Resolution of Pronominal Anaphora for Telugu Dialogues

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

Master of Science in Computer Science and Engineering by Research

by

HEMANTH REDDY JONNALAGADDA
201002100
HEMANTH.REDDY@RESEARCH.IIIT.AC.IN

International Institute of Information Technology Hyderabad
(Deemed to be University)
Hyderabad - 500 032, INDIA
JULY 2024

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled ’Resolution of Pronominal Anaphora for Telugu Dialogues’ by HEMANTH REDDY JONNALAGADDA, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Prof. RADHIKA MAMIDI

To my family Here and Above

Acknowledgments

I would like to express my deepest gratitude to my guide, Radhika Mamidi Ma'am, whose expertise and guidance introduced me to the fascinating fields of Natural Language Processing (NLP) and anaphora resolution. Your invaluable support, insightful discussions, and constant encouragement have been instrumental in the successful completion of this thesis. I am truly grateful for the opportunity to learn and grow under your mentorship.

I am profoundly thankful to my parents, my wife, and my sister for their unwavering support, encouragement, and understanding throughout this journey. Your love, patience, and belief in me have been my greatest source of strength, providing the motivation I needed to persevere through challenges and achieve my goals.

My heartfelt thanks to my friends at IIIT Hyderabad for their camaraderie, inspiration, and constant motivation. Your friendship and support have made this journey not only academically enriching but also enjoyable and memorable. I cherish the moments we have shared and the collective spirit that has driven us forward.

I would also like to extend my gratitude to Srikanth, Sravanthi, Saikrishna, and Prasanth. Working with you on various projects, especially in building QA systems, has been a rewarding and enlightening experience. Your collaboration, dedication, and innovative ideas have significantly contributed to the success of my endeavors, and I have learned immensely from our joint efforts.

Thank you all for your invaluable support and encouragement. Without you, this journey would not have been possible.

Abstract

The challenge of anaphora resolution has been a prominent research topic within the field of natural language processing (NLP) for many years. Anaphora refers to the linguistic phenomenon where a word or phrase refers back to another word or phrase previously mentioned in the discourse. Resolving these references accurately is crucial for understanding and generating coherent text. While significant progress has been made in anaphora resolution for several languages, there is a notable gap in research focusing specifically on dialogues in Telugu, a Dravidian language spoken primarily in India.

Telugu presents unique challenges for anaphora resolution due to its rich morphology and free word order. These characteristics complicate the identification of antecedents for pronominal references, which is essential for accurate language understanding. This thesis addresses these challenges by presenting a rule-based algorithm designed for pronominal anaphora resolution in Telugu human-to-human conversations.

The proposed algorithm consists of two main components: the creation of a comprehensive knowledge base and the development of a set of rules for resolving anaphora. The knowledge base includes an extensive list of Telugu pronouns along with their morphological information, which is crucial for understanding the grammatical relationships within sentences. The rule-based component leverages this knowledge base to identify and resolve pronominal references accurately.

A significant contribution of this research is the development of a new annotated corpus for Telugu dialogues. Due to the lack of existing resources, a new corpus was built, comprising 108 human-to-human conversations with a total of 509 pronouns. The corpus was annotated manually to ensure high-quality data for evaluating the algorithm.

The performance of the algorithm was tested on this corpus, and the results demonstrate its effectiveness. The algorithm achieved an overall accuracy of 61.1%, with the highest accuracy observed for 1st person pronouns (81.88%) and the lowest for 3rd person pronouns (43.56%). These results highlight the complexities involved in resolving 3rd person pronouns in Telugu dialogues and suggest areas for further improvement.

In conjunction with this research, we initially developed an interactive question answering system for the Hyderabad Multi-Modal Transport System (MMTS). Existing applications for querying train arrival times often rely on dropdown menus for selecting 'FROM Station' and 'TO Station,' which then display a comprehensive list of train schedules. These systems fall short in catering to users seeking specific time periods and additional contextual information. Our project addressed these limitations by creating a user-friendly interface that leverages advanced natural language processing techniques to interpret and respond to user queries. The system employs a frame-based approach, coupled with robust parsing and spell-checking mechanisms, to accurately handle user inputs, including those with spelling errors and code-mixed language. This innovative solution not only showcases our technical expertise in dialogue systems but also significantly enhances the commuter experience by providing precise and relevant train schedule information for Hyderabad MMTS users.

