Title of Your Project

Comprehensive Project
Semester VIII

A Project Report Submitted in Partial Fulfillment of the Requirements for Award
of
the Degree of Bachelor of Technology in Information and Communication
Technology

Submitted by
Your Name
Roll No. Your Roll No

Under the Supervision and Guidance of
Dr. Paawan Sharma
Professor in ICT Department
Pandit Deendayal Energy University

Under the Supervision and Guidance of
Dr. Santosh Kumar Bharti
Professor in CE Department
Pandit Deendayal Energy University

Submitted to
Department of Information and Communication Technology
School of Technology
Pandit Deendayal Energy University (PDEU)
Gandhinagar, INDIA, 382007

Declaration

I hereby declare that the project work entitled **"Project Title"** is an authentic record of my own work carried out in xyz industry/xyz university/xyz research institute as requirement of B. Tech dissertation for the award of **Bachelor of Technology in Information and Communication Technology**. I have duly acknowledged all the sources from which the ideas and extracts have been taken. The project is free from any plagiarism and has not been submitted elsewhere for any degree, diploma and certificate.

Signature:...............................

Your Name

(Roll No.)

Department of Information and Communication Technology
PDEU

Dr. Ganga Prasad Pandey
H. O. D. e-mail:

Certificate

This is to certify that the project entitled **"Title of the Project"** submitted by **Your Name**, Roll No. xyz123 to the Department of Information and Communication Technology under School of Technology, PDPU in partial fulfillment of the requirements for award of the degree of **Bachelor of Technology in Information and Communication Technology** embodies work carried out under the guidance and supervision of Dr./ MR. XXX YYYY, designation of Industry mentor/Professor, xyz industry/Department of XYz.

....
Dr. Ganga Prasad Pandey
(HOD, I.C.T. Department, PDEU.)

Industry Letterhead /XYZ industry/ Dept of ICT, PDEU

Address

DR./Mr. xxx yyy
Designation of Industry
mentor/ if Inhouse write about
Inhouse mentor

Phone: xxx - xxxx
Fax: xxxx - xxx
e-mail: ABC@xyz.com

Certificate

This is to certify that the project entitled **"Project Title"** submitted to the Department of Information and Communincation Technology under School of Technology, Pandit Deendayal Petroleum University in partial fulfillment of the requirements for award of the degree of **Bachelor of Technology in Information and Communication Technology** is a record of work carried out by **Your Name**, Roll No. XYZ123 under my supervision and guidance in the "Name of the Industry", "address of the industry", "Pin No".

Signature ....
Name of the supervisor(industry or PDEU):
(Designation)
Phone No:
Email ID:

Industry Letterhead /XYZ industry/ Dept of ICT, PDEU

Address

DR./Mr. xxx yyy
Designation of Industry
mentor/ if Inhouse write about
Inhouse mentor

Phone: xxx - xxxx
Fax: xxxx - xxx
e-mail: ABC@xyz.com

Certificate

This is to certify that the project entitled **“Project Title”** submitted to the Department of Information and Communincation Technology under School of Technology, Pandit Deendayal Petroleum University in partial fulfillment of the requirements for award of the degree of **Bachelor of Technology in Information and Communication Technology** is a record of work carried out by **Your Name**, Roll No. XYZ123 under my supervision and guidance in the ”Name of the Industry”, ”address of the industry”, ”Pin No”.

Signature ....
Name of the supervisor(industry or PDEU):
(Designation)
Phone No:
Email ID:

Certificate of Approval

The forgoing project entitled **“Project Title”** submitted by **Your Name**, Roll No XYX123 to the Information and Communication Technology under School of Technology, PDPU is hereby approved as project work carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite of **Bachelor of Technology in Information and Communication Technology** degree for which it has been submitted. It has been understood that by this approval of the undersigned do not necessarily endorse or approve every statement made, opinion expressed or conclusion drawn therein but approve only for the purpose for which it is being submitted.

. . . .
Signature of Panel Members

Contents

Contents **vi**

1 Introduction **1**
1.1 Problem statement 1
1.2 Motivation 1
1.3 Objectives 2
1.4 Literature review (Contd.) 3
1.4.1 Tree LSTMs with Convolution Units to Predict Stance and Rumor Veracity in Social Media Conversations 3
1.4.2 Have you lost the thread? Discovering Ongoing Conversations 4
1.4.3 Skip Thought Vectors 6
1.4.4 Knowledge Graph Embedding: A Survey of Approaches and Applications 7
1.4.5 Knowledge Graph Augmented Network Towards Multiview Representation Learning for Aspect-based Sentiment Analysis 8
1.5 Approaches 10
1.5.1 KG + Topic Discovery - Unsupervised 10
1.5.2 KG + Topic Discovery - Self - supervised 11
1.5.3 Hierarchical RNNs - Word Level and Sentence Level 12
1.5.4 Tree LSTMs 13
1.5.5 Attention based Graph Neural Networks 14
1.6 Data Collection (Contd.) 15

2 Experimental Methods and Results **19**
2.1 Methods 19
2.2 Algorithm 19
2.3 Results 19

3 Discussion and conclusion **21**

Bibliography **22**

Chapter 1

Introduction

1.1 Problem statement

"Identification of the critical point in a conversation and use it to identify and prevent cancel culture using Natural Language Processing techniques"

1.2 Motivation

It is not uncommon to observe how differences in the eccentricities of people to converse and present ideas conduces to cause confusion or disagreement even when the topic is something that both the individuals are quite familiar with. This phenomenon is almost never a sudden jump from a categorization like "**Strongly agree**" to "**Strongly Disagree**" but rather there are many such sporadic instances/points sprinkled throughout the conversation that lead the other individual into believing that they are against them.The aim of the undergoing study is to identify these clues and help mitigate such situations proactively.

