welling, 2016). From normalized features $= D^{-1}A$ to normalizing term $= D^{-1/2}\,AD^{-1/2}$.

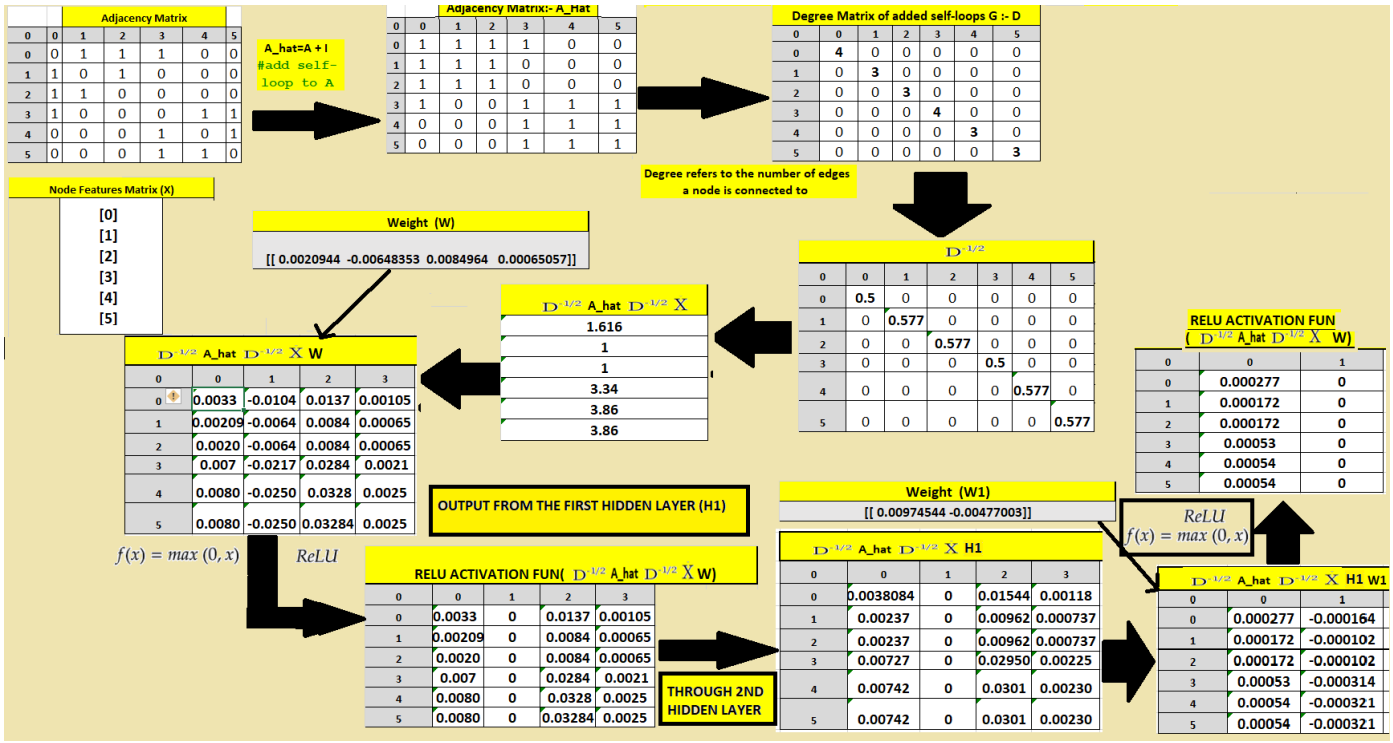
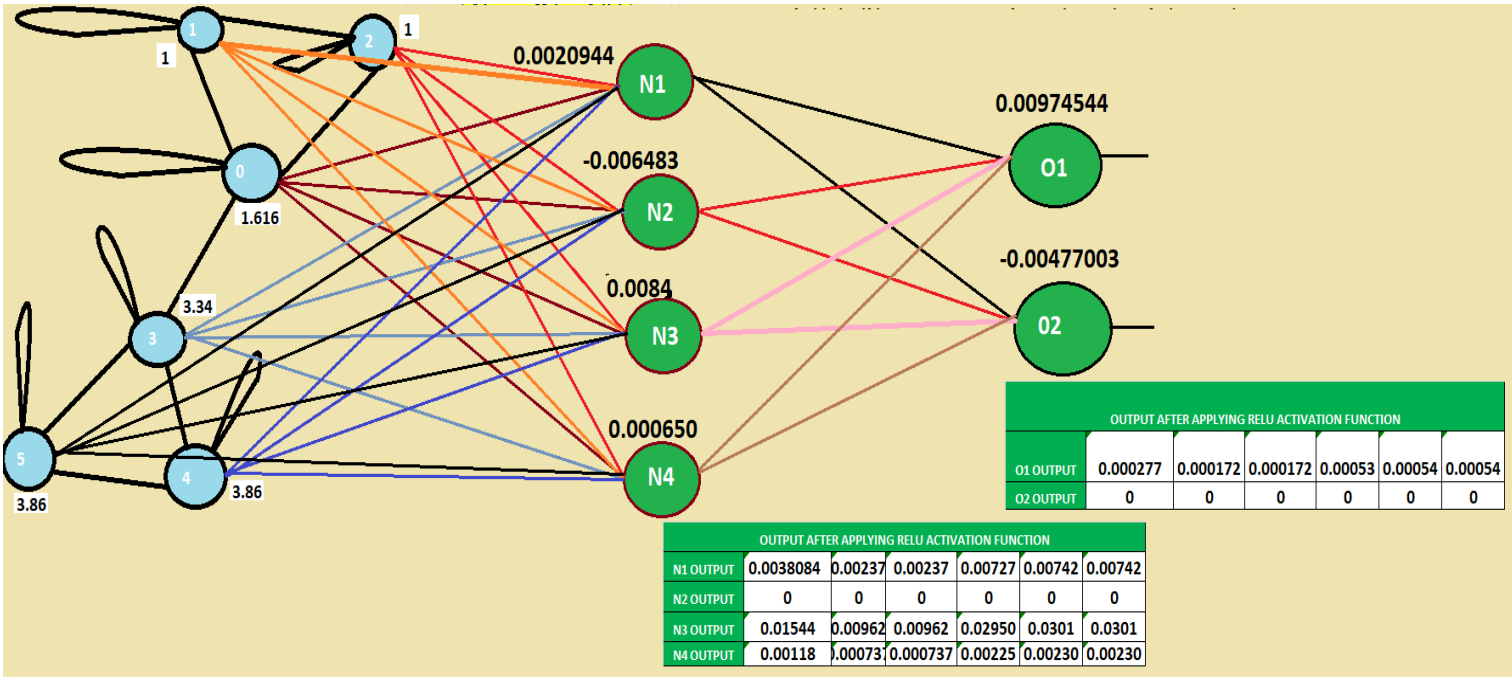


Figure 2.18 GCN Being passed on 2 hidden layer

Above figure shows after knowing the feature importance of the node a nonlinearity is induced through Relu activation function. Output from this will be passed to the second hidden layer where again normalised feature matrix will be multiplied with the weights and the final output shows which all nodes can be clustered.



OUTPUT AFTER APPLYING RELU ACTIVATION FUNCTION						
O1 OUTPUT	0.000277	0.000172	0.000172	0.00053	0.00054	0.00054
O2 OUTPUT	0	0	0	0	0	0

OUTPUT AFTER APPLYING RELU ACTIVATION FUNCTION						
N1 OUTPUT	0.0038084	0.00237	0.00237	0.00727	0.00742	0.00742
N2 OUTPUT	0	0	0	0	0	0
N3 OUTPUT	0.01544	0.00962	0.00962	0.02950	0.0301	0.0301
N4 OUTPUT	0.00118	0.00073	0.000737	0.00225	0.00230	0.00230

Figure 2.19 Normalised Feature Node Matrix Passed to 2 hidden layers

From the figure 2.19 and 2.20 , it can be clearly seen that there are 2 major groups, where the left group consists of nodes 0, 1, 2, and the right group consists of nodes 3, 4, 5. We can infer that the GCNs can already learn the feature representations even without training or backpropagation.

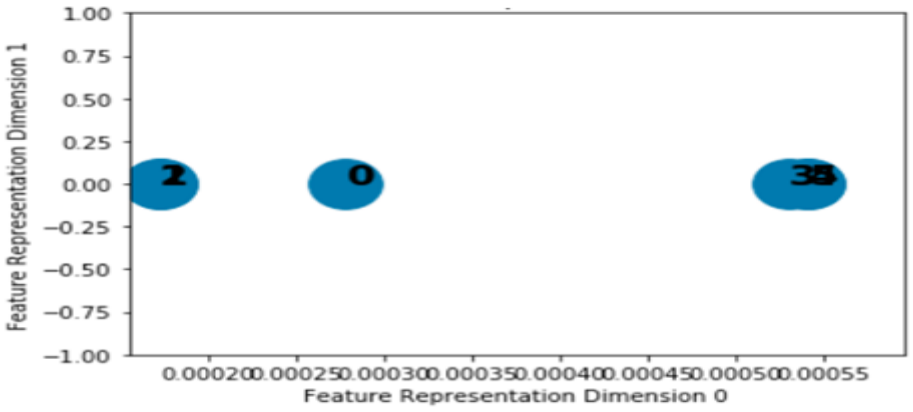


Figure 2.20 Feature Representation

2.7 Message Passing in Graph Neural Network

Goal of embedding is to take the data from the graph or original space and represent them in a new space. We showed the data in the new space from figure 2.9 as Z_u and Z_v this space is called embedding space. To make this transformation, we need the function which will call

encoder. The goal of the encoder is to take the data from the original space and then provide us with the data in the embedding space again in most of the applications. The goal of embedding is to find more simplified representation for the data while trying to preserve as much as information. In most of the application, the criteria is that we want the similarity between the nodes in the original space be as similar as possible to the similarity of the data in the embedding space(Klicpera et al., 2020). This means that we want the similarity between u and v in the original space be very similar to the similarity of the embedded representations of Z_u and Z_v in an embedding space.

Similarity of original space is represented as S_G and similarity in embedding space as S_E .Most of the times the definition of the similarity in an original space is different from the definition of the similarity in an embedding space. Message passing framework is a great general framework for Graph neural network methods. First, we initialize the data representation, which could be our graph in the original space x_v, we perform an aggregation function with the goal of summarizing the data and then we use this aggregation function result to update the state of our data. So if we assume H_v is our original graph and H_{v+1} is our graph, after applying this aggregation and update based on the encoding perspective, we could see that H_{v+1} is our embedding space. So the original data is H_v and H_{v+1} is our data on embedding space and our encoder is the combination of aggregation function and an update function. For multi-layer Graph Convolutional Network (GCN), Kipf and Welling (Kipf & Welling, 2016) introduced the following layer-wise propagation rule:

$$\boldsymbol{H^{(l+1)} = \sigma\left[\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right]}$$

The formula is derived from the Fourier Transform's first-order approximation of spectral graph convolutions. The adjacency matrix is normalised with its Degree Matrix in the first portion of the equation. The Degree Matrix is a diagonal matrix that displays how many neighbours each node has. The number of neighbours for unfixed-size kernels will not be a concern thanks to this normalisation. We multiply node characteristics (or node features) with the weight from the previous layer in the second component. Finally, we use a non-linear activation function, which is often known as Leaky ReLu.

2.7 Deep Walk embedding technique

We say that the similarity between two nodes, u and v is defined based on the probability of visiting u. If we do a random walk on a graph, starting with the node v. so the definition of the S_G is probability of u if we start with the v which is defined as $S_G(u, v) = p(u \mid v)$ in the

original space. This means we are going to perform a random walk on a graph, starting with the node v and we are trying to find what is the probability that we get to the node u. We define how many steps in a walk we should take to reach to u indicated by T. We need to repeat the process between these two nodes several times to see if we start with the node v and we perform a random walk for T steps. Are we going to get to the node, u or not? We can write this as a probability if we repeat this process for several times.

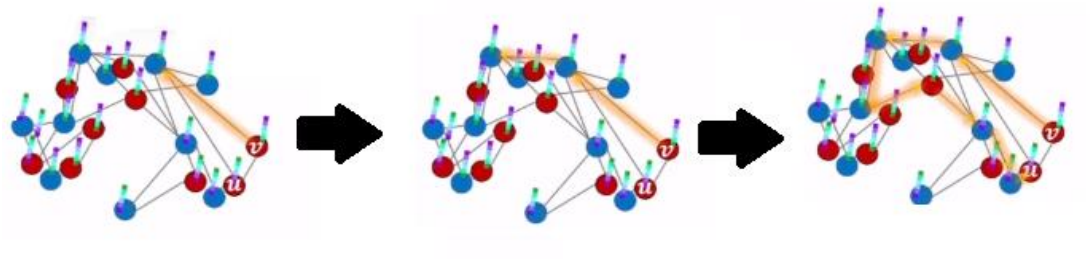


Figure 2.21 Random Walk on the Graph using DFS (Fouss et al., 2007)

In the above figure we will start with node v and now we should randomly select what is the next node we should go to select given the edges and the relations between the nodes in order to reach u. In this case, we have two options. First is going directly to the node u and the second is going to the blue node at the top left side let's say through the random walk, we get to the node in the blue. We again have a decision to make which will make it randomly on what is the next step we should take? Let's say we go to another node on the left side. That is another blue node and then we will repeat this process every time till we get to a node u, we have to randomly select the next step until we get to the node u or we may not get to the node u given that we had a less steps set through the parameters t , if we repeat this process for several times, starting with the node v and doing a random walk 40 steps, we will either get to the node u or we will not and if we have repeated this for several times, we can define this as a probability.

In the deep walk, the probability is the measure of a similarity so if there is a high chance to get to the node u if you start with the node v, the similarity between u and v will be high, if there is a low chance to get to the node u if we start with v the then the similarity between these two nodes will be low. We now have to define the similarity in an embedding space. In the most simplified version of the deep walk as S_E, which is a similarity and embedding space is defined as a DOT product between the vectors constructed for each node.

Our encoder will take the v and it will give us the Z_v, which is a vector and it will take the u and it will give us the Z_u, which is another vector. Similarity in an embedding space defined as $\boldsymbol{S}_E$ is a dot product of these two vectors.In order to define it as a probability, we're going to use a softmax as a form of a probability so we could have a similar measure $\boldsymbol{S}_E(\mathbf{z}_u, \boldsymbol{z}_v) = \frac{\exp(z_u^T z_v)}{\sum_{k \in V} \exp(z_u^T z_v)}$.

2.7.1 Deep Walk Encoder

In the case of Deep walk encoder is just a simple lookup table. as we said, the goal of the encoder is to take the nodes from the original space and represent them in a new space Z. In case of Deep walk, we said that this is a lookup table, this means that in an embedding space we will just have a D dimensional table where each node will have a row of the data representing. this means that the node V will become a Zv, which is a D dimensional vector, and node u will become a Zu, which is another D dimensional vector, so our goal is that we make the Zv such that it describes the node V with its features and all of its relation in the graph.

