Analysing online conversations to identify critical sub-problems

Report submitted in partial fulfillment of the requirements for the B. Tech. degree in Computer Science & Engineering

By

NAME OF THE STUDENT	**Roll No.**
1. Anurag Pal	*18BCP007*
2. Sagar Sinha	*18BCP094*
3. Sanskar Bhuwania	*18BCP099*

Under the supervision
Of
Dr. Santosh Kumar Bharti

SCHOOL OF TECHNOLOGY
PANDIT DEENDAYAL PETROLEUM UNIVERSITY
GANDHINAGAR, GUJARAT, INDIA

May 2022

Approval Sheet

This Project Report entitled "**Analysing online conversations to identify critical sub-problems**" by **Anurag Pal (18BCP007), Sagar Sinha (18BCP094) and Sanskar Bhuwania (18BCP099)** is recommended for the degree of Bachelor of Technology in Computer Science and Engineering.

Examiners

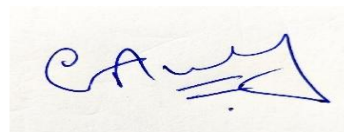
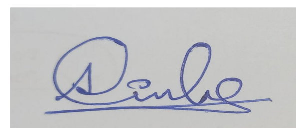
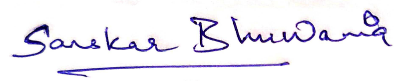
Supervisors

Date : 13–05-2022

Place : Pandit Deendayal Energy University, Gandhinagar

DECLARATION

I hereby declare that the project work entitled "**Analysing online conversations to identify critical sub-problems**" is an authentic record of my own work carried out in Pandit Deendayal Energy University as a requirement of B. Tech dissertation for the award of **Bachelor of Technology inComputer engineering**. 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.

Name of the student	Roll No.	Signature
1. Anurag Pal	18CP007	
2. Sagar Sinha	18BCP094	
3. Sanskar Bhuwania	18BCP099	

CERTIFICATE

This is to certify that the report on "**Analysing online conversations to identify critical sub-problems**" submitted by the students, as a requirement for the degree in Bachelor of Technology (B. Tech) in Computer Science & Engineering, under my guidance and supervision for the session 2019-2020.

Name of the student	Roll No.	Signature
1. Anurag Pal	18CP007	
2. Sagar Sinha	18BCP094	Sinha
3. Sanskar Bhuwania	18BCP099	Sanskar Bhuwania

Date: 13–05-2022

Place: Pandit Deendayal Energy University, Gandhinagar

Signature of the Supervisor

(Dr. Santosh Kumar Bharti)

ACKNOWLEDGEMENTS

The success and final outcome of this project required a lot of guidance and assistance from many people. It is my privilege to pledge the following few lines of dedication to those who helped us directly or indirectly in completing my project.

This work would not have been possible without the constant support, guidance, and assistance of our advisor Dr. Santosh Kumar Bharti and Dr. Paawan Sharma. Their levels of punctuality, knowledge, and ingenuity is something I will always keep aspiring to.

Furthermore, I am thankful to and fortunate enough to the faculties of the Department Computer Engineering for all the knowledge they have imparted, which I could put to use in this project in numerous contexts.They truly laid out the foundations for many of the tasks involved in the project.

Most importantly, last but not the least, I appreciate the cooperation and benevolence of my friends for helping me with the data recording and gathering all the insights from the numerous papers that we have read so far, it would have been a tedious journey to begin with, if it had not been the team members working together as a team.

Name of Student	Signature of Student
1. Anurag Pal	
2. Sagar Sinha	
3. Sanskar Bhuwania	Sanskar Bhuwania

Contents

Chapter 1

Introduction

1.1 Problem statement

"Analysing online conversations to identify critical sub-problems". We believe that it would be much easier to study complex social behaviours, for example - Collaboration or cancel Culture, when one can decompose them into smaller sub problems that are currently being worked on.Examples of these sub-problems would be measuring **Linguistic Alignment** or identifying **Hate speech**. This study could enable researchers to conglomerate ideas from these different seemingly independent research areas.

1.2 Brief of the Problem

Since their advent, social media platforms have been popular. In fact, over the past few years, several people have joined these forums at an increasingly high rate Today, these platforms have proliferated into almost every strata of society. Nevertheless, almost all of these suffer from all or some of the **critical sub-problems** we aim to study, which can have a significant psychological impact on an individual. The prevailing problem of biased ideologies, development of racial and sexist stereotypes, religious polarisation, etc.can all be attributed to the prevalence of the tasks we aim to capture. We have discussed the prevalence of the aforementioned concerns in the context of sub-problems ahead.

There is widespread prevalence of bias on social media platforms. *Tsikerdekis and Zeadally*[1] proposed a data-driven approach to discover language biases prevalent in the vocabulary of online discourse communities on Reddit. We believe that the platform has become a hotbed of biases of various types. There exist various sub-reddits promoting the problem in one form or the other. For example, a popular sub-reddit, called **r/AskReddit**, is synonymous with the promotion of bias, especially confirmation bias *Nickerson*[2]. People are exposed to the "feel good" effect, which is transient and does not serve a great purpose. Hence, capturing bias in online interactions becomes essential.

Online microaggressions are one of the most prominent ways to steer stereotype development. *Breitfeller et al.[3]* defined microaggressions as subtle, veiled manifestations of human biases. It is no surprise that there exist a large number of online servers and chat rooms, which are specifically used to deride a particular community or sex.

Online trolls and hate speech, by far, are the most prominent forms of sub-problems that occur on any social media platform. Hate-speech directly contributes to the problem of polarisation in different aspects of our social culture. Majority of the work in sub-problem detection has been published to study this task.

Linguistic alignment is one of the core components of constructive conversations. However, the opposite, i.e, misalignment, comes up in the case of incoherent discussions. *Niven and Kao*[4] reported the results of preliminary investigations into the relationship between linguistic alignment and argumentation at the level of discourse acts. Misalignment leads to idea misinterpretation, which in turn can heat up the conversation, leading to people vehemently berating each other. As the situation becomes extreme, there is a concern of polarisation in the conversation with people choosing sides.

The task of studying politeness is essentially the converse of what we essentially aim to study, i.e., impoliteness The prevalence of impoliteness is prevalent as one moves up the social ladder. In this regard, *Doyle and Frank* [5] proposed a computational framework for studying linguistic aspects of politeness. They showed that higher status and power is positively correlated with lower levels of politeness and vice-versa on Wikipedia and Stack Exchange. Condescending language is used to demean an individual's ability, which can lower the person's morale and self-esteem. It transforms collaboration between individuals into a toxic battle of who knows more than whom.

Stance is a significant component in conversations. We believe that this task can be enabled to differentiate conversations and debates with far extreme opinions than ones with moderate opinions. For example, the Wall Street journal used stance labels to separate news articles with opposing political ideologies. Through this, they aimed at exposing people to a diversity of opinions.

Deception can be widely found on editorials and op eds published by news agencies, journalists and other individuals or collective groups. There are many instances where manipulated content is shared online, to suit a particular political or ideological narrative. It is a major source of misinformation, and its prevalence can be worrisome in critical times, such as in the ongoing COVID-19 pandemic.

The problem we are working with actually draws from various disciplines - not only limited to computer sciences - Computational Linguistics, Natural Language

Inference, Graph Learning and Geometric Deep Learning but the study would also touch fields of social sciences like psychology or philosophy.

We borrow the insights from the well established models from Computational Linguistics that aid us in looking at textual data in the form of conversations as models as depicted in [6]and [7], also exploring sentence representations and models from the NLU Literature. Learning algorithms are picked up from the field of Geometric Deep Learning and Graph Machine Learning. The problem at hand deals with modelling of the human behaviour that revolves around their interactions on online social media platforms.These models have been established as theoretical models in the social science literature like [8]

The problem at hand is not one of trying to work out a specific problem but rather providing a test-bed or framework that can be used to improve on these specific yet complex problems. The focus of this study is to understand human social behaviour patterns in convoluted social phenomena using sub-problems that we believe if solved would constitute the solution.To the best of our knowledge such a bottom up approach for solving/learning from behaviour problems has not been given a shot yet. The closest someone has ever come across across is [9]

One might argue that this is nothing but an ensemble of various different models, they are not totally incorrect. Indeed, we would like to call it an intertwined learned model. The last feed -forward neural network layer will be responsible to learn the interaction between the various sub problem embedding with respect to the input graph using attention mechanism [10].

