Identification of Intent-Sentiment Co-reference from Conversational Texts

Thesis submitted in partial fulfillment of requirements

For the degree of

Master of Computer Science and Engineering

of

Computer Science and Engineering Department, Jadavpur University

by

Pinaki Karkun

Regn. No. - 154164 of 2020-2021

Exam Roll No. - M4CSE22040

under the supervision of

Dr. Dipankar Das

Assistant Professor

Department of Computer Science and Engineering

Jadavpur University

Kolkata, West Bengal, India

2022

Certificate from the Supervisor

This is to certify that the work embodied in this thesis entitled **"Identification of Intent-Sentiment Co-reference from Conversational Texts"** has been satisfactorily completed by **Pinaki Karkun** (Registration Number 154164 of 2020-21; Class Roll No. 002010502040; Examination Roll No. M4CSE22040). It is a bona-fide piece of work carried out under my supervision and guidance at Jadavpur University, Kolkata for partial fulfilment of the requirements for the awarding of the **Master of Engineering in Computer Science and Engineering** degree of the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Jadavpur University, during the academic year 2020-22.

Dipankar Das,
Assistant Professor,
Department of Computer Science and Engineering,
Jadavpur University.
(Supervisor)

Forwarded By:

Prof. (Dr.) Anupam Sinha,
Head,
Department of Computer Science and Engineering,
Jadavpur University.

Prof. (Dr.) Chandan Majumdar,
DEAN,
Faculty of Engineering & Technology,
Jadavpur University.

Department of Computer Science and Engineering
Faculty of Engineering And Technology
Jadavpur University, Kolkata - 700 032

Certificate of Approval

This is to certify that the thesis entitled **"Identification of Intent-Sentiment Co-reference from Conversational Texts"** is a bona-fide record of work carried out by **Pinaki Karkun** (Registration Number 154164 of 2021-22; Class Roll No. 002010502040; Examination Roll No. M4CSE22040) in partial fulfilment of the requirements for the award of the degree of **Master of Engineering in Computer Science and Engineering** in the **Department of Computer Science and Engineering, Jadavpur University**, during the period of Oct 2020 to June 2022. It is understood that by this approval, the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose of which it has been submitted.

Examiners:

(Signature of The Examiner)	(Signature of The Supervisor)

Department of Computer Science and Engineering
Faculty of Engineering And Technology
Jadavpur University, Kolkata - 700 032

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that the thesis entitled **"Identification of Intent-Sentiment Co-reference from Conversational Texts"** contains literature survey and original research work by the undersigned candidate, as a part of his degree of **Master of Engineering in Computer Science and Technology** in the **Department of Computer Science and Engineering, Jadavpur University**. All information have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Pinaki Karkun

Examination Roll No.: M4CSE22040

Registration No.: 154164 of 2020-22

Thesis Title: Identification of Intent-Sentiment Co-reference from Conversational Texts

Signature of the Candidate:

ACKNOWLEDGEMENT

I am pleased to express my gratitude and regards towards my Project Guide **Dr. Dipankar Das**, Assistant Professor, Department of Computer Science and Engineering, Jadavpur University, without whose valuable guidance, inspiration and attention towards me, pursuing my project would have been impossible.

Last but not the least, I express my regards towards my friends and family for bearing with me and for being a source of constant motivation during the entire term of the work.

Pinaki Karkun
MCSE Final Year
Exam Roll No. - M4CSE22040
Regn. No. - 154164 of 2020-22
Department of Computer Science and Engineering,
Jadavpur University.

Contents

List of Figures

List of Tables

Abstract

Our goal in this thesis work is to develop models based on judicial dialogues that can recognise the feelings of the utterances, the intent or intents of the speakers, the tokens that determine them, and any relationships between them.

One issue with a chatbot designed primarily for judicial help is that user intentions can become muddled and overlap. Therefore, judging intents solely from the words said by the user is difficult. The wording and context completely alter the intentions.

In order to correctly serve people, it's crucial for a chatbot or agent to keep track of their feelings during a conversation. When a dialogue veers off into an intense emotion, it loses all effectiveness. Therefore, it is a significant problem for the chatbot to understand which topics can cause those divisive feelings.

Chapter 1

Introduction

1.1 Natural Language Processing

Natural language processing (NLP) is a branch of computer science, to be more precise, a branch of artificial intelligence (AI) concerning the ability of computers to understand text and spoken words in the same manner that humans can. Several NLP activities help the machine understand what it's absorbing by breaking down human text and speech input in ways that the computer can understand. The following are some of these responsibilities:

- **Speech recognition**, also called speech-to-text, is the task of reliably converting voice data into text data.
- **Part of speech tagging**, also called grammatical tagging, is the process of determining the part of speech of a particular word or piece of text based on its use and context.
- **Word Sense disambiguation**, is the selection of the meaning of a word with multiple meanings through a process of semantic analysis that determine the word that makes the most sense in the given context.
- **Named entity recognition**, or NEM, identifies words or phrases as useful entities. NEM identifies 'Kentucky' as a location or 'Fred' as a man's name.
- **Co-reference resolution**, is the task of identifying if and when two words refer to the same entity.

- **Sentiment analysis**, attempts to extract subjective qualities—attitudes, emotions, sarcasm, confusion, suspicion—from text.
- **Natural language generation**, is sometimes described as the opposite of speech recognition or speech-to-text; it's the task of putting structured information into human language.

1.2 Discourse and Dialogue

Discourse is a one-way dialogue in which both parties participate. The purpose of a cooperative, two-way conversation is to transfer knowledge from the speaker/writer to the listeners/readers. Participants are encouraged to share knowledge and create bonds with one another.

Discourse study aims to answer two types of questions: (1) What information is provided in extended sequences of utterances that is not contained in the meaning of the individual utterances? (2) What effect does the context in which an utterance is used have on the meaning of individual utterances or parts of them?

The reason for considering discourse and dialogue rather than just the sentences that make up each one is that information is sometimes presented or requested over multiple sentences, and we want to recognise various phrases or relationships among them that identify the who, what, when, where, and why of the event. We could take material from articles in newspapers and magazines, as well as chapters from books, and save it in tables that are more easily searchable for dialogue. We might want to extract information from dialogue to complete activities like booking travel or making a restaurant reservation, or teaching a kid, for example. Alternatively, we might wish to be able to detect explicit or implicit demands in purely social exchanges that have no clear purpose.

1.3 Intent and Sentiment

Intents are the tags we can assign to words pr phrases in a dialogue or discourse dataset. It identifies what topic are being discussed in utterances or in dialogues. Intents are subjective to the conversation topic; depending upon them there can be unique sets of intents for different conversations.

Let's take two sentences from a chat; these are-

1. I am hungry.
2. I have not eaten anything since morning.

When addressing traditional Text Classification issues, these two statements have different syntactic meanings, which are treated as facts. But let us try to see things from a different perspective (intent of the sentences). Sentence 1 clearly indicates that the individual is hungry. Despite the fact that sentence 2 does not contain the word hungry, we may grasp the person's 'intent' that he or she is hungry at that point and is considering not eating anything.

Sentiment analysis, often known as opinion mining, is a natural language processing (NLP) technique for determining the emotional tone of a body of text. This is a common method for businesses to determine and categorise customer opinions about a product, service, or concept. It entails mining text for sentiment and subjective information using data mining, machine learning (ML), and artificial intelligence (AI).

Sentiment analysis tools assist firms in extracting information from unstructured and unorganised language found online in places like emails, blog posts, support tickets, web chats, social media channels, forums, and comments. Algorithms apply rule-based, automatic, or hybrid ways to replace manual data processing. Rule-based systems analyse sentiment using predetermined lexicon-based rules, whereas automatic systems use machine learning techniques to learn from data. A hybrid sentiment analysis combines the best of both worlds.

Opinion mining can extract the polarity (or the quantity of positive and negative) inside the text in addition to identifying mood. Additionally, sentiment analysis can be used on a variety of scales, including document, paragraph, phrase, and sub-sentence levels.

1.4 Co-reference

The objective of coreference resolution (CR) is to locate all linguistic expressions (called mentions) that relate to the same real-world thing in a given text. We can fix these mentions by replacing pronouns with noun phrases, as mentioned above.

"I wear Number 10 Jersey for the US National Team in honor of the Greatest athlete I have ever seen, Messi.", Usain Bolt said.

Original statement.

"Usain Bolt wear Number 10 Jersey for the US National Team in honor of Messi Usain Bolt have ever seen, Messi.", Usain Bolt said.

Statement after resolved co-reference.

Text interpretation, information extraction, machine translation, sentiment analysis, and document summarising are just a few of the NLP activities that can benefit from coreference resolution. It's a terrific method to get unambiguous statements that computers can understand much more readily.

1.5 Challenges

We have discussed about the works which already taken place by different authors on intent, sentiment classification and co-reference. Although all the topics we are discussing has not been yet implemented on judicial dataset.

Judicial dataset has its unique issues to address, mainly- it contains very low amount of positive statements, and most of the utterances are facts, so number of neutral statements tends to be more than others.

A chat bot mainly built for judicial assistance purpose possesses some problems like, the intents of the users can be convoluted and overlapped. So its hard to identify intents based on the words spoke by the user only. The context, phrases changes the intents entirely.

Its very important for a chat bot or agent to track the sentiments of the users while having a conversation with them to properly assist them. When a conversation screws to some extreme emotion the conversation becomes very inefficient. So it is an important challenge for the chat-bot to have clear connection to which topics can lead to those polarising emotions.

1.6 Research Objectives

Our objective in this thesis-work is to create models based on Judicial conversations which can identify intent / intents, sentiments of the utterances, identify tokens which determines them, and build a co-reference model to identify the relation between them if any.

1.7 Reasearch Contributions

We have created machine learning and deep learning models to classify and identify possible intents or topics that contains in the users and bot dialogues. We also have developed machine learning and deep learning based models to classify and identify the sentiments. We then developed deep learning base model to identify and classify if any utterances have topics or intent that potentially a co-reference to the sentiment of the utterance.

1.8 Thesis Outline

This thesis-work has been divided into seven chapters.

Chapter 1 has the introductions of the topics we are going to discuss.

Chapter 2 discusses the works which already been done on the topics.

Chapter 3 discusses how the raw dataset has been collected and prepared to be fed to different models.

Chapter 4 discusses how intents has been identified from the utterances and how tokens are identified which are responsible to tag those intents and how detection of intent-sentiment co-references taken place.

Chapter 5 discusses how utterance sentiments has been calculated and tokens that determines the sentiments has been identified.

Chapter 6 discusses how detection of intent-sentiment co-references taken place.

chapter 7 discusses the observations and the future work that can be done additionally on this domain.

Chapter 2

Related Work

2.1 Intent Classification

The study of intent classification has always piqued the curiosity of academics. Initially, a model for IC based on the frequency of each single phrase, extracted information from each term pair, and POS tagging was proposed in [8]. The click distribution for each query was utilised in a 2005 conference report [11] to classify whether it was a navigational or informative question. It was improved later in book chapter [12] by utilising statistical approaches on massive log data. In 2008, in the journal [7] intent classification methods based on basic rules matching and query feature manipulation were implemented. These solutions, however, did not use any machine learning or deep learning techniques. Many researchers have used neural network-based models to attain high IC performance in recent years.