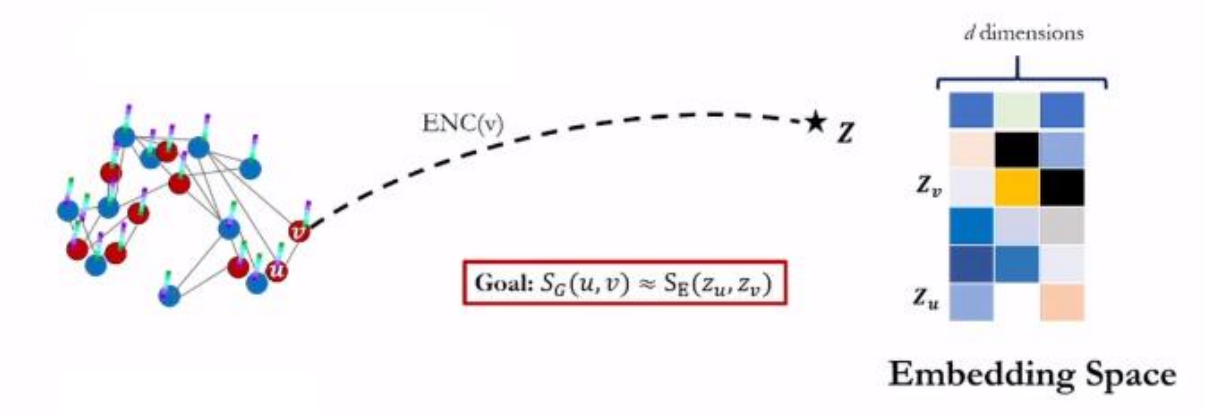


Figure 2.22 Encoder using Lookup Table (Fouss et al., 2007)

The fig 2.22 shows D dimensions table as another hyper parameter that you can define. This means that you could describe the node V with three dimensions or two hundred dimensions and you can use Deep walk to find those three dimensions or two hundred dimensions, so our encoder in the case of Deep walk is just a table so what happens is that if we have a V and we want to see what is the V in an embedded space, we will just go to the table. We will look at the Zv row and we will see what are the parameters or values that are defining the node.

From the above we can conclude S_G is a probability of visiting u starting with the v and S_E is just a dot product of the two vectors in the form of a softmax probability. We can use this two information to define a deep walk algorithm. Considering that we have defined the definition of the similarity in S_E, and S_G we can make a loss function that we aim to minimize.

2.8 Text Graph Convolutional Networks (Text GCN)

The text-based GCN model is novel semi-supervised learning concept that was recently suggested (Yao et al., 2018b) expanding on the previous GCN idea by Kipf et al. on non-textual data, it is capable of inferring the labels of unknown textual data with high accuracy given related known labelled textual data.

Text GCN is accomplished by embedding the whole corpus into a single graph, with documents (some labelled, some unlabelled) and words as nodes, and each document-word and word-word as edge having some preset weights depending on their connections with one another (eg. Tf-idf). The trained GCN model is then used to infer the labels of unlabelled documents.

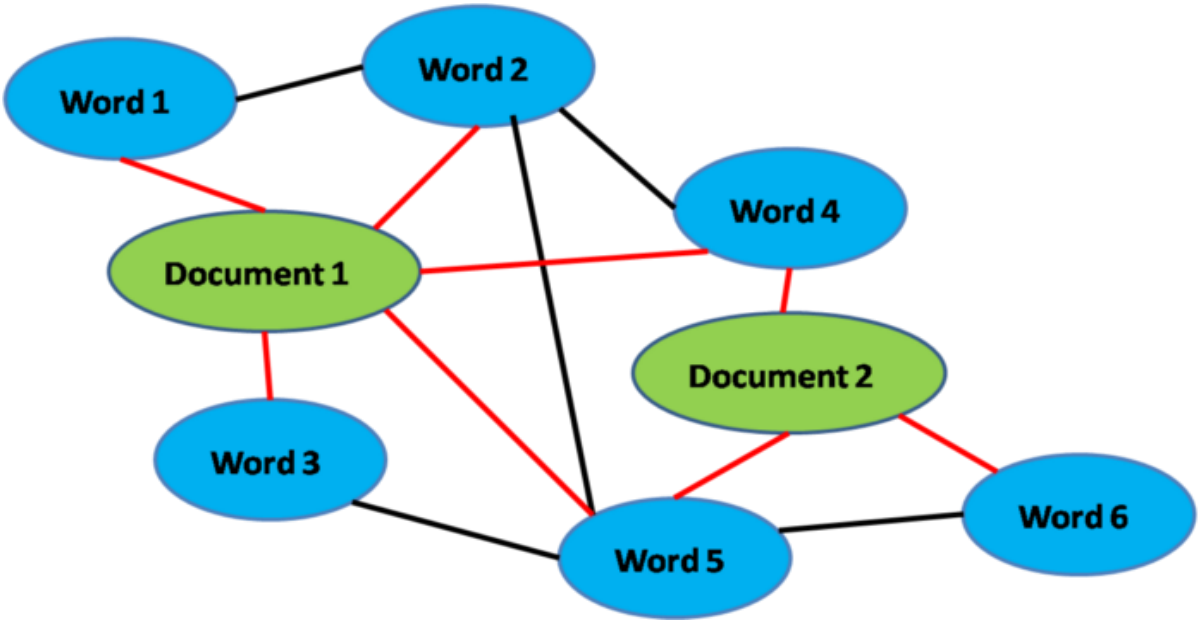


Figure 2.23 A graph representation of the corpus. The red lines indicate TF-IDF-weighted document-word edges, whereas the black lines show PMI-weighted word-word edges. (Yao et al., 2018b)

We create a graph with nodes and edges that reflect the links between document and words in order to allow GCN to understand the contexts. Documents and the whole vocabulary (words)

will form the nodes, which will be connected by weighted document-word and word-word edges. Their Aij weights are determined by below formula (Kastanos & Martin, 2021):

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

A word pairs for I,j, PMI value is calculated as:-

$$\begin{aligned} \text{PMI}\ (i,j) &= \log \frac{p(i,j)}{p(i)p(j)} \\ p(i,j) &= \frac{\#W(i,j)}{\#W} \\ p(i) &= \frac{\#W(i)}{\#W} \end{aligned}$$

From the above PMI denotes the Point-wise Mutual Information between pairs of co-occurring words over a sliding window #W of length 10 words. The number of sliding windows in a corpus that contain the word i is #W(i), the number of sliding windows that contain both the word i and the word j is #W(i,j), and the total number of sliding windows in the corpus is #W.

The phrase TF-IDF is frequency-inverse document frequency of a word in a document. Intuitively, a high positive PMI between words indicates that they have a high semantic connection, on the other hand, we do not create edges between words with a negative PMI. Overall, TF-IDF-weighted document-word edges capture context inside the document, whereas PMI-weighted word-word edges (which might span several documents) capture context across documents.

In contrast, such cross-document context information is difficult to supply as an input feature for non-graph-based models, and the model would have to learn it "from scratch" based on the labels. GCN should perform better since it provides more information on the relationship between documents, which is undoubtedly significant in NLP tasks.

2.8.1 Popular Graph datasets

Five widely used graph datasets are 20-Newsgroup (20NG), Ohsumed, R52 and R8 of Reuters 21578 and Movie Review (MR).

- The 20NG dataset (20NG News Group Dataset, n.d.) bydate version comprises 18,846 documents that are uniformly distributed across 20 categories. The training set has 11,314 documents, whereas the test set contains 7,532 documents.

- The Ohsumed corpus (Ohsumed Corpus, n.d.) comes from the National Library of Medicine's MEDLINE database, which is a bibliographic collection of major medical literature. We utilised the 13,929 unique cardiovascular disorders abstracts from the first 20,000. Each of the 23 illness categories has one or more related categories for each document in the set.
- The Reuters 21578 dataset has two subsets: R52 and R83 (all-terms version). R8 is divided into eight categories, including 5,485 training and 2,189 test docs. There are 52 categories in R52, which were divided into 6,532 training and 2,568 test doc.

2.8.2 Data Cleaning and Outcome

Table 2.1: Summary statistics of datasets.

Dataset	# Docs	# Training	# Test	# Words	# Nodes	# Classes	Average Length
20NG	18,846	11,314	7,532	42,757	61,603	20	221.26
R8	7,674	5,485	2,189	7,688	15,362	8	65.72
R52	9,100	6,532	2,568	8,892	17,992	52	69.82
Ohsumed	7,400	3.357	4.043	14.157	21.557	23	135.82

All of the datasets were initially pre-processed by cleaning and tokenizing the text (Kim 2014). For 20NG, R8, R52, and Ohsumed, we eliminated stop words specified in NLTK6 as well as low frequency terms occurring fewer than 5 times. It was seen that Text GCN significantly outperforms baselines on 20NG, R8, R52 and Ohsumed in terms of accuracy.

Table 2.2: Test Accuracy on document classification task

Model	20NG	R8	R52	Ohsumed
fastText	0.7938 ± 0.0030	0.9613 ± 0.0021	0.9281 ± 0.0009	0.5770 ± 0.0049
fastText (bigrams)	0.7967 ± 0.0029	0.9474 ± 0.0011	0.9099 ± 0.0005	0.5569 ± 0.0039
SWEM	0.8516 ± 0.0029	0.9532 ± 0.0026	0.9294 ± 0.0024	0.6312 ± 0.0055
LEAM	0.8191 ± 0.0024	0.9331 ± 0.0024	0.9184 ± 0.0023	0.5858 ± 0.0079
Graph-CNN-C	0.8142 ± 0.0032	0.9699 ± 0.0012	0.9275 ± 0.0022	0.6386 ± 0.0053
Graph-CNN-S	–	0.9680 ± 0.0020	0.9274 ± 0.0024	0.6282 ± 0.0037
Graph-CNN-F	–	0.9689 ± 0.0006	0.9320 ± 0.0004	0.6304 ± 0.0077
Text GCN	**0.8634 ± 0.0009**	**0.9707 ± 0.0010**	**0.9356 ± 0.0018**	**0.6836 ± 0.0056**

There are two primary reasons why Text GCN shows good results:

- The text graph may capture both document-word relationships and global word-word relationships.
- The GCN model computes a node's new characteristics as the weighted average of itself and its second order neighbours as a particular type of Laplacian smoothing (Li et al., 2018). The label information of document nodes can be transferred to their surrounding

word nodes (words inside the documents), which can then be transmitted to other neighbouring word nodes and document nodes. Word nodes can collect detailed document label information and function as connectors or important pathways in the graph, allowing label information to be shared.

2.9 Summary of this chapter

We thoroughly discussed the different aspects required to use Graph for text classification for social media posts. The literature study conducted here was in line with the first chapter's goal and aims. We began by reviewing what is graph and defining different types of graphs and how data is stored in graph. We moved on with discussing how convolution can be used on graph and what are its limitation and challenges that could arise. We then went through Graph convolution network to overcome limitations of convolution network. In details with an example, we also saw how the feature extraction happens on nodes in GCN using 2 hidden dense layer and with Relu activation function and softmax at the output layer.

In the next segment, we discussed various encoder and decode used on graph. We also covered random walk embedding and how it transforms the data from original space on to the embedding space. We also discussed about the Text GCN and how the document embedding happens and also we saw bench mark datasets and the outcome of it.

CHAPTER 3

RESEARCH METHODOLOGY

This section contains the entire methodology of our research. The flowchart below offers an overview of the research project.

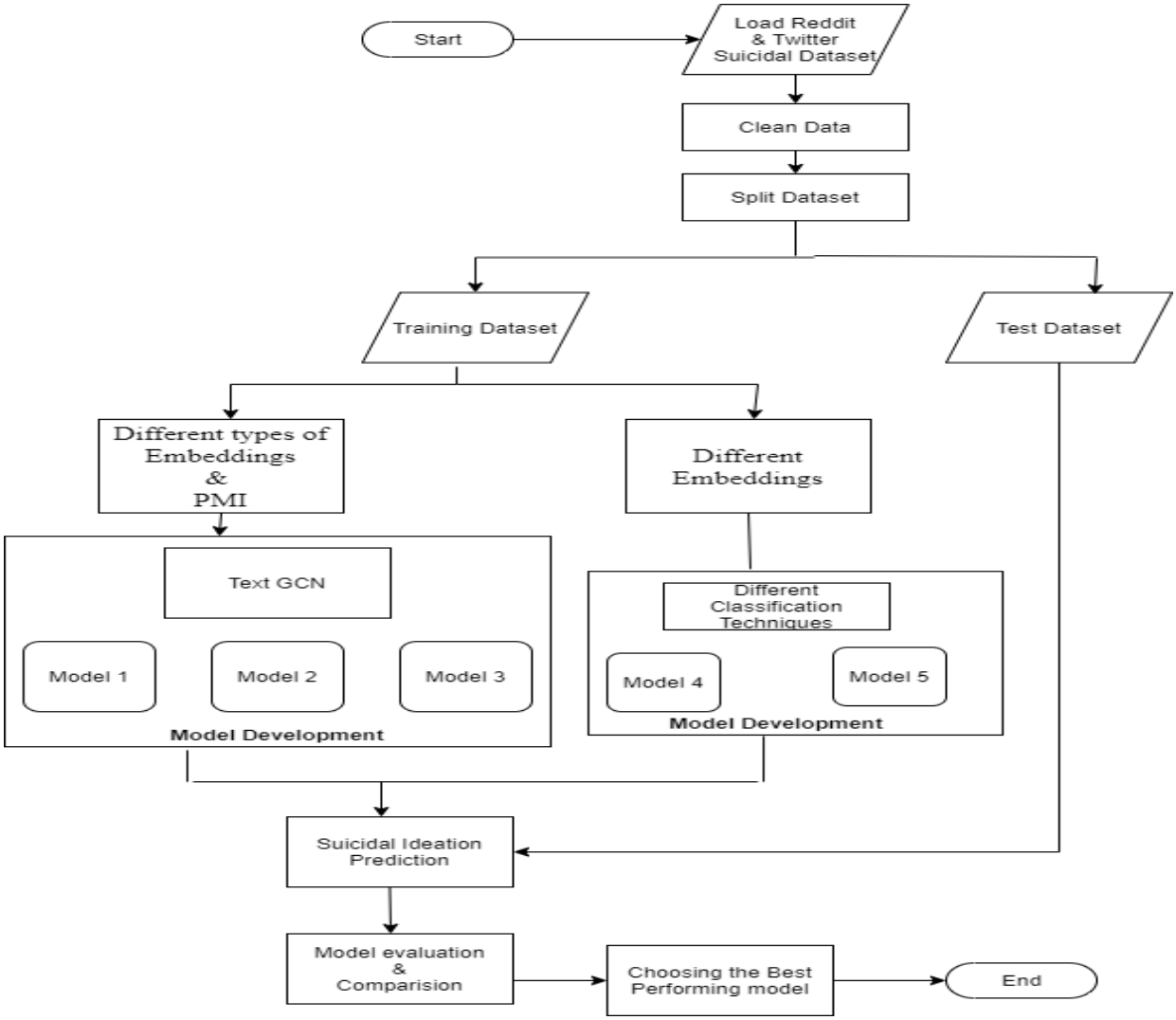


Figure 3.1 Research Methodology

3.1 Introduction

The study consists of a bunch of feature engineering to predict suicidal ideation detection on social media platforms based on text GCN (TGCN) model (Yao et al., 2018a). This will include loading the dataset, cleaning the dataset, preprocessing with few pretrained embeddings, prediction, model creations, comparison with other models and evaluation. Better performing model going to be picked based on the assessment metrics and analysis will then be decided upon same. Flowchart of Fig 3.1 shows how the proposed approach in which we load the data followed by text cleaning & preprocessing and splitting data in to train and validation set. This

step is followed by model evaluation & comparison with traditional classification models for ultimately choosing good performing model to conclude the research.

