LEVERAGING DEPENDENCY STRUCTURE FOR INFERENCE COMPUTATION, SUMMARIZATION, AND COMPREHENSION

Thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering by Research

by

ELIZABETH JASMI GEORGE
20163036
elizabeth.george@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2020
It is certified that the work contained in this thesis, titled “LEVERAGING DEPENDENCY STRUCTURE FOR INFERENCE COMPUTATION, SUMMARIZATION, AND COMPREHENSION” by ELIZABETH JASMI GEORGE, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. RADHIKA MAMIDI
Dedicated with love to Abraham, Alice, and Albert
Acknowledgments

This research is an outcome of passion and excitement in learning Computer Science and Engineering again after fourteen years, which was made possible by this fantastic research university. I want to thank my institute, International Institute of Information Technology, for giving me an opportunity to interact with intelligent classmates and listen to insights of knowledgeable professors.

I am incredibly grateful to my family, who understood my need for learning, helped me through the ups and lows, and offered practical insights. I want to thank my husband, Binu Idicula, for supporting and motivating me and my children Abraham Idicula, Alice Idicula, and Albert Idicula, for patiently compromising on their story-times. I want to thank my parents, uncles, and aunts for their encouragement. I thankfully acknowledge my father in law, late K. S. Idicula for his prayers and wishes.

I want to thank my company Advainet Solutions Private Limited, for being the motivation for researching in NLP.

I am indebted forever to my supervisor Dr. Radhika Mamidi for accepting, understanding, guiding, motivating, and directing me through my research journey. Her smiling face and comments of acknowledgment made me feel at ease and encouraged me to complete the tasks on time. From the day Radhika madam wrote the textbooks to refer to for understanding linguistics on the backside of my notebook, in my first Computational Linguistics-II class in 2017, she was always a gentle, humane mentor. She always answered my questions patiently, addressed my concerns effectively, and expressed her delight on my achievements. She granted me the freedom to explore my topic, persuading on converting the research to published work, and gently correcting and insisting on reading more research papers. I want to thank Radhika madam, for all the good things she brought to my life.

I want to thank Dr. Monojit Choudhury of Microsoft Research, Bangalore, for suggesting this exciting topic of research as part of the Computational Socio-pragmatics course he taught at IIIT-H in 2018. I would also like to thank all the anonymous reviewers of RANLP Stud 2019, PACLING 2019, Co-CoNet 2019, IntelliSys 2020, AIAP 2020, and TSD 2020 for carefully reading through the documents I submitted for review and for offering valuable suggestions for improvement.

I want to thank Mr. Santosh Raju Vysyaraju, my husband’s colleague at Amazon and an IIIT-H alumnus, for patiently addressing my research hurdles and pointing to resources. He helped me regularly to set realistic expectations on doing research and happily shared his memories of IIIT-H.
I want to thank all of my project mates and classmates Sai Nallapati, Ashutosh Mishra, Shyam Nandan Rai, Srikar Mantravadi, Aditya Srivastava and Venkata Ravindra Nittal, for their companionship in my research studies.

Most importantly, I want to thank my teachers Dr. Dipti Misra Sharma, Dr. Radhika Mamidi, Dr. Soma Paul, Dr. Manish Shrivastava, Dr. Naresh Manwani, and Dr. Monojit Choudhury, facilitators, and the lab mates of LTRC, for the guidance and help offered to me.

I want to thank the examiners of my thesis Dr. Rajeev Sangal and Dr. Kavita Vemuri, for their insightful comments and suggestions.
Abstract

In this research work, we present three syntax-based computational methods that use the dependency structure of sentences for inference generation, summarization, and machine reading comprehension. Besides, we present an introductory dataset to aid the production of inferences from a response in a context.

The methods described in this thesis utilize the dependency structure obtained from the parser that represents the dependency relations existing between the words of the sentence. This work explains how these relations are utilized to compute inferences from utterances, summarize documents, and answer questions based on context passages.

The first method describes computing pragmatic inferences from news headlines that are stand-alone units of text carrying no context information. The different types of pragmatic inferences are entailment, presupposition, and implicature. News headlines are some minimal utterances that contain maximum information for which the speaker is the news editor, and the receiver is the newsreader. The inferences, such as presuppositions and conventional implicatures that are independent of the context, depend mainly on syntax. Hence, this approach uses dependency trees of the headlines to find the syntactic structure and to compute inferences out of them. The generated inferences about the news headlines could be useful for assessing the impact of the same on readers, including children.

Utterances do not frequently exist in isolation, like in news headlines. Hence, we created a minimal dataset with implicated meanings of utterances in a context. In this work, 1k context-response utterance dialogue pairs having implicatures for the response utterance are collected and annotated with their implicated meanings. Since dialogues with implicatures are not easy to obtain, the data collection methods, challenges, findings, and sources of dialogue data of this genre are also explained. This dataset and further additions to it would improve the understanding capability of the machine and reduce the amount of conversational failures in human-computer interactions.

The second method describes summarizing a passage based on syntax and dependency structure. The extractive summary of the passage is obtained by finding overlapping sentences from the dependency graph constructed by attaching the dependency trees of individual sentences in the passage and simultaneously prioritizing content word nodes such as verbs, nouns, adjectives, and adverbs, and discarding translatives nodes such as prepositions and articles.

The third method uses dependency trees to simulate a human approach while answering a question about a passage. While reading a passage to answer a question, humans usually try to identify the verb
of action in the question and find the sentences with the same or synonymous verbs as the candidate answers. Simulating the human approach of finding the same or similar verb in the passage comparing to the verb in the question is followed for comprehending the passage to answer a question. The verb nodes of the dependency trees are compared to achieve this.
## Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Aim and Objectives</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Background and Definitions</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Our Approach</td>
<td>3</td>
</tr>
<tr>
<td>1.4 Contributions</td>
<td>4</td>
</tr>
<tr>
<td>1.4.1 Main Contributions</td>
<td>4</td>
</tr>
<tr>
<td>1.4.2 Other Contributions</td>
<td>5</td>
</tr>
<tr>
<td>1.5 Organization of this Thesis</td>
<td>5</td>
</tr>
<tr>
<td>2 Related Work</td>
<td>6</td>
</tr>
<tr>
<td>2.1 Related Inference Computation Works</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Related Summarization Works</td>
<td>7</td>
</tr>
<tr>
<td>2.3 Related MRC Works</td>
<td>8</td>
</tr>
<tr>
<td>2.4 Literature Review</td>
<td>8</td>
</tr>
<tr>
<td>2.4.1 The Literature on Inferences</td>
<td>8</td>
</tr>
<tr>
<td>2.4.2 The Literature on Dependency Grammar</td>
<td>8</td>
</tr>
<tr>
<td>3 Towards Computing Inferences from English Utterances</td>
<td>10</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>10</td>
</tr>
<tr>
<td>3.1.1 Presupposition and Conventional Implicature</td>
<td>11</td>
</tr>
<tr>
<td>3.1.2 Linguistic Definitions and Characteristics of Headlines</td>
<td>11</td>
</tr>
<tr>
<td>3.1.3 Importance of Headline</td>
<td>12</td>
</tr>
<tr>
<td>3.1.3.1 Pragmatic Function of a Headline</td>
<td>13</td>
</tr>
<tr>
<td>3.1.4 Relevance of this Work</td>
<td>13</td>
</tr>
<tr>
<td>3.2 Data</td>
<td>14</td>
</tr>
<tr>
<td>3.2.1 Format of the Data</td>
<td>14</td>
</tr>
<tr>
<td>3.2.2 Collecting the Data</td>
<td>15</td>
</tr>
<tr>
<td>3.3 Proposed Inference Generation Method</td>
<td>15</td>
</tr>
<tr>
<td>3.3.1 Extracting the Dependencies</td>
<td>16</td>
</tr>
<tr>
<td>3.3.2 Rule-Based System for Inference Generation</td>
<td>18</td>
</tr>
<tr>
<td>3.3.2.1 Presence of a Future Tense Verb</td>
<td>18</td>
</tr>
<tr>
<td>3.3.2.2 Presence of the Conjunction ‘but’</td>
<td>19</td>
</tr>
<tr>
<td>3.3.2.3 Presence of ‘again’ in a Clause with a Verb</td>
<td>20</td>
</tr>
<tr>
<td>3.3.2.4 Presence of ‘further’ as an Adjective</td>
<td>21</td>
</tr>
<tr>
<td>3.3.2.5 Presence of a ‘noun compound’</td>
<td>21</td>
</tr>
</tbody>
</table>
### 3.3.2.6 Presence of a ‘verb’ in the Past Tense

21

### 3.3.2.7 Presence of Nominal Modifier ‘of’

22

### 3.4 Results and Discussion

22

### 3.5 Annotation Guidelines

24

#### 3.5.1 Purpose of Annotation

24

#### 3.5.2 Guidelines for Annotating Presuppositions

24

##### 3.5.2.1 Definite Descriptions

24

##### 3.5.2.2 Factive Verbs

25

##### 3.5.2.3 Implicative Verbs

25

##### 3.5.2.4 Change of State Verbs

25

##### 3.5.2.5 Iterative

25

##### 3.5.2.6 Verbs of Judging

25

##### 3.5.2.7 Temporal Clauses

26

##### 3.5.2.8 Cleft Sentences

26

##### 3.5.2.9 Implicit Clefts with Stressed Constituents

26

##### 3.5.2.10 Comparisons and Contrasts

26

##### 3.5.2.11 Non-restrictive Relative Clauses

26

##### 3.5.2.12 Counterfactual Conditionals

26

##### 3.5.2.13 Questions

26

##### 3.5.2.14 More than Two Words in Quotes

27

##### 3.5.2.15 Future Tense Verb

27

##### 3.5.2.16 The Conjunction ‘but’

27

##### 3.5.2.17 Gender-Specific Statements

27

### 3.6 Conclusion

27

### 4 Conversational Implicatures in English Dialogue: Data Collection and Annotation

31

#### 4.1 Introduction

31

##### 4.1.1 Generating Conversational Implicatures

32

#### 4.2 Approaches Attempted for Creating Implicature Corpus

33

##### 4.2.1 Related Annotation Work

33

##### 4.2.2 Crowdsourcing the Implicature Generation Task

34

##### 4.2.3 Challenges in Crowdsourcing the Implicature Generation Task

35

#### 4.3 Implicature Corpus

36

##### 4.3.1 Collecting the Dialogue Snippets

37

##### 4.3.2 Evaluating the Annotation Task

38

#### 4.4 Conclusion

40

### 5 Using Dependency Trees for Summarization and Question Answering

43

#### 5.1 Dependency Graphs for Summarization

43

##### 5.1.1 Proposed Method for Extractive Summarization

44

##### 5.1.2 Constructing the Dependency Graph for a Passage

44

##### 5.1.3 Obtaining the Summary from the Dependency Graph

46

##### 5.1.4 Comparison with other Graph-based Summarization Methods

48

#### 5.2 Syntax-Based Machine Reading Comprehension

48

##### 5.2.1 Proposed Comprehension Method

49

##### 5.2.2 Passage and Question Data for Illustration

51
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2.3</td>
<td>Algorithm for Answering ‘Who’ Questions</td>
<td>51</td>
</tr>
<tr>
<td>5.2.3.1</td>
<td>Identify Question Word, Verbs and Nouns</td>
<td>51</td>
</tr>
<tr>
<td>5.2.3.2</td>
<td>Preprocessing the Question and the Passage</td>
<td>51</td>
</tr>
<tr>
<td>5.2.3.3</td>
<td>Find the Candidate Sentences by Matching the Verbs</td>
<td>52</td>
</tr>
<tr>
<td>5.2.3.4</td>
<td>Find the Candidate Sentences if Synonymous Verbs Match</td>
<td>52</td>
</tr>
<tr>
<td>5.2.3.5</td>
<td>Select Sentences with the Highest Cosine Similarity and Longest Matching Span</td>
<td>52</td>
</tr>
<tr>
<td>5.2.3.6</td>
<td>Find the Candidate Sentences with Matching Predicates</td>
<td>53</td>
</tr>
<tr>
<td>5.2.3.7</td>
<td>Answering ‘Who’ from the Answer Sentence</td>
<td>53</td>
</tr>
<tr>
<td>5.2.3.8</td>
<td>Answering ‘How’ and ‘What’ from the Answer Sentence</td>
<td>53</td>
</tr>
<tr>
<td>5.2.4</td>
<td>Results and Evaluation</td>
<td>53</td>
</tr>
<tr>
<td>5.2.4.1</td>
<td>Comparison with the other Syntax-based Systems</td>
<td>54</td>
</tr>
<tr>
<td>5.3</td>
<td>Conclusion</td>
<td>55</td>
</tr>
<tr>
<td>6</td>
<td>Conclusion and Future Work</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>62</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Hypothetical structure of a news schema [106] 13</td>
</tr>
<tr>
<td>3.2</td>
<td>The amount of information represented by headline and lead paragraph for a news story 14</td>
</tr>
<tr>
<td>3.3</td>
<td>Headline and associated data from BBC 15</td>
</tr>
<tr>
<td>3.4</td>
<td>Dependency tree for headline “Russian state television will not broadcast Olympics without national team” 18</td>
</tr>
<tr>
<td>3.5</td>
<td>Dependency tree for headline “Flybe to close Isle of Man base in 2020 but flights will continue” 19</td>
</tr>
<tr>
<td>3.6</td>
<td>Dependency tree for headline “Norway regulator again rejects “Donut” fish farm volume plan” 20</td>
</tr>
<tr>
<td>3.7</td>
<td>Dependency tree for headline “Catriona Matthew would captain Europe again in Solheim Cup” 20</td>
</tr>
<tr>
<td>3.8</td>
<td>Dependency tree for headline “UK economy to slow further” 21</td>
</tr>
<tr>
<td>3.9</td>
<td>Dependency tree for headline “Russia’s Olympic ban strengthens Putin’s reelection hand” 21</td>
</tr>
<tr>
<td>3.10</td>
<td>Dependency tree for headline “How women won the right to vote in 1918” 22</td>
</tr>
<tr>
<td>3.11</td>
<td>Dependency tree for headline “Governor of New Jersey meets PM Narendra Modi” 22</td>
</tr>
<tr>
<td>3.12</td>
<td>Relevant computations, total computations and appropriate expectations of inferences for different news genres 24</td>
</tr>
<tr>
<td>4.1</td>
<td>MicroWorkers’ customized TTV questions and answers template for conversation generation 36</td>
</tr>
<tr>
<td>4.2</td>
<td>MTurk good and bad examples for response generation in context 37</td>
</tr>
<tr>
<td>4.3</td>
<td>Implicature context from (a) movie Script of ‘Anastasia’; (b) TOEFL transcript [14] 38</td>
</tr>
<tr>
<td>4.4</td>
<td>Response annotation sheet 41</td>
</tr>
<tr>
<td>4.5</td>
<td>Polar response annotations by two annotators to non-polar responses 42</td>
</tr>
<tr>
<td>5.2</td>
<td>Dependency trees for the news article 46</td>
</tr>
<tr>
<td>5.3</td>
<td>Dependency graph for the news article 47</td>
</tr>
<tr>
<td>5.4</td>
<td>Strongly connected cluster in the dependency graph for the news article given in Figure 5.1 47</td>
</tr>
<tr>
<td>5.5</td>
<td>Modules in our syntax-based MRC 50</td>
</tr>
<tr>
<td>5.6</td>
<td>Dependency parsed question 51</td>
</tr>
<tr>
<td>5.7</td>
<td>Dependency parsed matching sentence from the passage 52</td>
</tr>
<tr>
<td>5.8</td>
<td>Answering a question from a passage excerpt 55</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Features of headline language with examples</td>
<td>12</td>
</tr>
<tr>
<td>3.2 Accuracy and generated percentage of inferences computed</td>
<td>23</td>
</tr>
<tr>
<td>3.3 Comparison of manually annotated inferences with computed inferences for a headline</td>
<td>29</td>
</tr>
<tr>
<td>3.4 Agreement options and their corresponding point values</td>
<td>30</td>
</tr>
<tr>
<td>4.1 Types of implicatures</td>
<td>33</td>
</tr>
<tr>
<td>4.2 Examples of situations provided for conversation generation task with ideal responses</td>
<td>35</td>
</tr>
<tr>
<td>4.3 Utterances and contexts collected from TOEFL listening comprehension section with their implicated meanings</td>
<td>39</td>
</tr>
<tr>
<td>4.4 Utterances and context collected from IMSDb movie dialogues with their implicated meanings</td>
<td>39</td>
</tr>
<tr>
<td>4.5 Interpretation of the agreement when Kappa statistic $\kappa$ varies from 0 to 1</td>
<td>40</td>
</tr>
<tr>
<td>5.1 Headline, noun nodes and verb nodes with their degrees</td>
<td>48</td>
</tr>
<tr>
<td>5.2 Evaluation results of our machine reader</td>
<td>54</td>
</tr>
<tr>
<td>5.3 Computed answer span from passages, questions and computed answers</td>
<td>57</td>
</tr>
<tr>
<td>5.4 Comparison of answers from our reader with those from other deployed readers for the same context passage</td>
<td>58</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Conversational AI is an exciting application of Artificial Intelligence (AI), which is advancing at a high pace and explored by research in many dimensions. We human beings do not hesitate to start a conversation with minimal assumptions about one another and use these assumptions in a conversation as a basis from which to draw specific inferences about one another’s intended meanings. But for a machine, making inferences on more than what is said is not possible unless its intent vocabulary contains those inferences. It is disappointing when a user eagerly asks a voice assistant for something, and it gives a bland reply, “Sorry, I do not understand that.” So we found this dimension of research on collecting situations where utterances mean more than the words in the uttered sentence as a very interesting direction of research and a need of the time. This work, when done in scale can reduce conversational failures in human-computer interactions. The work on generating inferences from headlines as well as the comprehension and summarization of news articles were done as a part of building a system for children to learn about current affairs. Considering the headline as the entry point to a piece of news and as a source rich in information about the news, we explored a headline’s potential of giving an idea about the current state of affairs, leveraging the syntax structure. Further exploration of the possibilities of using the dependency structure to simulate the human approach of finding the action in the passage sentence similar to the action mentioned in the question is done to achieve reading comprehension. A collective representation of the document is achieved by connecting the nodes in the dependency tree representation of sentences in the passage, and the summary is obtained from the sentence overlaps of nodes with the maximum number of out-edges.