1.3 The Task

To define the task, firstly we should formally introduce all the elements that constitute a canonical conversation over the online social media platforms. The study or the framework that we are preparing can be easily extended to have multiple data sources. Let the set of Data sources be $D = \{D_1, D_2, D_3, ...\}$. For our testbed we are dealing with three data sources namely Reddit - D_1, YouTube - D_2 and Twitter - D_3.

Every source D_i is an unordered collection of many conversations/submissions/threads c_i such that $D_i = \{c_1^{'}, c_2^{'}, c_3^{'}....c_N^{'}\}$ where N is the total number of conversations from that source and can vary from source to source. Data from all the D_i are then combined and segregated on the basis of their relevance with respect to the set of selected Sub-Problems $S \in \{S_1, S_2, S_3, ..., S_n\}$. Once separated based on the tests for each sub-problem, each sub problem S_i gets its own data, which is yet another unordered set of conversations c_i 's from the combined data i.e. $D_{S_i} = \{c_1, c_2, c_3..., c_n\}$.

Let's understand and tackle the process with one subproblem S_1 at a time. Consequently, the data associated with it would be D_{S_1}. Each of the conversations $c_i \in D_{S_1}$ are ordered collections of responses/replies/utterances u_i to a post p_i i.e. $c_i = \{p_i, \{u_1, u_2, u_3, ..., u_n\}\}$ where n is the total number of utterances.

Each of the c_i are converted into a graphical representation $G_{c_i} = (V, E, V_f, E_f)$ where V are the nodes, E stands for the edge relations between $\{V_1, V_2, ..., V_x\} \in V$. Each node and edge can have a set of features associated with it and they are represented using the Node feature matrix V_f and E_f respectively.

Edge Types
Asking
Informing
Asserting
Proposing
Summarising
Checking
Building
Including
Excluding
Self-promotion
Supporting
Disagreeing
Avoiding
Challenging
Attacking
Defending
Blocking

Node Types
Agreement
Announcement
Answer
Appreciation
Disagreement
Elaboration
Humour
NegativeReaction
Question

The integral part of the study is to figure out a spatial graph structure or representation $G_{S_i} \forall S_i \in S$. This final graph representation G_{S_i} is an amalgamation of all the graphs seen for S_i and has encoded all the information passed on from the various G_{c_i}graphs.

Let's zoom out a little, now that we have the models for each for the interactions of each of the nodes in the form of spatial graphs G_{S_i} , we can use these standard graphs to compare the unknown graphs G_{c_x} prepared from test conversations c_x and see how structurally similar they are with all the G_{S_i}

Learning the attention parameters (Node-Node) when building the model graph G_{S_i} itself and also the overall attention paid to the some of the particular subproblem grpahs $G_{S_1}, G_{S_3}, ...$ when training the last layer of the architecture.

1.4 Adaptations from existing solutions

One of the approaches that came to mind involved using structured spaces like "Knowledge graphs".We were planning to have a unit that consisted of a knowledge graph encapsulating the information of the context and a clustering sentence embedding space *Kalinowski and An* [11].The information from the former would be used to enhance the clusters in the sentence space. The cluster would encode information about the profile of the people(their interactions so far - Information from history based on the sentences(lexical based patterns)) whereas the context of these sentences was encapsulated in the form of a knowledge graph.

There is yet another variation of the approach that is mentioned above that looks at balancing forces between the two models that are meant to be learning the same structure. One would basically fill in for the mistakes for another.

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 Knowledge Graphs might not be able to capture the temporal aspects, but can be excellent at capturing relations.

Other approaches that we came across that could be adapted to our novel task were the Tree LSTMs *Tai et al.* [12] and their variations involving because the original study did not have hidden convolutional units that would essentially be used for recalling some parts of the previous hidden states in the LSTM as done by the authors Kumar and Carley[13]. The roots of these approaches have been discussed again in section 5 of the report.

1.5 Research Gap

They become really specific and fixated to the problem being solved at hand , ignoring other components that might be actually useful for a score boost. This causes the loss of generalizability, when the problem is seen in the light of other related sub problems. For example, if the researchers are dealing with a complex problem such as that of modelling the cancel culture, then it might help to view the problem along a greater breadth as well. Cancel culture or call-out culture is a modern form of ostracism in which some-one is thrust out of social or professional circles – whether it be online, on social media, or in person.

The reason or the cause of this phenomenon "Cancel culture" [8], [14] prevalent in the current social media can be attributed to many other smaller sub-problems. One might see this problem from the perspective of failure in the moderation of the Hate Speech over the platform. Also, the problem of Linguistic alignment can be seen as a factor in the process. The fact that the discussions online are not coherent and in sync with each other, might lead to confusion followed by misinterpretation of ideas.This could lead to heated conversations. It is important to note that people online are watching and even participating in this hot mess of a conversation that unravels itself. It is natural for people to start taking sides.This would lead to some degree of polarization in the conversation based on whether people agree or disagree with the opinion under observation.

Thus we have some idea of how these various sub problems could affect the cadence of the conversation.In our example, the linguistic dis-alignment when seen as leading towards increase in Hate speech component above makes a lot more sense together.That bundled with Polarization factor could be very useful in understanding how the conversational scene morphs itself into something completely different than what we began with.

At times it is important for us to take a step back, look at how far we have come in dealing with the Natural Language Understanding(NLU) problems.As a part of this study we are trying to aggregate the insights gained from previous attended problems to enhance the problem formulation framework for future research problems concerning people behaviours especially in online settings.

Chapter 2

Literature Review

Data structures Considering the benefits from either of the components, necessitates the form of data structure suited for it. The original textual data that we have, although structured in form of comment trees of long strings of text, was to be prepared for performing any meaningful analysis.

Beginning with the task of producing meaningful sentence representations [11], an efficient and elegant model for producing meaningful sub-word representations was the Bag of Words [15]. This however was not able to capture the sentence sentence interactions between the sentences to a greater extent. Skip-Thought [16] on the other hand solves this problem of context by a window of sentences to and using the centre sentence to predict "k" sentences to the left and right, where k is again the window size as in the skip-gram model.SkipThought works well in capturing the semantic relatedness, however we believe that this embedding could not perform well on our conversational data where there can be a turn of events at every other sentence, as it is trained on a corpus that mostly aligned in nature.Following this are the other adaptations where the focus is more on choosing the correct centre [17], or something that involves direct similarity checks[18].

Knowledge Graphs KGs are very popularly used for semi-supervised tasks, but in our case we were using them as a context placeholder that would take care of the latent topics and the interactions.It would have been tedious to use the same sentence embedding space to capture all these components.Hence we needed a workaround where we could exploit these contextual chunks of information from the knowledge graphs and limit dealing with sentences in very high dimensional space.

*Why might we need a combination of structures?*It is in fact a clever way of modularizing the components efficiently to work with this reduced set of context, however KGs don't seem to really work well with the temporal features in real time. Consider the example where the ideas from this paper are deployed for some particular task. The aforementioned knowledge graph will be built in real time- ad hoc - based on the sentence instances as they arrive. This information is reflected in our static data in the form of timestamps for each of the comments. If this analysis was to be done in real time, we would have to have a substitute for this time based component of the conversation.This necessitates the storage of the temporal information somewhere in the clusters or the other section. The idea behind these proposed models was to have the best of both worlds.KG part of the model structure would complement the user clusters. An essential component of these models is inspired by the rudimentary study [11]