In 2013, the authors of [20] employed CNN followed by triangular CRF to forecast intent and identify an entity using extracted characteristics from CNN, but in [9] Elman-type, Jordan-type, and bi-directional Jordan-type RNN followed by basic CRF were used for the job. Authors later employed a modified deep LSTM, followed by CRF and SoftMax, to explicitly represent the dependencies between semantic labels for greater understanding in an IEEE publication [21]. It was tested using bi-directional RNN, bi-directional RNN with attention mechanism, and Encoder-Decoder LSTM in [18], [17], and [13].

In conference article [15] a self-attentive shared encoder to produce better context-aware representations was used which applied the extracted and summarized features for IC at sentence and the token level. Recently in paper [2] authors 2019 have used a pre-trained BERT model and a fine-tuned BERT model for IC and joint IC - slot filling tasks respectively.

2.2 Sentiment Classification

Schukla [15] described a programme that assesses text quality based on annotations in scientific papers. Its methodology uses two approaches to obtain annotation sentiments. It determines total sentiment scores by counting all of the annotations produced by the documents. Its issue is that it indicates a complex link between annotations. A large query knowledge base containing metadata is required for the strategy. For hotel reviews, Kasper and Vela presented a ”Web Based Opinion Mining method” [17]. The research described a system for evaluating online user reviews and comments in order to aid quality control in hotel management systems. It can recognise and retrieve online reviews and works with German reviews. It has a multi-topic domain and is based on multi-polarity classification; the system could distinguish neutral words like ”don’t know” to ”classify sentiment polarity as neutral” and multi-topic cases found in their corpus.

Product reviews for mobile devices were investigated by (Zhang, et al) in [17]. This research can help determine precision. It can be used to evaluate the quality of a product and its position on the market [17]. Three machine learning approaches were used to compute the sentiment accuracy (Nave Base Classifier, K-nearest neighbour, and random forest). Using the random forest improves the performance of the classifier. Examining sentiments and opinions can be done in several ways. (Godbole et al.) investigated public opinion and blogs [18]. In the context of their individual task, it divides prior labour into two groups (sentiment analysis for news and blogs). The first is concerned with methods for automatically constructing sentiment lexicons, whereas the second is concerned with systems that analyse sentiment for entire documents.

Furthermore, Esuli & Srinivasiah’s research divides similar work into two categories [19]: the first focuses on detecting term direction, while the second focuses on detecting term subjectivity. These distinctions only apply to term/word level classification research studies, not document level categorization. The goal of this study is to determine the sentiment polarity (positive, negative, or neutral) of a text review’s data and calculate the sentiment score. A text evaluation is essentially divided into single phrases (”sentence-based”) and words (”words-based”) or extremely brief texts from a single source.

The prior research with Hearst on sentiment-based categorization of input documents implicated either the use of models primarily inspired by cognitive linguistics [20] or the manual or semi-manual creation of discriminant-word lexicons proposed by Das & Chen [21]. Turney’s study [22], for example, provided a new method for sentiment extraction in real time in the financial realm, which is based on posts from web-based stock message boards and attempts to automatically identify each message as a ”buy,” ”sell,” or ”neutral” recommendation. Its presented classifier has a 62 percent accuracy rate (the upper bound, human agreement rate, was 72 percent). Unfortunately, their suggested solution includes manually selecting and tagging terms from thousands of messages to create a discriminant-word lexicon.

Corpus-based approaches examine inclusion using seed words based on huge groups of text [22] or search for context-dependent labels based on local limitations (Argamon, et al) in [23]. People have also looked at the knowledge encoded in WordNet as relations (synonymy, antonymy, and hyponymy), as well as glosses. The majority of subjectivity detection studies for sentiment polarity classification assume that the input documents are opinionated. Many tools and applications require you to decide whether or not a particular document contains subjective information, or to distinguish which parts of the text are subjective. The impact of adjective orientation and grad-ability on sentence subjectivity was investigated before by (Etzioni, et-al) in [24]. Based on the adjectives in the text, the target told us whether or not a specific sentence is subjective.

A subjective sentence is one that reflects feelings, opinions, or beliefs. Rather than individual words, each sentence in a document is evaluated and tested for subjectiveness at the sentence level. The subjective statement might be categorised as having a positive or negative semantic direction if necessary. A subjectivity detector is used by Pang and Lee in [25] to remove objective sentences from a manuscript. They then combine inter-sentence contextual information with classic bag-of-words features using minimum cuts formulation. They claim to have made significant improvements over a standard word vector classifier. Word vector representations and categorization are two things that all recursive neural models have in common, according to the researchers. For analysing and evaluating online sentiments, they used a semantic relationship model. Their method creates a multi-topic domain. However, the feeling necessitates a larger pool of supervised training and evaluation resources.

Jin and Ho [26] conducted research on opinion mining on YouTube to demonstrate how social media might be used to radicalise a person. A global social networking platform, such as YouTube, has the ability to discover content and interaction geared at radicalising persons with little or no apparent prior interest in violent Jihadism, as seen in Crawling. Their research focuses on a method that has proven to be effective. They compiled a huge dataset from a YouTube collection that had been identified as having a radicalising purpose. Social network analysis and sentiment analysis techniques are used to analyse the data. It also looks at the topics mentioned and their polarity (good or negative) in terms of sentiment. They focused on gender disparities in this set of users, implying that female users have the most radical and intolerant opinions.

Using (automatically) labelled data obtained from online sources, the researchers used supervised learning algorithms to tackle the related challenge of detecting sentiment polarity in reviews. Our baseline trials on this topic, however, reveal that individuals may not always have the best intuition when it comes to selecting discriminating phrases. In the prior study by Brody & Elhadad [27], they did experiment with a number of various features, but their main focus was not on feature engineering.

The trend in sentiment analysis research is still untapped. Ambiguity, co-reference, Implicitness, inference, emotion detection, and other natural language overheads hampered sentiment analysis as well. To increase the accuracy and performance of the methodologies, applications, or algorithms proposed. It makes it easier for them to comprehend meaning and characteristics. However, there are still certain issues and limitations in analysing reviews and documents for sentiment scores.

2.3 Co-reference

2.3.1 Rule-based Models

The task of reference resolution in NLP has long been regarded as one that inevitably relies on certain hand-crafted rules. These principles are based on the syntactic and semantic characteristics of the text at hand. A constant point of contention has been which elements aid resolution and which do not. There have also been studies particularly designed to address this issue [1]. While most early AR and CR algorithms relied on a complex set of hand-crafted rules (often knowledge expensive) and so were knowledge-intensive, others attempted to reduce this dependency.

One of the first algorithms to deal with AR was Hobb's nave algorithm [6]. To find an antecedent, our algorithm used a rule-based, left-to-right breadth-first traversal of the syntactic parse tree of a sentence. Hobb's approach also used selectional constraints based on world knowledge for antecedent removal. Despite the fact that the majority of rule-based algorithms were knowledge-rich, some [5] tried to reduce the level of dependency of rules on external knowledge. The "knowledge-poor algorithms" were labelled as such. CogNIAC was a resource-constrained high-precision coreference resolver. This early strategy got us closer to understanding how humans resolve references.

This multi-sieve technique presented a CR architecture built on a sieve that applied tiers of deterministic rules in a one-by-one order. Each sieve was formed on the previous cluster's output. The sieve architecture ensured that the most essential constraints were prioritised. There were two stages to this method. The first was the mention processing phase, which involved extracting, sorting, and pruning mentions using several constraints. The multi-pass sieve phase uses numerous passes such as string match, head match, precise constraints such as appositives, and shared traits such as gender, animacy, and number, among others. This system outperformed most of the baselines when tested on the same datasets as the H and K models.

At the CoNLL 2011 shared task, an extension of the multi-sieve approach was presented. The addition of five extra sieves, a mention detection module at the start, and a post-processing module at the conclusion to offer the output in OntoNotes format were the primary changes made to the previous system. Both the open and closed tracks of the task placed this system top. Lee et al. reported a more full study and a more thorough review of this system, outlining the particular sieves used with an easy-to-understand and intuitive example.To facilitate CR, this technique, like the previous system, incorporates shared global entity-level information such as gender, number, and animacy into the system. The composition of several sieves utilised in this deterministic system is shown in 2.1.

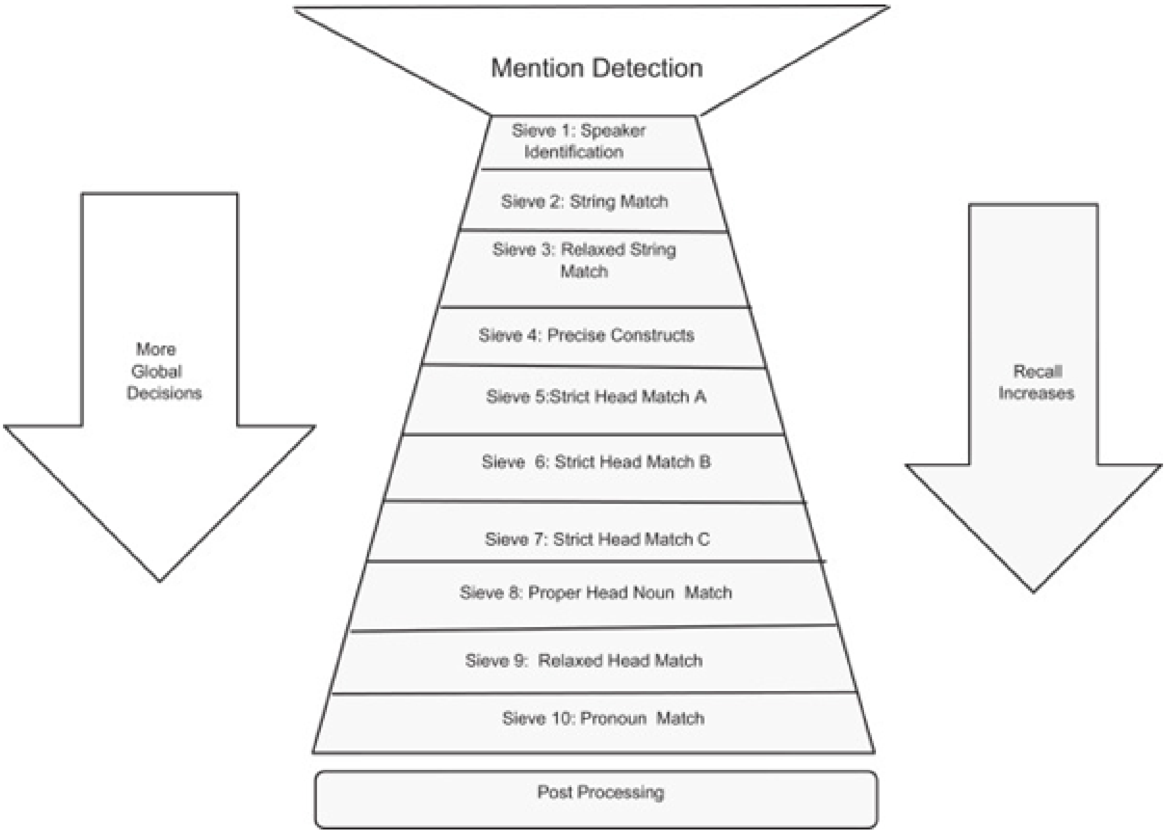

Figure 2.1: Coreference Resolution Sieve

Apart from the approaches outlined previously, which included a mix of salience, syntactic, semantic, and discourse restrictions, attempts to induce world knowledge into CR systems have also been attempted.