If left unattended, such conversation could become lopsided easily and in the process, as people vehemently express their opinions – people with differing views might feel or even be subject to some verbal attacks or scrutiny.Such conversations are practically dead ends where they **could have been fruitful if moderated properly**.

Cancel culture or call-out culture is a modern form of ostracism in which someone is thrust out of social or professional circles – whether it be online, on social media, or in person. This is getting quite exorbitant these days because of the anonymity bestowed upon us by the current online platforms that affords them this extraneous power over others. We do realize that such events can have its own consequences and do surely take a toll on the victim mentally and emotionally.

One of the key features of this project is to **identify and quantify the latent patterns** in the conversational data. This study could be a general eye-opener which will help us quantify the internal biases (confirmation bias, selection bias, etc.) that we subconsciously fall prey to. These insights will help us better understand how we communicate and whether we are victims to such biases.

This project takes off as a general idea/study, but we would hone in on a particular problem for validating our ideas. We would try to give special emphasis on the **generalizability of the project to the extent such that all we would need are little tweaks** could be transferred to other domains as well.One such example could be, a model that is trained to identify the vagueness or relevance of the sentences can later on be used a mould for a model that needs to predict whether there is scope for contention or conflicts later on due to these misunderstandings?

The study revolves around modelling a critical point that would be pivotal to understand these complex social interactions online.Complex in nature as they can be viewed very differently depending on the context of the problem at hand:

- Points of collaboration in a conversation
- Resolving conflicts
- Facilitating discussions that are vague
- Steering the conversations - amplify the seemingly weaker ideas
- Preventing cancel culture
- Preventing Groupthink
- Identifying suicidal tendencies

Hence, having a common ground for analysing such situations could be a great starting point for researchers in various fields such as Natural Language Understanding Computational Linguistics,Social Sciences,research psychologists A system that can be used to transfer its learned insights from one domain to another would be interesting to observe - maybe some new insights that we did not know earlier could be brought to light.

1.3 Objectives

Modeling the entire conversations in a computationally efficient manner, preserving the various relationships(both intrinsic and extrinsic) between the sentences of the conversation. the so that we can later on perform analysis on the various kinds

of conversations and henceforth use these observed patterns to train a generative model.
Once we have a set framework of how to store these sentences we can have our own custom metrics and study all the various categories of conversations - Conflicts/collaboration/mental health related, etc. and try to encode them into a single metric that we would aim to optimize.This final metric will take into consideration all the features(Relevance to the topic, history, sentiment etc.) that will be extracted from the EDA of the conversations.
This critical point in a conversation can help us pinpoint the causes(sentences) of contention and groupthink, and thereafter predict a sentence that could moderate the discussion which otherwise would have turned into a heated argument or herd thinking."

1.4 Literature review (Contd.)

1.4.1 Tree LSTMs with Convolution Units to Predict Stance and Rumor Veracity in Social Media Conversations

This paper proposes a better data-structure, **Binary Constituency Trees** for models dealing with conversations that takes into account the reply context that is important, which often diminishes with time as replies keep on stacking on each other. Tree LSTMs were already employed in [3], but they did not have hidden convolution units that would essentially be called recalling some parts of the previous hidden states in the LSTM as done by the authors [1].

The authors suggest that it would be good to use BCT over other previously work with tree LSTMs. **Branch LSTMs** [2] where the input to the LSTM were branches of trees, was not efficient because there was repetition of nodes and also no communication between the branches. [3] makes modification to the LSTM to accommodate unknown number of inputs by doing a child sum operation on the child nodes.

The current paper generalizes this further by performing more operations like Max Pooling of the child nodes. They propose that we use some fixed number of children for every node and include some important context-sentence like the title of the conversation/thread,if the task needs it. If the parent node has more than 2 children then another split is made as it is to be kept binary.

Idea : **Would it be worth to use Multiway trees in this scenario**? Even though we have these properties for BCTs but I feel there is still an overhead of storing the replies as each node can have only 2 children.As long as we preserve the dimensions of the hidden states h_i

The paper uses these representations:

1. **Glove Embedding** used for the words and the mean as a surrogate for sentences.

2. **BERT** - They tried to fine tune a pretrained BERT model because BERT is not a ready to use model to generate embeddings in its original form It is rather a model that can be tuned for a task.

3. **Skip Thought** - The model uses a neural-network that takes sentences as input and generate a 4800 dimension embedding for each sentence.

4. **Deep Moji(EMT) Embeddings** - Deep Moji is a neural network model that takes sentences as input and outputs a 64 dimension feature vectors.

5. **SKPEMT** - Concatenation of both Skip Thought and EMT embedding.