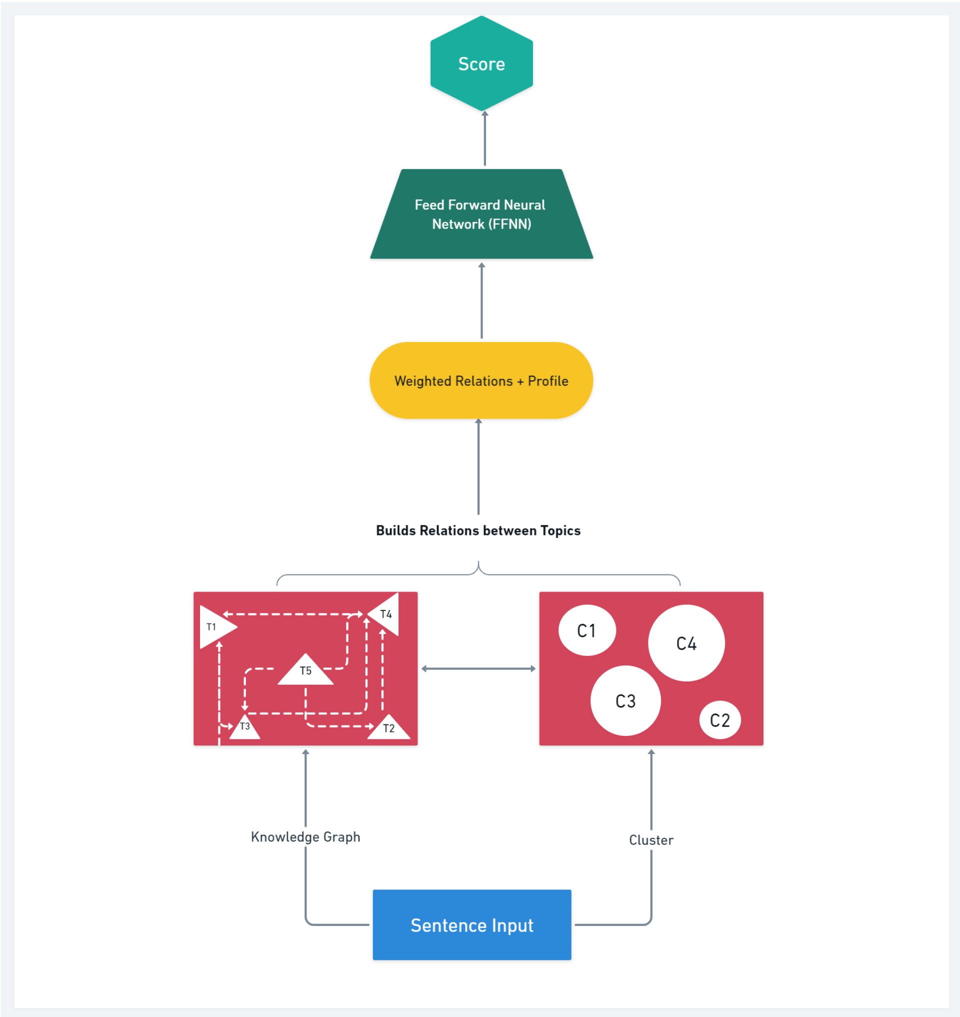


Figure 2.1: Combining Knowledge Graphs with clustering

To tackle this problem we thought of using algorithms that work fairly well with conversational data. One of the solutions that we came across was that using Tree LSTMs [13]. *Kumar and Carley*[13] showed that Tree LSTMs can enable us to use parts of the previous sentences as context present in the next sentences by using specific gates. Another noticeable solution is to use an embedding that works really well with conversational data. *Ahmed et al.*[19] shows that Tree Transformers, using attention along with the grammar information, provides us results that are at par with that of the tree LSTMs, if not better.

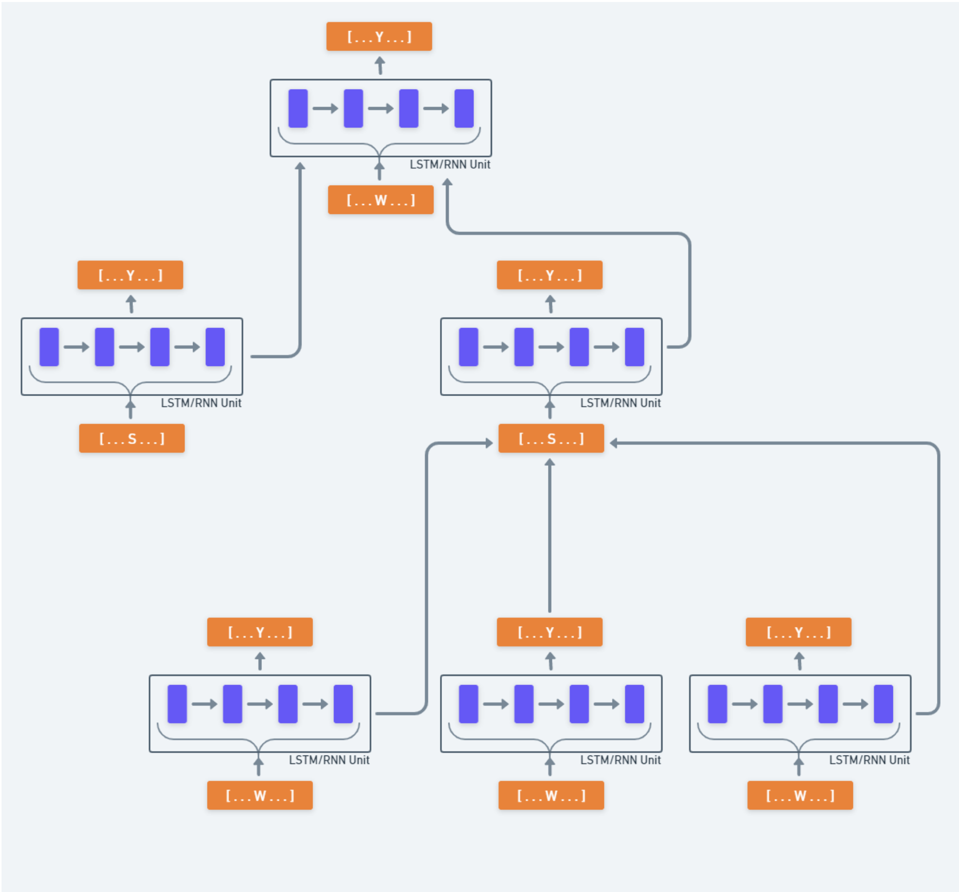


Figure 2.2: Tree LSTMs for working with conversational Data

The recent studies show that GNNs have been extremely useful at most of the patterns based problems. Hence as we are already dealing with knowledge graphs in our theoretical approach we speculate that GNNs would provide us a better model. The problem discussed earlier has been addressed

Subramani and Suresh[20] was particularly interesting as it was a successful attempt at linking unstructured data like Knowledge Graphs (comprising of the conversations insights in our case) to some structured data (like sentence spaces for each conversation).

Hamilton et al.[21] have provided an introductory survey for the application of machine-learning techniques to graph-based problems across a wide variety of domains such as text, images, science, knowledge graphs and combinatorial optimization for node classification, edge prediction and pattern matching. The authors have provided a comprehensive review of existing graph learning techniques, which are categorised into signal processing, matrix-factorization, random walk and deep learning-based approaches. As our task of sub-problem identification includes assigning node labels on unseen data, we have comprehensively surveyed this area.

Bhagat et al.[22] have described the node-classification problem as essentially being a node-labelling task. The essence of nodes is paramount in online social networks(OSNs), where they are used to represent friendship and follower-followee relations. In other cases, they can represent a community, interests and hobbies. We have utilised this task for the identification of discourse-markers, such as announcement, question, answer, etc. in unseen test-instances.

While surveying existing studies in graph-based node classification, we came across two traditional approaches to generalisation on unseen data- transductive and inductive. Here we will be providing a brief overview of these approaches followed by a systematic review of the most major works in this domain.

According to wikipedia, transduction is defined as learning derived by algorithms from specific training instances, and applied to specific validation and test instances. Although these algorithms don't learn the labels of test data, they are aware of the feature-representations of the test dataset since they have been trained upon them. The algorithms need to re-run through the entire dataset to predict the unseen labels. On the other hand, induction refers to the traditional supervised learning approach, where we are completely unaware of the validation and test dataset. This approach employs building predictive models to label the unseen instances.

Zhang et al.[23] proposed scalable Transductive Network Embedding(TLINE), a random-walk based transductive approach for node detection. They trained it on 90% training instances of Database Systems and Logic Programming(DBLP) dataset. *Zhang et al.*[24] proposed the Minimum Tree Cut Algorithm(MTC), and used it for image classification on the MNIST dataset and for web-spam detection on the classical Web-Spam dataset. The work by *Grover and Leskovec*[25] introduces Node2Vec, and the authors have used the model for multi-class node classification on BlogCatalog, Protein-Protein Interactions(PPI) and Wikipedia dataset. The research by *Perozzi et al.*[26], have presented DEEPWalk, a novel method for learning latent representations in a social network. They utilised this novel method for detecting social nodes on BlogCatalog, Flickr and YouTube datasets. *Cao et al.*[27] designed GraRep, a novel method for learning vertex representations of weighted graphs. They conducted their node-classification experiments across three types of networks - 20-Newsgroup language Network, BlogCatalog social network and DBLP citation network. *Lin et al.*[28] have proposed BerttGCN, and used it to construct a heterogeneous graph over the dataset and represent documents in the form of nodes. *Li and Pi*[29] have employed deep learning networks for node classification in social networks. They employed a DNNNC network for their task. The proposed framework could overcome the problem of suboptimal solutions based upon network embeddings+classifier. They have tested the proposed model on three representative network datasets, namely BlogCatalog, Flickr and Quora.