This work represents a significant step towards improving natural language processing for Telugu, with potential applications in dialogue systems, text summarization, machine translation, and other NLP tasks. The findings of this research contribute to the broader field of anaphora resolution and provide a foundation for future studies focusing on Telugu and other morphologically rich languages.

Contents

Contents

1 Introduction 11

1.1 Understanding Anaphora Resolution 11
1.1.1 Coreference 11
1.1.2 Types of Coreference 12
1.1.3 Types of Anaphoric Forms 12
1.1.4 Anaphora and Antecedent 12
1.1.5 Anaphora Resolution 13
1.1.6 Cataphoric References 13
1.1.7 Applications of Anaphora Resolution 13

1.2 Understanding Question Answering System 14
1.2.1 Evolution of Question Answering Systems 14
1.2.2 Architecture of Question Answering Systems 14
1.2.3 Applications of Question Answering Systems 15

1.3 Challenges and Literature survey 16
1.3.1 Challenges in Anaphora Resolution 16
1.3.2 Literature Survey on Anaphora Resolution 16
1.3.3 Telugu-Specific Studies 17

1.4 Contribution of this thesis 17

1.5 Thesis organization 17

2 Related Work 19

2.1 A Brief History of Question Answering Systems 19
2.1.1 Early Beginnings (1960s - 1980s) 19
2.1.2 The Advent of Information Retrieval and NLP (1980s - 1990s) 19
2.1.3 Rise of Statistical Models (1990s - 2000s) 19
2.1.4 Machine Learning and Deep Learning Era (2000s - Present) 20

2.2 A Brief History of Anaphora Resolution 20
2.2.1 Early Foundations (1960s-1980s) 20
2.2.2 Statistical and Machine Learning Methods (1990s-2000s) 20
2.2.3 Advancements with NLP and Deep Learning (2010s-Present) 20

2.2.4 Multilingual and Cross-Lingual Approaches 21

2.3 Telugu language 21
2.3.1 Telugu Language Overview 21
2.3.2 Linguistic Characteristics of Telugu 21
2.3.3 Similarities to Other Regional Languages 21
2.3.4 Dravidian Languages 22
2.3.5 Cultural and Historical Context 22
2.3.6 Overview of NLP in Regional Languages 22
2.3.7 Specific Work in Telugu NLP 23
2.3.8 Anaphora Resolution in Regional Languages 23
2.3.9 Key Challenges 23

3 An Interactive Question Answering System for Hyderabad MMTS 25

3.1 Creation of the Database 27

3.2 Our Approach 27
3.2.1 The Flow of the Process 28
3.2.2 Spell Checker and Knowledge base 28
3.2.3 Parsing 29
3.2.4 Frame Selection and Filling 30
3.2.5 Handling the adjectives and prepositions 32
3.2.6 SQL Query Generation 33

3.3 System Response and System Request Generation 34

3.4 Limitations and Future Work 35

3.5 Conclusion 36

4 Resolution of Pronominal Anaphora for Telugu Dialogues 37

4.1 Anaphora in the context of the Telugu language 37
4.1.1 Sandhi in Telugu 37
4.1.2 Verb Inflection in Telugu 37
4.1.3 Implications for NLP in Telugu 38

4.2 Pronouns in Telugu 38
4.2.1 Personal Pronouns 38
4.2.2 Non-Personal Pronouns 39
4.2.3 Reflexive Pronouns 39
4.2.4 Demonstrative Pronouns 39
4.2.5 Interrogative Pronouns 39

4.3 Types of Anaphora in Telugu Dialogues 40

4.3.1 Normal Pronominal Anaphora 40
4.3.2 Discourse Deictic Anaphora 41
4.3.3 One-Anaphora 41

4.4 Approach 42
4.4.1 Parsing 42
4.4.2 Knowledge Base 43
4.4.3 Algorithm 43
4.4.3.1 Personal Pronouns 44
4.4.3.2 Non-Personal Pronouns 45

4.5 Building the Corpus 46

4.6 Results 46

4.7 Issues in Telugu 47
4.7.1 Corpus Issues 47
4.7.2 Sandhi (Compound Words) 48
4.7.3 Subject Identification 48
4.7.4 Parser Errors 48
4.7.5 Summary 49

Conclusions and Future Scope 50
Future Scope 50

Related Publications 51

Bibliography 52

Chapter 1

1 Introduction

Natural Language Processing (NLP) is a subfield of artificial intelligence that focuses on enabling computers to understand, interpret, and generate human language in a way that is both meaningful and useful. The goal of NLP is to bridge the gap between human communication and computer understanding. This involves a wide range of tasks such as machine translation, sentiment analysis, speech recognition, text summarization, and more. NLP combines computational linguistics, machine learning, and deep learning techniques to process and analyze large amounts of natural language data.