The model in the paper was trained with cross-validation, with an objective function that is basically the **sum of cross entropy losses over all the nodes**.The learning algorithm was the standard **Stochastic Gradient Descent**.The metrics used for comparing the various combinations was **F1 Scores - Mean and Macro**

One of the observations was that the oversampling of the minority classes(for stance labels) was done to tackle class imbalance, for rumor labels - oversampling lead to changing in the structure of the tree hence they did not go through with it.For Tree LSTMs : Everything is same except that to enable multi task learning step wise training was done that alternate between rumor objective and stance objective.

The author concludes that for the task of the stance classification the Tree-LSTMs perform better than BCT-LSTM.For rumor classification the Tree LSTMs with the variation of using hidden convolutional units and max-pool units.

1.4.2 Have you lost the thread? Discovering Ongoing Conversations

This paper seems to be more relevant when seen in terms of real time intelligent systems that deal with conversations over many platforms simultaneously. The authors are hence trying to **link various isolated stored ongoing conversations**.The objectives of this paper include presenting a test bed for the novel task.

They further describe/justify how the task of discovering ongoing conversations is a novel task by contrasting it to the **automatic extraction of scripts**(where scripts are sequence of events that took place, for example "A search B," "A arrest B," "D convict B," "B plead C," and "D sentence B" where the roles are A = Police, B = Suspect, C = Plea, and D = Jury) from corpora, **dialog disentangling**and recognizing **textual entailment** or **natural language inference**.One of the fascinating features of the study is that they justify that the classical plays area a good surrogate for the real world conversations.

They explicitly stated that they have tried to come up with a context unaware model, hence the focus is on the lexical low-level features like :

1. **Simple features** such as the **average length of turns** (turns are the coment-reply pairs), **the questions per turn**, and the **exclamation marks per turn**

2. **Complex features** such as **"that" per turn** and **"it" per turn (economy features)** and **Type/Token ratio (elaboration features)**. The Type/Token ratio is an important feature as it describes how repetitive dialogues are: higher values for less repetitive dialogues.

They used these linguistic markers to justify that the similarity between corpus of online forums and that of the surrogate data - Plays. It is also further confirmed by a closer analysis of the **distribution of Type/Token** and **tested using Kolmogorov-Smirnov test**).One observation that I find particularly useful is that these plays are more or less mutually exclusive in terms of the content, this could or could not be the case with us

Idea : **What if we try and find inner scripts/stories in the conversations? Like is the story matching the outer story?** We might be able to do away with all the sentence space manipulations?

Learning was done using a **passive aggressive algorithm**.They had prepared their own feature vectors :
$x_p = [G(p), \{S(b_i), S(b_j)\}, \{DT(b_i), DT(b_j)\}, \{DS(b_i), DS(b_j)\}]$
Here $G(p)$ are the group features for similarity between speakers $s(b_i)$ and final vector $G(p)$ is normalization of vector $[g_1, g_2, g_3]$ where,

$g_1 = \frac{|s(b_i)s(b_j)|}{|s(b_1)s(b_j)|}$

$g_2 = \frac{|s(b_i)s(b_j)|}{|s(b_i)|}$

$g_3 = \frac{|s(b_i)s(b_j)|}{|s(b_j)|}$

For capturing the syntactic features they used syntax trees and their sub-trees to form a distributed tree (basically a weighted version)

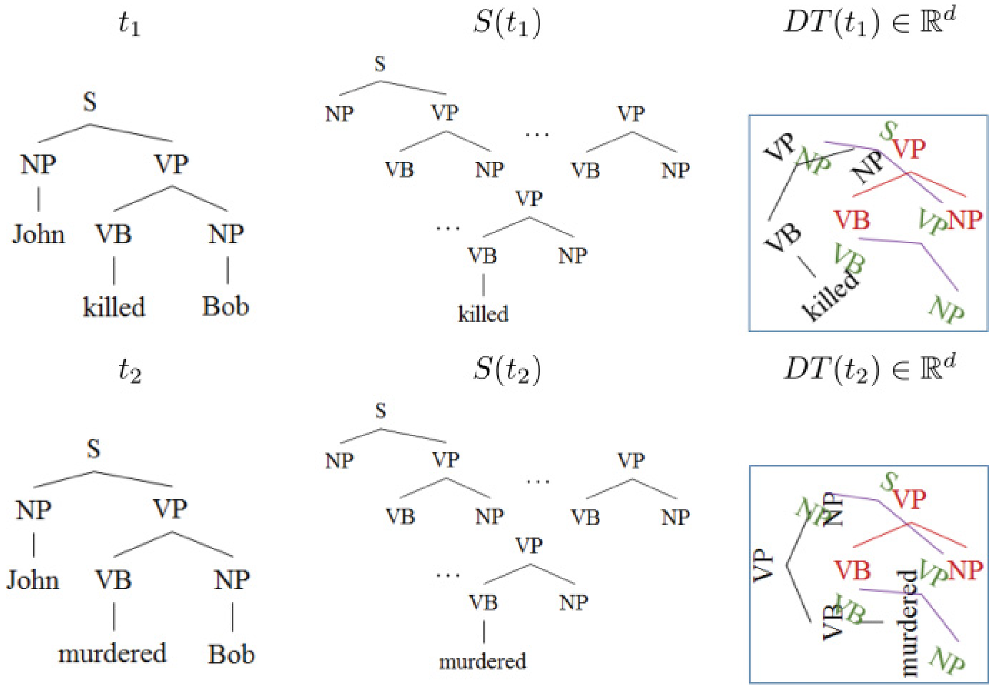

Figure 1.1: Intuition for Distributed trees -The common sub trees are encoded in a vector hence similar sub trees can be counted (intersection of S(ts)). The sub trees are then then turned into distributed trees by weights and this rotated sub trees are then used to see if there are similarities. This final representation is used to model the syntactic features.