1.1 Aim and Objectives

This work aims to explore the possibilities of using the dependency structure for computing inferences, obtaining summaries, and comprehending passages. We also aim to conduct a pragmatic analysis of utterances in stand-alone units such as news headlines and utterances in a context such as dialogues using computational techniques. Along with the three computational goals mentioned above, this work
aims to identify the pragmatic phenomenon of implicatures in conversations by collecting and annotating a dataset of conversations with implicated meanings.

We target the following outcomes from this research: (i) Pragmatic analysis of English news headlines using syntax-based computational techniques. (ii) Formulation of guidelines for annotating news headlines with the presuppositions they generate. (iii) Survey of the different kinds of conversational implicatures and a study of the scenarios in which they are created. (iv) Collection and annotation of a preliminary data set of dialogues with their corresponding conversational implicatures. (v) Summarization of documents leveraging the dependency structure. (vi) Machine reading comprehension of a passage for free form natural language questions using the dependency structure of the question and passage sentences.

1.2 Background and Definitions

Conversation or dialogue is the first kind of language we learn as children [58]. It is the kind of language use we commonly see in our daily life. Utterances are the smallest units of speech, which can also be represented in written language. Interlocutors in a conversation often express meanings in their utterances beyond the literal meaning of the sentences used. The recipients of the utterances are expected to make appropriate inferences in order to determine the intended meaning [81]. Interpretations of an utterance can consist of pragmatic inferences such as presuppositions, entailments, conventional implicatures, and conversational implicatures. In this research, we worked on computing the meanings of the utterances more than what is explicitly said and on producing precise summaries of long narratives and on simulating comprehension of documents given as context for answering questions.

Conversation is characterized by turn-taking [95]. The contributions made by their turns include a presentation phase of performing a kind of action and an acceptance phase of grounding the previous actions of the interlocutor [58]. Implicature is an important aspect of conversational competence because people often use indirect language for various purposes [81]. Grice [49] proposed that what enables hearers to draw these inferences is that conversation is guided by a set of maxims listed below, which are the general heuristics that play a guiding role in the interpretation of conversational utterances.

1. Maxim of quantity – Give neither more nor less information than is required.
2. Maxim of quality – Give information that is genuine and not spurious.
3. Maxim of relation – Give information that is appropriate to immediate needs.
4. Maxim of manner – Give information in the clearest, briefest, and most orderly manner.

Flouting of maxims is a situation wherein a speaker manipulates a particular maxim that results in a conversational implicature. Consider the scenario [56], “where Susan and Mary are talking about Mary’s boyfriend Peter.

(1) SUSAN: Is he good at buying you presents?
MARY: For my last birthday he bought me a pink scarf, even though I told him that I hate pink.

Mary clearly intends to communicate that there is some sort of contrast or incompatibility between Peter buying her a pink scarf and her telling him that ‘she hates pink’. Furthermore, in the scenario above Susan will have every justification to assume that Mary also means that Peter is not good at buying her presents.” These assumptions made by the hearer above from what is explicitly said are called inferences. Levinson [72] defines four kinds of inferences as follows.

1. **Entailment:**
   
   \[ A \text{ semantically entails } B \iff \text{every situation that makes } A \text{ true, makes } B \text{ true.} \]

2. **Presupposition:**
   
   An utterance \( A \) pragmatically presupposes a proposition \( B \) iff \( A \) is *appropriate* only if \( B \) is *mutually known* by the participants.

3. **Conventional Implicature:**
   
   Conventional implicatures are non-truth-conditional-inferences that are not derived from superordinate pragmatic principles like maxims, but are simply attached by convention to particular *lexical items* or *expressions*.

4. **Conversational Implicature:**
   
   A speaker \( S \)’s saying that \( p \) conversationally implicates \( q \) iff: (i) \( S \) is presumed to be observing the maxims (ii) in order to maintain this assumption it must be supposed that \( S \) thinks that \( q \) (iii) \( S \) thinks that both \( S \) and the addressee \( H \) mutually know that \( H \) can work out that to preserve the assumption in (i) \( q \) is in fact required.

Our work on inferences generation focuses mainly on presuppositions and conversational implicatures.

### 1.3 Our Approach

We use the dependency structure to compute inferences, summaries, and answer spans from documents. Dependency tree representation is considered as an optimal balance between representation and complexity. The tuples generated by the Stanford Parser [80] are easy to process and mostly accurate for achieving summarization and comprehension.

The common core of all varieties of dependency grammar is the assumption that syntactic structure consists primarily of binary asymmetrical relations, called dependency relations, that hold between words. The simplicity and hence practicality of dependency is increasingly acknowledged in the field of computational linguistics. Lucien Tesnière [102] elaborated a complete linguistic theory based on the dependency concept. Tesnière’s research is usually taken as the starting point of the modern theoretical tradition of dependency grammar. Around 2,500 years ago, Pāṇini, a revered scholar in ancient India
wrote Astādhyāyī [60] the oldest linguistic and grammar text of any language. Pāṇinian theory was formulated for Sanskrit. In the Pāṇinian framework [19], a sentence is analyzed in terms of dependency relations, more specifically, modifier-modified relations.

Summarization of a document is useful for feeding as input to many natural language processing tasks. The initial step in summarization is content selection, where the prominent words in the document are identified. A graph representation of the document is required to obtain the connections between the words in the document. The graph is a popular data structure used efficiently in many real-world applications. The dependency tuples obtained from the Stanford Parser [80] represent two nodes and an edge existing between them, labeled with a dependency relation. We found the representation to be suitable for constructing a document graph.

Teaching machines to read and comprehend is a popular research topic in this era of surplus information. Computing the answer span for a question asked on a passage is the essential task of machine reading comprehension. Comprehension is useful in educational software as well as question answering systems. Ideal software for this task should take any passage irrespective of its genre and answer the natural, free form questions asked about the content in the passage. We use the dependency structure of the sentences in the passage and the query to achieve comprehension.

Summarization and comprehension can be used in the conversation context, which is discussed at the beginning of this chapter, for answering with the summary of some specific passage requested and for answering questions based on a given passage. These are helpful for educational purposes in online tutoring systems.

1.4 Contributions

We have formulated a computational approach for generating inferences from the utterance, based on the syntax of the utterance. This is done by iterating through the dependency tuples obtained from the Stanford dependency parser [80]. We have introduced an extractive summarization method achieved by attaching the dependency trees of the sentences in the document. We have also attained fair answers from passages, for free form natural language questions using a syntax-based machine reader focusing on the verbs in the questions and context passage.

1.4.1 Main Contributions

- Ours is the first research attempt to compute inferences from English news headlines.

- This research addresses a novel and challenging data collection task of finding dialogues having implicit meanings. We collected a preliminary implicature dataset from listening comprehension sections of TOEFL [6] and dialogue snippets from movie scripts available at IMSDb [8].
• This research presents a novel method of connecting the dependency trees of sentences to construct a dependency graph for the passage and identify the maximally overlapping sentences containing the most number of well-connected content nodes.

• This research explores the possibility of using dependency structures of sentences to simulate the human approach of comprehending a passage by identifying the sentences with a similar verb as the verb of action in the question.

1.4.2 Other Contributions

This is a novel inference computation method of iterating through the dependency tuples obtained from Stanford parser to find the interrelation between words. Although, dependency tree representations are considered as the right balance of complexity and expressivity for use in NLP related tasks [36] and used for solving many NLP related tasks, this work is novel in iterating through the tuples to identify how the words are connected to generate an inference. Annotation guidelines for annotating an utterance with their presuppositions have been outlined. These guidelines are explained with customized examples for English news headlines based on the presupposition triggers enlisted by Levinson [72].

This research focuses on the inference generation task, which is gaining popularity due to the increased use of virtual assistants and the expectations on the same to understand the implicit meanings of what a user says. We studied and tabulated the different kinds of implicatures and a plethora of contexts generating them. This study is done to explore the inferences domain further than the headlines, by taking into consideration the conversational implicatures, which is totally independent of the syntax and very commonly used in the day-to-day conversations. The proposed question answering method obtains the answer span from a passage for a free form natural language question without using a trained model. It produces accurate answer spans for multiple genres of text.

1.5 Organization of this Thesis

This thesis is organized in five chapters. Chapter 1 is an introduction to the work conducted by this research. Chapter 2 explains the previous works related to our work. Chapter 3 elaborates on a computational approach to find inferences from English utterances explained with the examples of stand-alone utterances of English news headlines. It also presents annotation guidelines for annotating presuppositions based on the triggers listed by Levinson [72]. Chapter 4 surveys the domain of conversational implicature in the context of polar questions in particular. The chapter also presents and explains a dataset of implicatures and outlines the challenges, sources, and methods of collecting and annotating it. Chapter 5 explores the possibilities of using the dependency structure for document summarization and machine reading comprehension. The summary and conclusion of this research work are presented in chapter 6.
Chapter 2

Related Work

2.1 Related Inference Computation Works

Cianflone et al. [31] have introduced the new task of predicting adverbial presupposition triggers, and our research explores the scope of computing presupposition statements from the syntax structure provided by dependency trees of news headlines. The approach used in their paper uses deep learning, while we demonstrate a rule-based approach.

Most of the work about finding inferences focuses on entailments. Obtaining entailments is helpful in many NLP tasks such as question answering, semantic search, machine reading comprehension, and summarization. Entailments are inferences that can be generated from the knowledge about the semantic relationships in a language. An entailment is something that logically follows from what is asserted in the utterance [118]. In our research, we neither compute entailments nor predict relations of entailment, contradiction or neutrality exiting between two fragments of text, known as the premise and the hypothesis. We compute the inferences such as conventional implicatures and presuppositions that are solely based on the syntax.

The primary inference corpora available are the Stanford Natural Language Inference (SNLI) Corpus [25] containing around 550k hypothesis/premise pairs, Multi-Genre Natural Language Inference (MultiNLI) corpus [111] containing around 433k hypothesis-premise pairs and the SciTail entailment dataset [63] consisting of 27k pairs. The SNLI and MultiNLI are crowdsourced, while the SciTail dataset is created from sentences from the ‘wild’ using multiple-choice science exams and web sentences. In SciTail, each question and the correct answer choice are converted into an assertive statement to form the hypothesis. e-SNLI [27] is an extension of SNLI with explanations for each inference made and SWAG [119] is a dataset with 113k multiple-choice questions about a rich spectrum of grounded situations where they focus on whether a (multiple-choice) ending describes a possible (future) world that can be anticipated from the situation described in the premise.

The Recognizing Textual Entailment (RTE) challenge, which ran from the year 2006 onwards for eight years, presented a task of finding entailments after training on a dataset. The RTE task [32] dataset consisted of text(t)-hypothesis(h) pairs with the task of judging for each pair, whether t entails h. The
RTE task is defined as recognizing, given two text fragments, whether the meaning of one text can be inferred (entailed) from the other. This application-independent task is suggested as capturing significant inferences about the variability of semantic expression, which are commonly needed across multiple applications. The Stanford RTE system, which uses typed dependency trees as a proxy for semantic structure, seeks a low-cost alignment between trees for $p$ and $h$, using a cost model that incorporates both lexical and structural matching costs. This system is typical of a category of approaches to NLI based on approximate graph matching [78]. In our work, we attempt to generate hypotheses for news headlines rather than judging whether a hypothesis is correct or not. Burger and Ferro [26] attempted to generate a large corpus of textual entailment pairs from the lead paragraph and headline of a news article. This is a prominent work in news genre for dataset creation of entailments. To the best of our knowledge, ours is the first work towards computing presuppositions and conventional implicatures from English news headlines. We did not create a dataset for this task but instead evaluate the approach on headlines taken randomly from popular news websites.

Research on discovering event entailment knowledge [85] propose a method for automatic discovery of pairs of verbs related by entailment, such as $X \text{ buy } Y \Rightarrow X \text{ own } Y$ and $\text{ appoint } X \text{ as } Y \Rightarrow X \text{ becomes } Y$. Learning verb inference rules by Hila Weisman et al. [109] introduce linguistically motivated indicators that are specific to verbs and may signal the semantic relation between verb pairs and present a supervised classification model for detecting lexical entailment between verbs.

Potts [88] introduced an experimental dataset involving 215 indirect question-answer pairs collected from four sources and annotated with polarity using Amazon Mechanical Turk [1]. The data is obtained from CNN show transcripts, data from Hirschberg [52], Switchboard Dialog Act Corpus [48], and also those derived from highly restrictive regex searches over the Yahoo Answers corpus.

Lahiri [68] annotated a corpus of 7,032 sentences using Amazon Mechanical Turk with ratings of formality, informativeness, and implicature for each sentence on a 1-7 scale during which the annotators were asked to form implicatures for a given sentence.