One of the critical challenges in NLP is resolving references within a text, known as anaphora resolution. Anaphora resolution is the process of identifying the antecedent (the entity being referred to) for an anaphoric expression (the referring expression) within a discourse. This task is essential for maintaining coherence and understanding the meaning of the text, as it ensures that references to previously mentioned entities are correctly interpreted. Accurate anaphora resolution is crucial for various NLP applications, such as machine translation, information extraction, and dialogue systems, as it helps in maintaining the context and continuity of the discourse.

We built a question-answering system in English for Hyderabad MMTS (Multi-Modal Transport System). This foundational step involved developing algorithms and methodologies to accurately interpret and respond to user queries related to Hyderabad's local train services. Our focus was on ensuring the system could provide timely and precise information about train schedules, routes, and other related details.

Once we had a solid grasp of the question-answering system, we proceeded to the second phase, which focused on anaphora resolution. Anaphora resolution is crucial for maintaining coherence and understanding in dialogues, as it involves correctly identifying and linking pronouns to their respective antecedents. We developed and implemented techniques specifically tailored for Telugu, addressing the unique linguistic challenges posed by the language. By integrating these two components, we aimed to create a seamless and effective system capable of understanding and answering questions with contextual awareness and accuracy.

1.1 Understanding Anaphora Resolution

1.1.1 Coreference

Coreference occurs when two or more expressions in a text refer to the same person or thing. It is a fundamental concept in understanding natural language, as it helps maintain coherence and continuity in a dialogue or text. For example:

- **Example:** "Computational Linguists from many different countries attended the tutorial. The participants found it hard to cope with the speed of the presentation; nevertheless, they managed to take extensive notes."
    - In this example, "Computational Linguists" and "The participants" are coreferent, meaning they refer to the same group of people.

1.1.2 Types of Coreference

Exophora: Exophoric references require the reader or listener to use contextual or situational knowledge beyond the text to understand the referent. This type of reference is common in spoken language and real-time interactions where visual or situational cues play a critical role. For instance, if someone points and says, "Look at that," understanding what "that" refers to necessitates knowing what the speaker is pointing at or indicating.

Endophora: Endophoric references are those where the referent is within the text itself, making them more straightforward for text-based analysis. They ensure cohesion in writing by linking different parts of the text together. Endophoric references can be subdivided into anaphoric and cataphoric references, both of which are essential for maintaining narrative flow and coherence.

- **Anaphoric:** Anaphoric references point backward to something mentioned earlier in the text. They help avoid redundancy by allowing writers to replace previously mentioned nouns with pronouns or other substitute terms. This form of reference is crucial for maintaining the flow of information and ensuring that texts are not overly repetitive.
- **Cataphoric:** Cataphoric references point forward to something that will be mentioned later in the text. This type of reference can create suspense or emphasize a particular element in a narrative. It often requires the reader to hold the reference in memory until the antecedent is introduced, adding a layer of complexity to text comprehension.

1.1.3 Types of Anaphoric Forms

Repeated Form: Using the same expression again maintains clarity by avoiding ambiguity, especially in complex texts or when the referent is a key subject. Repetition ensures that the reader clearly understands who or what is being referred to, which can be particularly important in technical or academic writing.

Partially Repeated Form: This form maintains some clarity by repeating part of the original expression while varying other elements, such as titles or names. It strikes a balance between repetition and variety, ensuring that the referent remains clear while also avoiding monotonous repetition.

Lexical Replacement: Using a different word or phrase to refer to the same entity can add variety to the text and can also introduce nuances or additional information about the referent. Lexical replacements are often used to enrich the narrative and provide a more dynamic reading experience.