When there is no access to extensive training data in the desired target scheme, the change in CR research from rule-based systems to deep learning systems has resulted in a loss of the ability of CR systems to adapt to varied coreference phenomena and boundary definitions. Dependency syntax was also used as input in a recent rule-based algorithm. It attempted to target coreference kinds such as cataphora, compound modifier, i-within-i, and others that were not annotated by the CoNLL 2012 shared task. This method, known as Xrenner, was tested on two quite distinct corpora, the GUM corpus and the WSJ corpus.

We provide a summary of the rule-based approaches in Table 2.1.

Dataset	Evaluation Metric	Metric Value	Algorithm Rule Types
Fictional, Non-fictional, Books, Magazines, Part of Brown Corpus	Hobb's metric	88.3% (without selectional constraints) 91.7% (with selectional constraints)	Syntax-based rules + Selectional rules
Five Computer Science Manuals	Hobb's metric	74% (inter-sentential) 89% (intra-sentential)	Hybrid of: Syntax Rules + Discourse Rules + Morphological + Semantic
2 of the fiction and non fiction books same as Hobb's + 5 Human-keyword and task-oriented and task oriented databased	Hobb's metric	Overall: 77.6%	Discourse-based rules and constraints
Narratives	Precision and Recall	P:92% R:64%	Discourse rules+Syntax rules
Random Texts from Brown Corpora	Hoobs's metric	Overall:77%	Semantic constraints + Discourse constraints + Syntactic Constraints

ACE 2004 Roth-dev	MUC, B3, CEAF (F1 values)	MUC:75.9, B3:77.9,CEAF:72.5	Syntactic rules + Semantic rules
ACE 2004 Roth-dev	MUC, B3	MUC: 78.6, B3:80.5	Syntactic rules + Semantic rules(minimal)
ACE 2004 Culotta-test	MUC, B3	MUC:75.9, B3:81	Syntactic rules+Semantic rules
ACE 2004	nwire	MUC:79.6, B3:80.2	
MUC6-Test		MUC:78.4, B3:74.4	
CoNLL 2012	MUC, B3, CEAF, CoNLL	MUC:63.72, B3:52.08, CEAF:47.79,	
GUM corpus	MUC, B3, CEAF, CoNLL	MUC:55.95, B3:49.09, CEAFe: 44.47, CoNLL:49.84 Syntactic Rules	
Wall Street Journal Corpus		MUC:49.23, B3:41.52, CEAFe:41.13, CoNLL: 43.96	

Table 2.1: Rule-based Algorithms

2.3.2 Statistical and Machine Learning Models

During the late 1990s, the area of reference resolution shifted from heuristic and rule-based methods to learning-based methods. Decision trees, evolutionary algorithms, and Bayesian rules were some of the first learning-based and probabilistic techniques for AR.

These methods laid the groundwork for learning-based approaches to reference resolution, which improved over time and eventually surpassed rule-based algorithms. The availability of labelled coreference corpora like as MUC and ACE corpora was a major factor in this transition. From linguists to machine learning aficionados, CR's research community has grown. The mention-pair, entity-mention, and ranking models are three types of learning-based coreference models.

Coreference was considered as a set of pairwise linkages in the mention-pair [14] model. A classifier was employed to determine if two NPs are co-referent. This was followed by a step in which the linkages were reconciled using methods such as greedy partitioning or clustering to construct an NP partition. The decision tree classifier was initially used to suggest this notion for pronoun resolution in the early 1990s, and it is still recognised as a basic but successful model. The mention pair model featured three primary phases, each of which drew a lot of interest from researchers. It's worth noting that the training for the above classification and clustering phases is separate, and that bettering one stage's performance does not always imply bettering the accuracy of the other.

The construction of training examples was the first step in the mention-pair model. The goal of training instance construction was to lessen the skewness in the training samples because the majority of entities in the text were non-coreferent. Soon et alheuristic .'s mention creation approach [16] was the most widely used algorithm for creating mention instances. Soon's method generated a positive instance between an NP A1 and its nearest preceding antecedent A2, as well as a negative instance by pairing A1 with each of the NPs between A1 and A2. For instance creation, it just took into account annotated NPs. Another constraint imposed by this technique was that a positive instance between a non-pronominal instance A1 and an antecedent A2 could only be formed if A2 was non-pronominal as well.

The training of a classifier was the second phase of mention-pair models. As classifiers for CR, decision trees and random forests were commonly utilised. Statistical learners, memory learners such as Timbl, and rule-based learners were also quite popular.

Fernandes et alapproach .'s was another recently developed model that completely omitted the categorization phase. There were only two phases of mention detection and clustering in their model. A collection of document mentions x and the correct co-referent cluster y served as the training cases. The cluster features determined the training target (lexical, semantic, syntactic, etc.). This algorithm received a CoNLL score of 58.69 and was one of the top performers in the CoNLL 2012 shared-task closed track.

Despite being a frequently used model for CR, the mention-pair model has numerous fundamental flaws. The first was the transitivity constraint, which was imposed but did not always hold. This meant that if a mention A referred to a mention B and a mention B referred to a mention C, it was not always true that A co-referred with C. For example, consider the case where she is predicted antecedent of Obama and Obama is predicted antecedent of he, but because he is not co-referent with she due to a violation of the gender constraint, the transitivity condition should not be applied. This problem occurred mostly because the coreference classifier's previous decisions were not used to rectify future decisions. Because the pronoun here was semantically empty, the information from the two NP's here Obama and he was insufficient to make an educated judgement that they are co-referent. Furthermore, the NP Obama was ambiguous in its own right and could not be assigned any semantic features such as gender. Another drawback of the mention-pair model was that it only measured how good an antecedent was in relation to the anaphoric NP, not how good it was in relation to other antecedents. The entity-mention models and mention-ranking models were introduced to address the shortcomings of the mention-pair models.

In the entity mention model for CR, each referent in discourse has a single underlying entity. Instead of making coreference decisions independently for each mention-antecedent combination, this genre of algorithms was driven by the need to use prior coreference decisions to influence future ones. With the mention-pair model, the entity mention model attempted to categorise whether an NP was co-referent with a previous partially formed cluster instead of an antecedent, in order to address the "expresiveness" issue. As a result, the classifier's training examples were changed to a pair of NP N and cluster C, as well as a label indicating whether the NP's assignment to the partial cluster was positive or negative. Instead of pairwise characteristics, instances were represented as cluster-level features. The "ANY", "ALL", "MOST", and other predicates were used to specify cluster-level attributes such as gender and number over subsets of clusters. Many researchers have looked into the entity mention model.

The use of a binary classifier to determine whether an antecedent was co-referent with the mention was a problem in mention-pair models. The binary classifier could only give you a ”YES” or ”NO” answer, and it couldn’t tell you how good one antecedent was compared to the other. By ranking the mentions and selecting the best candidate antecedent, the ranking algorithms were able to avoid this issue. Because it reflected the struggle between distinct antecedents, ranking was seen to be a more natural technique to anticipate coreference relationships. The tournament models and the twin candidate model proposed by Yang et al. were two approaches developed to achieve this goal. On closer inspection, prior rule-based techniques similarly used hierarchical restrictions or sieves, starting with the most important and converge to the best antecedent. As a result, they, too, ranked the antecedents using restrictions that were prioritised by importance. Dennis and Baldridge’s algorithm, which replaced the classification function with a ranking loss, was a particularly well-known study that used mention-ranking. Another mention ranking model that solely employed surface features and used a log-linear model for antecedent selection surpassed the Stanford system, which won the CoNLL 2011 shared task, by 3.5 percent, and the IMS system, which was the top model for CR at the time, by 1.9 percent.

Despite its immense popularity, the mention rankers were unable to effectively use earlier decisions to make current decisions. This prompted the development of ”cluster ranking” algorithms. The greatest aspects of entity-mention and ranking models were combined in cluster ranking approaches. In recent deep learning models for CR [4], a mix of mention ranker and cluster ranker has been used. In addition, the mention-ranking model did not distinguish between anaphoric and non-anaphoric NPs. The difficulty is addressed by recent deep learning-based mention ranking systems [3], [19], which teach anaphoricity alongside mention ranking. One of the first machine learning methods to achieve this was through the work of.

On the CoNLL 2012 shared task, an entity centric model was previously the best performing model. It, like other machine learning algorithms, has a number of characteristics. Defining characteristics for mentions, let alone clusters, is a challenging task. Feature extraction is also a time-consuming operation. This began to alter gradually with the introduction of deep learning for NLP.

2.3.3 Deep Learning Models

The goal of reference resolution research has been to lessen reliance on hand-crafted features since its inception. Words could be represented as vectors communicating semantic dependencies, thanks to the development of deep learning in NLP. This boosted the popularity of approaches that used deep learning for CR.

By pre-training on these two independent subtasks, the first non-linear mention ranking model for CR intended to learn different feature representations for anaphoricity detection and antecedent rating. This method addressed two major issues in CR: the first was identifying non-anaphoric references in the text, and the second was the complicated feature conjunction in linear models, which was required because simpler features were unable to distinguish between truly co-referent and non-coreferent mentions. This approach addressed the concerns raised above by adding a new neural network model that only received raw unconjoined information as input and sought to learn intermediate representations.

The programme began with liberal mention extraction using the Berkeley Coreference resolution system, then used representation learning to better capture relevant features of the task.

The first non-linear coreference model that demonstrated that modelling global aspects of entity clusters can help with coreference. Added entity-level information obtained by a recurrent neural network (RNN) running over the candidate antecedent-cluster to the neural network-based mention-ranking model. The scoring function of the antecedent ranking model was modified by the addition of a global scoring term in this model. The goal of the global score was to determine how consistent the present mention was with the antecedent's partially completed cluster. Separate weight sharing RNNs were used to represent the clusters, which successively consumed the mentions assigned to each cluster. The goal was to record the history of past judgments as well as the mention-antecedent relationship. Instead, the Clark and Manning method, which was proposed about the same time, defined a very different cluster ranking model to induce global information.

This method was founded on the concept of integrating entity-level information, or characteristics specified across clusters of mention pairs. The mention-pair encoder passes features (described later) through a feed-forward neural network (FFNN) to produce distributed representations of mentions, a cluster-pair encoder uses pooling over mention pairs to produce distributed representations of cluster pairs, and a mention ranking model to primarily pre-train weights and obtain scores to be used further in cluster rank (Fig. 2.2).