Lasecki, Kamar, and Bohus [70] collected conversations focused around definable tasks using crowdsourcing methods. In their work, two annotators were assigned part of an agent and a user to a randomly given a task, and they were asked to engage in a conversation to complete the task. Reddy, Chen, and Manning [93] introduced CoQA, a 127k question-answer dataset for building conversational question answering systems. In the above research, pairs of annotators were given passage for reading and were asked to frame questions based on the passage and answer them from the passage consecutively. Another crowd-powered system utilizing asynchronous chat for efficient collection of dialogue dataset was designed by Ikeda and Hoashi [55].

### 2.2 Related Summarization Works

Recent summarization works [86, 42], are advancing around concept maps that are labeled graphs depicting concepts and their relations. Semantic similarity is considered [116] for achieving the best
results. The state of the art summaries are obtained using neural networks [76, 14]. We propose an easily obtainable graph representation of documents and the summarization achieved from that graph. There are other graph-based summarization methods such as [92, 45, 75] and [114].

2.3 Related MRC Works

The different datasets [30, 64, 93], their comparison [115], and approaches for MRC are described by [120]. The methods and trends for MRC are explained by [74], and an investigation on the popular benchmarks in MRC is done by [62]. There are popular non-neural methods used in MRC such as bag of words [53], SlidingWindow, Logistic regression, TF-IDF and Boosted method, Integrated Triaging [113] and the neural-based methods such as mLSTM+Ptr, DCN, GA [38], BiDAF, FastQA [110], RNET [108], ReasoNet [98] and QAnet [117] and those incorporating reading strategies [101] or discourse relations [82]. Feature matching for comprehending CNN/Daily mail passage to identify the place holder token using an entity-centric classifier is demonstrated in recent work on crowdsourced conversational data of 127k dialogues [93].

There are syntax-based question answering systems such as [107, 97, 90, 73] and [57]. The method we propose supports free form questions and can be applied successfully to multiple datasets and does not require any training with a specific dataset.

2.4 Literature Review

2.4.1 The Literature on Inferences

Inferences like presupposition, entailment, conventional implicature and conversational implicature are discussed in detail by Levinson [72] and implicatures are explained further by Potts [88, 89] reviewing the basic Gricean theory of conversational implicature [49], important consequences, known problems, and useful extensions and modifications. Benotti and Blackburn [18] view conversational implicature as a way of negotiating the meaning in conversational contexts and conveys that context and conversational implicature are highly intertwined. A series of papers by Bouton [22, 23, 24] explores the paradigm of implicatures in pragmatics.

2.4.2 The Literature on Dependency Grammar

Seminal work by Tesniere [102] written in French in 1959 and translated to English in 2015 contains a comprehensive approach to the syntax of natural languages called dependency grammar. Different dependency parsers are explained by Kubler et al. [66]. Annual Review of Linguistics [36] states that the use of dependency structures in NLP has consisted mainly of borrowing a formal representation that
strikes the right balance between expressivity and complexity. The common core of all varieties of de-
pendency grammar is the assumption that syntactic structure consists primarily of binary asymmetrical
relations, called dependency relations that hold between words. Our research represented by this thesis
uses dependency trees generated for English sentences from phrase structure parses by de Marneffe et
al. [34]. The dependency tree representation of the syntactic structure emphasizes the functional role of
a word in a sentence [36].
Chapter 3

Towards Computing Inferences from English Utterances

3.1 Introduction

An utterance is the smallest unit of speech, and written language representations of them can exist at the cost of losing their prosodic features. Levinson [72] states the order of addition of inferences to the context of an utterance as (i) the entailments of the uttered sentence S (ii) the clausal conversational implicatures of S (iii) the scalar conversational implicatures of S (iv) the presuppositions of S. This chapter explains an approach for the computation of inferences from newspaper headlines that are self-contained utterances in which the news editor is the speaker and the newsreader is the listener.

Newspapers are a popular form of written discourse, read by many people, thanks to the novelty of the information provided by the news content in it. A headline is the most widely read part of any newspaper due to its appearance in a bigger font and sometimes in colour print. In this chapter, we suggest and implement a method for computing inferences from English news headlines, excluding the information from the context in which the headlines appear. This method attempts to generate the possible assumptions a reader formulates in mind upon reading a new headline. The generated inferences could be useful for assessing the impact of the news headline on readers, including children. The understandability of the current state of social affairs depends significantly on the assimilation of the headlines. As the inferences such as presuppositions and conventional implicatures that are independent of the context depend mainly on the syntax of the headline, dependency trees of headlines are used in this approach, to find the syntactic structure of the headlines and to compute inferences out of them.

The headline of a news report appears at the top of the news report and is often printed in a bigger font and some times in a bright colour. The marketability of a news story depends to a great extent on the ability of the headline to attract readers. A headline generally tries to summarise the content of the news story, with a strong intention of communicating the context to the reader. Headlines also try to attract the attention of the newsreaders, prompting them to read on through the news story. A speech act is an utterance that performs an action when it is uttered. For example, an utterance “I invite you to my birthday party” is actually performing the action of invitation. Headline functions as a number of speech acts [13]. It urges, warns, and informs the reader [54]. This work views headline as a potential
source of rich information capable of generating multiple inferences relevant to the current social state, making it worthy of adding to the general knowledge.

This work was done as a part of building a system for children to learn about current affairs in a simpler way. In this work, we consider headline as a standalone unit of discourse, without any context or supporting background information and compute the inferences that arise from the headline alone. Our experiment attempts to compute inferences based on syntactic triggers. This work focuses on inferences, in particular presuppositions and conventional implicatures which are independent of context and omit conversational implicature which requires context information to formulate. The number of triggers used in this experiment is limited and the results include negatives in some cases.

### 3.1.1 Presupposition and Conventional Implicature

According to Levinson [72] presupposition is used to describe any kind of background assumption against which an action, theory, expression or utterance makes sense or is rational and conventional implicatures are non-truth-conditional inferences that are not derived from superordinate pragmatic principles like the maxims but are simply attached by convention to particular lexical items or expressions.

According to Fromkin et al. [44], presuppositions are implicit assumptions about the world, required to make an utterance meaningful or appropriate. Conventional implicatures are not based on cooperative principles or maxims. They don’t have to occur in a conversation, and they do not require special contexts for their interpretation. Not unlike lexical presuppositions, conventional implicatures are associated with specific words and result in additional conveyed meanings when those words are used, according to Yule [118]. Karttunen [59] views presuppositions as a special case of conventional implicatures. Presuppositions are denoted by ‘>’ and conventional implicatures are denoted by ‘≈’. For an utterance, “The king of France is wise” there can be a presupposition that > There is a present king of France. For an utterance, “Amelia is a toddler, but she is quiet” there can be a conventional implicature that ≈ Toddlers are not usually quiet [89].

As an example, upon reading a headline, ‘Schaeuble says British were ‘deceived’ in Brexit campaign’, a reader may make the following inferences. (i) Schaeuble exists. (ii) Schaeuble said something. (iii) Schaeuble believes that the British were ‘deceived’ in the Brexit campaign. (iv) The Brexit campaign happened. (v) Brexit can have a campaign (vi) The British government was deceived in the Brexit campaign (vii) The British citizens were deceived in the Brexit campaign. The inferences (vi) and (vii), which are conversational implicatures, need more contextual information along with the headline under consideration to support them. So generating inferences like (vi) and (vii) is not attempted in this work, and we try to generate inferences similar to those stated from (i) to (v).

### 3.1.2 Linguistic Definitions and Characteristics of Headlines

According to Dor [39], headlines are “the negotiators between stories and readers” and they have four functions of summarising, highlighting, attracting and selecting. The headline together with the lead or
the opening paragraph summarises a news story. Gattani [46] identifies three broad macro headline functions: (i) The informative headline, which gives a good idea about the topic of the news story, (ii) the indicative headline, which addresses what happened in the news story, and (iii) the eye-catcher headline, which does not inform about the content of the news story but is designed to entice people to read the story. The greater the mental effort required for processing a headline, the less relevant it becomes [39]. While reading a headline, the reader should be able to construct assumptions, either based on what can be perceived in their immediate environment or on the basis of assumptions already stored in their memory. The relevance of a headline is directly proportional to the amount of contextual effects and inversely proportional to the cognitive processing effort required to recover these effects [87].

Headlines are characterized by the density of the information present in them, and they have the syntactic characteristics of a telegraphic speech. They also contain bold expressions, polarisation, exaggerations, and provocative wording [65]. While processing headlines, more information should be expected from a shorter span of words. The grammatical rules for proper English sentences would be frequently violated either for filling more information in the short space available or for promoting the curiosity of the reader. News headlines use a special language called ‘block language’, a name first coined by Straumann [99]. Block language has a structure different from the normal clause or sentence structure but it often conveys a complete message. For example, the headline “Royal dog ill” conveys the same meaning as the fully grammatical sentence “The Queen’s pet dog Rex fell ill on Thursday”.

3.1.3 Importance of Headline

Headlines are the most widely read part of a newspaper. The hypothetical structure of news schema originally appeared in the book “News as discourse” [106] is shown in the following figure 3.1.

The headline and the lead paragraph together give a good idea about the event in the news story. The relation between the headline and the lead paragraph is given in Figure 3.2.

Most of the time, the headline acts as the abstract of an abstract. The exceptions for this happen when the usual headline writing strategies for generating curiosity in the reader are followed by flouting the maxims. The commonly used features of headline language with examples are given in Table 3.1.

<table>
<thead>
<tr>
<th>Headline language feature</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short words</td>
<td>White giraffe shot dead</td>
</tr>
<tr>
<td>Omitted words</td>
<td>Sky ours, Says comrade</td>
</tr>
<tr>
<td>Noun strings</td>
<td>Health minister’s phone track headache</td>
</tr>
<tr>
<td>Alliteration</td>
<td>Calamity care continues</td>
</tr>
<tr>
<td>Verb changes</td>
<td>Gates leaves Microsoft</td>
</tr>
</tbody>
</table>

Table 3.1 Features of headline language with examples

Newspaper headlines are relevance optimizers [39]. They are designed to optimize the relevance of their stories for their readers. Traditionally, newspaper headlines have been functionally characterized
as short, telegram-like summaries of their news items. This functional definition positions the headline in its appropriate role as a textual negotiator between the story and its readers.

### 3.1.3.1 Pragmatic Function of a Headline

Pragmatics focuses on the speaker-meaning at the level of utterances situated in a context [71, 118, 72]. In this case, the speaker is the writer of the headline, and the context is the current situation of the news event. A headline alerts the reader to the nature of the content of the text following it. The headline also enables the reader to grasp the meaning of the text and acts as a plurality of speech acts – urging, warning, and informing.

### 3.1.4 Relevance of this Work

This work computes inferences from the headlines. The inferences generated can be fed to a learning system that grades the impact created by the headline, based on sensitivity, child-friendliness, clarity, and various other parameters as required. It is advantageous to evaluate the impact because an ordinary reader naturally reads through the headlines in the newspaper before starting to read the whole news articles. The understandability of the headline contributes towards the ease of understanding of the news story following it.
3.2 Data

The dataset used in this work is comprised of around 350 headlines collected manually from different news websites [11, 2, 7] about four popular events which appeared continuously in news reports for a time span of a few months. The topics selected for including in the dataset are ‘Brexit’, ‘Disputes over the South China Sea’, ‘Syrian refugee crisis’, and ‘Pyeongchang Winter Olympics’. In the dataset, the headlines were arranged in chronological order to facilitate their use in studying the gradual evolution of the headlines, assuming that the reader has already read the previous headlines for the same news item.

3.2.1 Format of the Data

The data used as input for computing inferences using our rule-based system are in the format: Headline [source: News source Timestamp]. A subset of the same dataset is used for collecting human inferences for evaluation purposes. Some examples of headlines used as data are given below.

U.S. vows new North Korea sanctions ahead of Olympics face-off [source: Reuters February 07, 2018, 06:39 PM IST]

Schaeuble says British were “deceived” in Brexit campaign [source: Reuters June 23, 2017, 07:18 PM IST]
Olympics: IOC will not exclude Asian cities from 2026 Games bid [source: Reuters FEBRUARY 06, 2018, 02:58PM IST]

3.2.2 Collecting the Data

The data was collected by scraping from popular news websites and manually searching and picking interesting headlines. The headlines collected contained an image and a text snippet associated with them. Since this work is focusing on the analysis of headlines alone, the data is cleaned from images and extra text. A sample headline is given in Figure 3.3. The timestamp associated with headlines in the dataset is not used in the present work, though it might be useful for future developments to evaluate how headlines evolve as the news on that topic progresses in the course of time and how readers understand them based on their awareness of the previous headlines on the same topic. Although this approach will work on any sentence, for the benefit of studying the chronological progress of headlines for the same news event for later works, we insisted on collecting headlines for events that spanned for more than four months in the news. We selected headlines as our data because they are a source of rich information capable of generating multiple inferences from them.

![Figure 3.3](image.jpg) Headline and associated data from BBC

3.3 Proposed Inference Generation Method

In this work, it is assumed that only the headline is available to the reader for understanding the topic of the news and that the reader is entirely ignorant of the previous happenings under the same topic of news. The inferences of headlines are computed based on some logical conclusions attained, rooted in certain grammatical relations present in the headline. Rusu et al. [94] suggest subject-predicate-object triplet extraction from sentences that motivated this work. In the case of a news headline, the participants are the news editor of the headline, who is the speaker, and the common person reading the headline, who is the listener. For computing inferences, we begin with the extraction of nouns and verbs. The algorithm is outlined below.
Algorithm 1: Computing inferences from a news headline

**Result:** Inferences from headlines

Load the file containing headlines;

for each Headline in the file do
  Extract one headline from the dataset and preprocess it by removing the punctuations;
  Annotate the headline with POS tags for all tokens in it, using Stanford CoreNLP [80];
  Get all the verbs in the headline by comparing the POS tags of the tokens against the regular expression ‘V.+’;
  Get corresponding dependencies for all the verbs of the headline, using StanfordCoreNLP annotated with ‘depparse’ (Refer to section 3.3.1);
  Get all nouns and pronouns from the headline by comparing the POS tags of the tokens against the regular expression ‘N.+ | P.+’;
  Generate explicit inferences from the headline using Stanford OpenIE [12];
  Generate more inferences using the rule-based system (Refer to section 3.3.2 on grammatical relations held between tokens in the headline);

In the algorithm, we start with dependency parsing the headline, thus obtaining the verbs occurring in the headline with their dependencies. We get the headline tagged with POS tagger from Stanford and then extract the list of nouns and list of verbs in the headline. The verbs are also lemmatized to get the base form of the verbs present in the headline. The lemmatized form is used when a different form of the verb other than the tense form in which it appears in the headline, is required for a changed tense form in the computed inferences. A few rule-based approaches are implemented to get inferences from the headline. Stanford openIE [12] gives inferences that are directly stated in the headline. The headline “How the company kept out ‘subversives’” gives the inference “company kept out ‘subversives’” by openIE [12]. More inferences assumed from the syntactic structure of the headline are generated by the rule-based system.