3.2 Dataset Description

Our dataset is a collection of posts from “depression" and “SuicideWatch" subreddits of the Reddit platform. The posts are collected using technique called Pushshift API. All the posts that were made to "SuicideWatch" from Dec 16, 2008(creation) till Jan 2, 2021.(*Reddit Suicide and Depression Detection Dataset,* n.d.) are part of the dataset.

The data is having two columns as follows: -

- **text: -** It consists of the text posted by user on Reddit and Twitter which are to be binary classified.
- **label: -** It consists of 2 groups. It specifies if given text is either have Suicidal Ideation text or non-Suicidal.

text	class
Ex Wife Threatening SuicideRecently I left my wife for good because she has cheated on me twice and lied to me so much that I have decided to refuse to go back to her. As of a few days ago, she began threatening suicide. I have tirelessly spent these paat few days talking her out of it and she keeps hesitating because she wants to believe I'll come back. I know a lot of people will threaten this in order to get their way, but what happens if she really does? What do I do and how am I supposed to handle her death on my hands? I still love my wife but I cannot deal with getting cheated on again and constantly feeling insecure. I'm worried today may be the day she does it and I hope so much it doesn't happen.	suicide
Am I weird I don't get affected by compliments if it's coming from someone I know irl but I feel really good when internet strangers do it	non-suicide
Finally 2020 is almost over... So I can never hear "2020 has been a bad year" ever again. I swear to fucking God it's so annoying	non-suicide
I took the rest of my sleeping pills and my painkillersI canâ€™t wait for it to end, I ve struggled for the past 6 years and I am finally ending it.	suicide
Can you imagine getting old? Me neither.Wrinkles, weight gain, hair loss, messed up teeth and bones, health issues, menopause, hormones, hating new generations & the way world progress.Being a useless angry piece of shit who can't take care of itself. Being totally depended on people who secretly wants you to die already.Can you even imagine yourself there? Absolutely not. Even if I was happy, I'd take my life just to avoid this.	suicide

Fig 3.2 Dataset Overview

We plan to run our experiment combining two different datasets

3.2.1 Reddit Dataset

Reddit: This is an online group that aggregates online discussion & societal news. It contains lot of categories and area of interest in a topic is called a subreddit. Subreddit called “Suicide Watch”(*Reddit Suicide Watch*, n.d.) is the area which we have considered for our thesis.

3.2.2 Twitter Dataset

Dataset contains about 8200 rows of tweets., Data was extracted from Twitter API using query parameters such as Hopeless, Depressed, Vow to take care of, “I don’t belong here”, “Nobody deserve me”, “I want to die etc” (*Building a Suicidal Tweet Classifier Using NLP | by Israel Aminu | Towards Data Science*, n.d.).

3.3 Data Pre-processing

In the preprocessing, we have to take out the unwanted symbols and unwanted characters and emojis which may not be helpful in prediction of the target, if there are any completely missing values in class predictor that needs to be removed as well.

3.4 Word embeddings

We initially preprocessed our data by tokenizing & cleaning text corpus. We will be removing stop words and low frequency words appearing less than specific number of times. Each word or document are indicated as word vector as Text GCN input. Word vectors are generally numerical representation of the text data. We will then apply and explore below pre-trained embeddings.

- **Node2vec**:-Which is a variation of the deep walk that can be used to perform graph embedding. The main idea behind node2vec is to use a biased random walk rather than an unbiased random walk. When we perform random walk, there are two ways we can explore a network. We can perform depth first search know as DFS, or we can perform breadth first search known as BFS.

  In the depth first search. What we will do is we will try to go as far as possible. We want to explore the graph as much as we can. In DFS we tried to go far in the graph as much as possible and given that we had the T samples. On the other hand, in the breadth first search, we are trying to stay as low as possible. So we want to have a more chance on visiting the existing node again instead of going far in depth in the graph. Using both DFS and BFS are useful because DFS allows us to explore more global representation, while BFS allows us to explore more local representation. We can set two parameters Q1 and Q2 in the form of a probability so we could define how much we want to explore local information and how much we want to explore the global information and again, while there is more to the algorithm itself, the main idea is just to introduce the unbiased random walk in the form of two parameters when we are performing walk on the graph.

- **Tf-Idf** : It is statistic meant to show how important a word is to a document in text corpus. It is also used as a weighting component in searches of information recovery, user modelling and text mining.
- **Word2Vec**:- This creates vector space which are of several thousand dimensions, with every distinctive word in the text corpus such that words that have common perspectives in the text corpus lays close to each other in the vector space.

3.5 Dataset Split

We will be going with different train test split to show how graph can learn and perform with less train data. We will split the train test split in terms of 4 group to evaluate results. First split will have 80 – 20 % and second split will have 60-40% , Third split will have 50-50% ,fourth will have 30-70% and last split will have 20-80% and we are shuffling the data to make sure the class labels are divided proportionally to maintain bias & variance which will help in providing better models and also helps in avoiding from overfitting.

3.6 Models

We will be building edges amid nodes based on word presence in documents (document-word edge) and word co-occurrence in the whole text corpus (word-word edges). For the weight of the edge between a document node and a word node we will be using different embeddings as mentioned in the previous section. To use word co-occurrence information, we will be using fixed sliding window size on all documents in the text to collect co-occurrence statistics. We will use point wise mutual information (PMI), which is most known method to gauge for word connections, to calculate weights among 2-word node.

By use PMI, the graph will capture the correlation among some words with higher dependency. Also, for each node itself, we need to add a self-loop on it. After creating the graph, we should feed the graph into a GCN as in (Kipf and Welling 2017), by varying the dense layers , learning rates and also by ensuring embeddings are of the similar size as the label set, this then put across a SoftMax activation layer. We will compare the Text GCN model built with different traditional classification models such as logistic regression and Naïve Bayes. We will also visualize the word embeddings using t-SNE (Hinton & Roweis, 2002). The doc embeddings of dataset in different layers with many dimensions learned by TGCN.

3.7 Evaluation

When the models are built, we will make use below ways to assess:

- Accuracy, which is one of the good metrics to decide classifications on balanced dataset, we will compare the different models Accuracy as part of comparative study by varying number of training epochs for different learning rate.
- We will also vary windows size and calculate accuracy on Text GCN model.
- We will also measure F1 score for all models and we will plot ROC which is plotted on TPR Versus FPR. This helps to describe the model closest to the threshold is best one to choose.
- We will measure Test accuracy by varying training data proportions from 80%, 60%, 50%, 30% and 20%.

3.8 Required Resources

3.8.1 Software Requirements:

We will be needing following software resources to accomplish our implementation tasks:

- **Operating System**: Windows 10 (64-bit operating system)
- **Language**: Python 3.8.x
- **Python IDE**: Jupyter Notebook, Spyder
- **Python Libraries**: Numpy, Pandas, Seaborn, Matplotlib, Scikit-learn, XGBoost, imblearn, NLTK, Gensim, String, re, tensorflow 2.1.0, Keras 2.3.x ,Pytorch.
- **Pretrained Word Embeddings**: Word2Vec, Glove and Node2Vec.
- **Graph Libraries**: NetworkX, spectral or Graph4NLP
- **Others**: MS Office, Adobe PDF reader, Mendeley Desktop.

3.8.2 Hardware Requirements:

We will require a lot of processing resources because we will be training Deep NLP algorithms on a very large dataset. We will be working on huge text corpus so any of the below mentioned GPU powered computing infrastructure would be required.

- **Google Colab**: with Tesla K80 GPU 12GB and CPU 12GB;
- **Nimblebox**: with GPU 12GB and CPU 61GB.
- **Kaggle Kernel**: We can use GPU with quad core processor and up to 16 GB RAM

CHAPTER 4

ANALYSIS

This chapter explains the types of data, their nature, exploratory analysis, and the numerous hyper-parameters we utilised to tweak the model. We go through the technical specifics of these algorithms and how they helped us achieve optimal text categorization results.

4.1 Introduction

The research approach for this study was detailed in the preceding chapter. The findings, observations, and analyses made during the study approach are presented in this chapter. The data description is included in the first part. After that, we'll go over the data cleaning operations from the previous chapter and talk about our results and observations. The outcomes of the exploratory data analysis will also be discussed. The trained classification models and tweaked hyperparameters of each model will be discussed in the latter portion of this chapter

4.2 Data Description

Our data consists of postings from the Reddit subreddits "depression" and "SuicideWatch." The posts are gathered utilising a method known as Pushshift API. The dataset includes all postings to "SuicideWatch" from December 16, 2008 (creation) through January 2, 2021 (Reddit Suicide and Depression Detection Dataset, n.d.).

The data is having two columns as follows: -

- **text: -** It consists of the text posted by user on Reddit and Twitter which are to be binary classified.
- **label: -** It consists of 2 groups. It specifies if given text is either have Suicidal Ideation text or non-Suicidal.

text	class
Ex Wife Threatening SuicideRecently I left my wife for good because she has cheated on me twice and lied to me so much that I have decided to refuse to go back to her. As of a few days ago, she began threatening suicide. I have tirelessly spent these paat few days talking her out of it and she keeps hesitating because she wants to believe I'll come back. I know a lot of people will threaten this in order to get their way, but what happens if she really does? What do I do and how am I supposed to handle her death on my hands? I still love my wife but I cannot deal with getting cheated on again and constantly feeling insecure. I'm worried today may be the day she does it and I hope so much it doesn't happen.	suicide
Am I weird I don't get affected by compliments if it's coming from someone I know irl but I feel really good when internet strangers do it	non-suicide
Finally 2020 is almost over... So I can never hear "2020 has been a bad year" ever again. I swear to fucking God it's so annoying	non-suicide
I took the rest of my sleeping pills and my painkillersI canâ€™t wait for it to end, I ve struggled for the past 6 years and I am finally ending it.	suicide
Can you imagine getting old? Me neither.Wrinkles, weight gain, hair loss, messed up teeth and bones, health issues, menopause, hormones, hating new generations & the way world progress.Being a useless angry piece of shit who can't take care of itself. Being totally depended on people who secretly wants you to die already.Can you even imagine yourself there? Absolutely not. Even if I was happy, I'd take my life just to avoid this.	suicide

Figure 4. 1 Representation of the Data

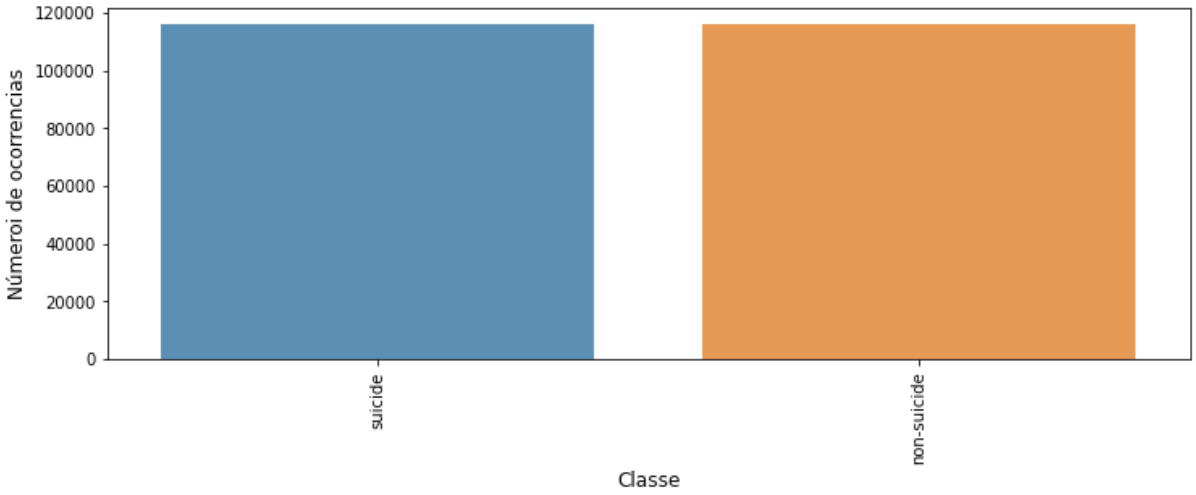


Figure 4. 2 Bar Chart of Target Lables Data Distribution

From the above figure we can observe there is no class imbalance in the graph, We have equal shares of both sucidal and non-suicidal data so no any kind of balancing techniques are used in this thesis.

4.3 Data Preparation

In this part, we will go through the data cleaning procedures to prepare the dataset for usage in word embeddings. We are executing the data cleaning stages listed below.

1. Special characters are Removed
2. URL's are Removed
3. Digits are Removed
4. Extra spaces and White spaces Removed
5. Stopwords are Removed
6. Removing those words which are appearing less than 5 times
7. Junk characters Removed

Below table shows the difference in the raw data and the cleaned dataset. We have kept use of only those words which appear more than 5 times in the corpus thus by ensuring rare words and less frequently used words are removed by this we can also filter out fillers or some rarely used words.