Pronominal Form: Pronouns are commonly used for efficiency and flow in writing, reducing the need for repeated noun phrases. They rely on the reader's ability to track referents through the text, which can enhance readability and make sentences less cumbersome.

Substituted Form: Substitute terms like "one" are useful for avoiding repetition while maintaining clarity, especially in comparative contexts. This form helps streamline sentences and can make descriptions more concise and easier to follow.

Ellided Form: Ellipsis, or the omission of part of the reference, relies on the reader's contextual understanding to fill in the gaps. This form can make texts more succinct and can also add a conversational or informal tone to writing.

1.1.4 Anaphora and Antecedent

Understanding the relationship between anaphora and antecedent is critical for discourse comprehension. Anaphora resolution is essential in natural language processing tasks such as machine translation and information extraction, where accurate interpretation of pronouns and referring expressions ensures coherence and continuity. For example, in "John went home because he was sick," correctly identifying that "he" refers to "John" is crucial for understanding the sentence.

1.1.5 Anaphora Resolution

Constraints on Anaphora: Constraints like number, gender, and person agreement help in correctly identifying the antecedent, ensuring the coherence and grammaticality of the text. Syntactic constraints, such as the syntactic role of the antecedent and anaphoric expression, also play a crucial role in resolution. For example, "John bought himself a new car" correctly matches "himself" to "John" based on syntactic and semantic agreement.

Preferences in Pronoun Interpretation: Preferences such as recency and grammatical role influence how pronouns are interpreted. These preferences guide the reader or listener in linking pronouns to their most likely antecedents, thereby facilitating smoother comprehension. For instance, the recency preference would more likely link "it" to "Scoda" rather than "Maruti" in the example provided.

1.1.6 Cataphoric References

Cataphoric references add complexity to discourse analysis as they require the reader to hold the reference in memory until the antecedent is introduced. They are often used stylistically in literature and speeches to create suspense or emphasize a particular element. Understanding cataphoric references is important for accurately parsing and interpreting texts, especially in natural language processing tasks.

1.1.7 Applications of Anaphora Resolution

Machine Translation: Accurate anaphora resolution ensures that pronouns and referring expressions are correctly translated into the target language, maintaining coherence and meaning in the translated text. Without proper resolution, translations can become ambiguous or misleading, affecting the overall quality of the translated content.

Information Extraction: Identifying and resolving anaphora helps in extracting precise information from texts, such as identifying entities, relationships, and events. This is crucial for applications like building knowledge graphs or extracting structured data from unstructured texts.

Dialogue Systems: In conversational agents and chatbots, resolving anaphora allows for maintaining context and providing relevant and coherent responses. This is essential for creating natural and engaging interactions in dialogue systems.

Text Summarization: Anaphora resolution helps in creating concise and coherent summaries by accurately linking references to their antecedents. This ensures that the summary accurately reflects the original text's meaning and maintains its coherence.

Question Answering: In question-answering systems, resolving anaphora enables the system to understand and retrieve the correct information related to the antecedent mentioned in the question. This improves the accuracy and relevance of the answers provided by the system.

Sentiment Analysis: Accurate resolution of anaphora is essential in sentiment analysis to correctly associate sentiments with the appropriate entities or subjects in the text. This helps in understanding the sentiment expressed towards specific entities and improves the overall accuracy of sentiment analysis.

1.2 Understanding Question Answering System

Question Answering (QA) systems are designed to automatically provide accurate and concise answers to questions posed by users in natural language. These systems leverage advancements in Natural Language Processing (NLP) and Artificial Intelligence (AI) to understand and process human language, transforming it into structured data that can be queried. QA systems are pivotal in various domains, including customer support, educational platforms, virtual assistants, and information retrieval systems.

1.2.1 Evolution of Question Answering Systems

The journey of QA systems began with simple keyword-based searches and has evolved into complex AI-driven models capable of understanding context, semantics, and user intent. Early systems were limited to specific domains with predefined databases, but modern QA systems employ sophisticated algorithms and vast datasets to handle a wide array of questions across various fields.

The introduction of machine learning and deep learning techniques has significantly enhanced the capabilities of QA systems. These techniques enable the systems to learn from vast amounts of data, improving their accuracy and ability to understand context over time. Pre-trained language models, such as BERT, GPT-3, and their successors, have further revolutionized the field by providing powerful tools for semantic understanding and context-aware responses.