3.3.1 Extracting the Dependencies

The Stanford dependencies are binary grammatical relations held between a ‘governor’ and a ‘dependent’ as specified in the Stanford dependencies manual [33], which provides documentation for the set of dependencies defined for English. The dependencies obtained from the Stanford CoreNLP dependency parser [80] are generated as a dependency tree that contains dependencies as tuples. The tuples generated for the headline ‘Rescue rules by Bank of England will divide Britain’ are given below.

```
```
For clarity, five tuples are shown below, each of which are triples consisting of a dependency type `dep`, a `governorGloss` and a `dependentGloss` from the list of tuples obtained from dependency parsing the headline ‘Rescue rules by Bank of England will divide Britain’ for explaining the terms. The current representation by Stanford contains approximately 50 grammatical relations. The dependencies are all binary relations: a grammatical relation holds between a governor (also known as a regent or a head) and a dependent. `governorGloss` is the word which is the governor in the grammatical relation and `dependentGloss` is the word which is the dependent in the grammatical relation. We iterate through these tuples to find interrelation between words, which give rise to inferences.

(i) `{'dep': 'nmod', 'governor': 2, 'governorGloss': 'rules', 'dependent': 4, 'dependentGloss': 'Bank'}`

(ii) `{'dep': 'case', 'governor': 6, 'governorGloss': 'England', 'dependent': 5, 'dependentGloss': 'of'}`

(iii) `{'dep': 'nmod', 'governor': 4, 'governorGloss': 'Bank', 'dependent': 6, 'dependentGloss': 'England'}`

(iv) `{'dep': 'aux', 'governor': 8, 'governorGloss': 'divide', 'dependent': 7, 'dependentGloss': 'will'}`

(v) `{'dep': 'dobj', 'governor': 8, 'governorGloss': 'divide', 'dependent': 9, 'dependentGloss': 'Britain'}`
3.3.2 Rule-Based System for Inference Generation

In this work, we use a rule-based system that is comprised of rules based on commonly occurring syntactic patterns. These patterns are modelled as inference triggers. Inference generation logic for an associated inference trigger is configured as a rule. Multiple iterations are performed on the dependency relations to generate inferences. Node JS tense conjugator [3] is used to find the required tense form of the verb to be attached in the computed inferences.

Since this work demonstrates the use of syntax structures to generate inferences using only a few triggers in the scope of inference triggers, the following types of verbs also could be included as triggers for producing more inferences. (i) iterative — anymore, return, another time, to come back, restore, repeat, etc. (ii) change of state verbs — stopped, began, continued, start, finish, carry on, cease, leave, enter, come, go, arrive, etc. (iii) Factive verbs — regrets, aware, realize, know, be sorry that, be proud that, be indifferent that, be glad that, be sad that, etc. (iv) Verbs of judging—accuse, criticize, blame, apologize, forgive, condemn, impeach, etc. As humans are better at making inferences these verbs should be included for more accurate results, by elaborating the rules using a string comparison of the verb under consideration with these above-mentioned triggers. The current set of inference triggers and rules used in computing inferences from headlines are elaborated in the following sections. The set of rules can be extended with more patterns to improve the quality of the generated inferences.

3.3.2.1 Presence of a Future Tense Verb

The presence of a future tense verb in the headline could suggest that we can infer that the event described by the noun is yet to happen. If the dependent is keeping an ‘aux’ (auxiliary) relation with the governor and if the ‘dependentGloss’ is the string ‘will’, then iterate once again through the dependencies to find a dependent ‘dobj’ (direct object) which is the noun phrase which is the (accusative) object of the verb where the ‘governorGloss’ of both dependency relations match. The dependency tree for a headline containing a future tense verb is given in Figure 3.4. The Algorithm 2 is applied to generate inferences for this future tense trigger.

For example, “Russian state television will not broadcast Olympics without national team” can have an inference “Olympics is not yet broadcast”.

![Figure 3.4](https://example.com/image3.4.png)  
**Figure 3.4** Dependency tree for headline “Russian state television will not broadcast Olympics without national team”
Algorithm 2: Computing inferences based on the presence of a future tense in a headline

Result: Inferences from headlines

Consider VD as the set of verbs in the headline with their dependencies, obtained from parser;

for each dependency tuple D in VD do
  if 'dep' of D is = 'aux' and 'dependentGloss' of D is = 'will' then
    for each dependency tuple ND in VD do
      if 'dep' of ND is = 'dobj' and 'governorGloss' of ND is = 'governorGloss' of D then
        output 'dependentGloss' of ND;
        output "is not yet";
        output past tense of ('governorGloss' of D);

3.3.2.2 Presence of the Conjunction ‘but’

The presence of the conjunction ‘but’ could suggest that we can infer that the subject was expected to undergo ‘negation’ of that which is mentioned in the part of the headline after the conjunction ‘but’. The dependency tree for a headline with a conjunction ‘but’ is given in Figure 3.5. The Algorithm 3 is applied to generate inferences for this conjunction-‘but’ trigger. For example, “Flybe to close Isle of Man base in 2020 but flights will continue” can have an inference >>" Closing Isle of Man base was expecting flights not continuing”.

Figure 3.5 Dependency tree for headline “Flybe to close Isle of Man base in 2020 but flights will continue”

Algorithm 3: Computing inferences based on the presence of conjunction ‘but’ in a headline

Result: Inferences from headlines

Consider VD as the set of verbs in the headline with their dependencies, obtained from parser;

for each dependency tuple D in VD do
  if 'dep' of D is = 'conj:but' then
    output "being ";
    output 'governorGloss' of D;
    output " was [not] expecting ";
    output Gerund of ('dependentGloss' of D );
### 3.3.2.3 Presence of ‘again’ in a Clause with a Verb

The presence of ‘again’ as an adverbial modifier in a clause with a verb could suggest that we can infer that the event described by the noun has already happened. The dependency trees for two headlines with ‘again’ as adverbial modifier are given in Figure 3.6 and Figure 3.7. The Algorithm 4 is applied to generate inferences that are made due to the presence of ‘again’. For example, the headline “Norway regulator again rejects “Donut” fish farm volume plan.” can have an inference >>“Norway regulator has rejected “Donut” fish farm volume plan before”.

![Figure 3.6](image_url) **Figure 3.6** Dependency tree for headline “Norway regulator again rejects “Donut” fish farm volume plan”

![Figure 3.7](image_url) **Figure 3.7** Dependency tree for headline “Catriona Matthew would captain Europe again in Solheim Cup”

**Algorithm 4**: Computing inferences of a headline which has the presence of ‘again’ in a clause with a verb

<table>
<thead>
<tr>
<th>Result: Inferences from headlines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consider VD as the set of verbs in the headline with their dependencies, obtained from parser ;</td>
</tr>
<tr>
<td>Consider N is the set of all nouns in the headline ;</td>
</tr>
<tr>
<td>for each dependency tuple D in VD do</td>
</tr>
<tr>
<td>if ‘dep’ of D is = ‘advmod’ and ‘dependentGloss’ of D is ‘again’ or if any noun in N is ‘dependentGloss’ with ‘dep’ of D = ‘nsubj’ then</td>
</tr>
<tr>
<td>for each dependency tuple ND in VD do</td>
</tr>
<tr>
<td>if ‘dep’ of ND is = ‘nsubj’ and ‘governorGloss’ of ND is = ‘governorGloss’ of D then</td>
</tr>
<tr>
<td>output ‘dependentGloss’ of ND ;</td>
</tr>
<tr>
<td>output past tense of (‘governorGloss’ of D) “before” ;</td>
</tr>
</tbody>
</table>

Get all the nouns in the headline and iterate through them until the ‘dependentGloss’ of a tuple is a noun in the headline and the dependent is ‘nsubj’(nominal subject) that is a noun phrase which is
the syntactic subject of a clause or if dependency relation is ‘advmod’ (adverb modifier). Then if the ‘governorGloss’ is ‘again’ follow from the inner For loop of Algorithm 4.

3.3.2.4 Presence of ‘further’ as an Adjective

The presence of ‘further’ as an adjective could suggest that we can infer that now it is already in the state described by the ‘noun’ succeeding the adjective ‘further’. For example, for the headline “UK economy to slow further.”, there can be an inference that “Economy is already slow”. The dependency tree for a headline containing ‘further’ as an adverb is shown in Figure 3.8.

![Figure 3.8: Dependency tree for headline “UK economy to slow further”](image)

3.3.2.5 Presence of a ‘noun compound’

Presence of noun compounds like ‘Brexit campaign’ could suggest that we may infer that ‘Brexit’ that is the first part ‘N1’ of the noun compound can be /can have a ‘campaign’, that is the second part ‘N2’ of the noun compound. The problem of computing semantic relation of the nouns N1 and N2 in the noun compound is not dealt with in this experiment. Only common sense assimilation that “N1 can be N2” or “N1 can have N2” is generated. For example, “Russia’s Olympic ban strengthens Putin’s reelection hand.” can have an inference “Olympic can be /can have ban”. The dependency tree for a headline containing a noun compound is shown in Figure 3.9.

![Figure 3.9: Dependency tree for headline “Russia’s Olympic ban strengthens Putin’s reelection hand”](image)

3.3.2.6 Presence of a ‘verb’ in the Past Tense

If the ‘verb’ is in the past tense in a headline, it could suggest that we can infer that the event has already happened. For example, “How women won the right to vote in 1918” can have an inference “women won the right to vote”. The dependency tree for a headline with a verb in the past tense is shown in Figure 3.10.
3.3.2.7 Presence of Nominal Modifier ‘of’

If there is a nominal modifier ‘of’ then, it could suggest that we can infer that the dependent ‘has’ governor. For example, “Governor of New Jersey meets PM Narendra Modi.” can have an inference “New Jersey has a Governor”. The dependency tree for a headline with a nominal modifier ‘of’ is given in Figure 3.11.

3.4 Results and Discussion

The unavailability of annotated inferences makes the comparison and evaluations difficult for this task. The inferences generated by the system are compared with human-annotated inferences for 100 randomly collected headlines. The two annotators are research scholars fluent in English researching Linguistics. They manually annotated a subset of the dataset for evaluation, based on the annotation guidelines provided to them. Annotation guidelines with explanatory examples for the inference triggers mentioned in section 3.5 were given to the annotators, and they were asked to look for the surface structure of the headline in general and use human judgment in making inferences. No upper limit on the number of generated human inferences was imposed. 11.8% of the inferences generated by the annotators were of the existential types, such as those beginning with a clause like “there exists”. The inference triggers other than the existential ones are occurring less in headlines compared to normal discourse, due to the peculiarity of block language used.

The percentages of computed inferences for some inference triggers used in this experiment are given in Table 3.2. For a headline ‘Britain takes step towards Brexit with repeal bill’ our system generates the following inferences (i) Britain takes step (ii) Britain takes step towards Brexit (iii) Britain takes step with repeal bill (iv) repeal can be/can have bill (v) Brexit has step.

Table 3.3 shows the comparative results of manually annotated inferences with the computed inferences for the three headlines in the first column and gives the percentage of correct computed inferences
Table 3.2 Accuracy and generated percentage of inferences computed

<table>
<thead>
<tr>
<th>Inference Trigger</th>
<th>Percentage of Accurate Inferences</th>
<th>Percentage of Inaccurate Inferences</th>
<th>Percentage of Missing Inferences</th>
</tr>
</thead>
<tbody>
<tr>
<td>But</td>
<td>69.3</td>
<td>0</td>
<td>30.7</td>
</tr>
<tr>
<td>Again</td>
<td>82.7</td>
<td>8.3</td>
<td>9</td>
</tr>
<tr>
<td>Further</td>
<td>94</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>Future Tense</td>
<td>93</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Noun Compound</td>
<td>54.4</td>
<td>40.2</td>
<td>5.4</td>
</tr>
</tbody>
</table>

and the percentage of incorrect results out of the computed inferences for those headlines. For example for the last headline — “Schaeuble Says British were “deceived” in Brexit campaign” only one of the manually annotated inferences — “Brexit can be/can have campaign” is computed by our Rule-Based system thus making the percentage of correct computed inferences to be 16.7%, and out of the three computed inferences “campaign has deceived” is wrong and thus the percentage of incorrect results in the computed inferences is 33%.

Though an extrinsic evaluation to evaluate the impact of the system, by investigating to what degree the system achieves the task for which it is developed is appropriate for this work, since we have not integrated it to the working system we wanted to do an intrinsic evaluation [105] on fluency, naturalness, meaning preservation, readability, clarity, informativeness, adequacy, syntactic correctness, appropriateness, etc. using a 7 point Likert scale with point values 1-7 assigned to responses. The agreement options and corresponding point values are given in Table 3.4. The evaluators were requested to register their agreement for each inference generated for a headline. The overall obtained score for appropriateness is 5.47, which indicates moderate agreement. The F-score is calculated as

\[
Precision = \frac{\# \text{ of relevant inferences}}{\text{Total} \# \text{ of computed inferences}} \tag{3.1}
\]

\[
Recall = \frac{\# \text{ of relevant inferences computed}}{\text{Total} \# \text{ of relevant inferences}} \tag{3.2}
\]

\[
F\text{-Score} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{3.3}
\]

Precision P measures how many computed inferences are relevant, and recall R measure how many relevant inferences are computed. F-Score is the harmonic mean of P and R. The F-score obtained is 0.56. In Figure 3.12, the \# of relevant inferences computed is shown in dark green colour, the \# of computed inferences is shown in blue colour and the \# of relevant inferences is shown in light green colour for different genres of news headlines.
3.5 Annotation Guidelines

3.5.1 Purpose of Annotation

This annotation task targets to provide the possible presuppositions for a news headline. Presuppositions can be any background assumption against which the headline makes sense or is rational. A ‘\(\gg\)’ symbol denotes presuppositions. A sentence and its negative counterpart share the same set of presuppositions, so the headline

“Karnataka CM meets prime minister Narendra Modi “ will have the following presupposition

\[ \gg \text{“Narendra Modi is the prime minister”} \]

which is true for the statement “Karnataka CM meets prime minister Narendra Modi” as well as its negative counterpart, “Karnataka CM does not meet prime minister Narendra Modi ”.

3.5.2 Guidelines for Annotating Presuppositions

For annotating, look for presupposition triggers, which are the linguistic items that are particular words or some aspects of the surface structure of the headline in general, which generates presuppositions. The following are some presupposition triggers with examples:

3.5.2.1 Definite Descriptions

Hunterston B: Pictures show cracks in Ayrshire nuclear reactor

\[ \gg \text{There are cracks in Ayrshire nuclear reactor.} \]
3.5.2.2 Factive Verbs

Factive verbs like regrets, aware, realize, know, be sorry that, be proud that, be indifferent that, be glad that, be sad that, etc.

Corbyn ‘regrets’ Labour MPs’ resignations
>> Labour MPs resigned.

3.5.2.3 Implicative Verbs

Implicative verbs like manage, remember, bother, get, dare, care, venture, condescend, happen, be careful, have the misfortune, have the sense, take the time, take the trouble, take the opportunity, etc.

How Russia Managed to Destroy Saudi Arabia?
>> Russia destroyed Saudi Arabia.

3.5.2.4 Change of State Verbs

Change of state verbs like stopped, began, continued, start, finish, carry on, cease, leave, enter, come, go, arrive, etc.

(i). Britain continued to struggle with Brexit
>> Britain was struggling with Brexit.

(ii). China has stopped stockpiling metals.
>> China had been stockpiling metals.

3.5.2.5 Iterative

Iterative like again, anymore, return, another time, to come back, restore, repeat, for the nth time, etc.

(i). HTC in talks with Micromax, Lava and Karbonn to return to Indian market
>> Micromax, Lava and Karbonn had been in the Indian market previously.

(ii). BoE’s Carney says will reassess outlook when there is Brexit clarity
>> Outlook has been assessed before.

3.5.2.6 Verbs of Judging

Verbs of judging like accuse, criticise, blame, apologize, forgive, condemn, impeach, etc.