Chapter 3

Building the Dataset

One of the major outcomes of this study would be a research test-bed that can be used for further work in the related areas.This test-bed would facilitate people from the kindred fields to exemplify their works on conversational behaviours.

There are a plethora of options when it comes to these sub problems, but many of them don't really capture the essence of the conversation because they constitute stand alone sentences.In order for us to be able to tap into this latent information, we needed the entire conversations that includes the "Comment Trees".

3.1 Reddit

Reddit as a social media platform can be called a collection of a large number of tight knit communities that share similar interests.These communities exist in the form of sub groups called "Subreddits". A subreddit consists of many threads or posts that are made by the members of the community. Each post on reddit is termed as a "submission" in the API documentation and contains all the comments.Most of the current social media platforms follow the nested comment style interface and reddit in this regard is no exception.

Most of the extraction was done using a python Wrapper around the Reddit API called PRAW and also PushShift API.The catch however was the limit of API calls that could be made using PRAW.This really slowed us down. Pushshift API overcame this situation for us because it has the reddit Data stored in a hosted database. Hence it was much more efficient and elegant to obtain the submissions through PushShift API.

Algorithm 1 Reddit Submission Extraction

Data: API key; list(Subreddits) : subs **;** list(duration): years PSAW custom URL : URL;

Result: A DataFrame of **MetaData** containing : sub_id, title, score, created, num_comments, permalink, flair, text

```
Create custom URL using required parameters;
Create Submission instance ;
Define UNIX timestamps for time period - "AFTER" and "BEFORE";
for subreddit in subs do
    Create a DataFrame ;
    while AFTER submission[created BEFORE do
        Data = get_PSAW_data_in_batches ;
        for len(Data)>0 do
            Extract submission data
        Add each extraction to DataFrame;
        Convert DataFrame to CSV;
```

Once we have collected all the submissions, we would need to weed out the ones that do not have significant information. We discovered that many of the conversations were in the band with <30 comments.Hence after cleaning them out we obtain a DataFrame that contains the submission IDs of choice.

Table 3.1: Reddit External Data Information

Subreddit	2020	2021	2018
r/unpopularopinion	16234	26196	8514
r/roastme	6994	7070	5772
r/unexpected	63700	10034	7643
r/askreddit	45644	50118	55484
r/FreeCompliments	27444	8063	255
r/funny	18597	10857	20266

Earlier we faced a problem of not being able to dig deeper into the comment tree as we were arbitrarily mentioning the level for the for loops.Hence the above code is the final iteration and a more robust solution that utilizes the breadth-first search traversal using a queue. The algorithm above will result into extraction of all the top-level comments, followed by second-level, third-level and so on.

Algorithm 2 Reddit Comment Extraction

Data: API key; submission ID's : sub_id

Result: A JSON file of **MetaData** containing : id, speaker,conversation_id, reply_to, timestamp, text,meta

Setup Reddit API access;
Import file = submissions.csv for a subreddit;
for *submission in file* **do**
    Create a submission instance in API;
    Intitalize a comment_queue for doing a BFS search over the comment tree;
    while *comment_queue* **do**
        comment = comment_queue.pop(0) ;
        Add the comment to the output JSON structure;
        comment_queue.extend(comment.replies);
    Save/Pickle the JSON file to disk;

3.2 YouTube

YouTube is a video-streaming platform that also allows users to leave comments on the videos. Users can also respond to the comments to the video and thus have a discourse. YouTube has a facility of level-1 replies only.

To extract YouTube comments, we utilised the YouTube Data API v3 from Google Developer Console. It provides us with a feature to extract comments by “videoId” as well as “channelId”. We used the “videoId” parameter for our purpose, as we ran out of hits while using the latter parameter. There exists an upper limit on the daily quota of number of responses.

We have tried to consider the major sub-problems that prevail in the field of affective computing.

3.3 Twitter

Twitter : Twitter is a social networking and news website where users exchange short messages/microblogs known as tweets. A tweet has the word limits of 180 characters, so if the message is quite long, users can also post the collection of tweets which is known as thread. Followers can reply to the original tweet/thread posted which can be called as level-1 replies. Level-1 reply can also.There can also be replies to level-1 reply which can be called as level-2 replies and so on. For the sake of simplicity, we extracted only level-1 replies for our project.

Algorithm 3 YouTube Data Extraction

Data: DevKey: DeveloperKey; VId: videoId; parameter: part="id, replies, snippet"; TF: textFormat="PlainText"

Result: A list of **metaData** containing username, comment, timestamp, likes, reply count, reply author, reply, published date, and updated date

```
Initialize request and response with suitable parameters  Execute response
while response do
    for item in response do
        Extract metaData
        if replyCount then
            nextPageToken=None
            while True do
                getReplies(parentCommentId);
    if nextPageToken in reponse then
        Execute response
    else
        break
```

To extract the twitter conversations, we utilised the Python library Twitter API. The reason for utilising this library instead of the official Twitter API is that the official Twitter API returns replies in a random sequence for any conversation. The Twitter API library comes in handy in this situation. It features a built-in function that arranges tweet replies in chronological order based on the 'timestamp' and 'reply_to_id'.

Algorithm 4 Twitter Data Extraction

Data: consumerKey; consumerSecret; accessTokenKey; accessTokenSecret; tweetId

Result: A csv file of **replies** having tweetId, text, createdAt, inReplyToUserId, replyCount, authorId

Import python library TwitterAPI;
Create its object 'twapi';
Authenticate using Twitter Developer credentials;
Import file = tweets.csv for tweetId;

for *tweetId in tweets.csv* **do**
  Extract **replies** by passing tweetId to twapi;

Save the response to the csv file

3.4 Selection choices of the data platforms

Although the data that we collected has been the entire conversations in all forms, we noticed that there are some inherent differences in how each of them function.

Reddit, for one shows a strict form of strict subreddit rules(with automated checks by bots) content moderation with assigned moderators for each subreddit and also the reddit space is generally considered to be self moderated by the people.The participants of a particular subreddit also watch over the content and if found incompatible in some sort then it is aggressively down-voted.However, there is no such mechanism for other platforms i.e. Twitter and YouTube.

When it comes down to the quality of conversations again reddit takes the crown, because firstly English is the major language and secondly also the average sentence length is seen in the order Reddit > Twitter > YouTube.

Some of the patterns that we have under study like Hate Speech and trolls would be dormant in a strict place like Reddit whereas it is quite prevalent in the comparatively lenient platforms like Twitter and Youtube.

Chapter 4

Design

4.1 Breakdown of the elements

We have sourced raw data from three primary sources, namely YouTube, Reddit and Twitter. After performing the requisite pre-processing, we obtain the pre-processed text data.

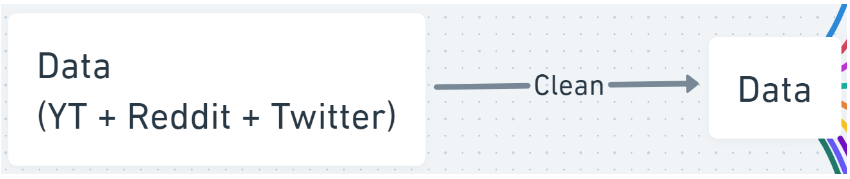


Figure 4.1: Data Block + Cleaning

4.1.1 Node Classification on Individual Threads

Since we aim to study the interactions in a conversation, we need to study the individual utterances. In this context, each utterance can be represented as a node and labels can be assigned to each one of them. A A description of the same can be inferred from the following figure.

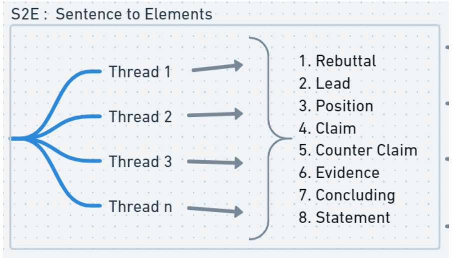


Figure 4.2: Sentence to Elements Block

4.1.2 Sub-Problems to Graph

Conversation threads belonging to each sub-problem are encoded into a larger graph embedding, which indicates an identification task in itself. Here we create eight graphs corresponding to each of the eight problems.