For semantic features, the evaluation measure used was **recall@k** as it works well in cases where positive hits are very few compared to the total hits. Finally, **statistical significance** was computed with **Pado's implementation of approximate randomization**.

1.4.3 Skip Thought Vectors

Skip-Thought model generates an encoding for a center sentence and uses that encoding to predict k sentences to the left and right, where k is again the window size as in the skip-gram model.They have used encoder-Decoder architecture from the standard machine translation models with added extra encoders(forward and backwards) for the next and the previous sentence.

According to me, one of the very strong assumptions that were made in the paper was that the sequence of the story captures all the emotions, dialogues and interactions between the characters and the plot. The use of books can for sure capture all the scripts(sequence of events but these features not so much) This may be true but for our problem statement this is not so subtle and yet glaring obvious. So we might need to just look at the idea and the relevance of having a two way decoder for the sentence generation with the window k as the emphasis.The pith of the paper is to capture the context based off the sentences that are in the vicinity of the centre sentence. But that might not work in our situation as the context might depend on something that some random fellow had stated in some other thread of the same conversation.

Idea : **What if, instead of trying to predict the "best sentence" we make the decoder have like 3 possible major variations learn at once?** (Data for such a thing could be easier to get as something like stance/sentiment can filter out the data pretty well).

One example could be : Positive/Negative/Neutral or based on the intensity: High/Medium/Low and then match if the next sentence actually typed is in conjunction with which of the following? This could be tried in form an additional gate with the RNN encoder decoder architecture used? Or some variation of an LSTM altogether?

The training They have used **orthogonal initialization** for the recurrent matrices(the parameters of the RNNs) i.e U W V. For non recurrent matrices they have weights initialized from a normal distribution over [-0.1,0.1] and Gradient clipping for vector exceeds 10. They have used mini-batches of size 128.(I believe is 128 sentences in a batch).

Their feature vector was based on the similar studies before that involved the concatenation of $U.V$ and $|U - V|$.The performance of this new embedding was tested against the standard sub tasks of NLP like Semantic relatedness, Paraphrase Detection, Image captioning, Classification tasks.Only simple tests like Logistic Regression and linear regression were used because they aimed to have the bare new embedding (without any prepossessing) tested against the current baselines.All the models were evaluated using cross validation for the L2 penalty.We see the metric **Recall @ K** again being used for comparison.

1.4.4 Knowledge Graph Embedding: A Survey of Approaches and Applications

Knowledge graph (KG) embedding is to embed components of a KG including entities and relations into continuous vector spaces, so as to simplify the manipulation while preserving the inherent structure of the KG. It can benefit a variety of downstream tasks such as KG completion and relation extraction, and hence has quickly gained massive attention. In this article, the authors have provided a systematic review of existing techniques, including not only the state-of-the-art but also those with latest trends. Particularly, they make the review based on the type of information used in the embedding task. The authors have initially proposed techniques that conduct embedding using only facts observed in the KG . After that, they have discussed techniques that further incorporate additional information besides facts. The paper serves a fair introduction to the application of knowledge graphs in tasks such as triple classification, link prediction, relation extraction, etc.

The authors have tried to incorporate additional representation techniques that could be incorporated with knowledge graph embedding. It consists of including entity types, relation paths and textual descriptions in conjunction with the traditional fact-based embedding. We could test out these techniques for our use-case to validate the propositions put forth by the paper.

1.4.5 Knowledge Graph Augmented Network Towards Multiview Representation Learning for Aspect-based Sentiment Analysis

Aspect Based Sentiment Analysis(ABSA) is a category of text analysis that categorizes opinions by aspect and tries to find the sentiment related to each aspect. For eg.,

"I love the design of this product".

Here the aspect "design" is captured in reference with the word "love", which could denote a positive sentiment.

Most papers focus on sentiment analysis using the semantic and context based approaches. In this paper, the authors have tried to complement the combiantions based upon syntactic, contextual and knowledge-based approaches through an efficient approach. They have proposed a knowledge graph augmented network which has the following architecture:

1. Different types of information are encoded as multi view representations and augmented, thus boosting the performance of ABSA.

2. They have also proposed a hierarchical fusion module to combine the different segments of knowledge in a bottom-up fashion. For example, the context, knowledge and syntactical representations wouldn't't be fused together at once, but would be fused in a pair of two. The resultant representations would be combined to produce a final sentiment score.

Their model has outperformed several existing state of the art features in the domain.

Differentiation from our approach : - Although the architecture is somewhat similar to the one proposed by us, the points of differentiation are:

1. They have used discrete sentences for modelling whereas our work will be modelled on conversations.

2. They haven't used any clustering based approaches for topic discovery.

Takeaways - Can we use the local-to-global fusion module in our context? Vector outputs from knowledge graphs and clustering approaches can produce the variety of embeddings. Though our work wouldn't revolve around extracting semantic representations, we need to assess how different latent patterns can be efficiently derived apart from knowledge-based representations.