(i). Trump blames financial market ‘disruption’ on Democrats
>> Trump thinks that financial market disruption is bad.

(ii). Amnesty criticises Hungary over treatment of migrants
>> Amity thinks that Hungary was not treating the migrants well.
3.5.2.7 Temporal Clauses

Temporal clauses like before, while, after, when, during, whenever, etc.

Britons were endlessly lied to during Brexit campaign

>> There was a Brexit campaign.

3.5.2.8 Cleft Sentences

Cleft sentences like (i) What he wanted to buy was a Fiat, (ii) It is Jaime for whom we are looking, (iii) All we want is peace, etc.

It is Jaime for whom we are looking

>> We are looking for someone.

3.5.2.9 Implicit Clefts with Stressed Constituents

Implicit clefts with stressed constituents like capital letters, or bold type, or underlined type can give rise to presuppositions.

3.5.2.10 Comparisons and Contrasts

Comparisons and contrasts like too, back, in return, etc. can give rise to presuppositions.

Russia is a better negotiator than Italy

>> Italy is a negotiator.

3.5.2.11 Non-restrictive Relative Clauses

John, who passed the test, was elated.

>> John passed the test.

3.5.2.12 Counterfactual Conditionals

If I had a guarantee, then I’d love them

>> I don’t have a guarantee.

3.5.2.13 Questions

What’s missing from your low carb breakfast?

>> Something is missing from your low carb breakfast.

Similarly, who can be replaced by someone, whereby somewhere, how by somehow to generate presuppositions. Yes/No questions will generally have vacuous presuppositions.

Are you living with mild or moderate depression?
Either you are living with mild or moderate depression, or you are not.

3.5.2.14 More than Two Words in Quotes

More than two words in quotes can give a presupposition that something is said. News headlines sometimes have quotes to emphasize words. So it may not be an utterance always. So we assume that more than two words in quotes mean something is said.

Merkel says May’s Brexit proposals “not the breakthrough”.

>> Merkel says, “not the breakthrough”.

3.5.2.15 Future Tense Verb

The presence of future tense verb in the headline can create a presupposition that the event described in the noun has not happened yet.

Russian state television will not broadcast Olympics without national team

>> Olympics is not yet broadcast by Russian state television.

3.5.2.16 The Conjunction ‘but’

The conjunction ‘but’ suggests a contrast.

Olympics-It’s ready but will they come?

>> Being ready was expecting them to come.

3.5.2.17 Gender-Specific Statements

New Zealand Prime Minister Jacinda Ardern gives birth to first child.

>> Jacinda Arden is a female.

Since the headlines use tricky language to attract readers, human intuition while listing the presuppositions is required. The format of the annotation is to write presuppositions preceding with a ‘>>’ following the headline and after writing all presuppositions for a headline, ending it with a ‘||’ with one presupposition statement in a line. Presuppositions should be expressed as simple sentences in simple English.

3.6 Conclusion

In conclusion, a syntax-based computation method for generating inferences from headlines generates many accurate inferences such as presuppositions and conventional implicatures. News headline as a stand-alone unit of text without attaching any information from the context in which it appeared in a news report holds the potential to generate multiple inferences from them, due to the rich information
content. Based on the observation that the presence of certain words and tense conditions can trigger inferences from a headline, we tried to generate inferences based on a set of rules, formulated based on certain grammatical relations present in the headline. The rule set could be expanded to include more observations and complex rules to compute more inferences. These inferences can be used to measure the impact and sensitivity of a headline mainly for checking the appropriateness when used in a platform designed for children.
<table>
<thead>
<tr>
<th>Headline</th>
<th>Manually Annnotated Inferences</th>
<th>Computed Inferences</th>
<th>Percentage of Correct Inferences</th>
<th>Percentage of Incorrect results</th>
</tr>
</thead>
</table>
| IOC extends North Korea deadline for Pyeongchang games                  | 1. IOC has power to extend deadline  
2. North Korea has deadline  
3. Deadline can be extended  
4. There exists North Korea  
5. There exists Pyeongchang games                                                | 1. Korea can have deadline  
2. Pyeongchang has games  
3. Games has deadline                                                              | 40%                               | 0%                              |
| Olympics: Medals at Winter Olympics through years                        | 1. There exists Winter Olympics  
2. Olympics has medals  
3. Olympics had been happening through years  
4. There are medals associated with years                                           | 1. Winter can have olympics  
2. Olympics has medals  
3. years had medals                                                                    | 75%                               | 0%                              |
| Schaeuble Says British were “deceived” in Brexit campaign               | 1. Schaeuble exists  
2. Schaeuble believes that the British were “deceived” in Brexit campaign  
3. Brexit can have campaign  
4. Schaeuble said something  
5. Schaeuble believes that the British were ‘deceived’ in Brexit campaign.  
6. Brexit campaign happened.                                                      | 1. Schaeuble Says British were “deceived”  
2. Brexit can be/can have campaign  
3. campaign has deceived                                                     | 16.7%                             | 33%                             |

Table 3.3 Comparison of manually annotated inferences with computed inferences for a headline
<table>
<thead>
<tr>
<th>Agreement Options</th>
<th>Point Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly Disagree</td>
<td>1</td>
</tr>
<tr>
<td>Disagree</td>
<td>2</td>
</tr>
<tr>
<td>Somewhat Disagree</td>
<td>3</td>
</tr>
<tr>
<td>Neither Agree nor Disagree</td>
<td>4</td>
</tr>
<tr>
<td>Somewhat Agree</td>
<td>5</td>
</tr>
<tr>
<td>Agree</td>
<td>6</td>
</tr>
<tr>
<td>Strongly Agree</td>
<td>7</td>
</tr>
</tbody>
</table>

*Table 3.4 Agreement options and their corresponding point values*
Chapter 4

Conversational Implicatures in English Dialogue: Data Collection and Annotation

4.1 Introduction

Human dialogue often contains utterances having meanings entirely different from the meanings of
the sentences used and are clearly understood by the interlocutors. But in human-computer interactions,
the machine fails to understand the implicated meaning unless it is trained with a dataset containing
the implicated meaning of an utterance along with the utterance and the context in which it is uttered.
In linguistic terms, conversational implicatures are the meanings of the speaker’s utterance that are
not part of what is explicitly said. In this chapter, we introduce a dataset 1 of dialogue snippets with
three constituents, which are the context, the utterance, and the implicated meanings. These implicated
meanings are the conversational implicatures. The utterances are collected by transcribing from listening
comprehension sections of English tests like TOEFL (Test of English as a Foreign Language) as well
as scraping dialogues from movie scripts available on IMSDb (Internet Movie Script Database). The
utterances are manually annotated with implicatures.

A conversation in the animation movie ‘Anastasia’ goes like this,

(1) ANYA: Is this where I get travelling papers?

    CLERK: It would be if we let you travel, which we don’t so it isn’t.

For a simple polar question asked by the character ANYA, the complex response by the CLERK
allows a human reader to conclude on many implicatures with a ‘yes’ or a ‘no’ as answers to that
polar question while reading that dialogue in the movie script. But in human-computer interactions,
such a human response will be complicated for the machine to conclude as a ‘yes’ or a ‘no’. The
basic assumption in a conversation is that, unless otherwise indicated, the participants are adhering to
the cooperative principles and maxims [118]. Conversational implicature is the linguistic term for
conveying more than what is said, and it is a highly contextualized form of language use that has a
lot in common with non-linguistic behaviour [18]. Conversational implicatures are cancellable, non-

1figshare.com/articles/Implicature_dataset/10315505
conventional, calculable, and non-detachable. An inference is cancellable or more exactly defensible if it is possible to cancel it by adding some additional premises to the original ones. Implicatures are not part of the conventional meaning of linguistic expressions [72]. As implicatures are not explicitly said, the speaker can always deny an implicature claiming that he/she did not intend to implicate something. As Grice [49] puts it, concerning utterances that carry conversational implicatures, “it is not possible to find another way of saying the same thing, which simply lacks the implicature in question”. If an utterance of $P$ conversationally implicates $q$ in $C$, then an utterance of $Q$ conversationally implicates $q$ in $C$, too, given that utterances of $P$ in $C$ and of $Q$ in $C$ say the same thing. This is the non-detachability test [21]. Conversational implicature is denoted by the symbol ‘$+$’. For example, in the following utterance by the Girl as a response to the Boy’s utterance, the implicated meaning is given after the ‘$+$’ symbol.

(2)  
**Boy**: Have you done the trigonometry and calculus problems?  
**Girl**: I did the trigonometry problems.  
$+$ I did not do the calculus problems.

We required data for training the model in our experiments on computing implicated meanings of utterances. Since datasets with implicatures of utterances are not available, except for a few introductory [88] and sentence level [68] experiments, we attempted to create a dataset of dialogues with their implicature. This chapter outlines the sources and methods of data collection for implicatures.

### 4.1.1 Generating Conversational Implicatures

Conversational implicatures are generated by a variety of situations like replying with a metaphor, idiom, irony, tauntology, hyperbole, sarcasm, indirect criticism, etc. PopeQ implicature is where a popularly well-known question is asked in response to a polar question to implicate that the same answer as that of the response question is the answer to the polar question initially asked. The name PopeQ is derived from the question “*Is the Pope Catholic?*” for which the answer is an obvious ’Yes’. There are classifications of conversational implicatures like scalar, generalized, particularized, relevance-based, etc. See examples in Table 4.1 where context refers to the most recent text in a conversational context preceding an utterance. The same response can have different meanings in different contexts. For example [79], the response utterance by B, “*I’ve cleared the table,*” has two different implicated meanings for the two different context questions asked by A.

(3)  
**A**: Have you cleared the table and washed the dishes?  
**B**: I’ve cleared the table.  
$+$ I have not washed the dishes.

(4)  
**A**: Am I in time for supper?  
**B**: I’ve cleared the table.  
$+$ No. You are late for supper.
<table>
<thead>
<tr>
<th>Type of Implicature</th>
<th>Context</th>
<th>Utterance</th>
<th>Implicature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalar</td>
<td>Who made these donuts?</td>
<td>I made some of these donuts.</td>
<td>I did not make all of these donuts.</td>
</tr>
<tr>
<td>Generalized</td>
<td>Did you call John and Benjamin?</td>
<td>I called Benjamin.</td>
<td>I did not call John.</td>
</tr>
<tr>
<td>Particularized</td>
<td>Did you drink the milk I kept on the table?</td>
<td>The cat seems to be happy.</td>
<td>No. I did not drink milk. The cat might have drunk the milk.</td>
</tr>
<tr>
<td>Relevance</td>
<td>How about going for a walk?</td>
<td>Isn’t it raining out?</td>
<td>No. I am not coming for a walk now.</td>
</tr>
<tr>
<td>PopeQ as a response</td>
<td>Are you sure you can take care of yourself this weekend?</td>
<td>Can a duck swim, mother?</td>
<td>Yes. I am sure I can take care of myself.</td>
</tr>
<tr>
<td>Metaphor as a response</td>
<td>Do you like her?</td>
<td>She is like cream in my coffee.</td>
<td>Yes. I like her a lot.</td>
</tr>
<tr>
<td>Tautology as a response</td>
<td>Do you want to taste my hamburger?</td>
<td>Hamburger is hamburger.</td>
<td>No. Hamburgers are not too good to taste.</td>
</tr>
<tr>
<td>Hyperbole as a response</td>
<td>Are you hungry?</td>
<td>I could eat a horse.</td>
<td>Yes. I am extremely hungry.</td>
</tr>
<tr>
<td>Idiom as a response</td>
<td>I could have been more careful.</td>
<td>It is useless to cry over spilled milk.</td>
<td>It is useless to be sad about what had already happened.</td>
</tr>
</tbody>
</table>

**Table 4.1** Types of implicatures

### 4.2 Approaches Attempted for Creating Implicature Corpus

We created a dialogue implicature dataset for which dialogues were collected by transcribing the listening comprehension sections of English language proficiency tests and dialogues from movie scripts. We then annotated them manually with the conversational implicatures to aid our research on generating conversational implicatures in human-computer interactions. A similar annotated resource is an experimental dataset [88] annotated with a definite/probable ‘yes’ or ‘no’ for 215 indirect polar questions. For ease and uniformity of creation and usage, we intended to create a dataset with dialogue triplets $<$ context, utterance, implicature $>$, where only one turn of dialogues in a scene for each pair of interlocutors is extracted, and only the single immediate context of an utterance is considered. The number of implicatures for an utterance in a context was not restricted to an upper limit.

#### 4.2.1 Related Annotation Work

Potts [88] introduced an experimental dataset involving 215 indirect question-answer pairs collected from 4 different sources and annotated with polarity using Amazon Mechanical Turk [1]. Lahiri [68]
annotated a corpus of 7,032 sentences using Amazon Mechanical Turk with ratings of formality, informativeness, and implicature for each sentence on a 1-7 scale during which the annotators were asked to form implicatures for a given sentence. Lasecki, Kamar, and Bohus [70] collected conversations focused around definable tasks using crowdsourcing methods. In their work, two annotators were assigned part of an agent and a user to a randomly given task, and they were asked to engage in a conversation to complete the task. Reddy, Chen, and Manning [93] introduced CoQA, a 127k question-answer dataset for building conversational question answering systems. In the above research, pairs of annotators were given passage for reading and were asked to frame questions based on the passage and answer them from the passage consecutively. Another crowd-powered system utilizing asynchronous chat for efficient collection of dialogue dataset was designed by Ikeda and Hoashi [55]. In their work, they collected data by giving a topic and asking contributors assigned with part of A or B to chat upon the given topic for up to 16 turns. Multiple contributors were taking the role of A and B, and the chat data was not collected in real-time, but instead completed when the contributors were available.

4.2.2 Crowdsourcing the Implicature Generation Task

Along the same lines of the related works mentioned above, we designed a methodology using crowdsourcing platforms to cooperate pairs of contributors and assign them to generate an implicature as follows. A situation will be given to crowdsource-contributors A and B. Many contributors will be sequentially assigned parts of A and B for the same given situation so that response utterances in different contexts and their implicatures can be collected. The contributor joining first to attempt the task will be assigned the role of A, and another contributor who joins at a later point of time will be assigned the role of B. Crowdsource-contributor A will be asked to provide a single utterance as a polar question based on the given situation. Crowdsource-contributor B will be asked to give two utterances as answers to the polar question posted by A. (i) An answer without an explicit ‘Yes/No’ and (ii) the real implicated meaning of the answer (i) with an explicit ‘Yes’ or an explicit ‘No’. The responding crowdsource-contributors were asked to be cooperative and to give an answer relevant to the question. Some ideal expectations about the question and response in a situation are given in Table 4.2.

Crowdsource-contributors who are native speakers of English, hailing from the US, UK, Australia, New Zealand, and Canada were given this task. This task, which demands three inputs, requires approximately 8-15 minutes to complete. As the dialogue data are obtained from 2 different contributors, it takes a turn around time of approximately 15 minutes. The crowdsourcing approach did not obtain high-quality dialogue data. The challenges in crowdsourcing this kind of dialogue requirement is discussed below.