Before CleanUp	After Cleanup
Ex Wife Threatening SuicideRecently I left my wife for good because she has cheated on me twice and lied to me so much that I have decided to refuse to go back to her. As of a few days ago, she began threatening suicide. I have tirelessly spent these paat few days talking her out of it and she keeps hesitating because she wants to believe I'll come back. I know a lot of people will threaten this in order to get their way, but what happens if she really does? What do I do and how am I supposed to handle her death on my hands? I still love my wife but I cannot deal with getting cheated on again and constantly feeling insecure. I'm worried today may be the day she does it and I hope so much it doesn't happen.	ex wife threatening left wife good cheated twice lied much decided refuse go back days ago began threatening suicide spent days talking keeps wants believe ill come back know lot people threaten order get way happens really supposed handle death hands still love wife cannot deal getting cheated constantly feeling insecure im worried today may day hope much doesnt happen
Am I weird I don't get affected by compliments if it's coming from someone I know irl but I feel really good when internet strangers do it	weird dont get affected compliments coming someone know irl feel really good internet strangers
Finally 2020 is almost over... So I can never hear "2020 has been a bad year" ever again. I swear to fucking God it's so annoying	finally almost never hear bad year ever swear fucking god annoying
i need helpjust help me im crying so hard	need help im crying hard
It ends tonight.I canâ€™t do it anymore. I quit.	ends tonighti cant anymore quit
I took the rest of my sleeping pills and my painkillersI canâ€™t wait for it to end, Iâ€™ve struggled for the past 6 years and Iâ€™m finally ending it.	took rest sleeping pills cant wait end ive struggled past years im finally ending

Figure 4.3 Difference in Raw data and Cleaned data

4.4 Exploratory Data Analysis

The exploratory data analysis outlined in the study technique is carried out in this part. To begin, we measure the difference in number of words in each line before and after clean up. Below figure shows the number of words in each line has reduced drastically after removing stop words, special characters, digits and unique words from the corpus

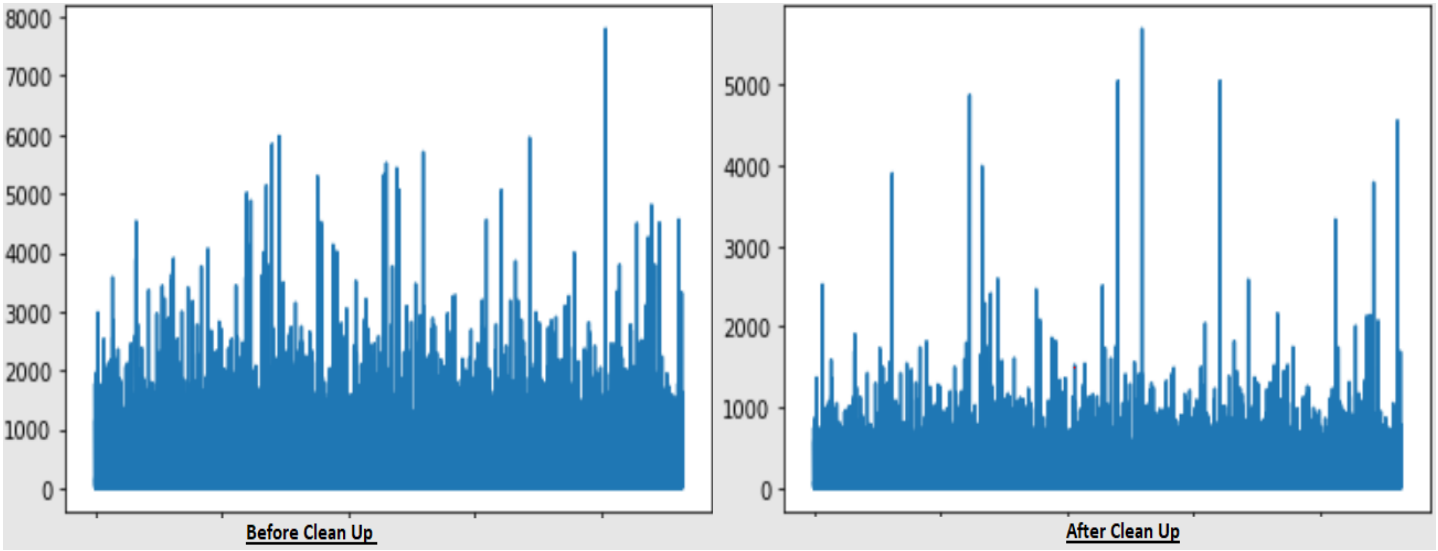


Figure 4.4 Number of words each line before and after clean-up

We used a most often used word analysis to get the top 15 most frequently used terms in suicidal labelled and non-suicidal labelled cleaned dataset. These phrases would produce extremely meaningful representations while avoiding the use of sparse matrices.

	Suicide		Non-Suicide	
Rank	Word	Frequency	Word	Frequency
1	im	266369	hate	5930
2	dont	178847	new	5794
3	like	130330	years	5746
4	want	127932	told	5745
5	feel	107497	parents	5744
6	life	107429	lot	5691
7	know	105560	doesnt	5586
8	ive	92063	best	5412
9	cant	89602	getting	5233
10	people	70384	started	5217
11	time	68121	reddit	5095
12	think	57089	pretty	5084
13	going	56889	stuff	5069
14	friends	47914	talking	4973
15	years	45484	better	4896

Table 4.3 Frequency count of top 15 words

Below figure shows the word cloud of suicidal and non-suicidal category data. We can already notice negative words are present on the suicidal part compared to the non-suicidal word cloud.

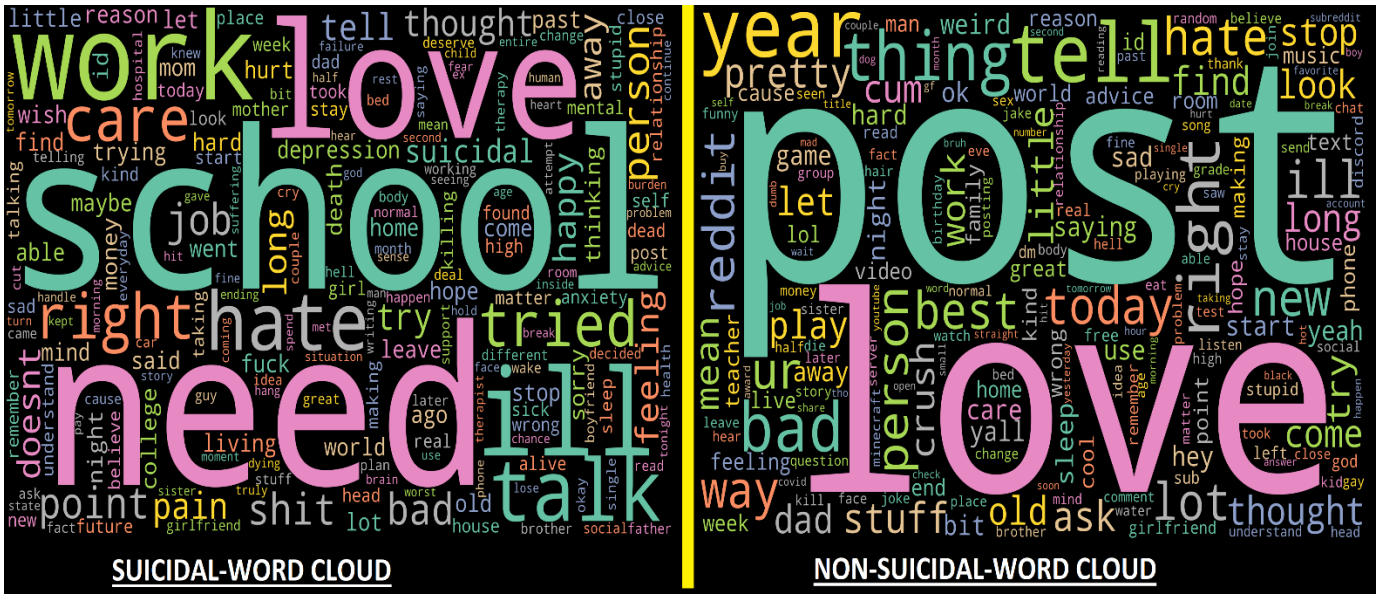


Figure 4.5 Suicidal word cloud vs Non-suicidal post word cloud

EDA Observations

1. Unlike other web resources, the Suicidal ideation dataset used in this research is not properly organised and it's written in very informal language since it's online post people many times remains anonymous and post what ever they are feeling or rough phase of their life.

2. Many of the uncommon words and junk words and emoticons are removed as part of the data pre-processing since these don't add any value during our model building process.

3. The distribution of word length is comparable for the Suicidal and non-suicidal categories before and after clean up.

4. From the word cloud we can notice the most commonly used words in both suicidal and non-suicidal post. We generally see most positive words present on the non-suicidal word cloud compared to the other.

4.5 Model Building

Text corpus and labels are separated out in two different files. Text corpus data will be processed initially to remove all the special characters. Target labels are then read and should contain the additional labels specifying whether target label is train or test.We then shuffle the target labels then the data also be shuffled correspondingly. We will then prepare vocabulary list and file containing all the words present in the corpus. We will prepare the word id map based on the position of the word in the vocabulary file.

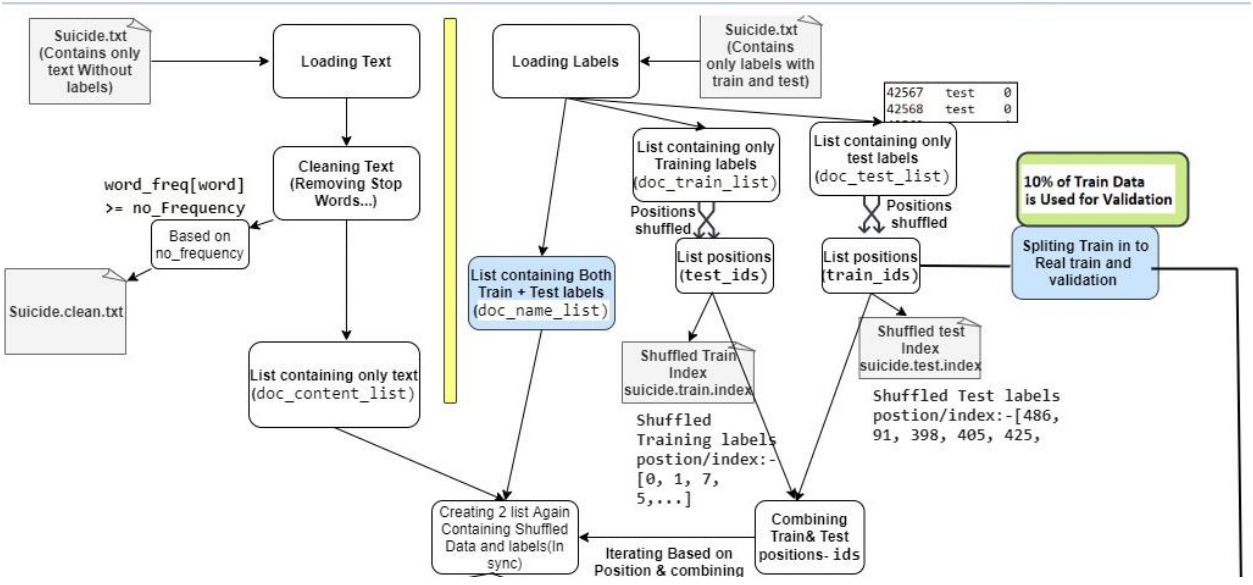


Figure 4.6 Data pre-processing & Shuffling text & labels

4.5.1 Graph Construction

We construct a large and heterogeneous text network that encompasses both word and document nodes to directly represent global word co-occurrence and adjust graph convolution. The number of nodes in a text graph |V | is equal to the sum of the number of documents (corpus size) and the number of unique words (vocabulary) in the corpus. We set feature matrix X = I to be an identity matrix, implying that each word or document is represented as a single-hot vector in the Text GCN input.

We create edges between nodes depending on the frequency of words in the data. Word co-occurrence and documents (document-word edges) throughout the whole corpus (word-word edges). The weight of the edge between a document node and a word node is called the term frequency-inverse document frequency (TF-IDF) of the word in the document. The term frequency refers to the number of times a word appears in a document. The inverse document frequency is the logarithmically scaled inverse fraction of the number of documents that include the term in the document.

We found that TF-IDF weight is better than using term frequency alone. We used fixed size sliding window mostly 15 on all documents in the corpus to collect co-occurrence data in order to leverage global word co-occurrence information. To compute weights between two-word nodes, we use point-wise mutual information (PMI), a prominent metric for word associations.

Weight of edge between node i and node j is defined as below (Yao et al., 2019)

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

A word pairs for i,j is calculated as:-

$$\begin{aligned} \text{PMI}\ (i,j) &= \log \frac{p(i,j)}{p(i)p(j)} \\ p(i,j) &= \frac{\#W(i,j)}{\#W} \\ p(i) &= \frac{\#W(i)}{\#W} \end{aligned}$$

Where #W(i) denotes the number of sliding windows in a corpus that contain the word i #W(i, j) denotes the number of sliding windows that contain both the words i and j, and #W is the total number of sliding windows in the corpus. A positive PMI number shows that words in a

corpus have a strong semantic correlation, whereas a negative PMI value suggests that the corpus has little or no semantic association. As a result, only edges between word pairs with positive PMI values are added.

4.5.2 Parameter Tuning for Graph Construction

In our thesis we have focused running experiments by varying train and test size data and tuning initial hyperparameters for construction of graph. After cleaning the data from stop words and junk text, we filter only those words which has appeared equal to greater than 5 times.

In the various experiments we have modified window size in the range 10 to 25 to see co-occurrence of word in that sliding window size and the impact on the test accuracy. In the experiments we have also modified embedding dimensions with which we create sparse matrix with the help of derived PMI values and i,j (row and column values).

4.5.3 GCN Neural Network

We input the text graph into a basic two-layer GCN as described in (Kipf and Welling 2017), with the second layer node (word/document) embeddings having the same size as the labels set and being fed into a softmax classifier.A two-layer GCN allows messages to be sent between nodes. The two-layer GCN allows information exchange between pairs of documents even though there are no direct document-document edges in the graph.

Below are some of the hyper parameters which was used in the implementation of GCN neural network.

- We use Softmax Activation function for output layer.
- Dropout Layer: By a proportion of the values supplied to it, this layer sets input units to 0 at random (Turning the neurons ON or OFF). This layer aids in the avoidance of overfitting. For dropout, we used the value 0.5.
- Loss Function: We have utilised the Categorical Cross entropy as it is a categorization. The cross-entropy loss between true and predicted labels is calculated with this function.
- Optimizer: We have used Adam optimizer as the optimally suitable with learning rate of 0.02 rate. For few of the experiments we have also used learning rate as 0.01.
- Epoch: We have stop early technique where if the loss doesn't gets improved then it stops.

4.6 Summary

We presented the findings and observations we made while using the study methods described in the previous chapter in this chapter. The source and type of the data, as well as columns and their descriptions, are discussed in the first portion of this chapter. The Exploratory Data Analysis was described in the next section, which validated the dataset's preparedness for text categorization. We also spoke about the data cleaning efforts we did to get the raw data into the format we needed.

We discussed about how the text corpus and the target labels are split and shuffled also we saw how the vocabulary file and list get formed using the text corpus. In the next section we saw the volume of train data being used to construct graph also discussed about the parameters that go in constructing graph and PMI values derivation between word-word and word-document while constructing the graph.

In the last section we discussed about the GCN neural network and the hyper parameters which is used to get the best result and variations which are done on different parameters during the experiments.

CHAPTER 5

RESULTS AND DISCUSSIONS

The outcomes of the experiments we conducted in this thesis are discussed in Chapter 5. These are in accordance with the research methods outlined in Chapter 3 and the hyperparameters selected for each model in Chapter 4.

5.1 Introduction

We conducted a series of experiments with various window size and varying learning rate and limiting the train data while constructing the graph as part of thesis. The outcomes of the implemented experiments are presented and discussed in the first part. In the last section, we compare the outcomes of the trials and attempt to understand the differences in performance across the traditional models.

5.2 Evaluation of the Models and Results

To complete the experiments outlined in Chapter 3's study approach, we generated several combinations by varying volume of train data, window size and embeddings dimensions for text classification. We used Text GCN by varying multiple parameters to see how best it can establish the relation between word and words and document and words,

We tested a total of 15 combinations by varying different parameters. As a metrics we have used Accuracy, Precision, Recall & F1-Score to measure against the volume of train and test split and window and embedding sizes.