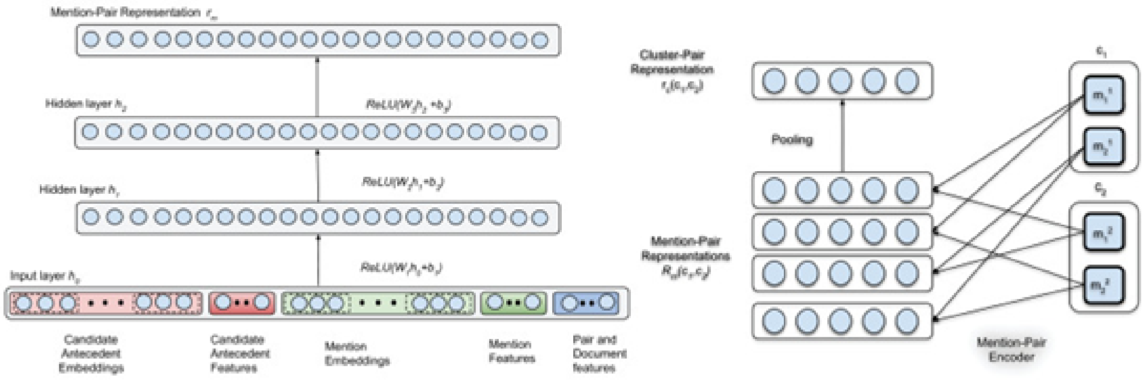

Figure 2.2: The Mention-pair and the Cluster-pair encoder

The following features were used in the entire model: the average of word embeddings in each mention, binned distance between mentions, head word embedding, dependency parent, first, last, and two preceding word embedding, average of 5 preceding and succeeding words of the mention, type of mention, position of mention, sub-mention, mention-length, document genre, string match, and so on. These characteristics were combined into an input vector and sent into an FFNN with three fully linked hidden rectified linear layers. The vector representation of the mention pair was the last layer's output. Given two clusters of mentions ci=mi1, mi2,..., mi—ci— and cj = mj1, mj2,..., mj—cj—, the cluster pair encoder provides a distributed representation rc(ci, cj) ∈ R2d. The max and average pooling over the mention-pair representations were used to create this matrix. Following that, a mentionpair model was trained on the mention pair encoder's representations in order to give pre-training weights for the cluster ranking job as well as a measure for coreference judgments. The slack rescaled goal that was explained before was used to train this mention ranking algorithm. Cluster ranking was the final stage, which used the mention ranking model's pre-trained weights to generate a score by feeding the cluster encoder's cluster representations to a single-layered fully-connected neural network. Merge (combine two clusters) and pass (no action) were the two options depending on scores. At each phase of the inference process, the highest-scoring (most likely) action was taken. This cluster ranking ensemble outperformed previous state-of-the-art techniques, scoring 65.39 on the CoNLL English task and 63.66 on the Chinese challenge.

Despite depending on few characteristics, the current state-of-the-art model is an end-to-end CR system that outperforms prior techniques. This end-to-end neural model [10] mentions detection and CR in the same sentence. This method began with the production of high-dimensional word embeddings to represent the words in an annotated document. The word embeddings were created using Glove, Turian, and character embeddings. Three different window sizes of a character-level convolutional neural network were used to train the character embeddings (CNN). To acquire effective word representations from the document's vectorized phrases, a bidirectional long short-term memory (LSTM) network was utilised (Fig. 2.3). Then, in each document, all possible mentions were retrieved and represented as a one-dimensional vector. This mention representation was created by combining the start word embedding, head word embedding, end word embedding, and a few more mention-level features. The attention mechanism was used to learn the head word embedding throughout the course of the whole mention span. gi = [xSTART I xEND(i), x′i, I was specified as the mention representation. The attention-weighted sum of the word vectors in span I was x′i, and the span bounds were xSTART I and xEND(i). Only spans of maximum width ten were examined in this technique, which trimmed candidates ruthlessly for training and evaluation. Only a subset of the highest scoring spans were saved for CR after the mentions were scored using an FFNN.

These top-scoring remarks were fed into the CR model (Fig. 2.4). From the past 250 references, candidate antecedents were picked. The scores for the mention-antecedent pairs were obtained using the formula below. The mention-antecedent pair representation was built up of individual mention representations gi and gj, similarity between the two mentions gigj, and paired features I j) representing speaker and distance qualities. As shown in the equation below, the final scoring function is the sum of the two individual mention scores of the candidate mentions, as well as the mention-antecedent pair score. sa(i,j)=wa · FFNNa([gi,gj,gi∘gj,(i,j)])

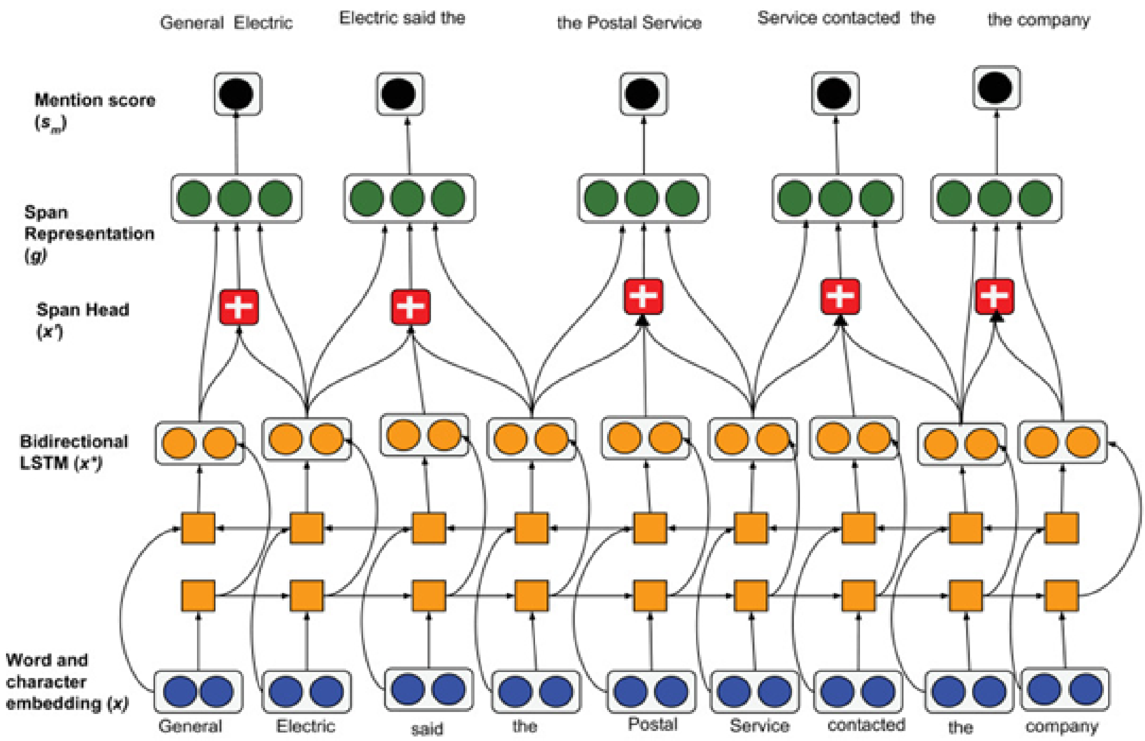

Figure 2.3: Bi-LSTM to encode sentences and mention scoring

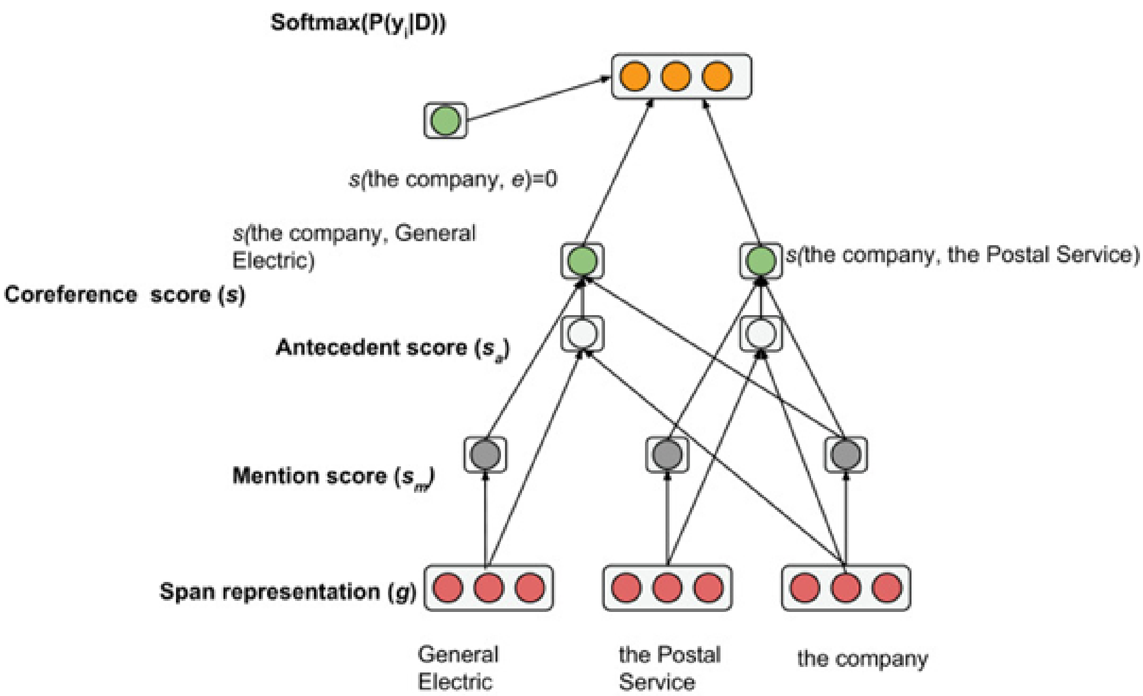

Figure 2.4: Antecedent Scoring

The marginal log-likelihood of all accurate antecedents on the basis of gold-clustering was utilised as the model's optimization function. log∏i=1N∑y′∈YiĜOLD(i)P(y′) During inference, the antecedent with the highest score was picked as the most likely antecedent, and coreference links were built using the transitivity characteristic.

The authors present the results of their experiments using five models with various initializations, and they prune the spans here by taking the average of the mention scores across each model. On MUC, B3, and CEAF measures, the suggested technique was thoroughly examined for accuracy, recall, and F1.

This model's computation time is lengthy, and it necessitates the saving of a large number of trainable parameters. This model is difficult to maintain since it relies on a large deep neural network. As a result, implementing the system as a ready-to-use off-the-shelf solution is tough.

Chapter 3

Dataset Preparation

3.1 Data Crawling

Initially, the raw data in the legal forum took the shape of a series of user-posted legal concerns and possible legal advice from Indian legal professionals. It was then turned into a conversational format for usage by a user and a trained conversational agent. BeautifulSoup, a Python library, is used to scrape the data. There were 350 distinct legal cases collected in all. Each of the raw examples is then processed further to get it into a conversational format. Table 3.1 shows a sample of the scrapped corpus in its raw form.