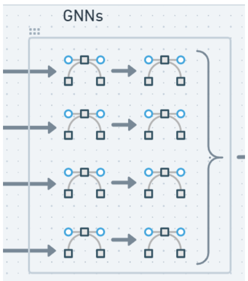


Figure 4.3: Running Graph instances through the network

4.1.3 Sub-Problem Graph to Graph-Attention model

The graph embeddings of each of the sub-problems are fed into a final model. For testing, a conversation would be compared with all the sub-problem embeddings and then an attention score would be given highlighting the relationship of the utterances with each of the sub-problems.

4.2 System Model

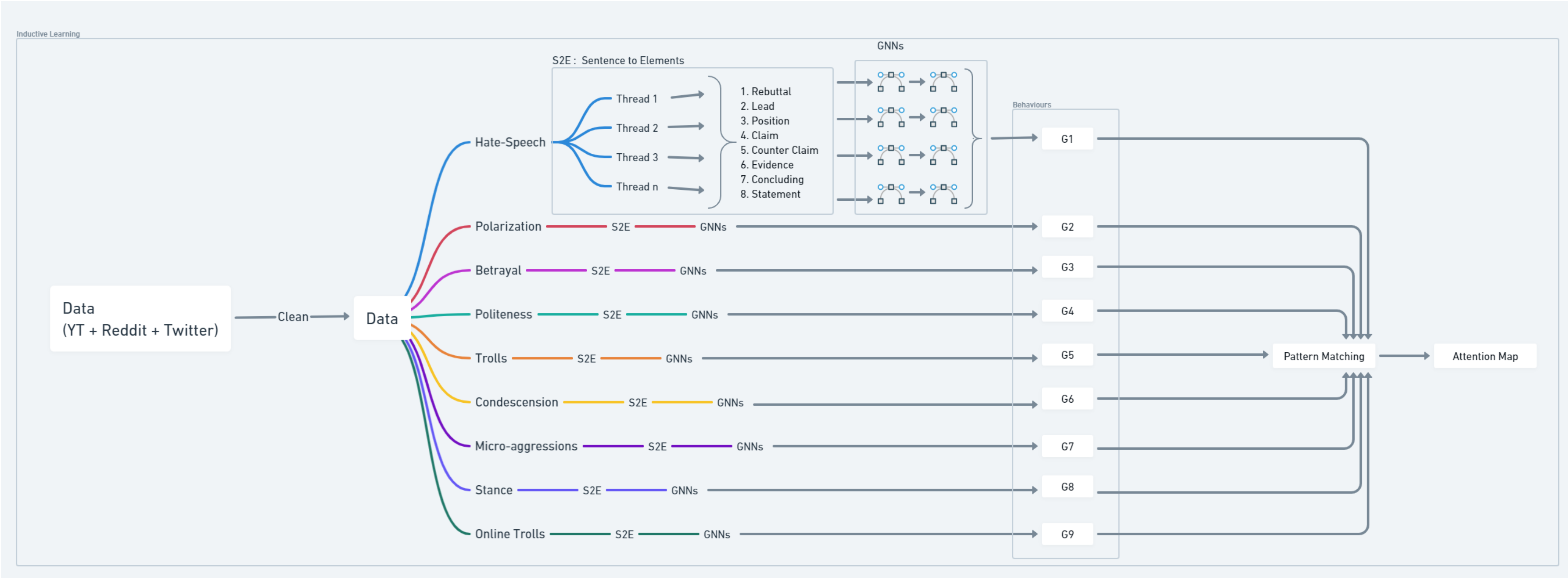


Figure 4.4: The Model Architecture

4.3 Tools/Technologies involved

4.3.1 Data Related

- **Pandas and Numpy** - used for Data manipulation and data processing.
- **Matplotlib and Seaborn** - used for Data visualisation.
- **NLTK and SPACY** - python libraries to work with NLP, provides tool for tokenization, parse tree visualisation, lemmatization, stop words removal, etc
- **SkLearn** - used for Machine learning and statistical modelling.
- **Jupyter**
- **VS code**
- **Convo-Kit** - A library that has a structured format for conversations and also has custom functions that can go along with it.

4.3.2 Graph Related

- **NetworkX** - used for manipulating and visualising graphs.
- **Pytorch** - A Deep Learning Framework used for building the Graph Models - GAT
- **WandB** - Weights and biases is used for Hyper-parameter tuning and experiment control to take care of all the runs and cancel runs if we consider then to be not fruitful.
- **HuggingFace** - An extensive library consisting of pre-trained language models such as transformers and BERT.

Chapter 5

Testing/Experimental Results

5.1 Node Classifier

Here we have tried to see every sentence as node and hence in order to facilitate the flow of knowledge between the nodes from other conversations, we needed something that is invariant to the conversation. For example, if we had made the design choice of nodes to be simply the users, then not only the number of users would vary from conversation to conversation but also the users themselves would be different. Hence classifying each of the sentence into one of the 10 dicourse components : ['announcement', 'elaboration', 'humor', 'appreciation', 'question', 'answer', 'agreement', 'negativereaction', 'disagreement', 'other']

5.2 Edge Classifier

We have referred to the CONVOKIT Corpus for our node labels. There are ten labels for all speaker utterances. $\forall N1, \exists(N2, N3, N4)$ that have the maximum connections. These pairs of nodes would be connected through edges. We intend to construct a morphological structure-based classification system depending upon syntactic analysis of text to study the following associations among different pairs of nodes:

1. **Asking**: Engaging and seeking information
2. **Informing**: Giving information
3. **Asserting**: Stating something as true
4. **Proposing**: Putting forward as argument
5. **Summarizing**: Reflecting your understanding
6. **Checking**: Testing understanding
7. **Building**: Adding to existing ideas

8. **Including**: Bringing in others
9. **Excluding**: Shutting out others
10. **Self-Promotion**: Boosting oneself
11. **Supporting**: Lending strength
12. **Disagreeing**: Refusing to agree
13. **Avoiding**: Refusing to consider argument
14. **Challenging**: Offering new thoughts to change thinking
15. **Attacking**: Destruction of their ideas
16. **Defending**: Stopping their attacks
17. **Blocking**: Putting things in the way of their arguments

5.2.1 The Heuristic Approach

Since we are following a heuristic approach, we have to make edges for 9x9 node pairs. However, possibility is not all the nodes occur together forming a pair. Therefore, we analysed the discourse components occurring together (constituting a node pair) by the number of utterances.

We found two ways in order to do this: analyse the entire conversation for a particular discourse component or analyse the direct replies to the discourse component in all the conversations.

Consequently the aim is to identify related nodes, we hypothesised analysing replies will be a better approach.

Approach 1: Analysing Nodes based on entire conversation

elaboration	7106
appreciation	3286
question	2899
answer	2395
announcement	2002
agreement	1410
disagreement	1045
negativereaction	931
humor	911
other	820

Figure 5.1: Total number of utterances of each discourse components from all the conversations where at least one utterance of "Announcement" is present

Looking at the number of utterances of each component in the conversations where "announcement" discourse component is present: (announcement, answer) and (announcement, question) node pair are important. Moreover, while looking at the relations for the "question" discourse component we get (question, answer) node pair.

answer	41100
question	17594
elaboration	17127
appreciation	7511
agreement	4578
disagreement	3102
humor	1992
negativereaction	1600
other	1577
announcement	1198

Figure 5.2: Total number of utterances of each discourse components from all the conversations where at least one utterance of "Question" is present

Since, ("Announcement","Question") and ("Question","Answer") were found related, we got a ("Announcement","Answer") transitive relation while analysing. So this problem is solved by approach 2.

Approach 2: Analysing Nodes based on direct replies

Depth:

1	31884
2	3464
3	3258
4	1024
5	765
6	297
7	259
8	93
9	84
10	34

Question

Category	Reply_depth-1	Reply_depth-2	Reply_depth-3	Total Replies
Answer	31813	2989	2606	39546
Question	2503	193	130	2936
Elaboration	508	165	116	870
Appreciation	420	75	18	535
Humor	346	49	20	433
Agreement	184	88	33	323
Other	236	19	14	284
Negative Reaction	219	26	11	266
Disagreement	49	25	9	98

Figure 5.3: Number of direct replies to "Question" tag bifurcated by different post depth

In this approach we are analysing the number of direct replies to a particular discourse component.After getting the total count of replies, we check how these replies are divided at each post depth.

For instance, a total of 31,884 utterances are of answer type at post depth 1, out of which 31,813 are direct replies to questions at post depth 0 i.e more than 90%.

So, for a node pair to exist we have kept a threshold of 33%. Therefore, for the question component, (question,answer) , (question,humour) , (question,question) and (question,agreement) pairs can be formed.