1.2.2 Architecture of Question Answering Systems

The architecture of a Question Answering (QA) system involves several key components, each responsible for specific tasks to process and deliver accurate answers. The system's architecture can be broadly divided into the following stages:

1. **Input Handling:**
    - **User Interface:** The front-end interface where users input their questions. This can be a text box on a web page, a voice input on a virtual assistant, or an API endpoint for other applications.
    - **Preprocessing:** The process of cleaning and normalizing the input text. This includes tokenization, stemming, and removing stop words to prepare the question for further analysis.
2. **Question Processing:**
    - **Question Classification:** Identifying the type of question (e.g., factoid, definition, list) to determine the appropriate answering strategy.
    - **Named Entity Recognition (NER):** Extracting key entities from the question to understand what the user is asking about.
    - **Dependency Parsing:** Analyzing the grammatical structure of the question to understand relationships between words and identify the focus of the question.
3. **Information Retrieval:**
    - **Document Retrieval:** Using search engines or databases to retrieve relevant documents or data sources that may contain the answer.

- **Passage Retrieval:** Narrowing down the retrieved documents to specific passages or sections that are most likely to contain the answer.

4. **Answer Processing:**
   - **Answer Extraction:** Extracting potential answers from the relevant passages using techniques like keyword matching, semantic similarity, and context analysis.
   - **Answer Ranking:** Evaluating and ranking the extracted answers based on relevance, accuracy, and confidence scores.
5. **Post-processing:**
   - **Answer Formulation:** Formatting the selected answer into a coherent response. This may involve natural language generation to ensure the answer is grammatically correct and contextually appropriate.
   - **Response Delivery:** Presenting the final answer to the user through the interface. This can be in the form of text, voice, or a structured data output.
6. **Feedback Loop:**
   - **User Feedback:** Collecting feedback from users to improve the system's performance over time. This feedback can be used to refine the models and algorithms used in the QA system.
   - **Performance Monitoring:** Continuously monitoring the system's performance using metrics such as precision, recall, and user satisfaction scores.

1.2.3 Applications of Question Answering Systems

QA systems have diverse applications across various sectors, enhancing user experience and efficiency. Some notable applications include:

1. **Customer Support:**
   - QA systems are widely used in customer service to provide instant answers to common queries, reducing the workload on human agents and improving response times.
     - **Example:** A user asks, "How can I reset my password?" and the QA system provides detailed instructions.
2. **Virtual Assistants:**
   - Virtual assistants like Siri, Alexa, and Google Assistant rely heavily on QA systems to understand and respond to user commands, perform tasks, and provide information.
     - **Example:** Asking Siri, "What's the weather like today?" and receiving a detailed weather report.
3. **Educational Platforms:**
   - In education, QA systems assist students by answering questions related to coursework, providing explanations, and offering tutoring services.
     - **Example:** A student asks, "What is the Pythagorean theorem?" and the system explains the theorem and its applications.
4. **Healthcare:**
   - QA systems help in healthcare by providing information on medical conditions, treatment options, and general health advice. They can assist both patients and healthcare professionals in making informed decisions.
     - **Example:** A user asks, "What are the symptoms of diabetes?" and the system lists common symptoms and advises consulting a healthcare professional.
5. **Information Retrieval:**
   - QA systems enhance information retrieval processes in search engines, digital libraries, and knowledge bases by delivering precise answers instead of a list of documents.
     - **Example:** Asking an academic database, "Who proposed the theory of relativity?" and getting the answer "Albert Einstein" along with relevant sources.

6. **E-commerce:**
    - In e-commerce, QA systems improve the shopping experience by answering product-related queries, offering recommendations, and assisting with purchase decisions.
        - **Example:** A user asks, "What are the best smartphones under $500?" and the system provides a list of recommendations with details and reviews.

1.3 Challenges and Literature survey

1.3.1 Challenges in Anaphora Resolution

Anaphora resolution faces significant challenges due to the inherent ambiguity of natural language. One primary difficulty is **pronoun ambiguity**, where identifying the correct antecedent for pronouns such as "he," "she," "it," and "they" is complex, especially when multiple potential antecedents exist. In languages like Telugu, the problem is compounded by **zero anaphora**, where subjects or objects are often omitted because they are implied by the context, making the resolution even more challenging.