Speaker	**Statement**
User	Q: My husband has been physically harassing me for years. Need help.
legal expert 1	1. You can apply for divorce
legal expert 2	1. File for divorce 2. Apply for maintenance.
legal expert 3	1. Make a police complaint 2. Shift to your maternal home 3. Send him legal notices 4. If you want a divorce then file for it

Table 3.1: Sample of the scrapped corpus

3.2 Data Annotation

3.2.1 Data Preparation Guidelines

The crawled data contains information as a set of question answers from user and multiple lawyers. Each of the unprocessed legal cases was turned into a series of legal talks between a client and an automated agent. We noticed that the majority of the raw cases lacked information, which could lead to incorrect solutions. As a result, one of the automated agent's key responsibilities would be to ask relevant questions depending on the given scenario in order to extract missing critical pieces of information via question answering. To accomplish so, diverse legal advice from different specialists was studied to identify the events' direction and chronology. Each incident is thereafter represented as a pair of issues with related legal advice.Finally, the dataset's informative ones are picked. Table 2 depicts the various conversation flows based on the raw text of Table 3.1, although only conversation 5 is appropriate for the dataset.

Solution id	**Solution text**
conversation 1	User: My husband has been physically harassing me for years. Need help. Bot: You can file for divorce.
conversation 2	User: My husband has been physically harassing me for years. Need help. Bot: Make a police complaint and shift to your maternal home for now.
conversation 3	User: My husband has been physically harassing me for years. Need help. Bot: Shift to your maternal home for now. Then apply for divorce and maintenance.
conversation 4	User: My husband has been physically harassing me for years. Need help. Bot: If possible try for mutual settlement else you can go for a police complaint.

conversation 5	User: My husband has been physically harassing me for years. Need help. Bot: How long are you married? User: 3 years Bot: First of all, file a written complaint at the nearest police station. User: We have tried to solve this mutually but he is not listening Bot: Do you have any children? User: Yes, One boy. He is only 1 year old. Bot: Do you want to live with him in the future? User: No. But I have no option. Bot: File a divorce case on the ground of mental cruelty User: But how will I survive if i divorce him? Bot: File a maintenance case too. User: Will it help? Bot: Yes. You may get divorce if harassment is proven and maintenance as well User: Thank you very much. Bot: Welcome.

Table 3.2: The possible conversations flow w.r.t. the raw text mentioned in Table 3.1

3.2.2 Description of the Dataset

We focus on understanding the user's intention as well as the gravity of the situation. Due to that reason each user utterance has been annotated with two attributes, intent name and sentiment score within the range of -5 to +5. In case of bot utterances annotation has been limited with intent name only. Figure 3.1 represents the hierarchical structure of the intent tree.

The intents are categorised as a hierarchical tree to show how they are linked and how different intents determines each others. the hierarchy of intents looks like this:

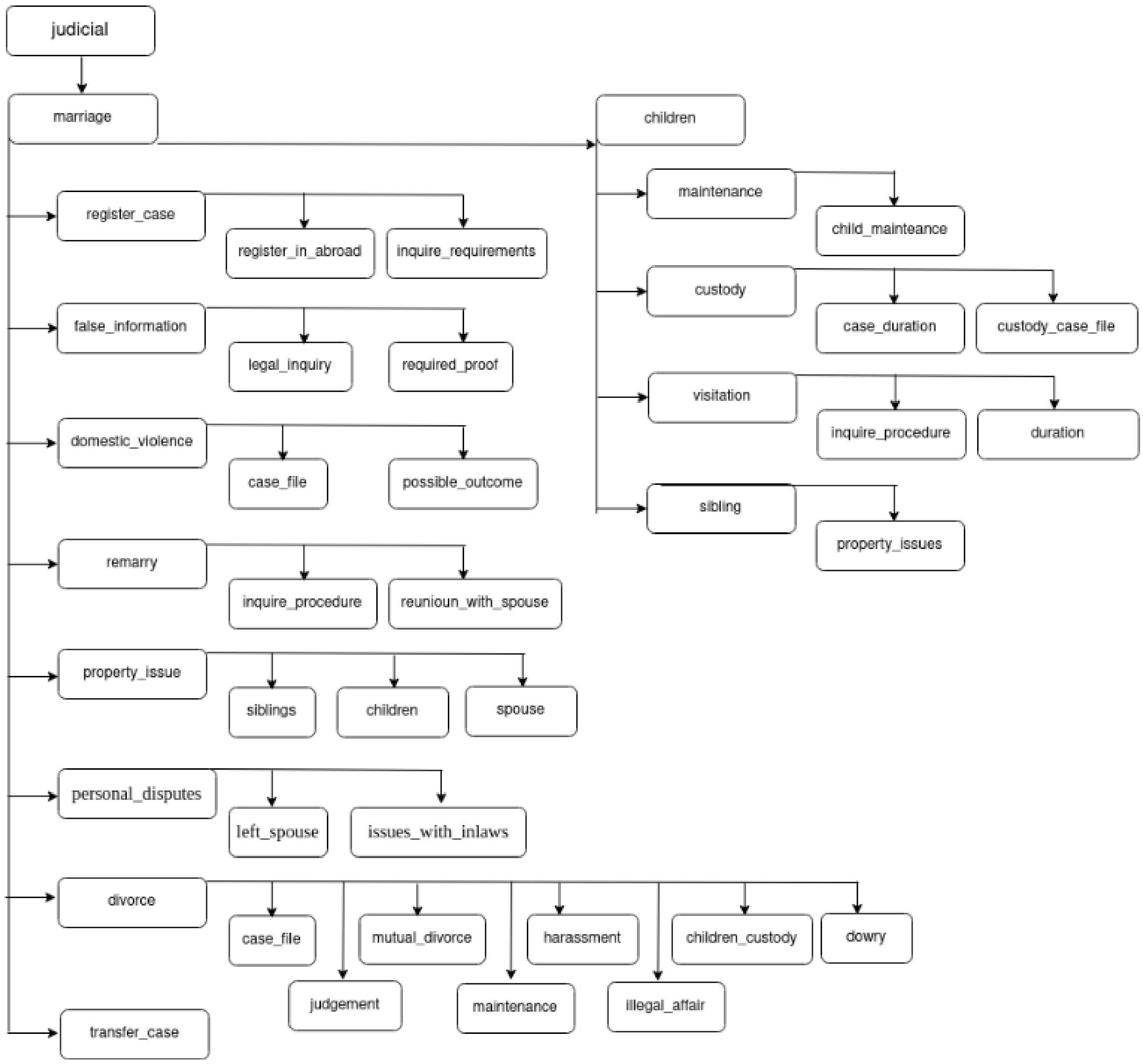

Figure 3.1: The hierarchical structure of the intent tree

Tag Name (occurrence)	**Parent Tag**	**Description**
judicial	-	-
marriage	judicial	-
divorce	marriage	-
case_file (1203)	divorce	If the user desires to apply for divorce.
mutual_divorce (101)	divorce	If the user desires to apply for a mutual divorce.
harassment (178)	divorce	If the user desires to apply for divorce because of harassment by the partner.

judgement	divorce	-
maintenance (283)	divorce	if the user is curious about maintenance related queries.
children_custody (125)	divorce	If the user is curious about custody of the child after divorce.
dowry (18)	divorce	If the user desires to apply for divorce due to the pressure of providing dowry.
illegal affair (33)	divorce	If the user desires to apply for divorce due to the illegal affair of their partner.
register_case	marriage	-
register_in_abroad (74)	register_case	If the marriage takes place abroad but needs to be registered in India or vice versa.
inquire_requirements (44)	register_case	If the user is curious about the requirements of the registration process.
false_information	marriage	-
legal_inquiry (405)	false_information	if the user is curious about the legal steps if false information is given before marriage
required_proof (99)	false_information	if the user is curious about the required proofs to handle case of false information
domestic_violence	marriage	-
case_file (46)	domestic_violence	if the user wants to file a case of domestic violence
possible_outcome (73)	domestic_violence	if the user is curious about the outcome of a registered case of domestic violence.
remarry	marriage	-
inquire_procedure (40)	remarry	if the user is curious about conditions of remarriage.
reunioun_with_spouse	remarry	if the user is curious to bring back a spouse.
-12		
property_issue	marriage	-

spouse (11)	property_issue	if the user is curious about the ancestral property distribution after divorce.
siblings (75)	property_issue	if the user is curious about ancestral property distribution between siblings.
children (16)	property_issue	if the user is curious about the share of a child in the parent's property.
personal_disputes	marriage	-
left_spouse (37)	personal_disputes	if the user wants to know how to take legal steps when their spouse has left home deliberately.
transfer_case (18)	marriage	if the user is curious about the legal process to transfer a case from one location to another.
children	judicial	-
custody	children	-
case_duration (126)	custody	if the user is curious about maximum time can be taken in child custody related cases.
custody_case_file (63)	custody	if the user wants to file a child custody related case.
maintenance	children	if the user wants to know about the child maintenance case after divorce.
visitation	children	-
duration (20)	visitation	if the user is curious about visitation rights after divorce.
inquire_procedure (56)	visitation	if the user is curious about the visitation procedure after divorce.
sibling	children	-
property_issues (6)	sibling	if the user wants to know how property should be shared among siblings after divorce.

Table 3.3: Description of intent tree mentioned in Figure 3.1

3.2.3 Statistics of the Dataset

Statistical analysis on table 3.4 signifies that sentiment of the legal corpus is negatively biased in general which in turn validates the psychological state of worried clients. A total number of 1440 user utterances have been taken in consideration. The data can be visualized by a normal distribution of N(-0.5211, 1.3479). Figure 3.3 and Table 3.5 represent details of the sentiment distribution. Here Table 3.4 presents the statistics of the overal dataset.

Feature name	**Value**
Maximum length of utterance by user (in character)	242
Minimum length of utterance by user (in character)	1
Average length of utterance by user (in character)	24.94
Total number of user utterances	1440
Maximum length of utterance by bot (in character)	220
Minimum length of utterance by bot (in character)	1
Average length of utterance by bot (in character)	19.58
Total number of bot utterances	1414
Total number of dialogues	430
Maximum length of a dialogue	15
Minimum length of a dialogue	3
Average length of a dialogue	7.81
Total number of positive utterance by a user	242
Total number of negative utterance by a user	622
Total neutral utterance by user	728
Maximum different intents appeared in a dialogue	13
Minimum different intents appeared in a dialogue	1
Average Number of different intents appeared in a dialogue	2.683795712
Number of time sentiment changes to +ve to -ve	20
Number of time sentiment changes to -ve to +ve	5