The crowdsourcing platform MicroWorkers’ [9] Questions and answers TTV (Template Test and Verification) customized for the implicature annotation task is given in Figure 4.1 and the examples given for response generation in Amazon Mechanical Turk are given in Figure 4.2.
Table 4.2 Examples of situations provided for conversation generation task with ideal responses

<table>
<thead>
<tr>
<th>Situation</th>
<th>Polar question</th>
<th>Indirect answer</th>
<th>Implicature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Situation 1: The TV had been on for a long time</td>
<td>Can I switch off the tv?</td>
<td>My favourite program will begin now.</td>
<td>No. I am going to watch it now.</td>
</tr>
<tr>
<td>Situation 1: The TV had been on for a long time</td>
<td>Is anyone watching this?</td>
<td>Oops! I forgot to switch it off.</td>
<td>No. You can switch it off.</td>
</tr>
<tr>
<td>Situation 2: Both of you, A and B, are dressed up to go out.</td>
<td>Should I take the umbrella?</td>
<td>It rained yesterday.</td>
<td>Yes. There is a chance of rain.</td>
</tr>
<tr>
<td>Situation 2: Both of you, A and B, are dressed up to go out.</td>
<td>Should I take the umbrella?</td>
<td>The sky is black as ink.</td>
<td>Yes. There is a chance of rain.</td>
</tr>
<tr>
<td>Situation 2: Both of you, A and B, are dressed up to go out.</td>
<td>Should I take the umbrella?</td>
<td>The sky is clear.</td>
<td>No. It will not rain.</td>
</tr>
<tr>
<td>Situation 2: Both of you, A and B, are dressed up to go out.</td>
<td>Should I take the umbrella?</td>
<td>I heard thunder.</td>
<td>Yes. There is a chance of thunderstorms and rain.</td>
</tr>
</tbody>
</table>

4.2.3 Challenges in Crowdsourcing the Implicature Generation Task

The first challenge in this approach is gathering and defining the situations which can give rise to implicatures in a conversation associated with them. The situation imagined by the task creators may not be entirely comprehended by the crowdsource-contributors. The reading and understanding of the provided situations require much cognitive effort and comprehension capabilities from the crowdsource-contributors. Crowdsourcing tasks are generally easy and do not require any specific knowledge and can be annotated directly without thinking much. Since this task demands the contributors to imagine more than what is required for an average task, it can become a less popular task to attempt and cost more to get contributors. They might also lose the natural dialogue flow in the process of comprehending the situation.

The inherent chance of the contributors giving irrelevant or indifferent answers like, ‘I don’t know’, ‘I am not sure’ or ‘do as you wish’, etc., despite the guidelines given creates another challenge and a requirement upon the completion of the task, to verify each utterance. The quality of the dialogue snippet is very much dependent on the question asked by the first crowdsource-contributor who attempted the task for a given situation. The difference in the understanding of the situation by the two participants A and B could compromise the quality of the generated utterances. All the questions from the first set
Figure 4.1 MicroWorkers’ customized TTV questions and answers template for conversation generation

of contributors that do not give rise to utterances from B with a possible implicature have to be deleted from the dataset, and that posed another challenge of cleaning up the data. Crowdsourcing platforms do not follow the FIFO (First In First Out) strategy for tasks, and so the tasks which are not completed quickly or those masked by filters have a high chance of getting forgotten.

The idea that there are clear indicators of implicatures in conversation when metaphors, ironies, indirect criticism, etc. are used, made us think of the possibilities of getting the dialogues from the movie scripts which are clean transcripts of hypothetical human dialogues and the listening comprehension sessions which are designed to create implicated meanings of utterances to test the English language proficiency of non-native speakers of English. Movie scripts had been a source of data for language research, particularly for identifying dialogue structures [15], speaker identification [67], and character modeling.

4.3 Implicature Corpus

The dialogues are collected from listening comprehension tapescripts of short conversation narrations available online for TOEFL, movie dialogues from the IMSDb [8] for 45 animation movies and other dialogues with metaphors, idioms, hyperboles, indirect criticisms, etc., that are extracted from the internet. Dialogues are also synthesized similar to the extracted ones with an interpretation of answers to polar questions that do not directly express a ‘yes’ or ‘no’ answer. We selected the animation genre
Figure 4.2 MTurk good and bad examples for response generation in context

of movies, considering the light tone of the scripts in this genre and the simplicity of the dialogues, as they target children as their primary audience. The movie script data have another advantage of being less noisy and devoid of spelling and grammatical errors compared to the real-time dialogues. The occurrences similar to the ones focused on by de Marneffe [33] involving scalar modifiers such as (5) and numerical answers such as (6) were also identified for the contexts with response utterances obtained from the movie scripts and listening comprehension questions.

(5) A: Was the movie wonderful?
   B: It was worth seeing.

(6) A: Are your kids little?
   B: I have a 10-year-old and a 7-year-old.

Online resources are used for the interpretation of idioms, metaphors, hyperboles, and tautologies.

4.3.1 Collecting the Dialogue Snippets

The dialogue snippets with a context and an utterance are identified as a sentence ending with a question mark and not containing multiple question marks in it together with the response sentence that follows it. After scraping those snippets from the IMSDb for the animation genre, they are inspected manually for being a polar question and the response holding a chance of implicatures. Those snippets with a ‘Yup!’, ‘Yes’, ‘Yep!’, ‘Nope!’, ‘No’, ‘Nay’ and similar, in the response are removed as they give a clear ‘yes’ or ‘no’ answer to the question asked. The remaining snippets are preprocessed by removing
the name of the movie character, making the utterance, and replacing it by $A$ for the questioner and $B$ for respondent. The preprocessed snippets are manually annotated with one or more implicated meanings, that we infer from the response utterance. The accuracy of the annotated implicatures can be verified by computing the similarity of annotations by different annotators for the same response utterance. Those annotations with high similarity scores can be prioritized for entry to the dataset.

**Figure 4.3** Implicature context from (a) movie Script of ‘Anastasia’; (b) TOEFL transcript [14]

Figure 4.3 shows excerpts from the script of ‘Anastasia’ movie from IMSDb and a transcript of listening comprehension question from TOEFL for which the part-A of the test is the narration of a short conversation between two people with a question about the conversation. Narratives of the TOEFL listening comprehension section are manually transcribed from the English Test Store website [5] for 500 dialogue narrations. Implicature generating dialogue situations are collected from 45 movie scripts of the animation genre from IMSDb and cleaned from other dialogues. The annotation is done manually by undergraduate students of linguistics, whose primary language of instruction is English. For TOEFL transcripts, the correct answer out of the four multiple-choice options provided is annotated as the implicature. For movie dialogues, the implicature is annotated depending on the utterance and the context. If the context is a polar question and the utterance is an answer without a ‘yes’ or a ‘no’, then a direct answer with an explicit ‘yes’ or ‘no’ is annotated as the implicature. Some implicatures annotated form listening comprehension sessions are given in Table 4.3, and those from IMSDb are given in Table 4.4.

**4.3.2 Evaluating the Annotation Task**

For evaluating the manual annotation of the context, utterance pairs collected from movie data, fourteen dialogue pairs from animation movie ‘CoCo’ where context is a polar question, is given to 2 annotators and their polarity assumption about the response utterances is recorded. The Cohen’s Kappa $\kappa$ inter-annotator agreement score is calculated from the observations. $\kappa$ takes into account the agreement happening by chance. As there are only two chosen annotators, Cohen’s Kappa is a preferred metric over Fleiss’ kappa. The response annotation sheet is given in Figure 4.4, and recorded observations for the fourteen dialogue pairs are given in Figure 4.5. $\kappa$ is calculated as follows,

$$\kappa = \frac{P_o - P_e}{1 - P_e} \quad (4.1)$$
<table>
<thead>
<tr>
<th>Context</th>
<th>Utterance</th>
<th>Implicature</th>
</tr>
</thead>
<tbody>
<tr>
<td>This calculator is not working, right?</td>
<td>I think you got the battery on upside down.</td>
<td>Yes. It is not working because the battery is not correctly positioned.</td>
</tr>
<tr>
<td>Would you like to go with us for coffee a little later?</td>
<td>I am off caffeine. Medical restriction.</td>
<td>No. I have to eliminate coffee from my diet.</td>
</tr>
<tr>
<td>Were you pleased with last week’s convention?</td>
<td>Nothing went as planned.</td>
<td>No. I was not pleased with last week’s convention.</td>
</tr>
<tr>
<td>Let me help you with those packages?</td>
<td>Thanks, but it is only three quarters to the block.</td>
<td>No. You don’t have to help me with those packages. It is not too far for me to carry the packages.</td>
</tr>
</tbody>
</table>

Table 4.3 Utterances and contexts collected from TOEFL listening comprehension section with their implicated meanings

<table>
<thead>
<tr>
<th>Context</th>
<th>Utterance</th>
<th>Implicature</th>
<th>Name of the movie</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can I call you in a little while?</td>
<td>It’s four in the morning... I’m going to go to sleep.</td>
<td>No. You should not call me.</td>
<td>Lost in Translation</td>
</tr>
<tr>
<td>It’s bad, isn’t it?</td>
<td>We should get you to the doctor.</td>
<td>Yes. It is bad.</td>
<td>Lost in Translation</td>
</tr>
<tr>
<td>And marriage, does that get easier?</td>
<td>It’s hard. We started going to a marriage counselor.</td>
<td>No. Marriage did not get easier.</td>
<td>Lost in Translation</td>
</tr>
<tr>
<td>How does that sound?</td>
<td>About as bad as you smell!</td>
<td>That does not sound good.</td>
<td>Anastasia</td>
</tr>
</tbody>
</table>

Table 4.4 Utterances and context collected from IMSDb movie dialogues with their implicated meanings

where:

- \( P_o \) = the relative observed agreement among raters.
- \( P_e \) = the hypothetical probability of chance agreement. There were a total of 14 responses to annotate.

Altogether, five responses were rated ‘Yes’ by both.

Altogether, four responses were rated ‘No’ by both.

Overall, Annotator-1 ticked Yes to 6 responses and No to 8.

Overall, Annotator-2 ticked Yes to 9 responses and No to 5.

Step 1: Calculate \( P_o \) (the observed proportional agreement): 5 responses were rated Yes by both. 4 responses were rated No by both. So,

\[
P_o = \frac{\text{number in agreement}}{\text{total}}
\]

\[
P_o = \frac{5 + 4}{14} = 0.64
\]
Step 2: Find the probability that the raters would randomly both say Yes. Annotator-1 said Yes to 6 out of 14 responses, giving a probability of 0.428. Annotator-2 said Yes to 9 out of 14 responses, giving a probability of 0.643. The total probability of the raters, both saying ‘Yes’ randomly is:

\[ 0.428 \times 0.643 = 0.275 \]  

(4.4)

Step 3: Calculate the probability that the raters would randomly both say ‘No’. Annotator-1 said No to 8 out of 14 responses, or 0.57. Annotator-2 said No to 5 out of 14 responses, or 0.357. The total probability of the raters, both saying ‘No’ randomly is:

\[ 0.57 \times 0.357 = 0.203 \]  

(4.5)

Step 4: Calculate \( P_e \). Adding answers from step 2 and 3 gives the overall probability that the raters would randomly agree. \( P_e = 0.275 + 0.203 = 0.478 \).

Step 5: Inserting the calculations into the formula (4.1) would generate the Kappa \( \kappa \) as:

\[
\kappa = \frac{(P_o - P_e)}{(1 - P_e)} = \frac{(0.64 - 0.478)}{(1 - 0.478)} = \frac{0.162}{0.522} = 0.3103
\]

(4.6)

So the calculated \( \kappa = 0.31 \), which indicates fair agreement. Refer Table 4.5 for the interpretation of the agreement when Kappa statistic \( \kappa \) varies from 0 to 1.

### 4.4 Conclusion

We observed that the dialogue data with implicatures is obtainable only through a close analysis of dialogues. We identified that non-polar answers to polar questions is a significant pattern of implicating utterances and identified them by inspecting movie dialogues. The obtained data [47] can be used for identifying conversational implicatures in dialogues and for synthesizing dialogues with implicatures. This is an ongoing data collection project, and when the collected data reaches a considerable scale with negative samples included, it can be used for designing a dialogue system utilizing utterances and context embedding for dialogue generation [17, 112].

<table>
<thead>
<tr>
<th>( \kappa ) score</th>
<th>Agreement Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>agreement equivalent to chance</td>
</tr>
<tr>
<td>0.1 – 0.20</td>
<td>slight agreement</td>
</tr>
<tr>
<td>0.21 – 0.40</td>
<td>fair agreement</td>
</tr>
<tr>
<td>0.41 – 0.60</td>
<td>moderate agreement</td>
</tr>
<tr>
<td>0.61 – 0.80</td>
<td>substantial agreement</td>
</tr>
<tr>
<td>0.81 – 0.99</td>
<td>near-perfect agreement</td>
</tr>
<tr>
<td>1</td>
<td>perfect agreement</td>
</tr>
</tbody>
</table>

**Table 4.5** Interpretation of the agreement when Kappa statistic \( \kappa \) varies from 0 to 1
What do you think the Response mean?
A **Yes** or a **No**. Give a ✅ in the respective column.

<table>
<thead>
<tr>
<th>Polar Question</th>
<th>Response</th>
<th>I think the response is Yes</th>
<th>I think the response is No</th>
</tr>
</thead>
<tbody>
<tr>
<td>You want to end up like that man, left off your family's ofrenda?!</td>
<td>I don't care if I'm on some stupid ofrenda!</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Can I still sign-up?</td>
<td>You got an instrument?</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Excuse me, can I borrow your guitar?</td>
<td>Sorry, muchacho.</td>
<td></td>
<td>✅</td>
</tr>
<tr>
<td>You thought we weren't?</td>
<td>I thought it might've been one of those made things that adults tell kids...</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Anything to declare?</td>
<td>Some churros... from my family.</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>You took my photo off the ofrenda?!</td>
<td>It was an accident!</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>You really hate music that much?</td>
<td>I will not let you go down the same path he did.</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>Is it too obvious?</td>
<td>I think it's just the right amount of obvious?</td>
<td>✅</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 4.4** Response annotation sheet
<table>
<thead>
<tr>
<th>Polar Question</th>
<th>Response</th>
<th>Annotator 1</th>
<th>Annotator 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>You want to end up like that man, left off your family's ofrenda?!</td>
<td>I don't care if I'm on some stupid ofrenda!</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Can I still sign-up?</td>
<td>You got an instrument?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Excuse me, can I borrow your guitar?</td>
<td>Sorry, muchacho.</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>You thought we weren't?</td>
<td>I thought it might've been one of those made things that adults tell kids...</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Anything to declare?</td>
<td>Some churros... from my family.</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>You took my photo off the ofrenda?!</td>
<td>It was an accident!</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>You really hate music that much?</td>
<td>I will not let you go down the same path he did.</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Is it too obvious?</td>
<td>I think it's just the right amount of obvious?</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Have you found him, Pepita? Have you found our boy?</td>
<td>A footprint!</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Is it too much?</td>
<td>it's all great.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Did you ever regret it? Choosing music over... everything else.</td>
<td>It was hard</td>
<td>Yes</td>
<td>yes</td>
</tr>
<tr>
<td>Can I get that doll?</td>
<td>Great! I'd love to get rid of it!</td>
<td>Yes</td>
<td>yes</td>
</tr>
<tr>
<td>Oh hey Dad, can we eat now?</td>
<td>Just wait till we get home.</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>C-can you can you unlock it?</td>
<td>Not in a million years. But it wouldn't matter;</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

**Figure 4.5** Polar response annotations by two annotators to non-polar responses
Chapter 5

Using Dependency Trees for Summarization and Question Answering

In this chapter, we discuss how Stanford dependency trees can be used for NLP applications other than inference generation. The use of these tree structures for summarization and machine reading comprehension are explored in the following sections.