1.5 Approaches

1.5.1 KG + Topic Discovery - Unsupervised

We have divided the tasks into two parts : Knowledge Graph that would be use for extracting relations and/or topic discovery form the sentences. The other part involves tracing or profiling the sentences(lexical based patterns) and its relations to the speaker's history.

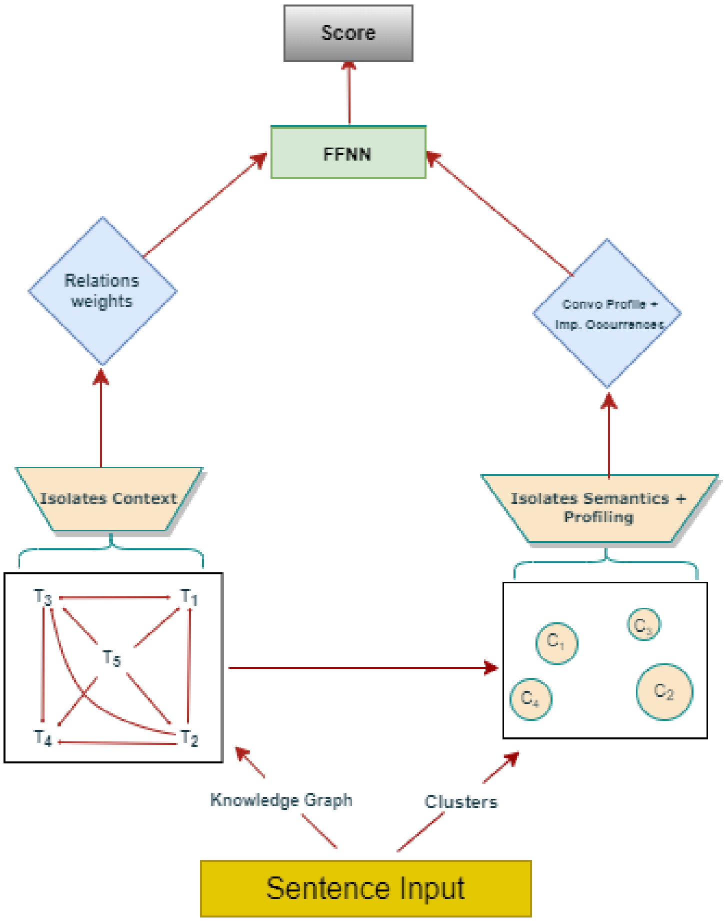

Figure 1.2:

1.5.2 KG + Topic Discovery - Self - supervised

Here we have modified the previous approach in such a way that both the parts - KG and Clusters will try and learn the same patterns. We hypothesise that the KG might not be able to capture the temporal aspects, but can be excellent at relations

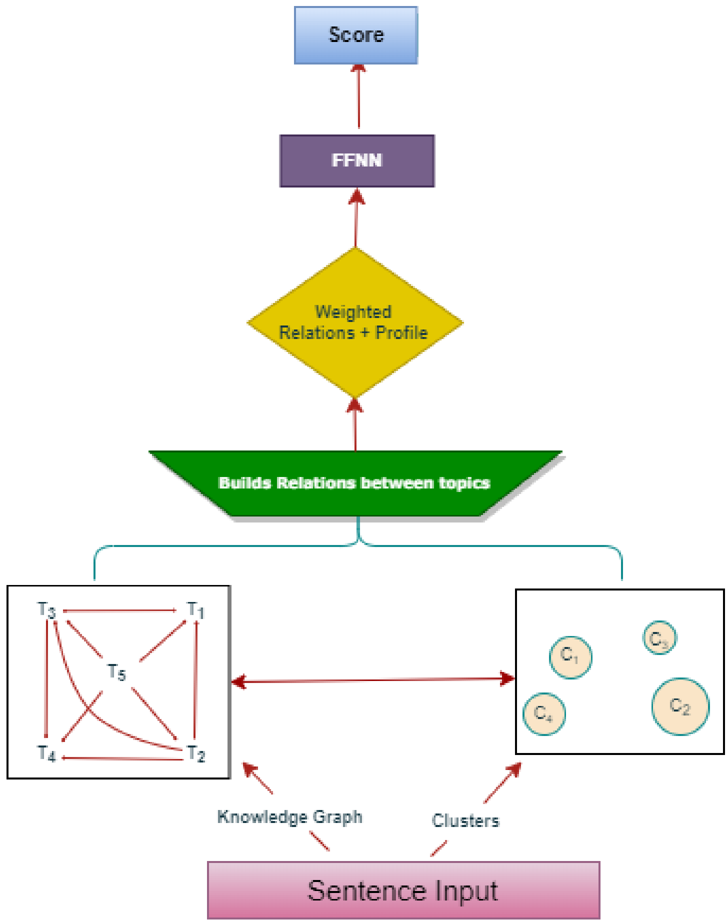

Figure 1.3:

1.5.3 Hierarchical RNNs - Word Level and Sentence Level

We will try to use the typical Sequence(sentences) to Sequence(sentences) generation encoder-decoder architecture.But it would be hierarchical in nature as the encoder would make use of the outer RNN for the word level and the output from these would be used as input to the inner RNN which is at the sentence level.Decoder will again have to be a RNN because we are generating yet another sequence of words - sentence suggestion.