		A	B	C	D	E	F	G	H	I
	Table 7	Agreement	Announcement	Answer	Appreciation	Disagreement	Elaboration	Humour	NegativeReaction	Question
A	Agreement	AA								
B	Announcement	BA			BD	BE	BF		BH	BI
C	Answer	CA			CD	CE	CF	CG	CH	CI
D	Appreciation				DD					
E	Disagreement	EA				EE				EI
F	Elaboration	FA			FD	FE	FF			
G	Humour							GG		
H	NegativeReaction									
I	Question	IA		IC				IG		II

Figure 5.4: All the node pairs

The figure below(Figure 5.5) shows our heuristic based selection of the kind of relations that could occur between the node pairs(from Figure 5.4). The selection is based on logical examples that can possibly happen over a conversation.

Table 8	Asking	Informing	Asserting	Proposing	Summarizing	Checking	Building	Including	Excluding	Self-promotion	Supporting	Disagreeing	Avoiding	Challenging	Attacking	Defending	Blocking
AA			✓					✓			✓					✓	
BA			✓				✓				✓						
BD							✓	✓			✓						
BE			✓			✓			✓			✓		✓	✓		✓
BF			✓	✓			✓	✓			✓						
BH			✓						✓			✓	✓	✓	✓		
BI	✓					✓	✓	✓									
CA			✓				✓	✓			✓						
CD			✓					✓			✓						
CE			✓							✓		✓		✓	✓		
CF			✓				✓			✓	✓					✓	
CG																	
CH			✓			✓			✓			✓		✓	✓		
CI	✓		✓			✓						✓		✓			
DD			✓				✓	✓			✓						
EA			✓			✓	✓	✓			✓						
EE			✓			✓			✓			✓		✓	✓		
EI												✓		✓			
FA			✓				✓				✓						
FD							✓				✓						
FE						✓			✓			✓		✓	✓		
FF				✓		✓	✓	✓			✓						
GG												✓			✓		✓
IA	✓										✓						
IC		✓		✓			✓										
IG													✓		✓		✓
II	✓			✓		✓	✓					✓					

Figure 5.5: Proposed edge pairs on the basis of contextual clues

5.3 Justification of the sub-problem data being used

5.3.1 Deception and Betrayal

To study this sub-problem, we have referred to the Convokit corpus on Deception in the Diplomacy dataset *Peskov and Cheng*[30]. The dataset comprises a series of utterances between the speaker and receiver in an online negotiation-based game called Diplomacy. The players of this game belong to seven different nations, namely Austria, Germany, Turkey, England, France and Russia. The utterances in this corpus have a meta-data feature named `Deception_Quadrant`, which is indicative of the message reception by the speaker and the receiver.

No. of speaker	83
No. of conversations	17289
No. of utterances	246
Actual lie count	887
Suspected lie count	667
Average Word Count	105.88

Table 5.1: Summary Statistics

You've whetted my thirst, lady. What's your take?	Truth	Truth	Straightforward
I don't care.	Truth	Lie	Cassandra
If I were telling the truth to you, I would smile and say "Yupp, that sounds great!" I am being open to you because I thought that we were partners.	Lie	Truth	Deception
I give you my word: I don't know what England is upto and I didn't even ask.	Lie	Lie	Caught

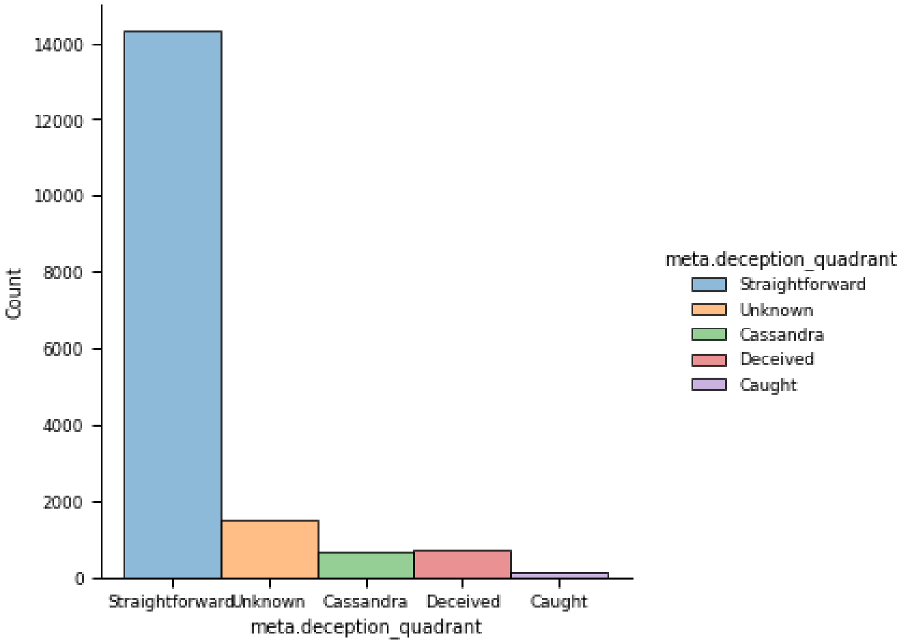


Figure 5.6: Distribution of deception label in corpus with respect to the final association between the speaker's intention and receiver's perception

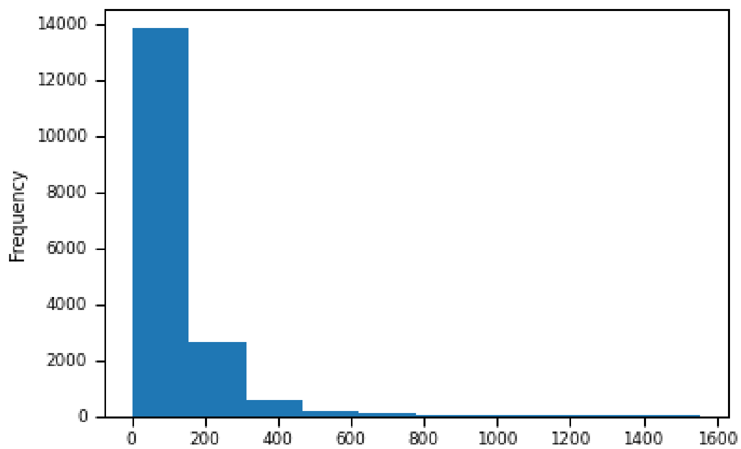


Figure 5.7: Distribution of Word Count per message

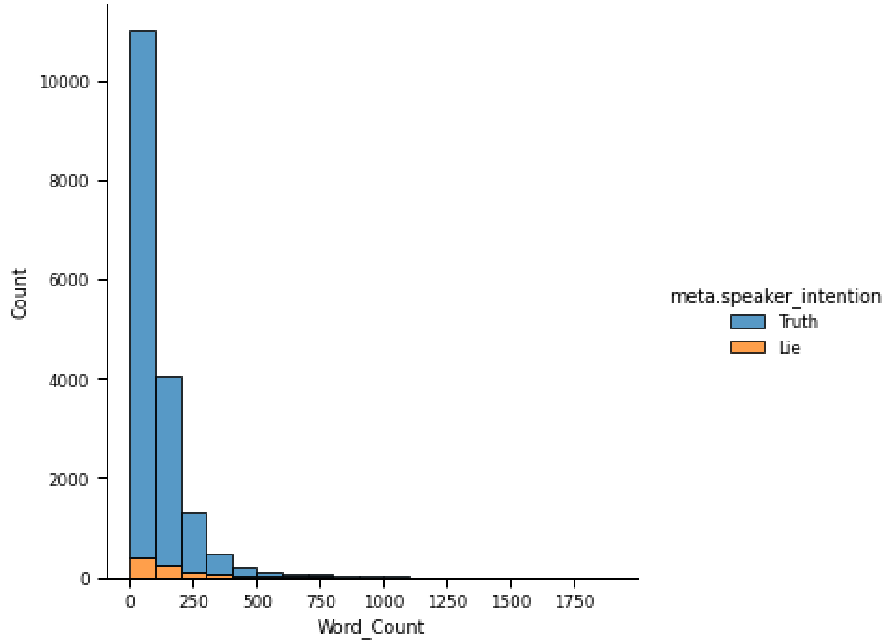


Figure 5.8: Distribution of word count in truth and lies as uttered by the speaker