5.1 Dependency Graphs for Summarization

Document summarization is a versatile application of natural language processing adaptable for generating input to context briefing, machine comprehension, meeting minute formation, and question generation. In this section, we propose a summarization method using dependency graphs. The dependency graph is a conglomeration of the dependency trees of the sentences for a passage obtained from the parser. Each word in the document becomes a node of the dependency graph, and the dependencies between the words are represented as the edges in the graph. We represent the document as a dependency graph and reduce the graph by prioritizing the word nodes, which hold the maximum edges associated with them, which we call as the fullest quivers. We then add the sentences containing the most number of filled quivers to the summary.

The graph data structure useful in representing pairwise relationship between objects had been experimented widely for natural language processing [28, 50, 43, 41, 20, 61, 83]. In this section, we explore the graph representation of a document connected by dependencies and find its use in finding the essential concepts expressed by the document.

Trees can efficiently represent sentences, but when it comes to the representation of the collective meaning of a passage, trees become insufficient, and graphs are required [103]. The easiest way for generating a document graph is by combining the trees obtained for individual sentences where each word becomes a node, and its dependencies represent edges. Since we have the dependency trees corresponding to individual sentences from the parser, connecting all of them together to obtain a graph representing the entire document is an effortless process. The major process is deciding whether to add a dependency tuple to the graph or to discard the tuple. The sentence is an organized set, the constituent elements of which are the words [102]. Each word in a sentence is not isolated as it is in the dictionary.
The mind perceives connections between words and their neighbours. Structural connections establish dependency relations between words [102]. The motivation for this approach to summarization is the ease of constructing a document graph from the dependency tuples obtained for the sentences from any off-the-shelf parser.

5.1.1 Proposed Method for Extractive Summarization

Summarization of a document starts with content selection, in which the essential content-bearing words are identified, and the sentences or the paraphrased or simplified version of those sentences are added to the summary. Tesnière [102] identifies four basic categories of content words (i) verbs (ii) nouns (iii) adverbs and (iv) adjectives that are semantically non-empty. In contrast, translatives such as prepositions, auxiliary verbs, and articles that transfer content words across syntactic categories are semantically empty. A graph representation of the entire document is required for finding the words which are connected with the maximum number of other words. Several techniques for dependency parsing are presented by Kübler et al. [66].

The Stanford parser [34]converts the parse into a dependency tree. For every element that one has in the utterance at hand, there is exactly one node in the syntactic structure that corresponds to that element [102]. One of the advantages of dependency parsers for NLP is that the parse can be easily encoded in a table. The output obtained from Stanford CoreNLP parser [80, 34, 35] consists of tuples like 
\{'dep': 'nmod:of', 'governor': 4, 'governorGloss': 'Bank', 'dependent': 6, 'dependentGloss': 'England'\} which mainly consist of a governorGloss, dependentGloss and a dependency relation existing between them. In the dependency tree obtained for a sentence, there will be edges labeled with the dependency relation from the GovernorGlosses to the DependentGlosses. The GovernorGloss is the word that determines the environments in which the governor and the dependent can appear together. So the number of out edges from a node represents the prominence of the node word for the document.

5.1.2 Constructing the Dependency Graph for a Passage

For finding the congruence of nodes in the sentences of a passage to identify their prominence for candidature to the summary, resolving co-reference is very much important. So we first process the news articles through the Hugging face neural coref [10] to resolve co-reference. Then the resolved sentences are parsed using Stanford parser to obtain the dependency tuples. The dependency graph is constructed as an adjacency list from these tuples, by adding the governorGloss and dependentGloss as nodes and the dependency relation between them as the outgoing edge label from the governor to the dependent. The singular and plural form of a word becomes the same node. Punctuations and stopwords are not added as nodes in the dependency graph. If a word for which a node exists in the dependency graph, reappears again in another sentence then it is considered as the same node, and the edges corresponding to the new occurrence of the word are added in the graph. Every node will be linked to the tuple consisting of the POS tag of the word associated with the node and a list of sentences containing that word in the form.
[POS, index-1, index-2...index-n]. The Construction of the dependency graph is explained in Algorithm 5.

\[ \text{NodeWeight}_i = \sum_i \text{POS Weight, Edge Count, Sentence Inclusion Weight} \quad (5.1) \]

**Algorithm 5: Constructing the Dependency Graph**

Result: Dependency graph of the passage
Install StanfordCoreNLP parser [80];
while there are more sentences in the passage do
Get the sentence \( i \) from the passage;
Run dependency parser on the sentence \( i \) with the option ‘depparse’;
Get tuples generated by the parser;
for each dependency tuple do
Get the pair of governorGloss and dependentGloss from the tuple;
if the POS of a word in the pair is Noun or Verb or Adjective or Adverb then
Feed the pair to the graph generator;
Add the POS tag of the words in the pair and the sentence index \( i \) to the respective newly added nodes;
Discard the words in the pair;

For the excerpt of 10 sentences shown in Figure 5.1, from a news article that appeared on BBC website [2], the dependency trees obtained from the Stanford parser are given in Figure 5.2, and the dependency graph constructed from the dependency trees is given in Figure 5.3. The dependency tree representation of the syntactic structure emphasizes the functional role of a word in a sentence [36].

1. The Pacific nation of Palau has become the first country to ban sun cream that is harmful to corals and sea life. 2. From Wednesday, sun cream that includes common ingredients, including oxybenzone, is not allowed to be worn or sold in the country. 3. Palau's President Tommy Remengesau said: "We have to live and respect the environment because the environment is the nest of life. 4. "The island nation markets itself as a "pristine paradise" for divers. 5. A lagoon in Palau's Rock Islands is a Unesco World Heritage site. 6. The country has a population of around 20,000 dotted across hundreds of islands. 7. The ban - which was announced in 2018 - prohibits sun cream containing any of 10 ingredients. 8. The list includes oxybenzone and octinoxate, which absorb ultraviolet light. 9. The International Coral Reef Foundation said the banned chemicals were "known environmental pollutants - most of them are... incredibly toxic to juvenile stages of many wildlife species". 10. Mr Remengesau told the AFP news agency: "When science tells us that a practice is damaging to coral reefs, to fish populations, or to the ocean itself, our people take note and our visitors do too.

**Figure 5.1** News article from BBC [https://www.bbc.com/news/world-asia-50963080](https://www.bbc.com/news/world-asia-50963080)
5.1.3 Obtaining the Summary from the Dependency Graph

From the dependency graph cluster shown in Figure 5.4, the strongly connected nodes are identified, and the intersection of sets of sentences that contain them are considered towards the summary. In the dependency graph cluster, the verbs are marked with blue bubbles and nouns with green bubbles. For each node with an edge count more than a threshold value, the node weight is calculated as a function of its part of speech tag weight, edge count, and sentence inclusion factor. A verb expresses the process [102], and a given verb attracts one or more actants to form a clause like how a given atom attracts other atoms to form a molecule [84]. So the verbs are given weights bigger than nouns. The weight $W$ is assigned in the order given in (5.2).

$$W(V.+)>>W(N.+)>>W(RB)>>W(JJ)$$ (5.2)

The obtained graph is traversed to collect the nodes with their weights crossing a threshold. From the lists associated with the highest weighing nodes, the intersection of the set of sentences involving a verb labeled node and a noun labeled node is selected as the summary. For the news article in figure 5.1, this approach produces sentences 1 and 7 as the summary “The Pacific nation of Palau has become the first country to ban sun cream that is harmful to corals and sea life. The ban - which was announced in 2018 - prohibits sun cream containing any of 10 ingredients.”.
Figure 5.3 Dependency graph for the news article

Figure 5.4 Strongly connected cluster in the dependency graph for the news article given in Figure 5.1
Summarization is a task which provides input to many other NLP applications. To measure the suitability of the generated summary for news articles, we compare the news headlines, which is a summary of the news article in most cases with the fullest noun nodes and fullest verb nodes existing in the dependency graph for that news article. The comparison is given in Table 5.1.

<table>
<thead>
<tr>
<th>#</th>
<th>News Headline</th>
<th>(Full nouns, #edges)</th>
<th>(Full verbs, #edges)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>China may send ducks to battle Pakistan’s locust swarms</td>
<td>(ducks,7), (locust,6)</td>
<td>(fight,5)</td>
</tr>
<tr>
<td>2</td>
<td>Parents in Japan and Hong Kong struggle as schools shut</td>
<td>(school,31)</td>
<td>(deny,12), (close,9)</td>
</tr>
<tr>
<td>3</td>
<td>Israel election: Netanyahu claims ‘biggest win’ amid vote count</td>
<td>(election,13), (party,11), (Netanyahu,8), (Gantz,8)</td>
<td>(win,18), (need,12)</td>
</tr>
</tbody>
</table>

Table 5.1 Headline, noun nodes and verb nodes with their degrees

5.1.4 Comparison with other Graph-based Summarization Methods

There are different graph-based summarization methods [92, 45, 75, 114], considering words, sentences, embeddings, or paragraphs as nodes and cue phrases, salience, or similarity measures as edges. The ranking strategies such as path scoring, salience scoring, and tree refinements are used in these graph-based summarization methods. Our summarization method uses words as nodes and dependency relations obtained from the dependency tuples of the parser as the edges. The ranking is based on the weight of the nodes, which is influenced by the degree of the node.

5.2 Syntax-Based Machine Reading Comprehension

Comprehending a news story depends profoundly on understanding “Who did What!” from the statements in that news article. This section explains an approach in answering questions using a machine reading comprehension (MRC) method, which leverages the syntax structure of the sentences to find the verb of action in the passage corresponding to the action in the question. This method generates correct answers for factoid questions on descriptive reading comprehension passages like news articles. The proposed method finds all the sentences in the passage, which has the same or synonymous verb as the verb in the question, processes the dependencies of the verbs obtained from the dependency parser and proceeds with further rule-based filtering for matching the other attributes of the answer span.

MRC is the basic task of textual question answering, in which each question is given a related context from which the answer should be inferred. MRC paradigm covers a broad set of categories such as (i) multi-passage MRC where the given context C is not a single passage, but a collection of
documents (ii) conversational machine reading comprehension (CMRC) where the conversation history also acts as part of the context to help with answer prediction and (iii) knowledge-based machine reading comprehension (KBMRC) in which additional related knowledge extracted from knowledge bases is necessary to answer the question [74].

This section presents a method for answering a natural language question from a single descriptive passage, in which the verb describing the action in the question is identified, and the passage is analyzed for candidate sentences having verbs synonymous with the verb in the question out of which the answer can be deduced. Some syntax-based comprehension methods like [107] replace the options from the multiple-choice answers in the question and compare it with the sentences in the context passage. Our approach does not need any answer options to compute the answer span but instead reads from the beginning to the end of the passage comparing the verbs and their dependencies with the verb in the free form natural language question. Machine comprehension systems are particularly suited to high-volume, rapidly changing information sources like news. The NewsQA dataset [104] is a curated set of news articles from CNN/DailyMail [29] with a crowd-sourced natural language question added for each news article. We use recent news articles from popular news websites with crowd-sourced questions similar to NewsQA, to evaluate our machine reader. The most effective way of understanding a passage is by answering multiple questions on the passage, and it requires domain knowledge [51]. News often contains the main verb in all the statements in the news article.

The human reader starts to comprehend by skimming the passage to get a general idea about the text, followed by scanning the passage to get some specific information [96]. We speculate that one of the approaches during scanning is attempting to identify the verb in the question and finding a similar verb in the passage. A question will often have a main verb in it. Our machine reader focuses on pruning the passage text based on the required action comparable to the one that is mentioned in the question.

5.2.1 Proposed Comprehension Method

We propose a method for answering free form natural language English questions on any random passage. Our approach begins with dependency parsing the question and the passage using Stanford-CoreNLP [80] to obtain the typed dependency relations existing in them. The question word is identified, and all the nouns and verbs in the question are filtered along with their corresponding dependencies.

Structural connections establish dependency relations between words [102]. The Stanford parser [80] converts the parse into a dependency tree. For every element that one has in the utterance at hand, there is exactly one node in the syntactic structure that corresponds to that element. One of the advantages of dependency parsers for NLP is that the parse can be easily encoded in a table. The output obtained from Stanford CoreNLP parser [80] consists of tuples like \{‘dep’: ‘nsubj’, ‘governor’: 6, ‘governorGloss’: ‘celebrated’, ‘dependent’: 5, ‘dependentGloss’: ‘Koch’\}¹ which mainly consist of a governorGloss, dependentGloss and a dependency relation existing between them. In the dependency tree obtained for

¹This is one of the tuples obtained for the sentence “Christina Koch celebrated with a thumbs up as she got out of the Soyuz capsule”
a sentence, there will be edges labeled with the dependency relation from the GovernorGlosses to the DependentGlosses.

The modules in our syntax-based MRC are shown in Figure 5.5. The **Question handler** does all the processing on the question utterance like POS tagging, verb and noun filtering, finding the dependency relation associated with verbs, and question word identification. The **Passage handler** processes the passage by preprocessing and separating the sentences, finding the dependency relations of verbs in it, getting the list of lemmatized verbs, and POS tagging the sentences in the passage. The **Answer handler** identifies candidate answers, which either matches the verb in the question or has a synonymous verb as that in the question or otherwise has the maximum text span matching with the question. The **Answer selector** module pinpoints the answer to the ‘Who’ question by iterating through the dependency tuples obtained from the dependency parser. For the named entities that could not be filtered from the candidate answer, the whole candidate sentence is output as the answer. The illustrated algorithm for processing a CNN news article from the NewsQA [104] and answering a natural language question about it is given in the following section.
5.2.2 Passage and Question Data for Illustration

The following news article passage is from the non-anonymized version of the CNN/Dailymail dataset [29].

WASHINGTON (CNN) – One of the Marines shown in a famous World War II photograph raising the U.S. flag on Iwo Jima was posthumously awarded a certificate of U.S. citizenship on Tuesday. The Marine Corps War Memorial in Virginia depicts Strank and five others raising a flag on Iwo Jima. Sgt. Michael Strank, who was born in Czechoslovakia and came to the United States when he was 3, derived U.S. citizenship ... on top of Mount Suribachi on February 23, 1945.”

The question asked is, “Who was born in Czechoslovakia?”

5.2.3 Algorithm for Answering ‘Who’ Questions

The algorithm is demonstrated with the passage and the question mentioned in the previous section in the following seven subsections.

5.2.3.1 Identify Question Word, Verbs and Nouns

The question word is identified from the question, and all the nouns and all the verbs in the question are filtered with their corresponding dependencies. The dependency parsed question is given in Figure 5.6.

POS tagged question: [('who', 'WP'), ('was', 'VBD'), ('born', 'VBN'), ('in', 'IN'), ('czechoslovakia', 'NNP'), ('?', '. ')]

5.2.3.2 Preprocessing the Question and the Passage

The question and the passage sentences are converted to lowercase and preprocessed by converting the verbs in the question to their lemmatized form. WordNet Lemmatizer in NLTK [77] is used for lemmatizing the verbs.

Question Word: Who
Nouns: ‘czechoslovakia’ : ‘NNP’
Verbs: [('be', 'VBD', 1), ('bear', 'VBN', 2)]

2CNNStories/ 644a3f79470d3b457efacc7d4ea33577d59e69c1 .story.