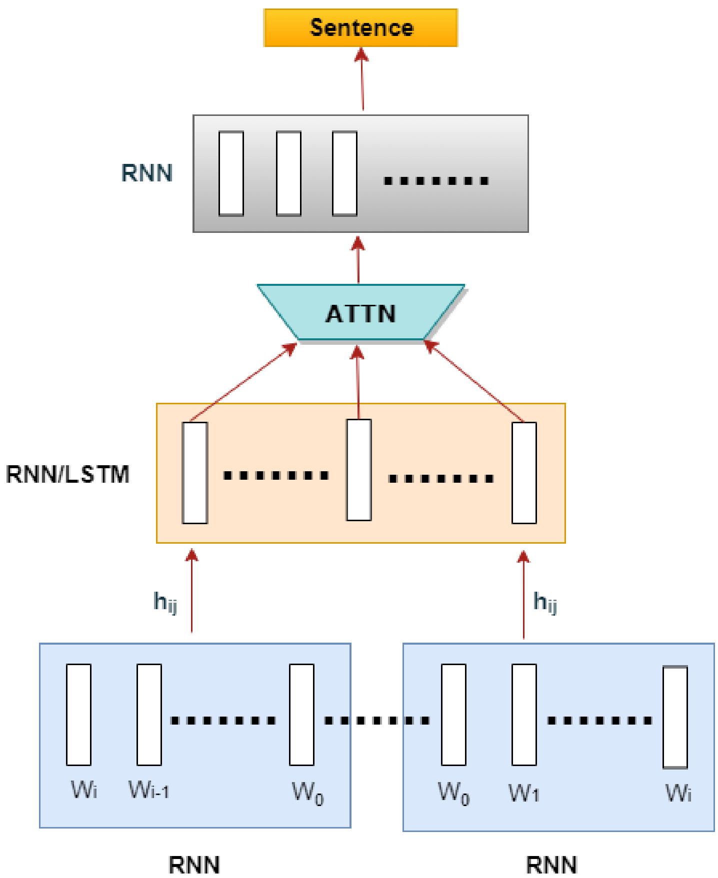

Figure 1.4:

1.5.4 Tree LSTMs

Tree LSTMs work really well when dealing with conversational data that is well represented by a tree structure. The use of recurrent neural networks enables us to have sentences of variable lengths. The LSTMs have two outputs : the state vector h_i and the output vectory_i. The input vector in our case may be the word embedding or the sentences itself (based on experiments and data available)