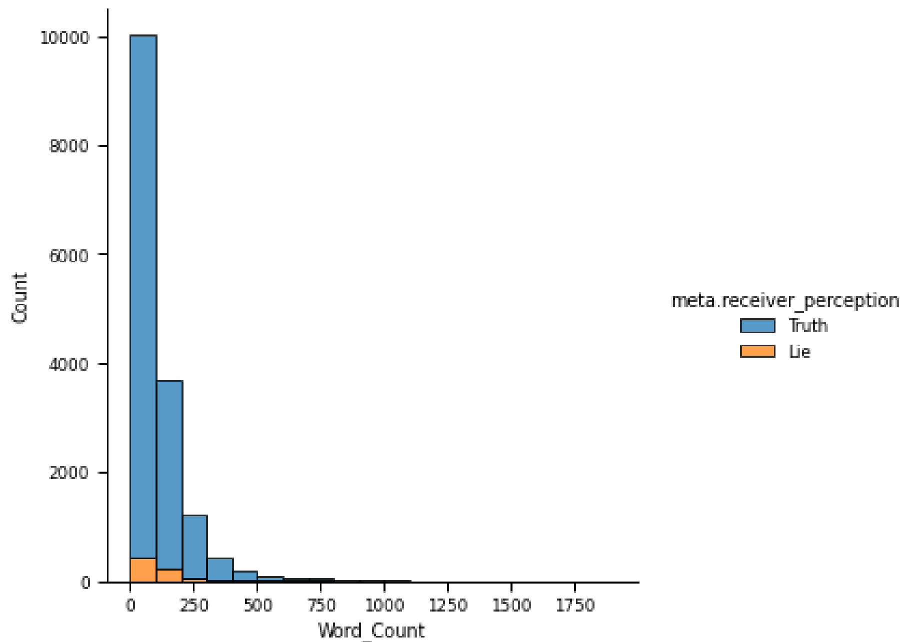


Figure 5.9: Distribution of word count in truth and lies as perceived by the receiver

5.3.2 Linguistic Alignment

Linguistic alignment is a subtle indicator of whether the conversation at hand is well coordinated or not. This could be an impactful element in various kinds of problems. For example, if one has to solve the task of incentivizing collaboration, then we think that Linguistic Alignment is something that is a major player for this task. Imagine it this way, less the linguistic alignment, less will be the coordination and coherency of the ideas. Even though both the people are talking about the exact same thing, they might lack coordination and hence no productive conversation.

One of the problems with studying the alignment is that we cannot have it calculated on the fly as the comments drop in. Alignment is inherently backward looking, hence one should have previous knowledge of some sort, in our conversational data, this info is available to us beforehand.[4]

Some of the metrics listed by *Niven and Kao* and *Doyle et al.* can be broadly categorised into 2 parts : Distribution and Conditional. Distributional measures such as LSM (Linguistic style matching) [32] uses the user styles or Zelig Quotient - [33] which looks over word frequencies and word category within conversations .These measures are more concerned with similarity and less of an alignment, whereas conditional measures like LLA(Local Linguistic Alignment -[34]) or "By-word conditional methods" that covers the problem of length and sees the portion of the sentence for pronouns are much better at quantifying Linguistic Alignment.

For now we have restricted our scope of verification to just the coordination that been discussed by *Danescu-Niculescu-Mizil et al.*, and is implemented in Convo-Kit as per [35] because of the varied data sources, but we plan to prepare a comparison table for all the metrics discussed above for this sub-problem as well. This implementation is rudimentary because it takes into account only the occurrence of the function words from various speakers.

```
j__sandra_day_oconnor 0.04285
j__david_h_souter 0.04059
j__antonin_scalia 0.03841
j__potter_stewart 0.03784
j__stephen_g_breyer 0.03635
j__john_paul_stevens 0.03552
j__anthony_m_kennedy 0.03482
j__john_m_harlan 0.0342
j__samuel_a_alito_jr 0.03338
j__thurgood_marshall 0.0327
j__byron_r_white 0.0325
j__felix_frankfurter 0.03209
j__william_j_brennan_jr 0.03155
```

Figure 5.10: Co-ordination scores on a Court case scenario

5.4 Node Classifier

The BERT pretrained model i.e bert-base-uncased has been already trained on wikipedia content and hence is already aware of how sentences are structured. Although we had planned to fine tune hyperparameters of the BERT, but we could not fit the suggested batch sizes and the epochs due to constraints of computational power.

This is clearly seen in the graphs above wherein we could not reach the end stage for many of the runs as the memory would be insufficient and hence we had to stop early, but we did see significant gains for the ones that ran completely. Hence we believe that given the resources we can in fact improve the accuracy of the classifier.

Figure 5.11: Average Training loss for BERT

Algorithm 5 Fine -tuning BERT for Sentence classification

Data: Comments: str, Model Configurations
Result: Labels for sentences

Loading the Comments
while *len(Comments)* **do**
  Split the sentence into tokens;
  Add the special [CLS] and [SEP] tokens;
  Map the tokens to their IDs;
  Pad or truncate all sentences to the same length;
  Create the attention masks which explicitly;
  Differentiate real tokens from [PAD] tokens;
Perform Train-Test-Split;
Define the train_DataLoader and Val_DataLoader;
Download the pre-trained model;
Define optimizer, scheduler;
while *training* **do**
  Create instances of Train : DataLoaders,Model,optimizer,scheduler;
  Intitalize a comment_queue for doing a BFS search over the comment tree
  for *epoch in range(0,epochs)* **do**
    model.zero_grad();
    model.train();
    loss, logits = model(inputs);
    Calculate the loss;
    Back-propagate using SGD;
    Track train-Loss;
while *Validation* **do**
  Create instances of Val : DataLoaders,Model,optimizer,scheduler;
  for *batch in Val_DataLoader* **do**
    while *No gradient calculation* **do**
      loss, logits = model(inputs);
      Track Val-Loss;

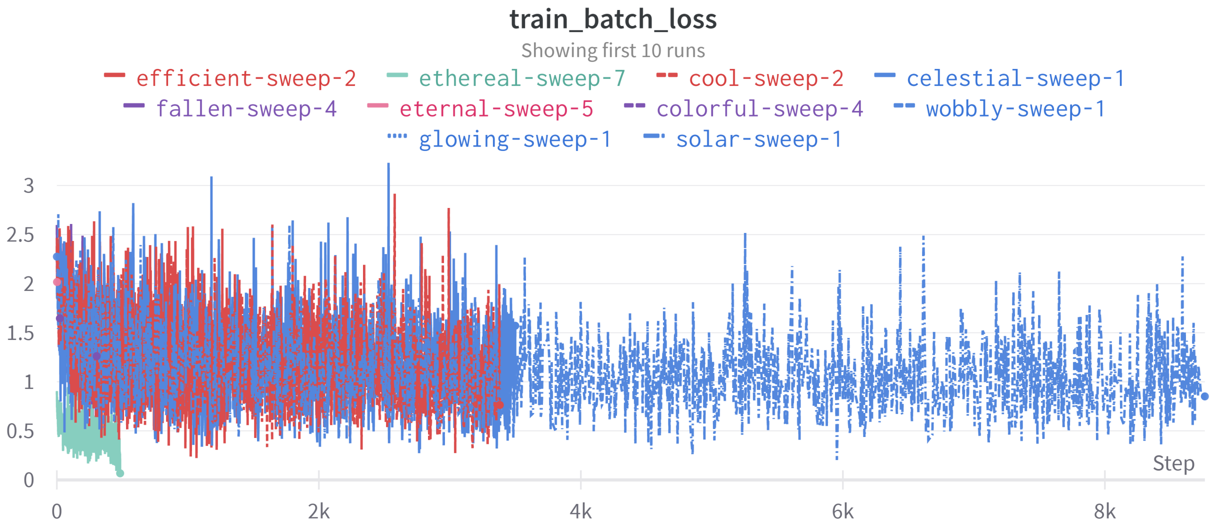


Figure 5.12: Average Batch loss for BERT

Chapter 6

6.1 Impact Statement

6.1.1 Benefits of the study

The outcome of this on-going study is to provide researchers a framework or a paradigm to facilitate them to study the problems in a more structured manner. This could also lead to more explainable results as now the behaviours/patterns of novel models of say, online content moderation can be explained in terms of these other factors. These models would not remain a complete black box and can be easily tuned/looked upon by an external expert to fine tune its performance, not just the architectural hyper parameters, but also how they favour solving one kind of problem over others. Consequently, it would lead in more generalizabilty of these models.

Emphasis is laid on these palpable models because they are quite exorbitant in the daily lives of an average human.It could have major consequences on lives of the people mentally, if we remiss seemingly not harmful errors or peculiar patterns in the behaviour of these models.