5.2.1 Evaluation of the Model varying the volume of Train & Test Data

Huge corpus which had 231973 lines after cleaning was divided in to 115000 and 116973 for train and test respectively. Among 115000 line 10% of data 11500 sentences was used for validation. We have fixed window size of 15 and learning rate to 0.02 .

In the later experiments we varied the train data from 50% to 10% and then to 20% keeping the window size and learning rate as constants.

The table below lists the experiments we conducted, as well as the results of each experiment's evaluation varying the volume of train and test data each time.

Table 5.1: Evaluation of each Exp on Varying the volume of Train & Test Data

Train & Test split (Train %|| Test %)	Text Corpus size	Training Data	Test Data	window Size	Learning Rate	Embedding Dim	Training accu	Val accu	Test accu	Precision	Recall	F1-Score
50% || 50%	231973	115000	116973	15	0.02	300	0.95	0.93	0.93	0.93	0.93	0.93
50%||50%	115000	57500	57500	15	0.02	300	0.95	0.92	0.92	0.92	0.92	0.92
10%||90%	231973	23197	208776	15	0.02	300	0.97	0.92	0.92	0.92	0.92	0.92
20%||80%	231973	46394	185579	15	0.02	300	0.96	0.92	0.93	0.93	0.93	0.93

From the above comparison table, we can see various models have been evaluated mainly varying the volume of train and test data keeping learning rate as 0.02 and window size as 15 while constructing the graph also we have kept embedding size as 300 to construct the sparse adjacent matrix for the above experiments.

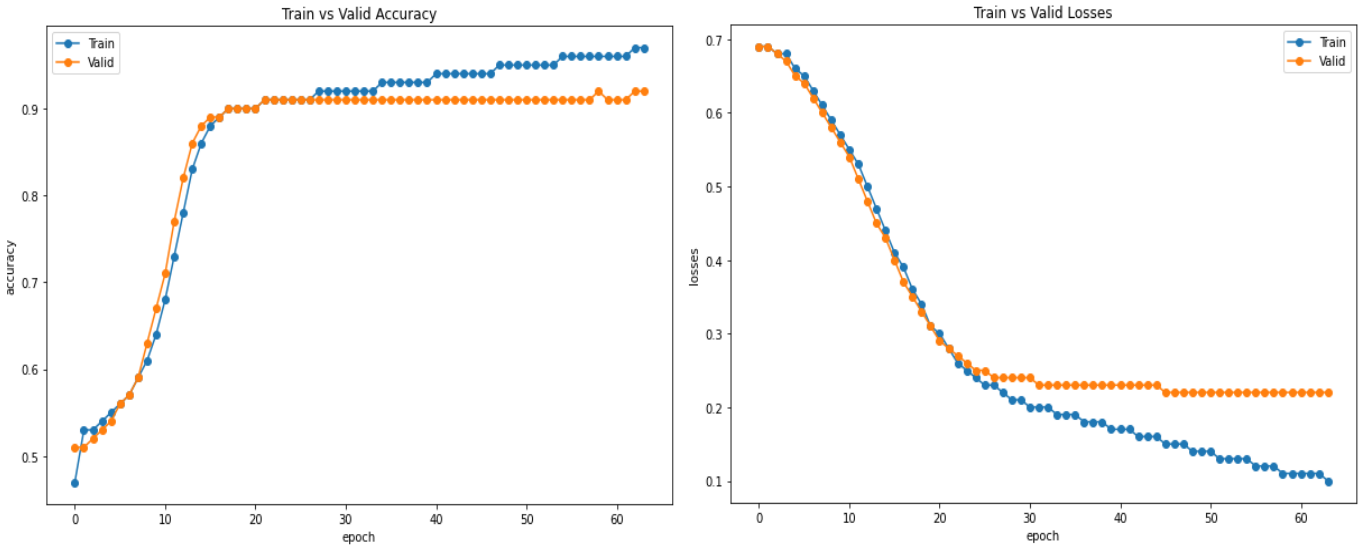


Figure 5.2 Accuracy and Loss with 50% of Train and Test Data

As seen in the above figure training accuracy increases along with the validation accuracy even with 50% of training data we get 95% accuracy on training 92% on validation and 92% on Test data which is considerably very good since we have trained our model with only 50% of trained data.

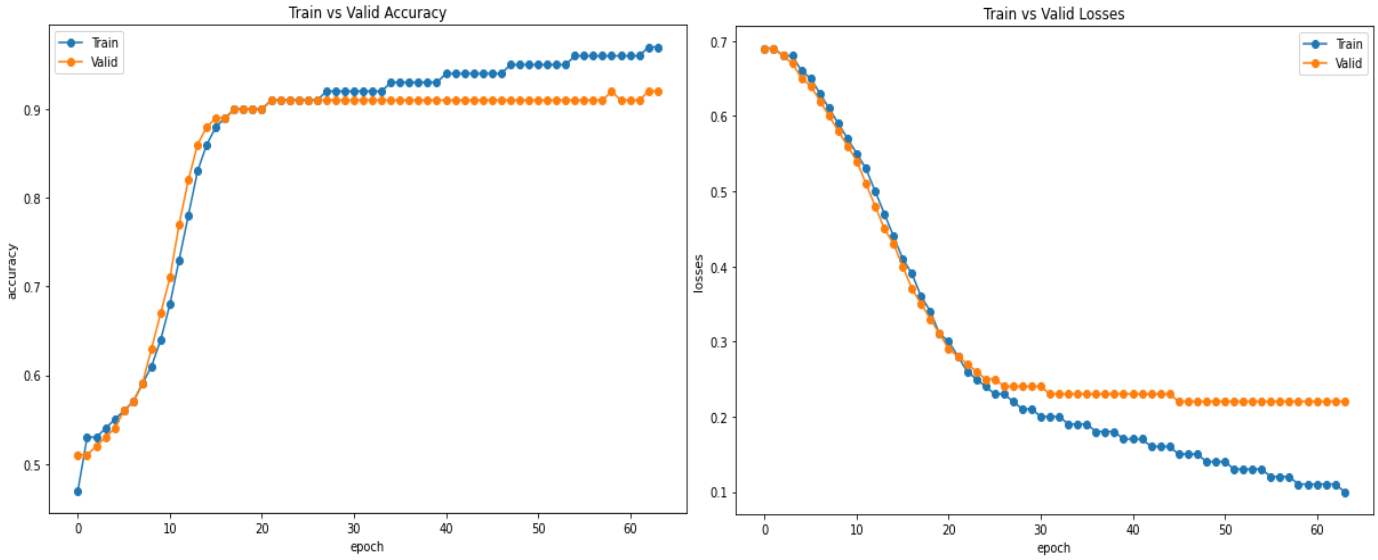


Figure 5.3 Accuracy and Loss with 10% Train and 90% Test Data

From figure 5.3 we can see that even when we have trained our model only with 10% of training data it still gives very good result on the unseen test data. We can clearly see the training accuracy going up to 96% validation accuracy to 91% and test accuracy to 92% though the model was trained on only 10% of the whole data. This is one of the most key point of our thesis where Text GCN can be trained on very less data and still it performs very well on the unseen data. We can also see even on increasing the training data from 10% to 20% still the accuracy of test doesn't vary much thus by showing Graph can perform very well on less training data of social media posts.

5.2.2 Evaluation of the Model varying the Window size

Window size helps in deciding the co-occurrence of words relations ship which helps in calculating PMI values between words-words and document-words. In our experiment we use corpus size of 10k records since whole corpus of 2 lakhs records takes more time to clean and process it. With 10k records we have divided training and test data in 40% and 60% ratio respectively. We have kept the learning rate as 0.01 and embedding dimension as 300.

The table below lists the experiments we conducted, as well as the results of each experiment's evaluation varying the window size during the graph building.

Table 5.4: Evaluation of each Exp by Varying windows size

Train & Test split (Train %|| Test %)	Text Corpus size	Training Data	Test Data	window Size	Learning Rate	Embedding Dim	Training accu	Val accu	Test accu	Precision	Recall	F1-Score
40%||60%	10000	4000	6000	15	0.01	300	0.97	0.90	0.90	0.90	0.90	0.90
40%||60%	10000	4000	6000	20	0.01	300	0.96	0.90	0.90	0.90	0.90	0.90
40%||60%	10000	4000	6000	25	0.01	300	0.96	0.88	0.89	0.89	0.90	0.89

From the above comparison table, we can see various models have been evaluated mainly varying the window size keeping learning rate as 0.01 while constructing the graph also we have kept embedding size as 300 for the above experiments.

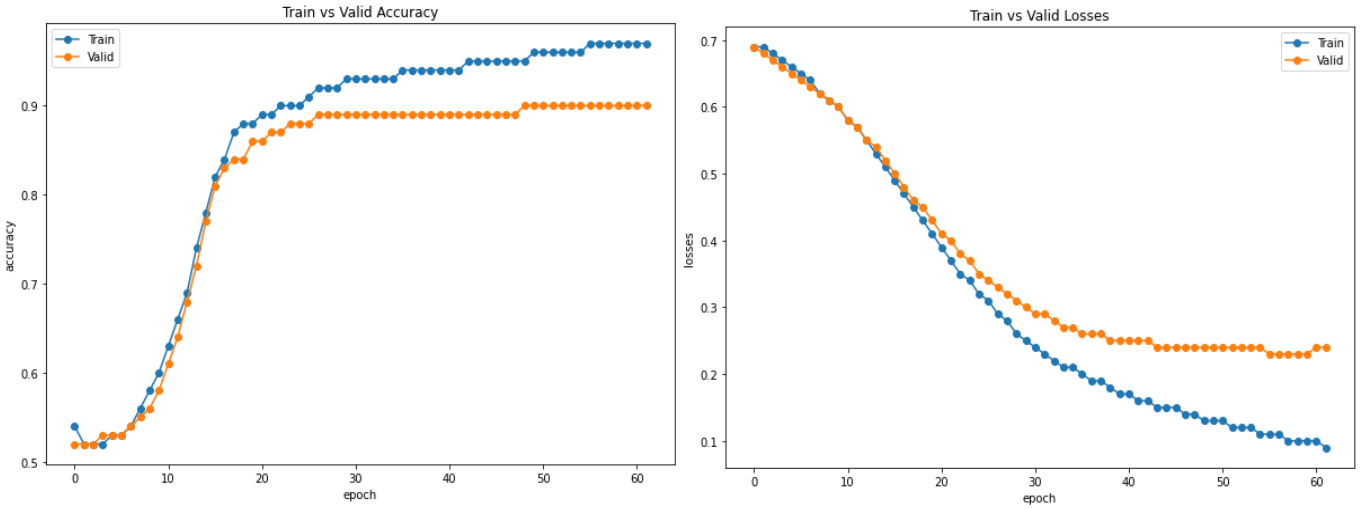


Figure 5.5 Accuracy and Loss with window size of 15

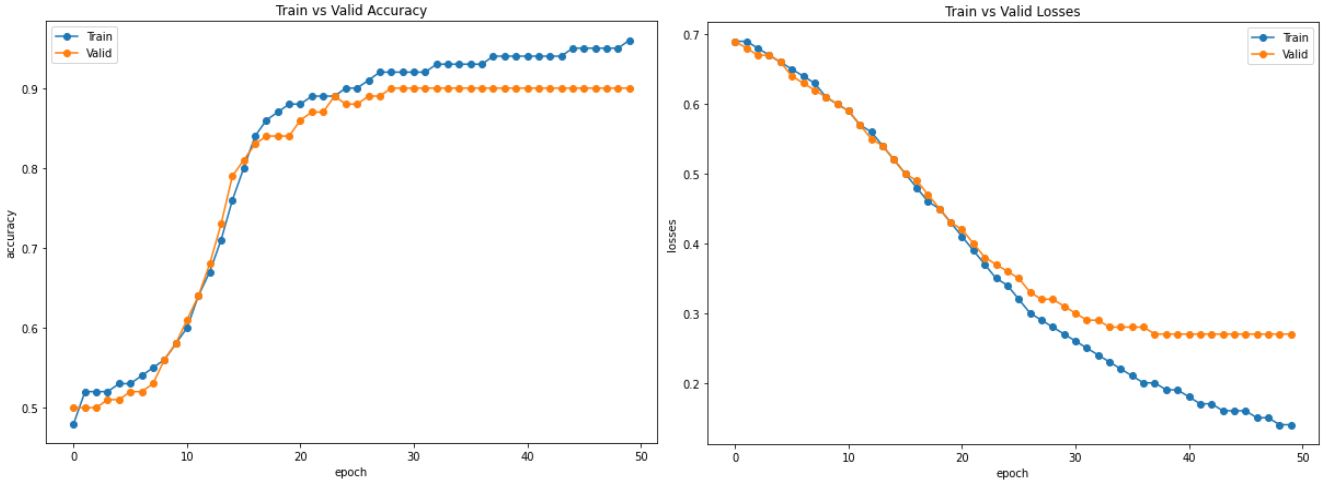


Figure 5.6 Accuracy and Loss with window size of 20

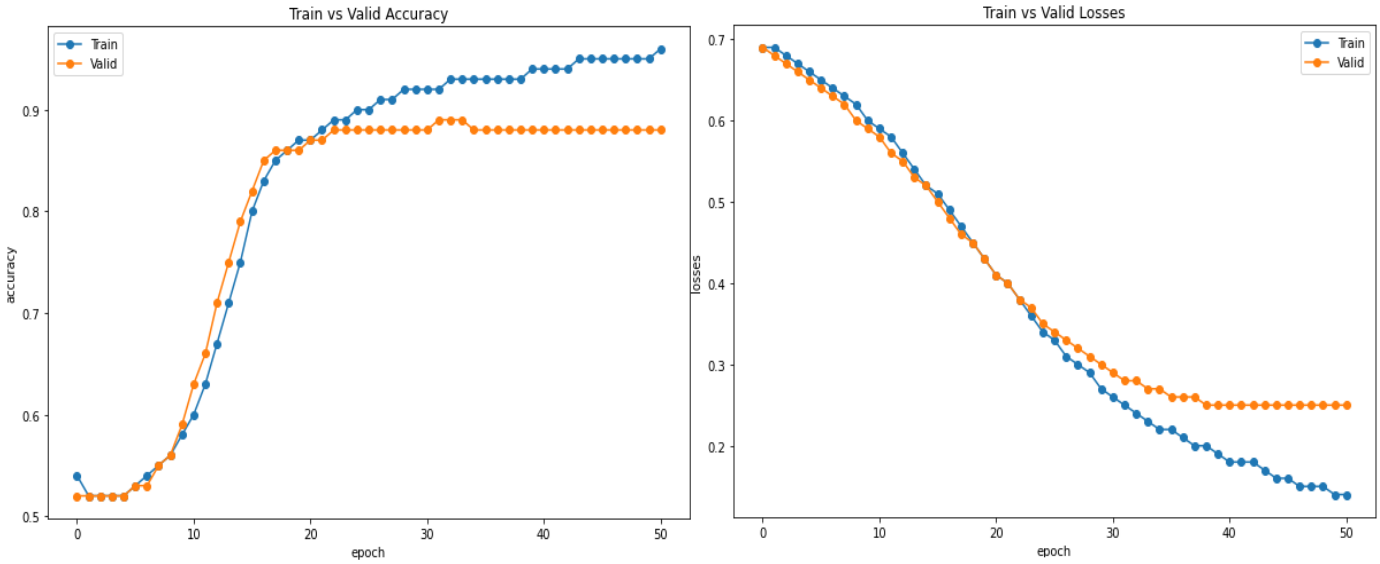


Figure 5.7 Accuracy and Loss with window size of 25

From figure 5.4,5.5 and 5.6 we can see that even when window size is varied from 15 to 25 with learning rate as 0.01 and embedding size as 300 . we see training accuracy going up to 97% and test accuracy with 89%. For other window size we could see any big difference in test accuracy for the window size of 15,20 and 25 we consistently got the accuracy and F1 score as 89%.There by indicating after optimal window size even after increasing window size will not yield any better results but it just increases processing time since the bigger sentences needs to be moved multiple times with that fixed window size.