Table 3.4: Statistical details of the dataset

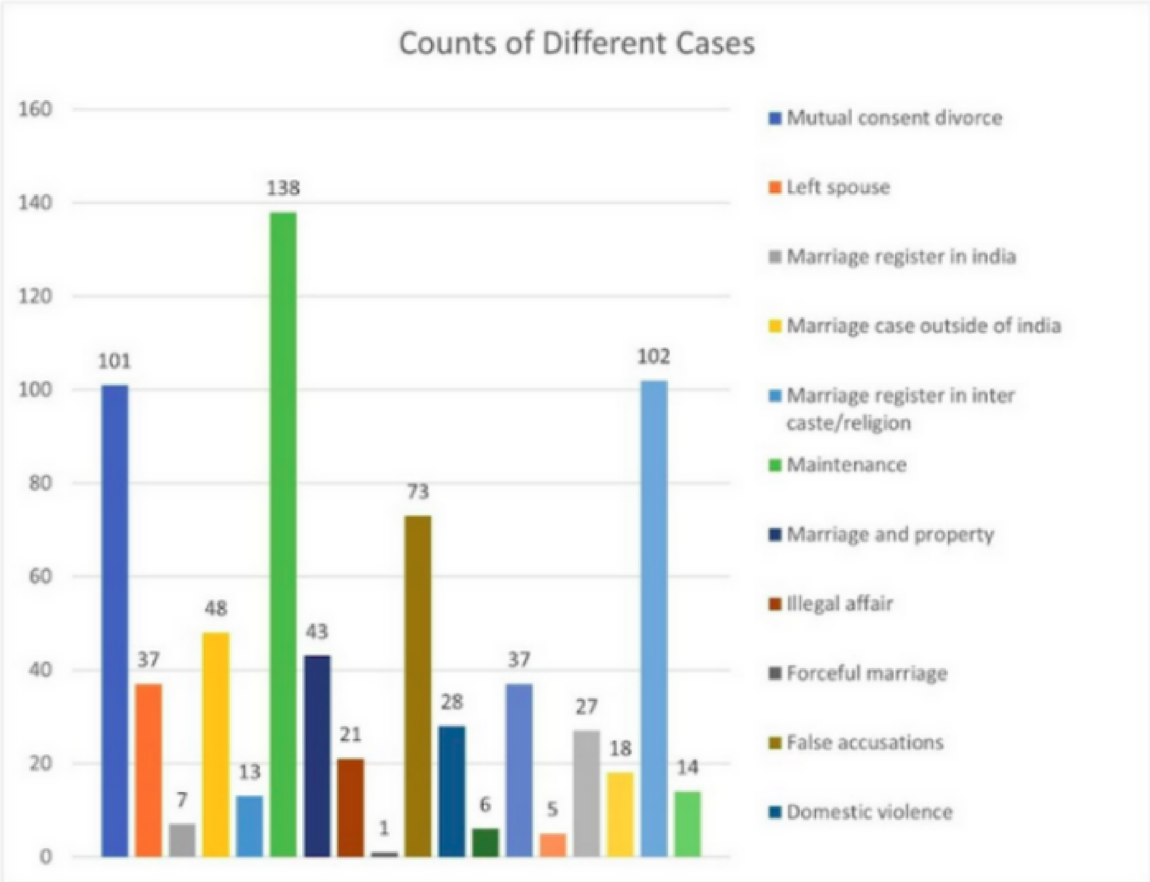

Figure 3.2: A histogram plot of the intent tree mentioned in Figure 3.1

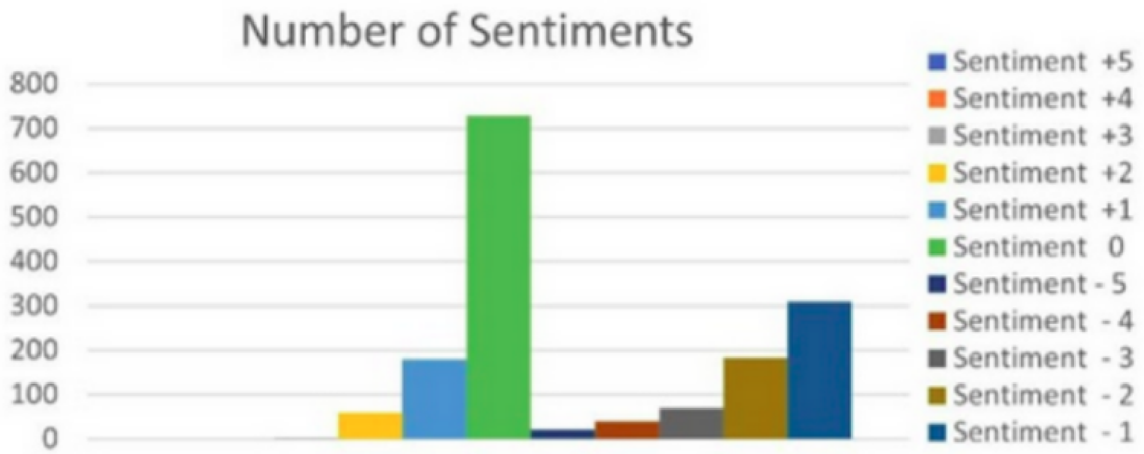

Figure 3.3: A histogram plot of sentiment data in the dataset.

Feature name	**Value**
Total count (N)	1440
Mean	-0.5211
Standard deviation	1.3479
Variance	1.8167
Median	0
Skewness	1.21
Kurtosis	1.1584
A-Squared	73.79
p-Value	< 0.005
Minimum sentiment score	-5
Maximum sentiment score	5
First Quartile	-1
Median	0
Third Quartile	0

Table 3.5: Statistics report of sentiments at sentence level

Chapter 4

Intent Classification

4.1 Subtask 1 : Identification

The second task is to identify words or tokens that re responsible for that perticular utterance to relate to the intent. There can be one or more than one word that can be marked as intent-word.

4.1.1 Task Description

Our task is to identify potential words that better represent intent classes. The utterances can contain one or more than one word that can be potentially intent word. So that we later on can deduct relationship between intent and sentiment using these words.

4.1.2 Dataset Description

The first thing we need to identify the feature words that determines the intent classes, we need to do some feature extraction. For that we first use TF-Idf to extract feature words that potentially can be used for identifying intent-word. We perpare a dataset and mark the feature word that are present in the utterances as potential-intent words.

Secondly we prepared a dataset to mark tokens in the utterances if they are intent-words are not to feed into an entity recognition model based on BERT to identify possible intent classes in the conversation. The utterances has been broken into tokens; so each row of the dataset contains the tokens, their corresponding sentence number their pos tag and entity recognition tag. The snippet of the dataset is given follows:

SENTENCE	WORD	POS	TAG
Stentence1	hello	NN	Other
Stentence2	hello	VB	Other
Stentence2	how	WRB	Other
Stentence2	can	MD	Other
Stentence2	i	VB	Other
Stentence2	help	NN	Inten

Table 4.1: Dataset snippet for intent word identification

4.1.3 Experimental Setup

The TF-Idf model is developed on the primary dataset to identify potential intent-words.

- Size of the primary dataset : 3178 utterances
- Intent word found in : 2841 utterances
- Intent word not found in : 337 utterances

Then the dataset fed into the entitity recognition model based on BERT to have a neural network identify the pattern and classify the tokens if they are intent or not.

The description and the hyper-paremeters of the entity recognition model are as follows:

- Model Name : bert-base-uncased
- Training data size : 64%
- Validation data size : 16%
- Test data size : 20%
- Number of epochs : 20
- Learning rate : 1e-5
- Train batch size : 16

- Eval batch size : 16
- Number of output classes : 39

4.1.4 Results and Observations

Metrics	Result
Precision	0.64
Recall	0.59
F1-score	0.61

Table 4.2: Experimental result of intent identification model

4.2 Subtask 2 : Classification

Intent classification is the automatic classification of text in dialogues or in coversations. the categorization depends on the domain it is based on. The intents are used to tag keywords in conversation so users can easily identify different topics discussed in a conversation.

4.2.1 Task Description

The dataset contains conversational utterances and each utterances consist of question-answer on different topics. There are two speaker of each utterances, namely the user and the bot. The dataset has been annotated to have one or more than one intent tagged to each utterances irrespective of the speaker. Our task is to classify those intents based on the given hierarchy of intent classes. Intent classification uses NLP and different ML and DL algorithms to classify:

4.2.2 Dataset Description and Experimental Setup

4.2.2.1 ML Models

Understanding the context of any natural language text is one of the challenges in dealing with text data. As a result, we need to contextualise texts in vectorized format, which is why, while constructing machine learning models, we tried to deploy embeddings using the TF-IDF vectorizer first. For text feature extraction, we utilised the TfidfVectorizer module from the scikit-learn package. The following are the machine learning classifiers that we used to assess the performance of our intent categorization in this study.

The BERT-based architecture, on the other hand, beats the rest of the machine learning and deep learning models substantially.

To feed machine learning models the dataset has been created in the following way:

SPEAKER	STATEMENTS	INTENT
User	No, I have decided to end this relationship.	@marriage@divorce
Bot	else If you want to terminate the matrimonial relationship with him and become free from him without bothersing divorce him and file domestic_violence FIR and seek alternative accommodation , maintenance, compensation for mental torture from husband.	@marriage@divorce@file @harassment@cruel@case_file @domestic_violence @maintenance @residence
Bot	yes police can frame FIR but the written statement will be used as a defense when police will take your statement.	@marriage @personal_disputes @left_spouse @false_accusations @proofs@defamation @case_file@divorce @suggestion
User	ok.	acknowledge

Bot	The decree granted by a Foreign Court is considered to be legal, validity and binding in the Indian Courts by the virtue of Section 14 of the Civil procedure Code.	marriage@divorce @legal_enquiry
Bot	Kindly share the details.	marriage@divorce @mu- tual@legal_enquiry
Bot	Where did you get married?	marriage@divorce @validity
Bot	There is no such law which rescribes to use husband only.Depending upon the con- venience she may choose her surname.	marriage@modify_docs
User	He recorded her voice/call conver- sation with that person in mobile. He has sufficient audio recordings of both which clearly shows that they are in physical relation.	proofs
Bot	welcome, have a nice day	thank

Table 4.3: Snippet of dataset for intent classifiaction for ML models

- **Multinomial Naive Bayes** The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts captured from TF-IDF vectorizer). The multinomial distribution normally requires integer feature counts. We have used the MultinomialNB module from scikit-learn package with default parameters.

- **Stochastic gradient Descent** This estimator implements regularized linear models with stochastic gradient descent (SGD) learning: the gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule (aka learning rate). SGD allows minibatch (online/out-of-core) learning via the partial_fit method. For best results using the default learning rate schedule, the data should have zero mean and unit variance. We have used the SGDClassifier module from scikit-learn package with hinge loss and l2 penalty and learning rate of 1e-5.

- **Support Vector Machine** Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. We have used the svm module from scikit-learn package with linear kernel.

- **Logistic Regression** Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. We have used a logistic regression module from scikit-learn package with learning rate of 1e5, maximum iteration of 100.

4.2.2.2 BERT

BERT makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. In its vanilla form, Transformer includes two separate mechanisms — an encoder that reads the text input and a decoder that produces a prediction for the task. We have fine tuned the 'bert-base-uncase' model from the hugging-face library to do multi-label classification.