Another challenge is dealing with **complex sentences**. These sentences often contain multiple clauses, where the antecedent might be distant from the anaphor. This complexity requires robust mechanisms to maintain and resolve **coreference chains** across sentences and paragraphs, ensuring coherence in longer texts. Additionally, **contextual understanding** is crucial; resolving anaphora correctly often demands a deep understanding of the context and sometimes world knowledge.

Cross-linguistic differences further complicate the task. Different languages have unique syntactic and semantic structures, necessitating tailored anaphora resolution strategies. From a machine learning perspective, challenges include the lack of large, annotated corpora in many languages, including Telugu, which hampers the development of effective models. Extracting the right features for these models and applying **transfer learning** from one language or domain to another also presents significant hurdles.

1.3.2 Literature Survey on Anaphora Resolution

Early approaches to anaphora resolution were primarily syntactic and rule-based. For instance, Hobbs (1978) introduced methods for resolving pronouns, while Lappin and Leass (1994) developed a comprehensive rule-based approach. These foundational works paved the way for more advanced techniques.

The advent of machine learning brought new methods. Soon, Ng, and Lim (2001) proposed using decision trees for coreference resolution, a significant shift from purely rule-based methods. Ng and Cardie (2002) enhanced this approach by refining feature sets, demonstrating the potential of machine learning in this field. The evolution continued with the application of neural networks. Clark and Manning (2016) utilized deep reinforcement learning for mention-ranking coreference models, and Lee et al. (2017) introduced an end-to-end neural network approach, marking a substantial leap forward in anaphora resolution capabilities.

Addressing multilingual and low-resource challenges, Rahman and Ng (2012) integrated world knowledge into coreference resolution models, while Kumar, Agarwal, and Kumar (2020) focused on Indian languages, including Telugu. These studies highlighted the importance of incorporating external knowledge and adapting methods to specific linguistic contexts.

Evaluation and datasets have also played a crucial role in advancing anaphora resolution. The CoNLL-2012 Shared Task, as detailed by Pradhan et al. (2012), provided a benchmark dataset and evaluation metrics, facilitating standardized comparisons and improvements in the field.

1.3.3 Telugu-Specific Studies

For Telugu, specific studies have adapted general methods to address language-specific challenges. Sivasankaran and Sobha (2009) explored anaphora resolution in Tamil using machine learning, providing insights applicable to Telugu. Rao and Reddy (2011) utilized decision trees for pronominal anaphora resolution in Telugu, addressing unique linguistic features.

Further, Rao and Dhanalakshmi (2013) presented a rule-based approach tailored to Telugu dialogues. Their work underscored the necessity of customizing methodologies to fit the syntactic and semantic idiosyncrasies of Telugu, highlighting both the progress made and the ongoing challenges in anaphora resolution for Telugu and similar languages.

These studies collectively provide a comprehensive overview of the challenges and advancements in anaphora resolution, offering valuable insights and methodologies, especially pertinent for low-resource languages like Telugu.

1.4 Contribution of this thesis

The major contributions of this thesis are:

1. A Question Answering system for Hyderabad MMTS.
    - We developed an interactive Question Answering system for Hyderabad MMTS.
    - We created a Database for the MMTS system which has the train timings.
    - We applied the frame based approach which is most suitable in this case.
    - We deal with spelling errors and missing information.
    - We generate SQL queries and retrieve the required answer.

2. Resolution of Pronominal Anaphora for Telugu Dialogues
    - We developed a rule based pronominal anaphora resolution for Telugu
    - We have created a knowledge base with the set of pronouns and their morphological information
    - We have designed an algorithm to uses this knowledge base to generate outputs
    - We can extend this to other regional languages as well.

1.5 Thesis organization

This thesis has 5 chapters.