5.2.3 Evaluation of the Model varying the Embedding size

Embedding dim help in constructing sparse adjacent matrix by mapping rows columns against PMI values found for word-word and document-word. In this experiment we use corpus size of 10k we have divided training and test data in 40% and 60% ratio respectively. We have kept the learning rate as 0.02 , window size as 10 and embedding dimension as 600 and 100.

The table below lists the experiments we conducted, as well as the results of each experiment's evaluation varying the embedding dim during the graph building.

Table 5.8: Evaluation of each Exp by Varying Embedding Dim

Train & Test split (Train %|| Test %)	Text Corpus size	Training Data	Test Data	window Size	Learning Rate	Embedding Dim	Training accu	Val accu	Test accu	Precision	Recall	F1-Score
40%||60%	10000	4000	6000	10	0.02	600	0.93	0.89	0.89	0.90	0.90	0.89
40%||60%	10000	4000	6000	10	0.02	100	0.96	0.89	0.90	0.90	0.90	0.90

From figure 5.9 and table values we can see that even when embedding dimensions varied from 300 to 600 and then to 100 with learning rate as 0.02 and window size as 10. we see training accuracy going down thought there is no much variation in the test accuracy which remains at 89%.There by indicating lesser dimension embedding matrix carries important details, Due to time constraint we could not find optimal embedding size where the train and test accuracy changes drastically.

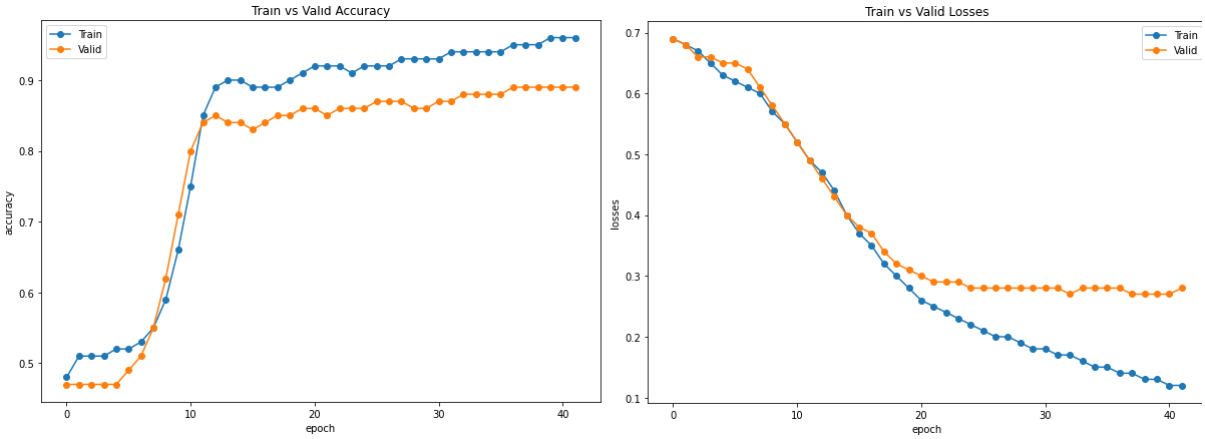


Figure 5.9 Accuracy and Loss with embedding size as 100

5.2.4 Evaluation Traditional model Against Text GCN

We have used whole corpus which had 231973 lines to compare Text-GCN against logistic regression with 90% of test data and 10% of train data. We have fixed window size of 15 and learning rate to 0.02 for text GCN. While processing the data for Logistic regression we are using porter stemmer for stemming and also, we are cleaning all those words which are not alphabets there by eliminating junk characters and digits.

We have used TFIDF to converts words in to vector, TF-IDF score compares the most common and rarest words in the corpus to the rest of the words in the corpus, assigning low scores to the most common and rarest terms. When compared to its predecessor, this approach gathers more data.

The table below lists the experiments we conducted, as well as the results of each experiment's evaluation done for Text GCN and Logistic Regression.

Table 5.10: Comparison of traditional model against Text GCN

Perc of Train & Test split (Train %|| Test %)	Text corpus size	Training data	Test Data	window size	Learning Rate	Embedding Dim	Training accuracy	Val accuracy	Test accu	Precision	Recall	F1-Score
10%||90%	231973	23197	208776	15	0.02	300	0.96	0.92	0.92	0.922	0.921	0.921
10%||90%	231973	23197	208776						0.86	0.86	0.86	0.86

5.3 Discussion

We explored the limitations of social network text categorization in Chapter 2 of the literature review since not much research are carried out on suicidal ideation using social network posts. Earlier text classification research necessitated the creation of manual features from text with a large quantity of labelled data, which necessitated manual intervention.

Text Graph Neural Network performs very well and generate tangible outcomes despite the little data available to train by constructing word-word and document-word graph, the embeddings we have used in this work can retrieve contextual information. This Research also demonstrates that tradition algorithm like logistic regression doesn't perform well compared to text GCN when the volume of train data is less. This work demonstrated the efficacy of Text GCN with relatively less training data, in terms result in less hardware and time require to build and train and deploy model to production.

5.4 Summary

We understood the components required to implement text GCN in the introduction. In the next section we saw the parameters required to construct graph. We further experimented with the different volume of train and test data and the impact on the unseen test data accuracy. Following section, we experimented with varying window size to find the co-occurrence of words in the corpus which is then used to find the PMI values to calculate the weights between word-word edge and document-word edge which in turn was used to create adjacent matrix.In the next section we also experimented on the embedding dimensions and the impact on the test accuracy. In the last section we saw how traditional algorithm gets trained on low volume of train data and performs on the unseen data.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

This chapter is the thesis's final and concluding chapter. It discusses the research work's summary as well as the research's last remarks.

6.1 Introduction

This chapter focuses on a comprehensive evaluation and analysis of the complete study project. In the first portion, the chapter reviews the aims and research questions from chapter 1 and responds to them by explaining the study methods used.

The outcomes addressed, as well as the limitations and obstructions encountered throughout the investigation, are detailed in the following section. We also explore the work's contribution and significance in the field of natural language processing. The chapter finishes with a discussion of the chapter's future scope and recommendations.

6.2 Discussion and Conclusion

In this section, we will explore the queries raised in Chapter 1 and answer to them considering the study findings.

- *What are the common emotional words in the people who has got suicidal ideations?*

From the horizontal bar chart and word cloud (Figure 6.1 & 6.2) of repeated words we can broadly see the pattern in the suicidal ideation text where topics either revolves around one of the below: -

- **Desires**: Suicidal text mainly contains words such as “Want”, “Like”, “Need” & “Deserve”,” Right”,” Use”,” Able” and so on.
- **People**: Contains text mainly contains words such as “Father “, “Mom”, “Friends”, “Family”, Girlfriend”,” Boyfriend” and so on.
- **State of Mind or Body:** Contains text mainly contains words such as “Hurt”,” Depression”,” Sick”,” Ill”,” Anxiety”,” Tired”, “Pain”, “Fear”, “Hate”, “Suffering”, “Failure”,” Die”, “Life” and so on.

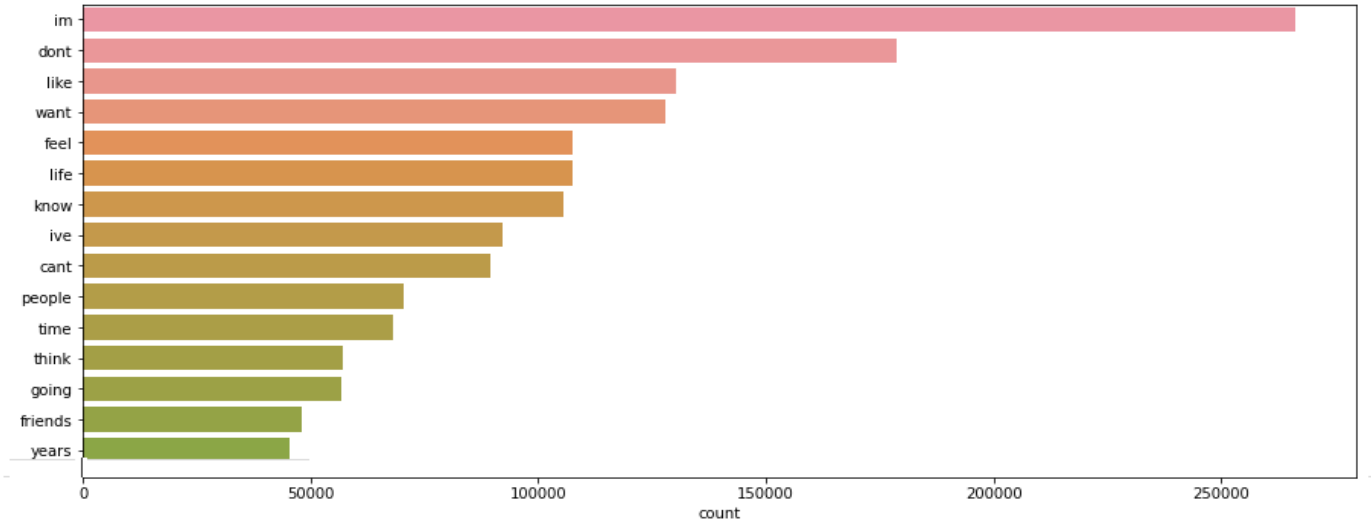


Figure 6.1 Word Count of Suicidal Ideation Text

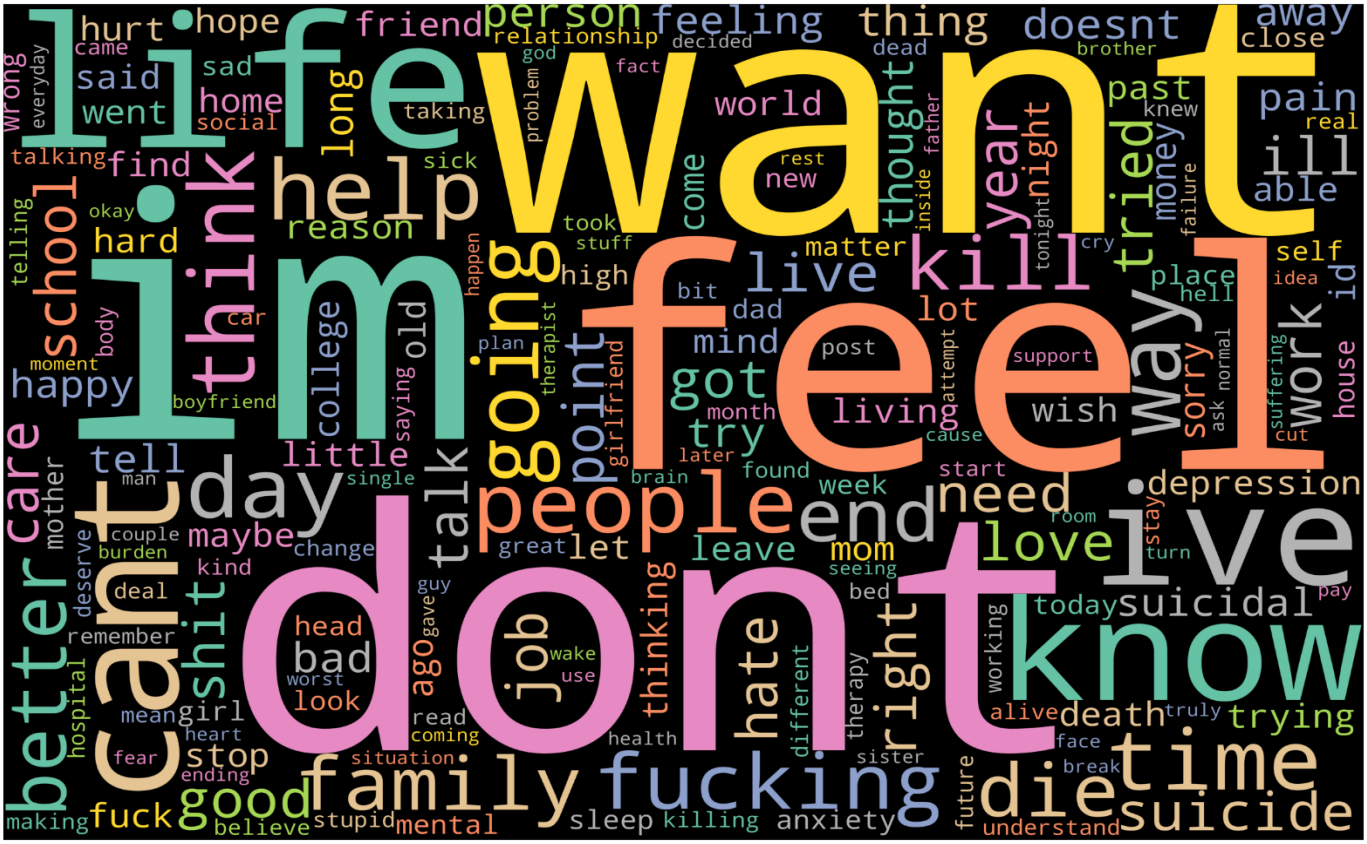


Figure 6.2 Word Cloud of Suicidal Ideation Text

- *Does Text GCN better than traditional classifications models on social media posts?*

From the model results and evaluation metrics, we could see Text GCN with window size of 15 embedding dimension of 300 showed the performance gain of ~7%-8% over traditional models. Even making the model better by 1% can save some one life hence, we infer that the Text GCN perform better on social media post which usually contains huge text.

- *How does Text GCN evaluate on test set compared to traditional models when the size of trained data varies from 80%, 60%, 50% , 30% and 20%?*

The methodology in the research uses huge text which was posted by user as input features. In Text GCN we filter down only those words which appears at least more than 5 times thus removing those words which are not common in the corpus then we use window size of 15 where each sentence will be moved over the window size of 15 to find the co-occurrence between the words.

We use PMI between words indicates that they have a high semantic connection and build edges between words on the other hand, we do not create edges between words with a negative PMI. Overall, TF-IDF-weighted document-word edges capture context inside the document, whereas PMI-weighted word-word edges (which might span several documents) capture context across documents there by understanding contextual, syntactic, or pragmatic features. If we still had lesser training data due to window size text keywords get repeated in the context and even when train data is less word to word connection happens through PMI and document with words happens with TF-IDF. Whereas traditional algorithms like Logistic Regression and Naïve Bayes doesn't consider the contextual embeddings also relationship between words and documents are not found.