To feed BERT for multi-label classification the dataset has been created in the following way:

STATEMENTS	marriage	divorce	legal	file	legal_enquiry	thank	greet	advice	maintenance
hello.	0	0	0	0	0	0	1	0	0
hello, how can i help you?	0	0	0	0	0	0	1	0	0

I have been married for 6 years & having kids....	1	0	0	0	1	0	0	0	0
My mothers in law is giving fuel to that.Alway...	1	0	0	0	1	0	0	0	0
You can not prove that your mothers in law is ...	1	0	0	0	0	0	0	0	0

Table 4.4: Snippet of dataset for intent classifiaction for BERT

The description and the hyper-paremeters of the intent recognition model are as follows:

- Model Name : bert-base-uncased
- Training data size : 64%
- Validation data size : 16%
- Test data size : 20%
- Number of epochs : 20
- Learning rate : 1e-5
- Train batch size : 24
- Eval batch size : 24
- Threshold : 0.2

4.2.3 Results and Observations

Metrics	MNB	SGDC	SVM	LR	MBERT
Accuracy	0.27	0.41	0.41	0.40	0.95
Precision	0.15	0.38	0.36	0.38	0.59
Recall	0.26	0.38	0.42	0.42	0.49
F1-score	0.17	0.37	0.37	0.39	0.53

Table 4.5: Experimental result for different intent classification models

4.2.4 Error Analysis

In the case of intent data, it has been found that a single utterance can include many intents. For instance, the utterance ”Can I visit my daughter after the divorce” combines the intents of ”visitation” and ”case file” of divorce. In these circumstances, intent detection by intent classification models is only partially successful.

A single utterance may occasionally have many intentions. For instance, when I brought up divorce, she said, ”I can't endure his abuse.” I want to file a case,” the user does not specify whether she wants to launch a lawsuit for divorce, harassment, or both.

Utterance	Actual Intent	Predicted Intent
Model 1: Multinomial Naive Bayes		
So, we were separated for 3 yrs. Now she has an affair with another person. I need divorce.	left_spouse, illegal_affair, case_file (divorce)	left_spouse, child_visitation
My wife uses bad and abusive language to me and my old father and mother. She has asked about divorce and 20L for maintenance	harassment, case_file(divorce) , maintenance	child_custody, transfer_case

My mother-in-law is forcing me to sell our ancestral bunglow. But I have four brothers, they are not ready for it.	inlaw_harassment, property_issues	property_issues, remarry
Model 2: Stochastic Gradient Descent Classifier		
So, we were separated for 3 yrs. Now she has an affair with another person. I need divorce.	left_spouse, illegal_affair, case_file (divorce)	left_spouse
My wife uses bad and abusive language to me and my old father and mother. She has asked about divorce and 20L for maintenance	harassment, case_file(divorce) , maintenance	harassment, legal_inquiry
My mother-in-law is forcing me to sell our ancestral bunglow. But I have four brothers, they are not ready for it.	inlaw_harassment, property_issues	property_issues, harassment
Model 3: Support Vector Machine		
So, we were separated for 3 yrs. Now she has an affair with another person. I need divorce.	left_spouse, illegal_affair, case_file (divorce)	illegal_affair, case_file (divorce) , dowry
My wife uses bad and abusive language to me and my old father and mother. She has asked about divorce and 20L for maintenance	harassment, case_file(divorce) , maintenance	inlaws_harassment, dowry, mutual

My mother-in-law is forcing me to sell our ancestral bunglow. But I have four brothers, they are not ready for it.	inlaw_harassment, property_issues	inlaw_harassment, property_issues
Model 4: Logistic Regression		
So, we were separated for 3 yrs. Now she has an affair with another person. I need divorce.	left_spouse, illegal_affair, case_file (divorce)	left_spouse, illegal_affair, remarry
My wife uses bad and abusive language to me and my old father and mother. She has asked about divorce and 20L for maintenance	harassment, case_file(divorce) , maintenance	inlaws_harassment, harassment, child_custody
My mother-in-law is forcing me to sell our ancestral bunglow. But I have four brothers, they are not ready for it.	inlaw_harassment, property_issues	property_issues, transfer_case, left_spouse
Model 5: Multi-label Classification using BERT		
Things have come to a standstill as regards your matrimonial relationship is concerned for which you shall have to decide as to what do you want to do now if he continues to act the way he is acting.	marriage, divorce, file, harassment, cruel, case_file, seeking_advice, suggestion	marriage, divorce, file, maintenance, case_file

But when they filed the domestic_violenceA on me and my family members they did attach the Original invitation with Marriage Hall and M.Sc. B.Ed as the degree.Can I file a FIR on my Wife and her Parents on the above ground?	marriage, divorce, maintenance, eligibility, case_file	marriage, divorce, file, maintenance, case_file
2005 left for 6 months with kids, 2008 july again left with kids, filed diverse case returned back in 2012, 2016 may again left with kids, Sept 16 Filed 498a , I went HQ for quashing.	marriage, divorce, children, case_file, file	marriage, divorce, file

Table 4.6: Error analysis of intent classification models

Chapter 5

Sentiment Classification

5.1 Subtask 1 : Identification

The second challenge is to determine which words or tokens are responsible for that specific utterance's sentiment. There can be one or more words that are identified as sentiment-words.

5.1.1 Task and Dataset Description

Our goal is to find words that better convey different sentiment classes. So that we can later use these words to deduce a correlation between intent and sentiment. We must first identify the feature words that determine the sentiment classes.

5.1.2 Dataset Description

To achieve this task, we first utilise textblob to extract feature words that could be used to identify sentiment of whole utterance. We create a dataset and label feature words that appear in utterances as potential sentiment-words.

Second, we created a dataset to label tokens in utterances as sentiment-words or not, which we used to feed into a BERT-based entity identification model to discover potential sentiment classes in the dialogue. Tokens have been created from the utterances, and each row of the dataset contains the tokens, their related sentence number, pos tag, and entity recognition tag. The following is a sample of the dataset:

SENTENCE	**WORD**	**POS**	**TAG**
Sentence1	hello	NN	Sentiment
Sentence3	I	PRP	Other
Sentence3	have	VBP	Other
Sentence3	been	VBN	Other
Sentence3	married	VBN	Other

Table 5.1: Dataset snippet for sentiment word identification

5.1.3 Experimental Setup

The description and the hyper-paremeters of the entity recognition model are as follows:

- Model Name : bert-base-uncased
- Training data size : 64%
- Validation data size : 16%
- Test data size : 20%
- Number of epochs : 20
- Learning rate : 1e-5
- Train batch size : 16
- Eval batch size : 16
- Number of output classes : 1

Metrics	Result
Precision	0.32
Recall	0.70
F1-score	0.44

Table 5.2: Experimental result of sentiment identification model

5.1.4 Results and observations

5.2 Subtask 2 : Classification

The technique of identifying positive or negative sentiment in text is referred to as sentiment classification. Businesses utilise it to identify sentiment in social data, assess brand reputation, and better understand their consumers. It is the analysis of utterances polarity towards different types emotions.

Sentiment analysis is quickly becoming a crucial tool for monitoring and understanding sentiment in all forms of data, as humans communicate their thoughts and feelings more openly than ever before.

Brands can learn what makes customers happy or frustrated by automatically evaluating customer feedback, such as comments in survey replies and social media dialogues. This allows them to customise products and services to match their customers' demands.

5.2.1 Task and Dataset Description

Using sentiment analysis to evaluate the responses of users using any chat-bot is important to provide necessary responses and to control the flow of conversation by handling the emotions of users such a way a fruit-full result comes out of the conversation.

The dataset has been annotated with values 0, 1, 2. When the utterance is positive the sentiment is set to 1, if it is negetive it set to 2 and in neutral cases it set to 0. Snippet of the dataset that is being used is shown in Table 5.3:

STATEMENTS	**SENTIMENTS**
"What does it mean Stage of Case is ""FOR APPERA..."	0
okay.	1
After 15 years of my marriage me and my wife ...	2
okay.If there is no validity evidence against ...	1
My married life is not happy , what will I do ...	2
Kindly tell the way with which she will not st...	1
I am not legally marriage. I don't want any t...	2
I am currently working with an MNC in USA. I w...	1
My wife has asked for Rs. 30000/month and Rs.1...	1
okay.thankyou bot.	0

Table 5.3: Snippet of dataset for sentiment classifiaction for BERT

5.2.2 Experimental Setup

5.2.2.1 ML Models

- **Multinomial Naive Bayes** An NB classifier based on the Bayes' theorem, is a probabilistic ML model used for the task of classification. It is fast to implement and yields promising results. The classifier is based on the conditional probability that a document d belongs to a class c. Bayes' formula lies in the foundation of the algorithm.

- **Support Vector Machine** A multi-class SVM has the objective to allocate the hyperplane that can effectively divide the input data into multiple separate classes. The type of hyperplane depends on multiple features. If the number of features is two, trivially, the hyperplane is a line. If the number of features is three, the hyperplane is in the form of a two-dimensional plane. If the number of features exceeds three, the hyperplane takes a complex form.

- **Logistic Regression** An LR predicts the probability of an outcome by fitting data to a logistic function. The classifier uses a linear function f (x) = w0+w1x1+...+wrxr, where w0, w1, ... , wr are the predicted weights or coefficients. The LR function p(x) is a sigmoid function,

- **Random forests** Random forests is an ensemble learning method for text classification. In the RF classifier a bunch of independent trees is built. Every document is classified by the trees independently. The class of the document is defined by the largest number of votes of all trees.

5.2.2.2 DL Models

- **CNN** In the case of CNN, 1D convolutional layers have been used. Each of the convolution layers is followed by a Max-pooling layer. A layer called Max-pooling comes after each convolution layer. The first and second convolutional layers, respectively, have 256 and 128 filters set up, while the size of the filter for both layers is set to 5. The training approach is conducted with a 64-person batch size and a 0.0001 learning rate. In both convolutional and fully connected layers, the exponential linear unit (ELU) was applied. In the first of the three dense layers, there are 128 hidden neurons with a dropout value of 0.7, and there are 64 hidden neurons with a dropout value of 0.5. For the first dataset, six softmax units and eleven softmax units were utilised to categorise each user input.

- **RNN** We chose an embedding layer of size 256 as the first layer in the RNN model-based classification framework, and then a 256-layer RNN layer. There were two further LSTM layers placed after these first two. Both of the two LSTM layers have 256 hidden neurons, with the first having a dropout value of 0.3 and the second having a dropout value of 0.3 and recurrent dropout 0.2. A dense layer with softmax activation function is present at the end. Using a learning rate of 0.001 and Softmax units, each user input for the dataset was classified.

- **BERT** As we know, it is a transformer based model which can be used for various types of classification problems. It uses stacked encoders to train it's language model so well that if it fine-tuned with even a small training dataset it produces wonderful results.