5.2.3.3 Find the Candidate Sentences by Matching the Verbs

The verbs in the question are compared with the verbs in the passage. If they match, then that sentence from the passage is added to the candidate sentence list. The auxiliary verbs such as ‘am’, ‘are’, ‘is’, ‘was’, ‘were’, ‘can’, ‘could’, ‘may’, ‘might’, ‘must’, ‘shall’, ‘should’, ‘will’, ‘would’, ‘do’, ‘does’, ‘did’, ‘has’, ‘have’ can frequently occur in most English passages and they can often overshadow the main verbs. So the auxiliary verbs are deprioritized while considering the passage to identify the presence of the verb from the question. The possible answer from the original passage is obtained as ‘sgt. michael strank, who was born in czechoslovakia and came to the united states when he was 3, derived u.s. citizenship when his father was naturalized in 1935.’

5.2.3.4 Find the Candidate Sentences if Synonymous Verbs Match

If a sentence contains a verb that is synonymous with the verb in the question, then that sentence is also added into the candidate sentence list. Thesaurus and Synonyms are generated using NLPCompromise packages [3]. The sense of the word given for finding synonyms is also given to avoid confusion between noun and verb senses. In the example considered here, the animal bear should not be considered for finding the synonyms of verb bears-bore-born in the question. If candidate sentences are not obtained on the exact verb match, then they are obtained from the synonymous verb match by replacing the question verbs with each verb in the synonym list and rechecking the passage for a match. For the verb ‘bear’ the thesaurus obtained are [‘buck’, ‘carry’, ‘convey’, ‘deliver’, ‘ferry’, ‘fetch’, ‘lug’, ‘move’, ‘pack’, ‘take’, ‘tote’, ‘transfer’, ‘transport’] and synonyms obtained are [‘bear’, ‘accept’, ‘assume’, ‘hold’]

Figure 5.7 Dependency parsed matching sentence from the passage

5.2.3.5 Select Sentences with the Highest Cosine Similarity and Longest Matching Span

Out of the candidate sentences collected, those with the longest matching span and the best cosine similarity are selected to be the answer sentence. For passages describing many actors doing the same action in different contexts, all the passage sentences with the action verb will be matching. The order
of their occurrence in the passage is not significant for deciding their candidature for being the answer to
the question. So, in that case, the cosine similarity of the matching passage sentence with the question
is evaluated for making the best choice. The longest matching span is also evaluated for reinforcing the
decision.

5.2.3.6 Find the Candidate Sentences with Matching Predicates

If there is no matching verb in the passage, then the predicate of the question is matched with the
sentences in the passage, to get the answer. The sentences with the maximum cosine similarity to the
question are considered as a candidate answer. The nouns in the best-matched sentence are compared
with the nouns in question to get the most suitable candidate sentence as the answer sentence.

5.2.3.7 Answering ‘Who’ from the Answer Sentence

Once the answer sentence is identified, the answer to the ‘Who’ question is found out by iterating
through the dependency relation tuples obtained from the dependency parser and matching ‘governor-
Gloss’ of the dependency tuple with the verb in the question and the having ‘dep’ as ‘nSubj’ or ‘amod’.
The sentence with the verbs converted to their lemmatized form is used to identify ‘Who’. If the answer
is a noun compound, the chain of nouns in the answer is obtained by following the dependency relation
‘compound’ until it reaches the last modifier from the head noun. A possible answer span with stemmed
verbs is ‘sgt. michael strank , who be bear in czechoslovakia and come to the united states when he
be 3 , derive u.s. citizenship when his father be naturalized in 1935’. Figure 5.7 shows a part of the
dependency tree for that span and the answer is obtained as ‘sgt. michael strank’.

5.2.3.8 Answering ‘How’ and ‘What’ from the Answer Sentence

To find the answer to ‘How’ questions, ‘dependentGloss’ of dependencies ‘advmod’, ‘amod’ are
found out. The matching verb’s dependent ‘dependentGloss’ with a dependency relation ‘dobj’ is found
out for answering ‘What’.

5.2.4 Results and Evaluation

This approach of utilizing the syntax structure and matching the verbs is efficient in the cases where
the passage is action-oriented like news articles. However, this method is not efficient in identifying
inferences and causal relations. The efficiency of our comprehension method depends on the accuracy
of the parser used, and paraphrases and coreferences will not be resolved by the current implementation.
This approach shows lesser efficiency when a chain of verbs is embedded in the sentence. Comparing
to the deep-learning approaches in MRC requiring long passages as training data, this approach using
syntax obtains answers with fewer data and answers many questions effectively without learning the
meaning. The question answering of a ‘Who’ question from an excerpt from ‘Dracula’ taken from
<table>
<thead>
<tr>
<th>Data Source</th>
<th>#Documents considered</th>
<th>#Questions</th>
<th>Genre</th>
<th>EM p@1</th>
<th>p@3</th>
<th>Hit@3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORD-19</td>
<td>2</td>
<td>14</td>
<td>Scholarly Articles Crowd sourced questions on news articles</td>
<td>- 25%</td>
<td>0.64</td>
<td>0.71</td>
</tr>
<tr>
<td>NewsQA</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
<td>0.5</td>
<td>0.75</td>
</tr>
</tbody>
</table>

Table 5.2 Evaluation results of our machine reader

Gutenberg [100] is shown in Figure 5.8. For evaluating our system, we use some extractive metrics mentioned in a survey on machine reading comprehension systems [16]. The metrics applied are (i) Exact Match (EM) or accuracy - the percentage of answers that exactly match with the correct answers (ii) Precision@K - the number of correct answers in the first K returned answers without considering the position of these correct answers (iii) Hit@K - count of the number of samples where their first K returned answers include the correct answer. The evaluation results of our reader on a few samples from NewsQA [104] and CORD-19 [4] are given in Table 5.2.

The answers computed for questions from the computed answer candidate sentences from multiple passages are given in Table 5.3.

Finding out the answer span is the first step in reading comprehension, which is attained in this work. Even when the verb in the question is not present in the passage, the matching span congruence and cosine similarity identifies the best candidate answer if the question is answerable. In the future, we will be working on incorporating paraphrasing, inferences, and sentence reduction techniques.

The comparison of answers obtained from our reader with two other readers deployed in the web (i) from Microsoft Research Montreal ³ and (ii) a demo on fine-tuning ALBERT [69] for the task of question answering ⁴, for five random questions for the same context passage “Canada: A Brief Overview Geography And Climate” ⁵ is given in Table 5.4. For the third question, our reader does not identify the answer because the POS obtained by the parser for ‘border’ is ‘NNS’ instead of ‘VBP’. We hope that the accuracy of POS tagging would increase with the state-of-the-art performance parser Stanza [91] from Stanford.

5.2.4.1 Comparison with the other Syntax-based Systems

Syntax-based question answering systems [107, 97, 90, 73] existed even in the pre-ML era [57]. Some methods [97] use syntactic relation pattern matching by scoring hypernyms and hyponyms. Methods [90] using tree-matching find the answer by evaluating the distance between a question and each of

⁴https://littlealbert.now.sh//
⁵https://www.canada.ca/content/dam/ircc/migration/ircc/english/pdf/pub/welcome.pdf
5.3 Conclusion

After analyzing the results obtained from the experiments, we conclude that the dependency trees obtained from Stanford parser \[80\] can be used for various language-processing tasks such as document summarization and machine reading comprehension. This summarization method can give results in low-resource settings due to its low requirement on training data. As this is a syntax-based approach, it can be applied to other languages also. This method can also work in multi-document summarization.
because the occurrence of the same content word in multiple documents can connect the document graphs of those documents to form a single graph representation.

The low-resource approach used in reading comprehension does not require extensive training data for building the final system, as is the case with many DNN-based systems today. However, the tools used to analyze the questions and the data are trained in a supervised way, and the accuracy of the tools impacts our reader’s efficiency.
<table>
<thead>
<tr>
<th>Context</th>
<th>Passage Sentence</th>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>News article</td>
<td>Earlier this month, the Asma Jahangir Foundation organised the second Asma Jahangir Conference in Lahore to commemorate the human rights lawyer who died on February 11, 2018.</td>
<td>Who organised the conference in Lahore?</td>
<td>Asma Jahangir Foundation</td>
</tr>
<tr>
<td>News article</td>
<td>Mao Zedong, founder of the People’s Republic of China (PRC), reportedly called Australia the “lonely continent”.</td>
<td>Who is the founder of PRC?</td>
<td>-</td>
</tr>
<tr>
<td>CORD-19</td>
<td>UID: 41jqgsv0</td>
<td>What are the precautions needed while using hydroxychloroquine as a drug?</td>
<td>caution and contraindication with chloroquine and hydroxychloroquine expectedly, some precautions will be need while use both these drugs that include frequent monitoring of hematological parameters (rbc, wbc and platelet counts), measurement of serum electrolytes, blood glucose (because of hypoglycemic potential of hcq) and hepatic as well as renal functions.</td>
</tr>
<tr>
<td>Dracula - Project Gutenberg</td>
<td>The women looked pretty, except when you got near them, but they were very clumsy about the waist.</td>
<td>How did the women of budapest look like?</td>
<td>pretty</td>
</tr>
<tr>
<td>Baba Yaga - Project Gutenberg</td>
<td>when baba yaga moved away from the window, the little girl gave some ham to the cat and asked her whether there was any escape.</td>
<td>What did the girl give?</td>
<td>ham</td>
</tr>
<tr>
<td>NewsQA</td>
<td>7843646a31a1ca87834dc4c72a092b36c693baff</td>
<td>when does paul die?</td>
<td>les paul die thursday of pneumonia.</td>
</tr>
<tr>
<td>CORD-19</td>
<td>UID: cpu3q9o6</td>
<td>What is the recommended dose?</td>
<td>retinopathy be a dose-limiting adverse effect of hydroxychloroquine, and a safe daily dose appear to correspond to 6.5 mg/kg of ideal body weight and 5.0 mg/kg of actual body weight [8].</td>
</tr>
</tbody>
</table>

Table 5.3 Computed answer span from passages, questions and computed answers
<table>
<thead>
<tr>
<th>Question</th>
<th>Answer from MSR Montreal Reader Demo</th>
<th>Answer from Little Albert Reader</th>
<th>Answer from Our Reader</th>
</tr>
</thead>
<tbody>
<tr>
<td>How is the weather in Canada?</td>
<td>Welcome to Canada — Canada: A brief overview</td>
<td>warm to hot</td>
<td>like Canada’s landscapes, the climate vary across the country.</td>
</tr>
<tr>
<td>Which is the second largest country on earth?</td>
<td>Welcome to Canada — Canada: A brief overview</td>
<td>Canada</td>
<td>Canada be the second largest country on earth, cover an area of 10 million square kilometres (3.9 million square miles).</td>
</tr>
<tr>
<td>How many oceans border Canada?</td>
<td>Three oceans border Canada: the Pacific Ocean in the west, the Atlantic Ocean in the east, and the Arctic Ocean to the north.</td>
<td>Three</td>
<td>-</td>
</tr>
<tr>
<td>What should you buy to survive Canadian winter?</td>
<td>Be sure to buy a winter coat, boots, gloves and hat to keep you warm. With the right clothing, you will be prepare...</td>
<td>winter coat, boots, gloves and hat</td>
<td>be sure to buy a winter coat, boots, gloves and hat to keep you warm.</td>
</tr>
<tr>
<td>How much coastline does Canada have?</td>
<td>Altogether, Canada has over 200,000 kilometres (125,000 miles) of coastline.</td>
<td>over 200,000 kilometres (125,000 miles)</td>
<td>altogether, Canada have over 200,000 kilometres (125,000 miles) of coastline.</td>
</tr>
</tbody>
</table>

**Table 5.4** Comparison of answers from our reader with those from other deployed readers for the same context passage
Chapter 6

Conclusion and Future Work

In this thesis, we presented rule-based approaches leveraging the dependency structure for generating inferences from utterances, summarizing documents, and answering free form natural language questions from passages.

We attempted to generate inferences based on a set of rules, formulated based on certain grammatical relations present in the utterance. This approach of applying logic on the syntactic structure to generate inferences stands different from alternative approaches using deep learning techniques because of the lesser data, time, and compute requirement. A set of seventeen presupposition triggers for annotating news headlines is formulated based on the triggers mentioned by Levinson [72] and explained with examples in section 3.5 of Chapter 3. As the framework is already made to manipulate the tuples obtained from dependency parsing, the set of rules can be elaborated with more triggers and complex rules.

A dataset was collected and annotated containing dialogues with implicatures associated with the response utterance. The collected dataset can be accessed from the figshare repository [47]. The data is useful as a reference for identifying and synthesizing conversational implicatures. The dialogues are collected from 74 listening comprehension short conversation practice sections of the TOEFL English proficiency test and 45 movie scripts of the animation genre. Chapter 4 outlines the common kinds of conversational implicatures and the sources, methods, and challenges of crowd-sourcing the dialogue collection and annotation for implicatures. As implicatures are generated in a wide range of situations and are highly dependent on the hearer’s understanding, we have primarily focused on the polar questions where an indirect answer without an explicit ‘Yes’ or ‘No’ generates implicatures. In the future, we are planning to add more contexts along with the polar question context considered in this work and annotate the identified dialogues with implicatures. In our future work, scalar implicatures which can be identified with the comparison keywords [72] such as <all, most, many, some, few>, <always, often, sometimes>, <must, should, may> would be focused in particular using a similarity-Judgement method like that proposed by Degen [37]. Scalar implicatures are easy to isolate and notice, and a lot of research on implicature such as [40] is focused on those implicatures. Extra context features where
context/i refers to the i\textsuperscript{th} most recent additional context would also be considered where the utterance gets its meaning from multiple-context-Utterances going back in time during a conversation.

We summarized documents by constructing the dependency graph representing the document. We define the document dependency graph as an interconnected collection of dependency trees representing the sentences in the document. The obtained summary is extractive, and the summary could be used for feeding as input to other NLP tasks. The advantage of this summarization method is the smooth construction of the graph representation for any document from the tuples obtained from the parser. Simplification of sentences can be attempted to obtain summaries nearing the gold standard. We also wish to work with the recently released state-of-the-art fully neural NLP toolkit Stanza [91] from Stanford, which supports 66 languages. We believe that sentence reductions would be easier to implement on the tree structure of the sentences received as tuples from the parser.

We simulated the human approach of reading comprehension in this research, where the verb in the question is identified, and a sentence from the passage with a similar action as the verb in the question is identified as a candidate answer sentence. This method accurately identifies the answer span in many cases, even though the actual comprehension of the meaning is not happening. But we find it useful in processing action-oriented genres like news where many statements contain a main verb. We wish to use this as a baseline for obtaining answer spans and filter those spans to select the correct answer span by incorporating deep learning methods.
Related Publications

1. Elizabeth Jasmi George.

Verb Focused Answering from CORD-19, 23rd International Conference on Text, Speech and Dialogue (TSD), 2020.

2. Elizabeth Jasmi George.

Quiver to Target: Document Summarization using Dependency Graphs, 7th International Conference on Artificial Intelligence and Applications (AIAP), 2020.

3. Elizabeth Jasmi George, Radhika Mamidi.

Conversational implicatures in English dialogue: Annotated dataset, 8th Symposium on Natural Language Processing (NLP), co-affiliated with the Third International Conference on Computing and Network Communications (CoCoNet'19), 2019.


(http://www.sciencedirect.com/science/article/pii/S1877050920312436)

4. Elizabeth Jasmi George, Radhika Mamidi.

Towards Computing Inferences from English News Headlines, 16th International Conference of the Pacific Association for Computational Linguistics (PACLING), 2019.


https://doi.org/10.1007/978-981-15-6168-9_36
Bibliography


[63] T. Khot, A. Sabharwal, and P. Clark. In AAAI.