As pe the analysis done in the previous chapter by modifying different parameters we see Text GCN can be trained even with 10% of the data and still it can perform very well on the unseen data with this observation we can infer that the Text GCN bring in contextual information which tradition algorithms fails and outperforms the traditional techniques also when Text GCN techniques combined with Neural Network, they give the state-of-the-art performance.

6.3 Limitation

We had problems while working on our thesis, which we attempted to address with the best answer possible given our resources and time constraints. The first issue we ran into was relying on RAM availability. When windows size is moved over the huge sentences it gets replicated multiple times in the set and calculating the PMI and TF-IDF score takes load on RAM. The process of loading and training the word vectors for the full dataset was hindered by a lack of RAM. At times it took more than 6 hours to process the full dataset with 12 GB RAM. We also utilized the less volume of text corpus to solve this problem and trained the model on a smaller train and test dataset in different experiments.

Apart from above technical limitations, another limitation we faced is GCN neural network algorithms require a lot of GPU power, we ran into another issue in terms of extended training. Due to the limited availability of GPUs, thorough training for deep learning algorithms was limited. To get around this, we only trained the models on a small number of epochs and used a small number of hyper-parameter adjustment options. The acquired results are the best in terms of the resource and infrastructure available.

6.4 Future Recommendation

In this part, we'll go through some potential recommendations that might help the classifier perform even better. Text GCN capture global word cooccurrence data and making good use of limited labelled targets. Due to time restriction, we refrained using multiple dense layers and refrained to only 2 hidden layers however using multiple dense layer can improve the performance if used in the future.

Other intriguing future initiatives include enhancing classification performance utilising attention mechanism and constructing an unsupervised 'text GCN framework for representation learning on largescale unlabelled text data. We recommend to explore different word embeddings and different embedding dimensions to construct sparse matrix to see how well they work also we also recommend to explore pre-trained embeddings such glove to construct graphs.

6.5 Summary

The closing remarks regarding the research conducted in this study are presented in this chapter. This chapter started with responding to the research questions posed in Chapter 1. This chapter supports the research technique that was previously discussed. In the next part, we explored the study's limitations, as well as the problems and limits encountered throughout its execution. Finally, the chapter finishes with future recommendations and the study's research possibilities.

References

20NG News Group Dataset. (n.d.). http://qwone.com/~jason/20Newsgroups/

Building a suicidal tweet classifier using NLP | by Israel Aminu | Towards Data Science. (n.d.). Retrieved July 23, 2021, from https://towardsdatascience.com/building-a-suicidal-tweet-classifier-using-nlp-ff6ccd77e971

Cao, S., Lu, W., & Xu, Q. (2015). Grarep: Learning graph representations with global structural information. Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, 891–900.

Corrigan, P. (2004). How stigma interferes with mental health care. American Psychologist, 59(7), 614.

Dehmamy, N., Barabási, A.-L., & Yu, R. (2019). Understanding the representation power of graph neural networks in learning graph topology. ArXiv Preprint ArXiv:1907.05008.

Fouss, F., Pirotte, A., Renders, J.-M., & Saerens, M. (2007). Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Transactions on Knowledge and Data Engineering, 19(3), 355–369.

Goyal, P., & Ferrara, E. (2018). Graph embedding techniques, applications, and performance: A survey. Knowledge-Based Systems, 151, 78–94.

Graham, B. (2014). Spatially-sparse convolutional neural networks. ArXiv Preprint ArXiv:1409.6070.

Hinton, G., & Roweis, S. T. (2002). Stochastic neighbor embedding. NIPS, 15, 833–840.

Kastanos, A., & Martin, T. (2021). Graph Convolutional Network for Swahili News Classification. ArXiv Preprint ArXiv:2103.09325.

Kim, P. (2017). Convolutional neural network. In MATLAB deep learning (pp. 121–147). Springer.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. ArXiv Preprint ArXiv:1609.02907.

Klicpera, J., Groß, J., & Günnemann, S. (2020). Directional message passing for molecular graphs. ArXiv Preprint ArXiv:2003.03123.

Li, Q., Han, Z., & Wu, X.-M. (2018). Deeper insights into graph convolutional networks for semi-supervised learning. Thirty-Second AAAI Conference on Artificial Intelligence.

Liquiere, M. (n.d.). Graal A machine Learning model for some Object approach problems.

Luo, Y., Uzuner, Ö., & Szolovits, P. (2016). Bridging semantics and syntax with graph algorithms—state-of-the-art of extracting biomedical relations. Briefings in Bioinformatics, 18(1), 160–178. https://doi.org/10.1093/bib/bbw001

Moreno, M. A., Jelenchick, L. A., Egan, K. G., Cox, E., Young, H., Gannon, K. E., & Becker, T. (2011). Feeling bad on Facebook: Depression disclosures by college students on a social networking site. Depression and Anxiety, 28(6), 447–455.

Ohsumed Corpus. (n.d.). http://disi.unitn.it/moschitti/corpora.htm

Reddit Suicide and Depression Detection Dataset. (n.d.). Retrieved July 25, 2021, from https://www.kaggle.com/nikhileswarkomati/suicide-watch

Reddit Suicide Watch. (n.d.). Retrieved July 25, 2021, from https://www.reddit.com/r/SuicideWatch/

Rickwood, D., Deane, F. P., Wilson, C. J., & Ciarrochi, J. (2005). Young people﷽s help-seeking for mental health problems. Australian E-Journal for the Advancement of Mental Health, 4(3), 218–251.

Rippel, O., Snoek, J., & Adams, R. P. (2015). Spectral representations for convolutional neural networks. ArXiv Preprint ArXiv:1506.03767.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2008). The graph neural network model. IEEE Transactions on Neural Networks, 20(1), 61–80.

Shaw, B., & Jebara, T. (2009). Structure preserving embedding. Proceedings of the 26th Annual International Conference on Machine Learning, 937–944.

Singh, H., & Sharma, R. (2012). Role of adjacency matrix \& adjacency list in graph theory. International Journal of Computers \& Technology, 3(1), 179–183.

Stokes, J. M., Yang, K., Swanson, K., Jin, W., Cubillos-Ruiz, A., Donghia, N. M., MacNair, C. R., French, S., Carfrae, L. A., Bloom-Ackermann, Z., & others. (2020). A deep learning approach to antibiotic discovery. Cell, 180(4), 688–702.

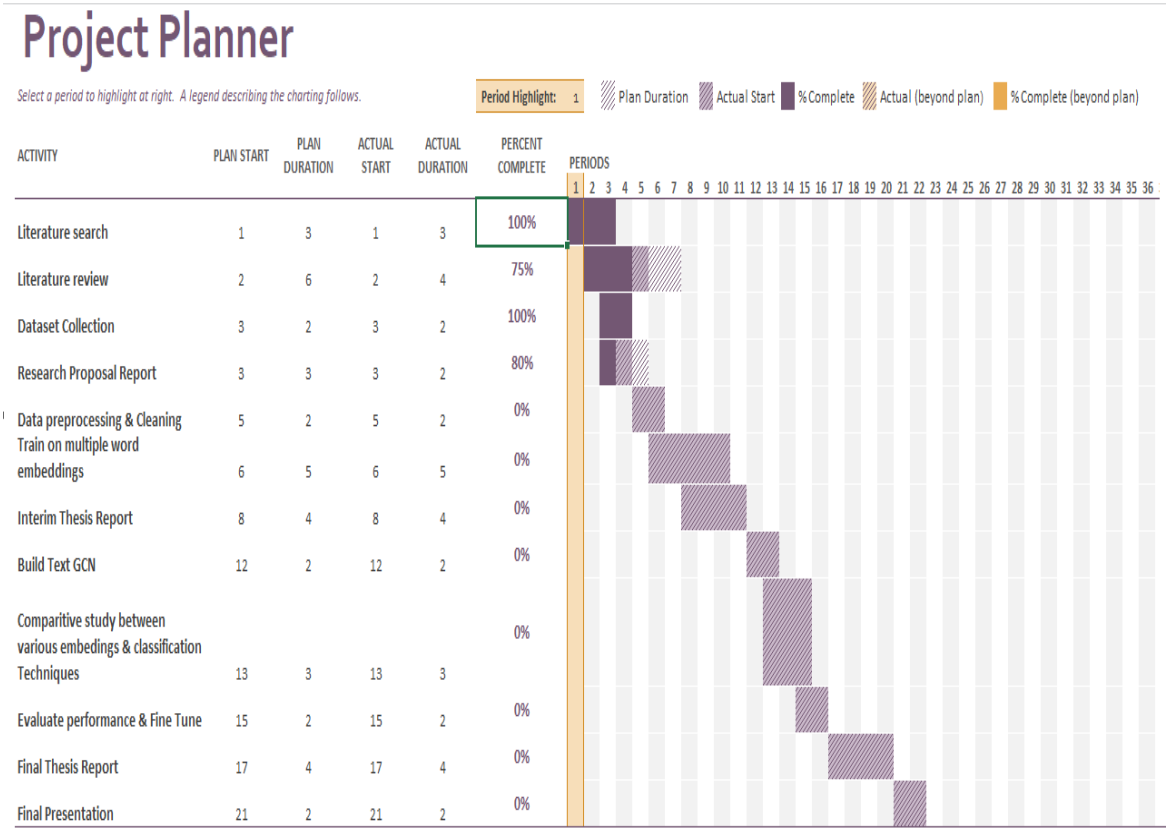
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need.

Wang, G., Li, C., Wang, W., Zhang, Y., Shen, D., Zhang, X., Henao, R., & Carin, L. (2018). Joint embedding of words and labels for text classification. ArXiv Preprint ArXiv:1805.04174.

Yao, L., Mao, C., & Luo, Y. (2018a). Graph Convolutional Networks for Text Classification.

APPENDIX A
RESEARCH PLAN

The timeframes that have been followed up until the mid-term thesis submission are depicted in the Gantt chart below. It also includes a thorough strategy for the final report submission and presentation in the following weeks.



ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Literature search	1	3	1	3	100%
Literature review	2	6	2	4	75%
Dataset Collection	3	2	3	2	100%
Research Proposal Report	3	3	3	2	80%
Data preprocessing & Cleaning	5	2	5	2	0%
Train on multiple word embeddings	6	5	6	5	0%
Interim Thesis Report	8	4	8	4	0%
Build Text GCN	12	2	12	2	0%
Comparitive study between various embedings & classification Techniques	13	3	13	3	0%
Evaluate performance & Fine Tune	15	2	15	2	0%
Final Thesis Report	17	4	17	4	0%
Final Presentation	21	2	21	2	0%

Figure A.1 Research Plan

APPENDIX B
RESEARCH PROPOSAL

Suicidal Ideation Detection using graph convolutional Network

By Sunil Sharanappa

1. Background

Suicide is one of the most pressing health problems in our modern era. Each year, close to 700,000 carry out suicide & suicide is the 4th top cause of death amongst young people (15 to 29 year) as per WHO 2019. (Suicide, n.d.). Especially in this pandemic suicide rate has increased due to people stressing a lot about their career and family. Many factors can cause persons to take the extreme steps, for instance, personal problems, such as anxiety, despair, impulsivity or factors, like social seclusion and adverse events in life, and also suicide attempts done in the past, etc.

Millions of peoples in the world are victims to suicide each year, becoming the prevention is an important world-wide community health goal. There are increasing proofs that the Cyberspace and social media can affect suicide connected behavior (social media and Suicide - Wikipedia, n.d.). Suicidal Detection indicates whether someone has suicidal ideation or thoughts by identifying the textual content which are created by the individual. Since the increase in social media platforms & anonymity, lot of people interact with other people on the Internet. Online mediums have now become way for individuals to vent their suffering, feelings, & suicidal tendencies. Due to this now, online mediums have set out as a inspection tool for suicidal thoughts and mining social platform content to enhance suicide avoidance (Lopez-Castroman et al., 2020) & mental health of the society.

Very bizarre phenomena are rising, online groups are getting into an agreement on self-harms & suicides. For instance, some years back social platform group called “Blue Whale Game” in 2015-2016 which used various tasks of self-wounding & made the people to take extreme step of committing suicide by the finish. Suicide is a pressing social issue & it has taken out thousands of lives each year. Thus, it is very much necessary to detect suicidal thoughts and to prevent suicide before victims end their life. Detection and treatment in early stages are viewed as the effective ways to prevent suicide bids, victims with suicidal thoughts might convey their thoughts of their extreme steps in brief suicide plan, thoughts and also kind of role changing.

Suicidal thoughts recognition is to find the possibilities of motive or events before unexpected tragedy occurs. An analysis done by McHugh(McHugh et al., 2019) showed limitations of thoughts as a screening tool, but also indicated that individual expression of suicidal thoughts symbolizes their psychological suffering. Successful detection of initial signs of suicidal thoughts can identify individuals and open a consultation portal to let social workers or family members to resolve their mental issues.

Social platforms with its psychological health groups have become an new study area in Natural language processing. It gives a good research platform for the growth of new methods and improvements which may bring about a new dimension in suicide recognition and further suicide probability inhibition Marks (Marks, 2019). Choudhury (De Choudhury et al., 2016) researched the change from a mental health issue to suicide thoughts in Reddit platform. He created a propensity score based statistical methods to derive the distinctive markers of this shift. Traditional Social media post classification techniques involve extracting features from the text corpus followed by is a classifier to generate predictions and they are all computationally intensive. The most critical issue with tradition models is with the volume of labeled data. Accuracy of the traditional models gets better with the huge volume of train data but that comes with huge computational overhead, Whereas Text GCN will give better accuracy with a smaller number of labeled data.

The developments in Deeplearning have raised research interest on suicidal thoughts discovery. More novel techniques such as graph neural networks & attention mechanism can also be used for suicidal text depiction learning. Few other techniques such as reinforcement, adversarial and transfer can also be used. For example, knowledge of the mental wellbeing detection can be transferred for suicidal thoughts detection and GAN can be used to generated samples for data augmentation. The main goal of our research is to share information of suicide thoughts on Reddit & twitter (micro blogging) social platforms using Deeplearning model. Our primary work is to look at the Graph convolution network (GCN) with different pretrained embeddings. Techniques. Graph Convolution Neural Networks (GCNs) are designed to operate on irregularly structured graphs, have seen a recent raise in popularity in several domains. Although Social media posts represent a sequence of words, a document contains an implicit graph structure in the form of syntactic and semantic relationships (Mihalcea & Tarau, 2004) GCN is not explored on social media posts. A text corpus can be arranged into graph by making use of both intra-document and inter-document relationships. (Yao et al., 2019) creates graph from the complete corpus while Huang (Huang et al., 2019) creates graph on per each

document basis. We will build Text GCN model for suicidal ideation prediction. Which can be made as part of any online forums and blog for early detection of suicidal ideation.