Again, the expected outcome of the study is a neutral framework and not a solution to solve a particular task. Keeping this in mind we can apply the insights from this study to many applications.

- **Identifying the points of collaboration in a given conversation** - Our study can help in identifying the points in the conversation in which users agree with each other.

- **Resolving conflicts** - Our study can help in detecting the exact point in the conversation at which conflict is occurring. After detection, some kind of AI system can be developed which can resolve the conflict.

- **Facilitating discussions that are vague** - Vague point in the conversation can be detected by our study. Later, by some AI technology, this vague points can be made clear.

- **Steering the conversations - amplify the seemingly weaker ideas** - If an individual is not able to emphasize his/her idea in the group conversation, that can be detected and later using AI that idea can be amplified.

- **Preventing cancel culture** - If majority of the group participating in the conversation is trying to outcast an individual, it can be detected through our studies.

- **Preventing Group-think** - Group-think is when someone expresses opinion and rest of the group gets influenced and start having the same opinion.

- **Identifying suicidal tendencies** - It can be identified if the person is having suicidal tendencies on the basis of his/her behaviour in the conversation.

The challenges faced while collecting data:

- Unavailability of Edge level information in utterances.

- Unavailability of data with replies(hierarchical data).

- Lack of large human annotated Training data.

- Collecting data from discrete sources is a challenge, unavailability of a single source.

- Robust datasets are available for various problems like sentiment classification, question answering, summarizing but not for the problem statements that analyse conversations as in whole.

Limitations in Raw Data data

- Data set is imbalanced for the node level information.

- Prevalence of Selection bias – We are collecting data based on our knowledge regarding the sub-problem which leads to selection bias.

- Data set is extracted from only 3 sources. This is enough for an initial study but we suspect could still result in a little variance when we try to apply it to other platforms.

Limitation in Graph Design

- The graphs are made on the basis of established node pairs. This would result in extra connections among the utterances (even though one of them is not a reply to another).

  For instance, there are three utterances U1,U2,U3. U2 (answer) is reply to U1 (question) and U3 (agreement) reply to U2 (answer). Even though U3 is not a reply to U1, we will get a connection because the (question,agreement) pair is already established.

- While making a graph it is possible that for the given utterance pair (U1,U2) an edge from U1 to U2 and U2 to U1 is possible.

  For instance, U1(question) and U2(answer) are two utterances where, U2 is reply to U1. Since, (Question,Answer) and (Answer,Question) pair exists. Edge from U1 to U2 and vice versa is possible. In this case we need to take into consideration the post depth of utterance. Since, U2 is a reply to U1, its depth would be higher and hence U2-¿U1 edge won't be there. In such cases, few relationships will be missing.

- Node pairs might be different for different sub-problems.

Bibliography

[1] Michail Tsikerdekis and Sherali Zeadally. Online deception in social media. *Communications of the ACM*, 57(9):72–80, 2014.

[2] Raymond S Nickerson. Confirmation bias: A ubiquitous phenomenon in many guises. *Review of general psychology*, 2(2):175–220, 1998.

[3] Luke Breitfeller, Emily Ahn, David Jurgens, and Yulia Tsvetkov. Finding microaggressions in the wild: A case for locating elusive phenomena in social media posts. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 1664–1674, 2019.

[4] Timothy Niven and Hung-Yu Kao. Detecting argumentative discourse acts with linguistic alignment. In *Proceedings of the 6th Workshop on Argument Mining*, pages 104–112, 2019.

[5] Gabriel Doyle and Michael C Frank. Investigating the sources of linguistic alignment in conversation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 526–536, 2016.

[6] Kaiping Chen and Lucas Alexander DiSanto Sanchez. Conversation structure and quality of political discourse: Evidence from reddit. *Available at SSRN 3362008*, 2019.

[7] Jacki O'Neill and David Martin. Text chat in action. In *Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*, pages 40–49, 2003.

[8] Thomas S Mueller. Blame, then shame? psychological predictors in cancel culture behavior. *The Social Science Journal*, pages 1–14, 2021.

[9] Paul A. Jargowsky. Omitted variable bias. In Kimberly Kempf-Leonard, editor, *Encyclopedia of Social Measurement*, pages 919–924. Elsevier, New York,

2005. ISBN 978-0-12-369398-3. doi: https://doi.org/10.1016/B0-12-369398-5/00127-4. URL `https://www.sciencedirect.com/science/article/pii/B0123693985001274`.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL `http://arxiv.org/abs/1706.03762`.

[11] Alexander Kalinowski and Yuan An. A comparative study on structural and semantic properties of sentence embeddings. *arXiv preprint arXiv:2009.11226*, 2020.

[12] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.

[13] Sumeet Kumar and Kathleen M 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, 2019.

[14] Meredith D. Clark. Drag them: A brief etymology of so-called "cancel culture". *Communication and the Public*, 5(3-4):88–92, 2020.

[15] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[16] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. *Advances in neural information processing systems*, 28, 2015.

[17] Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*, 2018.

[18] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

[19] Mahtab Ahmed, Muhammad Rifayat Samee, and Robert E Mercer. You only need attention to traverse trees. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 316–322, 2019.

[20] Nishant Subramani and Nivedita Suresh. Discovering useful sentence representations from large pretrained language models. *arXiv preprint arXiv:2008.09049*, 2020.

[21] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[22] Smriti Bhagat, Graham Cormode, and S Muthukrishnan. Node classification in social networks. In *Social network data analytics*, pages 115–148. Springer, 2011.

[23] Xia Zhang, Weizheng Chen, and Hongfei Yan. Tline: Scalable transductive network embedding. In *Asia Information Retrieval Symposium*, pages 98–110. Springer, 2016.

[24] Yan-Ming Zhang, Kaizhu Huang, and Cheng-Lin Liu. Fast and robust graph-based transductive learning via minimum tree cut. In *2011 IEEE 11th International Conference on Data Mining*, pages 952–961. IEEE, 2011.

[25] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[26] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[27] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 891–900, 2015.

[28] Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu. Bertgcn: Transductive text classification by combining gcn and bert. *arXiv preprint arXiv:2105.05727*, 2021.

[29] Bentian Li and Dechang Pi. Learning deep neural networks for node classification. *Expert Systems with Applications*, 137:324–334, 2019.

[30] Denis Peskov and Benny Cheng. It takes two to lie: One to lie, and one to listen. In *Proceedings of ACL*, 2020.

[31] Gabriel Doyle, Dan Yurovsky, and Michael C Frank. A robust framework for estimating linguistic alignment in twitter conversations. In *Proceedings of the 25th international conference on world wide web*, pages 637–648, 2016.

[32] Molly E. Ireland, Richard B. Slatcher, Paul W. Eastwick, Lauren E. Scissors, Eli J. Finkel, and James W. Pennebaker. Language style matching predicts relationship initiation and stability. *Psychological Science*, 22(1):39–44, 2011. doi: 10.1177/0956797610392928. URL `https://doi.org/10.1177/0956797610392928`. PMID: 21149854.

[33] Simon Jones, Rachel Cotterill, Nigel Dewdney, Kate Muir, and A Joinson. Finding zelig in text: A measure for normalising linguistic accommodation. pages 455–465, August 2014. 25th International Conference on Computational Linguistics ; Conference date: 23-08-2014 Through 29-08-2014.

[34] Kate Niederhoffer and James W. Pennebaker. Linguistic style matching in social interaction. *Journal of Language and Social Psychology*, 21:337 – 360, 2002.

[35] Cristian Danescu-Niculescu-Mizil, Lillian Lee, Bo Pang, and Jon Kleinberg. Echoes of power: Language effects and power differences in social interaction. In *Proceedings of WWW*, pages 699–708, 2012.

Project Group Members Personal Details

1. Name of the Student: Anurag Pal

   Email: anuprag812@ gmail.com

   Mobile no: 6355393253

   Permanent Address: 116/A, C Site,
   Freelandganj,
   Dahod - 389160

2. Name of the Student: Sagar Sinha

   Email: sagar.sce18@sot.pdpu.ac.in

   Mobile no: 9537418856

   Permanent Address: A-21, Meera Park Society,
   Dahej Bypass Road,
   Bharuch-392011

3. Name of the Student: Sanskar Bhuwania

   Email: sanskatbhuwania07@ gmail.com

   Mobile no: 9725118277

   Permanent Address: 704, Ghanshyam Tower,
   Shreeji Sadan Residency,
   Zadeshwar, Bharuch - 392011