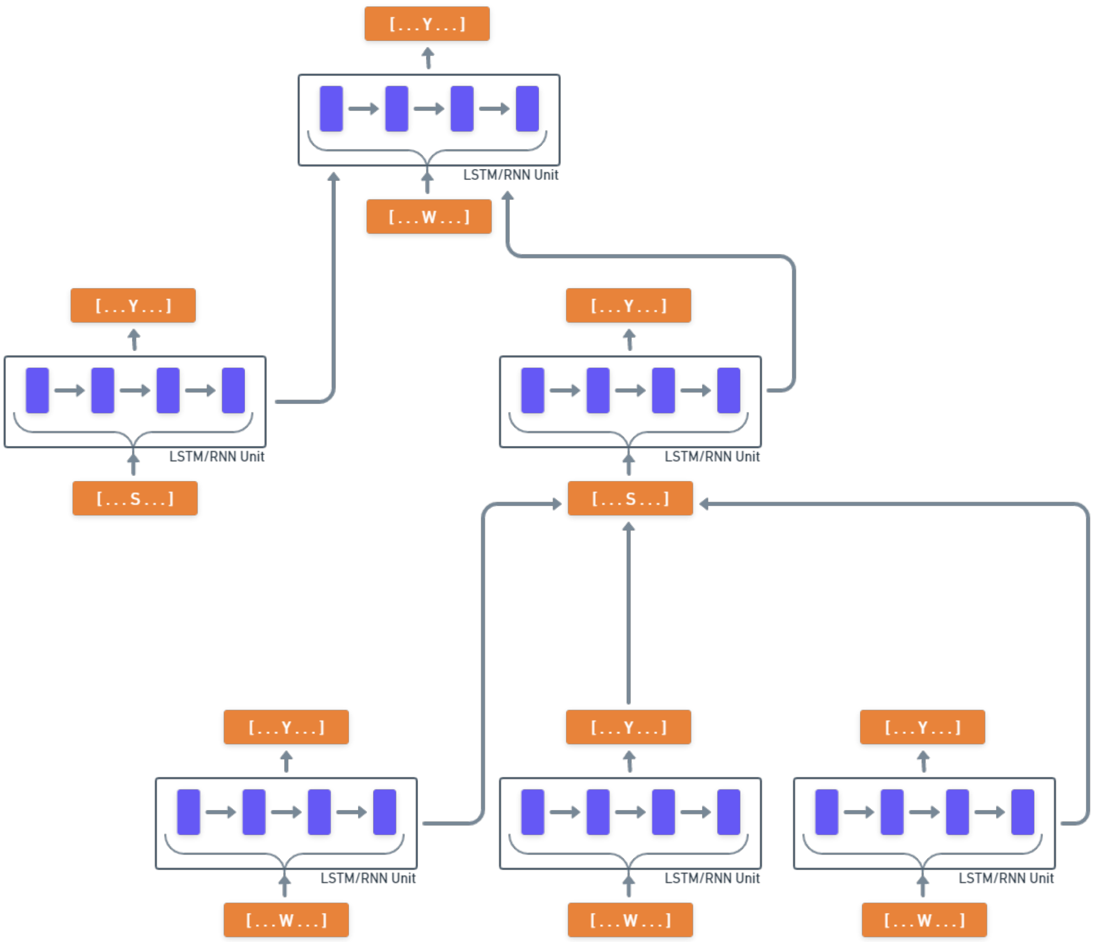

Figure 1.5: A tree structured LSTM with arbitrary branching factor

1.5.5 Attention based Graph Neural Networks

This topic is relatively new from all aspects so we would initially start tinkering with this flavour and as we discover better versions, we might adapt the same to suit the needs of the problem.As we aim to train in an unsupervised fashion we would use only the graph structure: similar nodes have similar embedding. Unsupervised loss function can be a loss based on node proximity in the graph, or random walks.Attention mechanism would make sure that more emphasis is laid on the sentences that the model thinks are most relevant to the problem at hand.

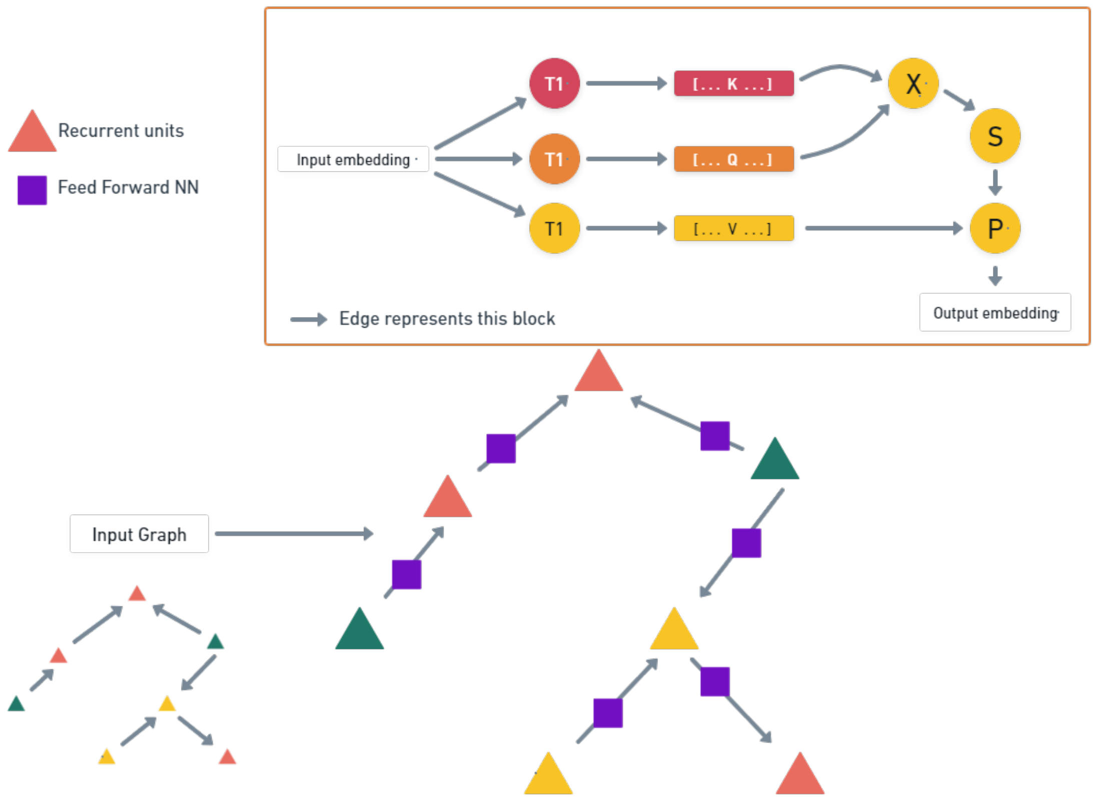

Figure 1.6: A Fundamental structure is formed based on the observations made in the EDA which are then used to tune the graph edges based on the observations seen. The input is the graph embedding at each stage represented by the arrow heads