2. Problem Statement OR Related Research OR Related Work

Unexpected change in behavior is 1 of key suicide warning sign. An individual's suicide thought is bigger if a behavior has changed rapidly, mainly if it is associated to a heartbreaking event, loss, or some big change in life. Considering this in concurrence with social platforms, In which people keep constantly publish messages and deliberately express their feelings. Despite the significance of this application area and Suicidal being one of the major causes of death in the world, there is a scarcity of published work on text classification for Suicidal ideation. Some of the reasons contributing to this under-representation include a shortage of freely available labeled datasets, Due to data privacy policy being implemented by social networking sites it’s not easy to extract some one’s post or messages.

There are different methods to solve text classification problem. Traditional studies primarily focus on feature engineering. They use feature engineering to generate well designed characteristics, such as the bag-of words, ngrams & graph features (Luo et al., 2016).However, these methods need set up text representations manually, which will cause training them more difficult. Besides traditional methods, deep learning methods are also widely used in text classification problem. Traditional deep learning methods for classification can be divided in two. One way of studies gives more attention on training data. These studies proved that reasonable use of word embeddings can achieve great success in deep learning methods for text classification. Wang (Wang et al., 2018) further introduced an attention framework that combined text and text label embeddings. Besides training data, other studies focused on architecture improvement. Two commonly used deep learning models are CNN and RNN. In order to give more flexibility of representing sentence in these deep learning models, attention mechanism (Vaswani et al., 2017) was used as an internal sub-module.

Classification is a basic and critical challenge in natural language processing, A necessary intermediary step during text classification is text representation. There are several traditional ways for text presentation such as IDF-TF vectors, bag-of-words, n-grams and so on. Recently, recurrent neural networks (RNN) have received wide attention from scientists. However, these methods mainly focus on sentence in short distance, but ignore the global word relationships in a whole document also traditional models have a poor performance in some long-term or non-consecutive sequences analysis. Traditional models usually need numerous training data

to reach high accuracy, which cause constraints in industrial applications. Whereas neural networks for text classification (Kipf & Welling, 2016) can even show good accuracy on less training data. When building the graph for Text GCN we consider the document & word as node & it doesn't require any inter-document relationships.

Mental health text classification has attracted more research attention in recent years. Unlike other text classifications, Mental health research need more robust model. We would like to take inspiration from graph neural network and build Our model on text GCN (Yao et al., 2018a) and will be using social media suicidal ideation posts as the data.

3. Research Question

1. What are the common emotional words in the people who has got suicidal ideations and people who don't have suicidal ideations?
2. Does Text GCN better than traditional classifications models?
3. How does Text GCN evaluate compared to traditional models when the size of labeled data varies by 1%, 5%, 10% and 20% ?

4. Aim and Objective

We intend to follow the text graph neural network proposed by Yao(Yao et al., 2018a) and we will create models on Social media(Reddit and twitter) labeled suicidal datasets, where each words or documents will be indicated as a word vector for Text GCN input. Word vectors are numerical representation of text data. We will apply different pre-trained embeddings. We will also vary the dense layers and drop outs in Text GCN on the constructed graph using both words (from vocabulary) and documents.

There are four goals in our thesis:

- To evaluate how Text GCN performs on social media corpus.
- To assess effect of the size of the target data with different sizes of the training data against traditional models.
- To evaluate the results on using different pretrained models while building graph on Text GCN.
- To evaluate the results by varying the dense layers, drop out and learning rates and compare the results with traditional classification models.

5. Significance of the Study

Recognizing the key factors are the initial step in suicide avoidance. Though, the societal stigma around mental health means that individuals might prevent specialist help (Corrigan, 2004). they might become less inclined on fewer formal resources for support (Rickwood et al., 2005).

In Recent Times, online social platforms have turn out to be informal source for research. Exploration has shown that people of these kind of risk are turning towards modern-day technologies (social media, online-forums, microblogging site) to express their hardship without having to face somebody one on one (Moreno et al., 2011). Due to this risk factors and warning indications have seen in these online spaces. We imagine that huge data of individuals' emotions and actions can be used effectively for quick discovery of behavioral variations in risk persons and also it may even help to avoid casualties.

The main contributions of our thesis are twofold. Firstly, empirically compare the Text Graph Convolutional Network (Text GCN) solution with other traditional models. Text GCN was initially developed for news corpus classifications. To our understanding this is the first illustration of a GCN being used for classification on any social media dataset. Secondly Focus on the semi-supervised situation where the number of labels is highly limited, and the compute is restricted.

6. Scope of the Study

The scope of study involves predicting Suicidal Ideation based on Reddit and Twitter posts which are already labeled. Apart data subtext, characters, habits, emotions etc., few individuals may not want to express on social platforms, Especially, when they are immersed in loneliness, misery and in suicidal thoughts, Individuals tend to be isolated and have very low motivations to create content or leave some info on social platform. Due to rapid changes in emotional fluxes, individuals having suicidal motives have a tendency to remove their old posts related to suicidal thoughts, attempting to hide the true innermost intentions. However, the study only focuses on Suicidal Ideation but does not focus on other mental health issues which can also be part of corpus and wrongly labeled either suicidal or non-suicidal.

7. Research Methodology

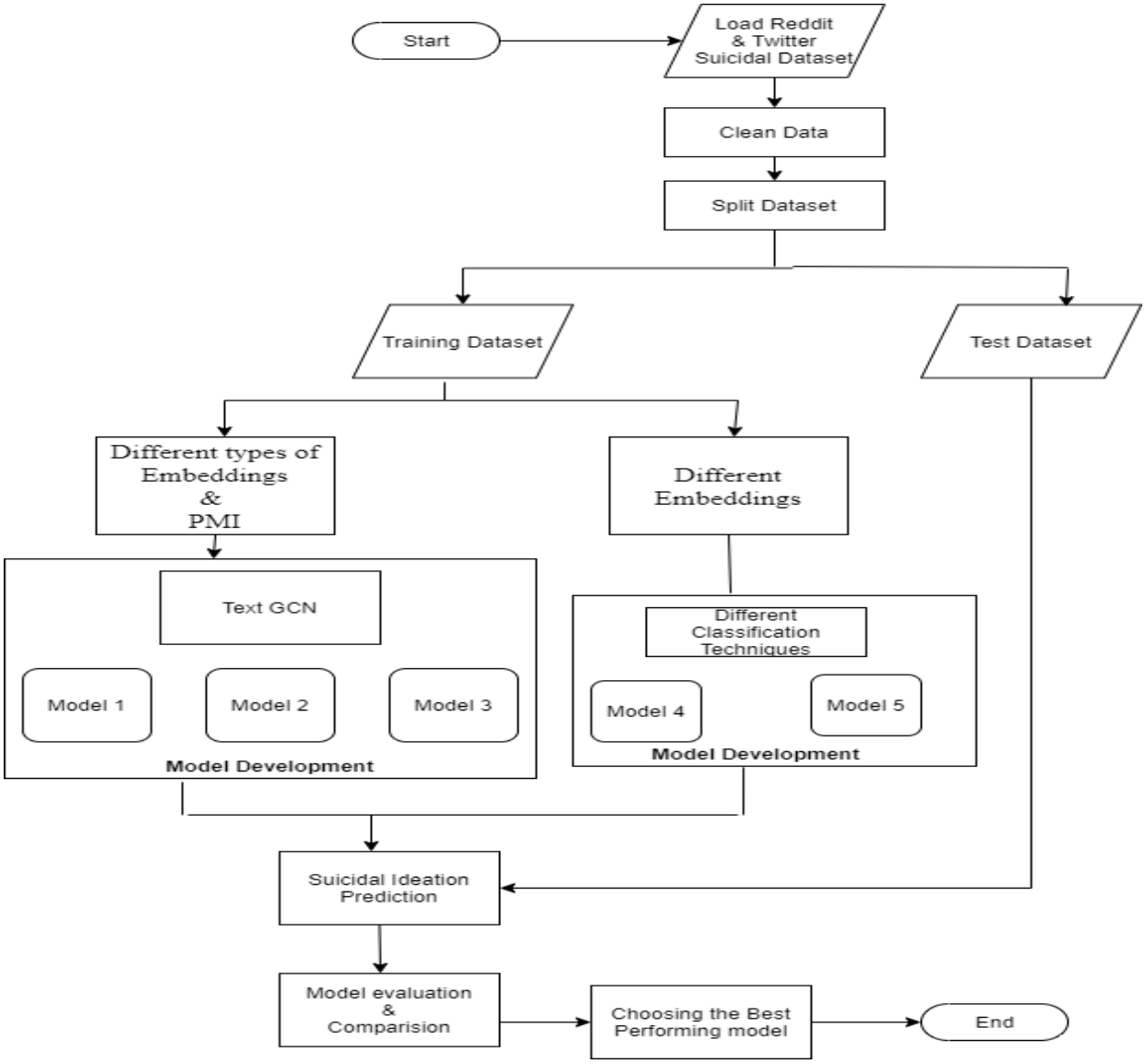


Fig 7.1

a) Introduction

The study consists of a bunch of feature engineering to predict suicidal ideation detection on social media platforms based on text GCN (TGCN) model from Yao (Yao et al., 2018a). This will include loading the dataset, cleaning the dataset, preprocessing with few pretrained embeddings, prediction, model creations, comparison with other models and evaluation. Better performing model going to be picked based on the assessment metrics and analysis will then be decided upon same. Flowchart of Fig 7.1 shows how the proposed approach in which we load the data followed by text cleaning & preprocessing and splitting data in to train and validation set. This step is followed by model evaluation & comparison with traditional classification models for ultimately choosing good performing model to conclude the research.

b) Dataset details

The data is having two columns as follows: -

- **text: -** It consists of the text posted by user on Reddit and Twitter which are to be binary classified.
- **label: -** It consists of 2 groups. It specifies if given text is either have Suicidal Ideation text or non-Suicidal.

text	class
Ex Wife Threatening SuicideRecently I left my wife for good because she has cheated on me twice and lied to me so much that I have decided to refuse to go back to her. As of a few days ago, she began threatening suicide. I have tirelessly spent these paat few days talking her out of it and she keeps hesitating because she wants to believe I'll come back. I know a lot of people will threaten this in order to get their way, but what happens if she really does? What do I do and how am I supposed to handle her death on my hands? I still love my wife but I cannot deal with getting cheated on again and constantly feeling insecure. I'm worried today may be the day she does it and I hope so much it doesn't happen.	suicide
Am I weird I don't get affected by compliments if it's coming from someone I know irl but I feel really good when internet strangers do it	non-suicide
Finally 2020 is almost over... So I can never hear "2020 has been a bad year" ever again. I swear to fucking God it's so annoying	non-suicide
I took the rest of my sleeping pills and my painkillersI canâ€™t wait for it to end, I ve struggled for the past 6 years and I am finally ending it.	suicide
Can you imagine getting old? Me neither.Wrinkles, weight gain, hair loss, messed up teeth and bones, health issues, menopause, hormones, hating new generations & the way world progress.Being a useless angry piece of shit who can't take care of itself. Being totally depended on people who secretly wants you to die already.Can you even imagine yourself there? Absolutely not. Even if I was happy, I'd take my life just to avoid this.	suicide

Fig 7.2 The actual data can be seen in below screenshot:

According to our plans, we plan to run our experiment combining two different datasets

a) Reddit: This is an online group that aggregates online discussion & societal news. It contains lot of categories and area of interest in a topic is called a subreddit.Subreddit called “Suicide Watch”(*Reddit Suicide Watch*, n.d.) is the area which we have considered for our thesis.

Our dataset is a collection of posts from “depression" and “SuicideWatch" subreddits of the Reddit platform. The posts are collected using technique called Pushshift API. All the posts that were made to "SuicideWatch" from Dec 16, 2008(creation) till Jan 2, 2021.(*Reddit Suicide and Depression Detection Dataset*, n.d.) are part of the dataset.

b) Twitter:- Dataset contains about 8200 rows of tweets., Data was extracted from Twitter API using query parameters such as Hopeless, Depressed, Vow to take care of, “I don’t belong here”, “Nobody deserve me”, “I want to die etc” (*Building a Suicidal Tweet Classifier Using NLP | by Israel Aminu | Towards Data Science*, n.d.).

c) Data Processing

In the preprocessing, we have to take out the unwanted symbols and unwanted characters and emojis which may not be helpful in prediction of the target, If there is any completely missing values in class predictor that needs to be removed as well.

d) Dataset Split

We will be going with train test split of 80 – 20 % and will be using stratified split to make sure the class labels are divided proportionally to maintain bias & variance which will help in providing better models and also helps in avoiding from overfitting.

e) Creating Word vectors with pre-trained word embeddings

We initially preprocessed our data by tokenizing & cleaning text corpus. We will be removing stop words and low frequency words appearing less than specific number of times. Each word or document are indicated as word vector as Text GCN input. Word vectors are generally numerical representation of the text data. We will then apply and explore below pre-trained embeddings.

- Tf-Idf : It is statistic meant to show how important a word is to a document in text corpus. It is also used as a weighting component in searches of information recovery, user modelling and text mining.
- fastText: It is a simple and efficient text classification method proposed by Juolin [5]. The document embeddings are represented by the avg of word or ngrams. The classifier is linear. Two baselines with bigram and without bigram are evaluated.
- Word2Vec:- This creates vector space which are of several thousand dimensions, with every distinctive word in the text corpus such that words that have common perspectives in the text corpus lays close to each other in the vector space.

f) Models

We will be building edges amid nodes based on word presence in documents (document-word edge) and word co-occurrence in the whole text corpus (word-word edges). For the weight of the edge between a document node and a word node we will be using different embeddings as mentioned in the previous section. To use word co-occurrence information, we will be using fixed sliding window size on all documents in the text to collect co-occurrence statistics. We will use point wise mutual information (PMI), which is most known method to gauge for word connections, to calculate weights among 2-word node.

By use PMI, the graph will capture the correlation among some words with higher dependency. Also, for each node itself, we need to add a self-loop on it. After creating the graph, we should feed the graph into a GCN as in (Kipf and Welling 2017), by varying the dense layers , learning rates and also by ensuring embeddings are of the similar size as the label set, this then put across a SoftMax activation layer. We will compare the TGCN model built with different traditional classification models such as logistic regression and Naïve Bayes. We will also visualize the word embeddings using t-SNE (Hinton & Roweis, 2002). The doc embeddings of dataset in different layers with many dimensions learned by TGCN.

g) Evaluation

When the models are built, we will make use below ways to assess:

- Accuracy, which is one of the good metric to decide classifications on balanced dataset, we will compare the different models Accuracy as part of comparative study by varying number of training epochs for different learning rate.
- We will also vary windows size and calculate accuracy on TextGCN model.
- We will also measure F1 score for all models and we will plot ROC which is plotted on TPR Versus FPR. This helps to describe the model closest to the threshold is best one to choose.

8. Requirements

We will be using the Python 3.8 above along with any of the deep learning frameworks like Pytorch, Keras and Tensor Flow. We will be working on huge text corpus so any of the below mentioned GPU powered computing infrastructure would be required.

- Google Colab, with Tesla K80 GPU 12GB and CPU 12GB;
- Floydhub, with Tesla K80 GPU 12GB and CPU 61GB.
- Nimblebox with GPU 12GB and CPU 61GB.