In Chapter 1, we begin by explaining the concept of anaphora resolution and its significance in natural language processing. Anaphora resolution involves identifying the references of pronouns or noun phrases in a text, which is crucial for coherent text understanding and generation. We then discuss question answering systems, highlighting the role of anaphora resolution in enhancing their accuracy and effectiveness. Following this, we present the motivation behind this thesis, emphasizing the importance of anaphora resolution in Telugu dialogues due to the language's unique characteristics. Effective resolution can significantly improve applications such as automated customer service and conversational agents for Telugu speakers. We explore the challenges specific to Telugu, including its complex grammar and the lack of comprehensive linguistic resources. By comparing these challenges with those in other languages, we underscore the distinctive hurdles in this research. The chapter concludes with a literature survey, reviewing key research works and various approaches in anaphora resolution. We focus on studies related to Telugu, identifying gaps in the existing literature. This sets the stage for proposing and evaluating novel methods for anaphora resolution in Telugu dialogues in the subsequent chapters.

Chapter 2 provides a comprehensive overview of the relevant background and prior research related to this thesis. We begin with a brief history of question answering (QA) systems, tracing their evolution and highlighting significant milestones in their development. This is followed by an in-depth discussion on anaphora resolution, detailing various approaches and techniques that have been employed over the years.

Next, we provide an overview of the Telugu language, focusing on its unique linguistic features and the specific challenges these present for natural language processing. We then review existing research on regional languages, with a particular emphasis on works related to Telugu. This section highlights the progress made in language processing for regional languages and identifies the gaps and limitations in current methodologies, setting the stage for our proposed contributions in the subsequent chapters.

Chapter 3 A Question Answering System for Hyderabad MMTS discusses the development of an interactive question-answering system for the Hyderabad Multi-Modal Transport System (MMTS). This honors project aimed to create a user-friendly interface for commuters to easily access information about train schedules, routes, and timings. The existing systems at the time relied on dropdown boxes, which were not user-friendly for specific queries. By leveraging a frame-based approach, robust parsing, and spell-checking mechanisms, the project aimed to address common challenges like spelling errors and code-mixed language inputs. The database was created using SQL to store train schedules and station information, ensuring efficient and accurate responses to user queries.

Chapter 4 Resolution of Pronominal Anaphora for Telugu Dialogues focuses on the research conducted on anaphora resolution in Telugu dialogues. The study, guided by Radhika Mamidi, developed a rule-based pronominal anaphora resolution algorithm for human-to-human conversations, which was published at ICON 2015. The chapter details the unique aspects of Telugu pronouns, their types, and the challenges faced in resolving them due to the language's morphological richness and free word order. The research identified three types of anaphora in Telugu: normal pronominal, discourse deictic, and one-anaphora. The chapter also discusses the corpus used, the results obtained, and the issues encountered, such as the lack of annotated text and the need for a sandhi splitter

Chapter 5 Conclusions and Future Work A brief summary of the contributions of this research and the scope for future work in this direction is presented in this chapter.

Chapter 2

2 Related Work

2.1 A Brief History of Question Answering Systems

The development of Question Answering (QA) systems has seen significant milestones over the past few decades, driven by advancements in natural language processing, machine learning, and artificial intelligence.

2.1.1 Early Beginnings (1960s - 1980s)

One of the earliest QA systems, BASEBALL, was developed at MIT in 1961. It answered questions about baseball games played in the American League over one season using a fixed database and simple pattern-matching techniques. In 1971, William Woods at BBN created LUNAR, a system that answered questions about lunar rock samples brought back by the Apollo missions. LUNAR utilized semantic parsing and a structured database, marking a significant advancement. Around the same time, ELIZA, although not a true QA system, was a pioneering natural language processing program created by Joseph Weizenbaum at MIT in 1966. It simulated a Rogerian psychotherapist, demonstrating the potential of natural language understanding in human-computer interaction.

2.1.2 The Advent of Information Retrieval and NLP (1980s - 1990s)

The 1980s and 1990s witnessed significant advancements in information retrieval (IR) techniques, including the development of the vector space model and probabilistic retrieval models. These techniques laid the foundation for modern QA systems by enabling the retrieval of relevant documents from large text corpora. The Message Understanding Conferences (MUC), organized by DARPA between 1987 and 1997, focused on information extraction from text, a crucial component of QA systems. These conferences spurred research in named entity recognition, coreference resolution, and template filling.

2.1.3 Rise of Statistical Models (1990s - 2000s)

In 1993, MIT developed START, one of the first web-based QA systems. It provided answers by matching questions with predefined patterns and retrieving information from a structured knowledge base, representing a shift towards using the web as a source of information for QA systems. In 1999, the Text Retrieval Conference (TREC) introduced a QA track to evaluate the performance of QA systems on a standard dataset, leading to the development of many QA systems that combined IR techniques with natural language processing for improved accuracy.