1.6 Data Collection (Contd.)

In the report for IA1, we demonstrated the code to mine the 'Post' and the 'comments' to it of social media platform 'Reddit'. In this report we are demonstrating the code to mine the replies to the tweets of another popular social media platform 'Twitter'.

```
from TwitterAPI import TwitterAPI, TwitterOAuth,
                                    TwitterRequestError,
                                    TwitterConnectionError,
                                    TwitterPager

import pandas as pd
import tweepy
import time

#authentication
consumer_key= "consumer_key"
consumer_secret= "consumer_secret"
access_token_key= "access_token_key"
access_token_secret= "access_token_secret"


twapi = TwitterAPI(consumer_key, consumer_secret, access_token_key,
                                     access_token_secret, api_version
                                    ='2')


auth = tweepy.OAuth1UserHandler(
   consumer_key, consumer_secret, access_token_key,
                                       access_token_secret
)
api = tweepy.API(auth)


# Placeholder screen name for the handles you want to retrieve
                                    tweets from.
names = ["Reuters"]


""" Retrieves the most recent 200-ish tweets from a  list of given
                                    screen names.
    Returns a data frame of the tweets with [id, screen_name, text,
                                         timestamp].
"""
def get_new_tweets(names):
    print("Retrieving tweets")
    corpus = []

    for name in names:
```

```
        tweets = api.user_timeline(screen_name=name, include_rts=
                                    False, count=200,
                                    tweet_mode="extended")
        time.sleep(4)
        corpus.extend(tweets)
    data = [[tweet.id_str, tweet.user.screen_name, tweet.full_text,
                                tweet.created_at,] for tweet
                                in corpus]
    tweets = pd.DataFrame(data, columns=['tweet_id', 'screen_name',
                                'text', 'timestamp'])

    return tweets


""" Adds conversation_ids to the tweets retrieved from
                            get_new_tweets
    Returns a data frame of the tweets with [id, screen_name, text,
                                timestamp, conversation_id]
"""
def add_data(tweets):
    print("Retrieving additional data")
    ids = tweets.tweet_id
    conv_ids = []
    for id in ids:

        TWEET_ID = id
        TWEET_FIELDS = 'conversation_id'

        try:
            r = twapi.request(f'tweets/:{TWEET_ID}', {'tweet.fields
                                        ': TWEET_FIELDS})

            for item in r:
                conv_ids.append(item['conversation_id'])

        except TwitterRequestError as e:
            print(e.status_code)
            for msg in iter(e):
                print(msg)

        except TwitterConnectionError as e:
            print(e)

        except Exception as e:
            print(e)

    tweets['conversation_id'] = conv_ids
    return tweets
```

```python
# RETRIEVE TWEETS FROM THE SCREEN NAMES(S)
tweets = get_new_tweets(names)
# RETRIEVE CONVERSATION IDs OF THE RETRIEVED TWEETS
tweets = add_data(tweets)
# WRITE THE RETRIEVED TWEETS TO CSV
tweets.to_csv("tweets.csv")
# VIEW THE DATAFRAME HEAD WITH THE TWEETS RETRIEVED
tweets.head()
```

	tweet_id	screen_name	text	timestamp	conversation_id
0	1504097744490835969	Reuters	Biogen says Aduhelm reduces Alzheimer's indica...	2022-03-16 14:10:21+00:00	1504097744490835969
1	1504097654325723136	Reuters	Marina Ovsyannikova, the Russian woman who bur...	2022-03-16 14:10:00+00:00	1504097654325723136
2	1504096482034696195	Reuters	China stocks leap after State Council pledges ...	2022-03-16 14:05:20+00:00	1504096482034696195
3	1504095333021896704	Reuters	Three million barrels of Russian oil, products...	2022-03-16 14:00:46+00:00	1504095333021896704
4	1504095157762764806	Reuters	A Philippine designer makes dresses 👗 and gown...	2022-03-16 14:00:05+00:00	1504095157762764806
5	1504093972607774721	Reuters	House Republicans who challenged Biden's win a...	2022-03-16 13:55:22+00:00	1504093972607774721
6	1504092713561567239	Reuters	Two British-Iranians leave Iran after years of...	2022-03-16 13:50:22+00:00	1504092713561567239
7	1504091448899211265	Reuters	Lloyd's of London fines Atrium Underwriters ov...	2022-03-16 13:45:20+00:00	1504091448899211265
8	1504090188259205125	Reuters	EXCLUSIVE Tesla halts work at Shanghai factory...	2022-03-16 13:40:20+00:00	1504090188259205125
9	1504089119370227713	Reuters	With affordable housing scarce and real estate...	2022-03-16 13:36:05+00:00	1504089119370227713

Figure 1.7: Data frame of the tweets with [id, screenName, text, timestamp, conversationId]

As shown in the figure, the data in the above format gets stored in the "tweets.csv" file. Using the attribute "conversationId", we can mine the replies to the given tweet.

Chapter 2

Experimental Methods and Results

In this chapter....

2.1 Methods

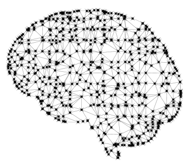

Figure 2.1: The ...

2.2 Algorithm

2.3 Results

Table 2.1: Results

Number	20	30	40
Test	0	1	3

Algorithm 1 Off-line Training of Traditional AutoEncoder(Considering single Layer):

Read the $Training\_Folder$ ▷ # There are 11 files in the Training dataset
for $files \leftarrow$ in $Training\_Folder$ **do**
    $current\_file = files$
    map the data in $current\_file$ to Bit dynamic range

    for $x \leftarrow$ in $current\_file$ **do**
        minimize $L = \|x - x^{/}\|^2 = \|x - \sigma^{/}(W^{/}\sigma(Wx + b) + b^{/})\|^2$
    End_for ▷ # End of first for loop
End_for ▷ # End of 2nd for loop

Chapter 3

Discussion and conclusion

In this work......

Finally..........

The future prospect......

Bibliography

[1] S. Kumar and K. Carley. Tree LSTMs with convolution units to predict stance and rumor veracity in social media conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5047–5058, Florence, Italy, July 2019. Association for Computational Linguistics.

[2] J. Ma, W. Gao, and K.-F. Wong. Rumor detection on Twitter with tree-structured recursive neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1980–1989, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[3] K. S. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. *CoRR*, abs/1503.00075, 2015.