The description and the hyper-paremeters of the sentiment recognition model are as follows:

- Model Name : bert-base-uncased
- Training data size : 64%
- Validation data size : 16%
- Test data size : 20%
- Number of epochs : 20
- Learning rate : 1e-5
- Train batch size : 32
- Eval batch size : 32
- Number of output classes : 1

5.2.3 Results and Observations

Sentiment class -1	**Model 1**	**Model 2**	**Model 3**	**Model 4**	**Model 5**	**Model 6**	**Model 7**
Precision	0.69	0.59	0.64	0.65	0.64	0.69	0.91
Recall	0.42	0.89	0.75	0.71	0.85	0.81	0.9
F1-score	0.52	0.71	0.69	0.68	0.73	0.74	0.9
Sentiment class 0							
Precision	0.54	0.78	0.76	0.79	0.76	0.72	0.89
Recall	0.63	0.72	0.78	0.75	0.78	0.86	0.91
F1-score	0.58	0.75	0.77	0.77	0.78	0.78	0.89

Sentiment class 1							
Precision	0.14	0.25	0.39	0.35	0.62	0.5	0.57
Recall	0.21	0.31	0.21	0.33	0.1	0.45	0.63
F1-score	0.17	0.27	0.28	0.34	0.17	0.473	0.59

Table 5.4: Experimental result for sentiment classification models(before sampling)

Sentiment class -1	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
Precision	0.59	0.5	0.53	0.57	0.53	0.57	0.69
Recall	0.44	0.75	0.65	0.62	0.65	0.64	0.78
F1-score	0.504	0.6	0.58	0.59	0.58	0.6	0.73
Sentiment class 0							
Precision	0.39	0.84	0.76	0.79	0.71	0.6	0.84
Recall	0.62	0.51	0.6	0.58	0.75	0.56	0.89
F1-score	0.48	0.64	0.67	0.67	0.73	0.58	0.86
Sentiment class 1							
Precision	0.31	0.45	0.54	0.52	0.5	0.57	0.66
Recall	0.21	0.43	0.56	0.62	0.36	0.55	0.72
F1-score	0.25	0.44	0.55	0.57	0.42	0.56	0.68

Table 5.5: Experimental result for sentiment classification models(after under-sampling)

5.2.4 Error Analysis

Class range	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
-1 : 0	282	6	16	16	20	17	0
-1 : 1	134	8	5	6	2	11	0
0 : -1	66	34	22	20	18	13	0
0 : 1	208	19	19	21	6	10	0
1 : 0	418	18	11	10	7	3	52
1 : -1	166	11	2	3	5	6	43

Table 5.6: Error analysis of different models

Where, Positive sentiment represented as Class 1, Negative sentiment represented as Class -1, Neutral sentiment represented as Class 0.

Chapter 6

Co-reference Detection

6.1 Task and Dataset Description

The objective of co-reference resolution is to locate all phrases in a text that refer to the same thing. It's a crucial stage in a variety of higher-level NLP activities involving natural language comprehension, such document summarising, question answering, and information extraction.

Although co-reference resolution has been used in noun-noun or noun-pronoun based resolution, we in this thesis work are trying to detect co-reference between the intent of user in a dialogue or conversation to the sentiment of the user. The main purpose of detecting intent-sentiment co-reference is to better understand how sentiment of a user depends on the topics they are discussing. Better scoring means the those topics are more sensitive to users, hence the conversation needs better monitoring. This significantly improves chat-bot AI's performance to mimic human interactions.

We divided detection of co-reference into two section. First, we will discuss how detect if utterances have any sentiment-word co-relates to intent-word. We are going to achieve that by already prepared intent-word and sentiment-word contained database. And we parse the word dependency using Stanford NLP parser(Stanza) to detect if any relation between them exists.

6.1.1 Co-reference Detection by Stanford Parser

We have used the dependency parser developed by Stanford namely "Stanza". The dependency parser takes an utterance as input and generates all the dependencies possible between each and every token. From the input phrase, the dependency parsing module creates a tree structure of words that depicts the relationships between words' syntactic dependencies.

6.1.1.1 Pipeline and Processors Used

- Language : ”en”
- Tokenize Processor : ”tokenize”
- Part of Speech Processor : ”tokenize”
- Lemma Processor : ”tokenize”
- Dependency Parser Processor : ”depparse”

Let's take an utterance :

- ”okay , thank you , for sharing the information.”

The parsed dependancy result is shown in the Table 6.1

id: 1	word: okay	head id: 3	head: thank	deprel: discourse
id: 2	word: ,	head id: 1	head: okay	deprel: punct
id: 3	word: thank	head id: 0	head: root	deprel: root
id: 4	word: you	head id: 3	head: thank	deprel: obj
id: 5	word: ,	head id: 7	head: sharing	deprel: punct
id: 6	word: for	head id: 7	head: sharing	deprel: mark
id: 7	word: sharing	head id: 3	head: thank	deprel: advcl

id: 8	word: the	head id: 9	head: information	deprel: det
id: 9	word: information	head id: 7	head: sharing	deprel: obj
id: 10	word: .	head id: 3	head: thank	deprel: punct

Table 6.1: Result of the Stanford Dependency Parser

The dataset after detecting co-reference are given in Table 6.3

6.2 Experimental Setup

Now we created an classification model based on BERT to classify if the utterances have any sentiment-intent co-reference or not thus achieving our thesis objective.

The experiment has been carried out in three stages.

6.2.1 Experiment 1:

In first experiment we annotate the dataset to have co-reference label attached to all the utterances . We assigned 1 to utterances where pairs of co-references are there and 0 where there isn't. Then the utterances and the the co-reference labels are fed into BERT.

6.2.2 Experiment 2:

In second experiment we take the dataset already prepared in the previous experiment and add the intent words and the sentiments words extracted from the dataset produced by the dependency parser. Then the concatenated utterances with the intent word and sentiment words are used as input vector and the the co-reference as labels to feed into BERT.

SPEAKER	STATEMENTS	INTENT	SENTIMENTS	intent detected	intent word	senti words	co-reference
User	hey bot i have a problem.	@greet	0	TRUE	bot,hey	hey,bot, i,have,a, problem,	bot:have, hey:have
User	hey bot,I have a legal_enquiry.	@greet	0	TRUE	hey	hey,bot, ,i,have,a, legal enquiry,	hey:have
User	My present wife is supportive and is ready to help. My only fear is that will the court interfere and take Suo motto cognizance because of this issue? What will be the safest way?	legal_ enquiry	3	TRUE	cognizance, court,fear, interfere,motto, present,ready, safest, supportive, take,wife	supportive	ready:supportive, supportive:supportive, wife:supportive
User	she threat to file dowary FIR on his family.	marriage @divorce @threat	0	TRUE	dowary,file, threat	she,threat, to,file, dowary,fir, on,his,family,	dowary:dowary, file:threat, threat:threat
User	She has two kids to support. Can she ask for alimony ?	children @maintenance	0	TRUE	alimony,ask, two	she,has, two,kids,to, support, ,can,she, ask,for, alimony,	alimony:ask, ask:ask, two:kids

Table 6.3: Co-reference Dataset

6.2.3 Experiment 3:

In third experiment we take the dataset already prepared in the previous experiment but instead of concatenating utterances with the intent and sentiment word directly we concatenate them by special character used in BERT for separation of input token '[SEP]'. Then the concatenated utterances with the intent word and sentiment words are used as input vector and the the co-reference as labels to feed into BERT. The model descriptions and hyper-parameters are given follows:

- Model Name : bert-base-uncased
- Training data size : 64%
- Validation data size : 16%
- Test data size : 20%
- Number of epochs : 20
- Learning rate : 1e-5
- Train batch size : 32
- Eval batch size : 32
- Number of output classes : 1
- Number of utterances : 4914
- Number of co-referenced utterances : 2378

6.3 Results and Observations

Metrics	**Result**
Experiment 1	
Precision	0.91
Recall	0.91
F1-score	0.92

Experiment 2	
Precision	0.93
Recall	0.93
F1-score	0.93
Experiment 3	
Precision	1.0
Recall	1.0
F1-score	1.0

Table 6.4: Experimental result of co-reference model

Chapter 7

Conclusion and Future Work

7.1 Conclusion

The goal of co-reference resolution is to find all phrases that refer to the same entity in a text. It's an important step in a number of higher-level NLP tasks requiring natural language understanding, such as document summarization, question answering, and information extraction. Despite the fact that co-reference resolution has been employed in noun-noun or noun-pronoun based resolution, we are attempting to identify co-reference between the user's purpose in a dialogue or discussion and the user's sentiment in this thesis study. The basic goal of intent-sentiment co-reference detection is to better comprehend how a user's sentiment varies depending on the subjects they're talking. Better ranking indicates that certain issues are more sensitive to users, necessitating more careful monitoring of the discourse. This boosts the ability of chatbot AI to simulate human interactions dramatically.

The detection of co-reference was split into two sections. To begin, we identified all of the utterances in which a sentiment-word is related to an intent-word. We were able to accomplish this by using a database that already contained intent-word and sentiment-word information. We then used the Stanford NLP parser (Stanza) to analyse the word dependence and discover co-reference between sentiment and intent.

Bibliography

[1] E. Bengtson and D. Roth. Emnlp'08 understanding the value of features for coreference resolution. pages 294–303, 01 2008.

[2] Q. Chen, Z. Zhuo, and W. Wang. Bert for joint intent classification and slot filling, 2019.

[3] K. Clark and C. D. Manning. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262, Austin, Texas, Nov. 2016. Association for Computational Linguistics.

[4] K. Clark and C. D. Manning. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Berlin, Germany, Aug. 2016. Association for Computational Linguistics.

[5] A. Haghighi and D. Klein. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1152–1161, Singapore, Aug. 2009. Association for Computational Linguistics.

[6] J. R. Hobbs. Resolving pronoun references. *Lingua*, 44(4):311–338, 1978.

[7] J. Jansen, D. Booth, and A. Spink. Determining the informational, navigational, and transactional intent of web queries. *Information Processing Management*, 44:1251–1266, 05 2008.

[8] I. Kang and G.-C. Kim. Query type classification for web document retrieval. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003.

[9] B. Kundu and S. Choudhury. Demystifying topology of autopilot thoughts: A computational analysis of linguistic patterns of psychological aspects in mental health. In *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, pages 435–446, Kolkata, India, Dec. 2017. NLP Association of India.

[10] K. Lee, L. He, M. Lewis, and L. Zettlemoyer. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics.

[11] U. Lee, Z. Liu, and J. Cho. Automatic identification of user goals in web search. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, page 391–400, New York, NY, USA, 2005. Association for Computing Machinery.

[12] Y. Liu, M. Zhang, L. Ru, and S. Ma. Automatic query type identification based on click through information. In H. T. Ng, M.-K. Leong, M.-Y. Kan, and D. Ji, editors, *Information Retrieval Technology*, pages 593–600, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[13] S. Louvan and B. Magnini. Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 480–496, Barcelona, Spain (Online), Dec. 2020. International Committee on Computational Linguistics.

[14] V. Ng and C. Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 104–111, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[15] L. Qin, W. Che, Y. Li, H. Wen, and T. Liu. A stack-propagation framework with token-level intent detection for spoken language understanding. 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 2078–2087, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[16] W. M. Soon, H. T. Ng, and D. C. Y. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544, 2001.

[17] J. Su, K. Duh, and X. Carreras, editors. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, Nov. 2016. Association for Computational Linguistics.

[18] N. T. Vu, P. Gupta, H. Adel, and H. Schütze. Bi-directional recurrent neural network with ranking loss for spoken language understanding. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, page 6060–6064. IEEE Press, 2016.

[19] S. Wiseman, A. M. Rush, and S. M. Shieber. Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 994–1004, San Diego, California, June 2016. Association for Computational Linguistics.

[20] P. Xu and R. Sarikaya. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 78–83, 2013.

[21] K. Yao, B. Peng, Y. Zhang, D. Yu, G. Zweig, and Y. Shi. Spoken language understanding using long short-term memory neural networks. *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 189–194, 2014.