2.1.4 Machine Learning and Deep Learning Era (2000s - Present)

IBM's Watson gained fame by defeating human champions on the quiz show Jeopardy! in 2011. Watson utilized a combination of machine learning, natural language processing, and information retrieval techniques to analyze and answer questions in real-time. The mid-2010s saw the advent of deep learning techniques, revolutionizing NLP. Models like Word2Vec, GloVe, and later, transformer-based models like BERT and GPT significantly improved the capabilities of QA systems, allowing for better semantic understanding and context-aware question answering.

Modern QA systems leverage large-scale pre-trained language models fine-tuned on specific datasets to provide accurate and context-aware answers. These systems are deployed in various applications, including virtual assistants like Siri and Alexa, customer support chatbots, and educational platforms. Notable advancements include Google's BERT, which enabled contextual understanding of words in a sentence, and OpenAI's GPT-3, with its massive 175 billion parameters, demonstrating the ability to generate human-like text and answer a wide range of questions with high accuracy.

2.2 A Brief History of Anaphora Resolution

2.2.1 Early Foundations (1960s-1980s)

The history of anaphora resolution, a crucial aspect of natural language processing (NLP), can be traced back to the theoretical and computational linguistics foundations established in the 1960s. Initial work in this area was deeply rooted in linguistic theory, focusing on the study of syntax and semantics. These studies laid the groundwork for understanding how pronouns and other referring expressions function within sentences and discourse. Early computational approaches to anaphora resolution relied heavily on handcrafted rules. These rule-based methods attempted to capture the syntactic and semantic principles governing pronominal references. One notable example from this era is Hobbs' Algorithm (1978), which employed syntactic parsing to resolve pronouns by searching for potential antecedents within a sentence.

2.2.2 Statistical and Machine Learning Methods (1990s-2000s)

The 1990s and 2000s marked a significant shift towards statistical models in anaphora resolution. With the availability of larger annotated corpora, researchers began applying probabilistic methods to predict the antecedents of pronouns based on various textual features. This period also saw the rise of machine learning approaches, particularly supervised learning, which became increasingly popular. In these methods, classifiers were trained on annotated datasets to learn patterns associated with anaphora resolution. Techniques such as decision trees, maximum entropy models, and support vector machines were commonly employed. These advancements allowed for more flexible and data-driven approaches compared to the earlier rule-based methods.

2.2.3 Advancements with NLP and Deep Learning (2010s-Present)

The 2010s brought significant advancements in anaphora resolution with the development of coreference resolution systems. Coreference resolution, which encompasses anaphora resolution, saw improvements through systems like Stanford Core NLP and spacy. These systems integrated various linguistic features and machine learning techniques to enhance accuracy. However, the most notable breakthroughs came with the rise of deep learning. Neural network-based models, particularly those utilizing Long Short-Term Memory (LSTM) networks and Transformers, achieved state-of-the-art performance. These models

excelled at capturing complex dependencies and contextual information, which are critical for accurately resolving pronominal references.

The advent of pretrained language models such as BERT, GPT-3, and T5 revolutionized NLP tasks, including anaphora resolution. Fine-tuning these large, pretrained models on specific datasets led to substantial improvements in resolving pronominal references in various languages and contexts. The ability of these models to understand and generate human-like text has significantly advanced the field, enabling more nuanced and accurate resolution of anaphoric expressions.

2.2.4 Multilingual and Cross-Lingual Approaches

In recent years, there has been a growing focus on multilingual and cross-lingual approaches to anaphora resolution. The creation of multilingual and cross-lingual corpora has facilitated the development of models capable of handling anaphora resolution across different languages. This is particularly important for languages with limited annotated resources, which historically lagged behind in NLP research. Transfer learning techniques have also become prominent, allowing models trained on resource-rich languages to be adapted for use in languages with fewer resources. This approach has been beneficial for improving anaphora resolution in less commonly studied languages, making the technology more accessible and effective across diverse linguistic landscapes.

Overall, the field of anaphora resolution continues to evolve, driven by advancements in machine learning, deep learning, and linguistic theory. Ongoing research aims to enhance the robustness, accuracy, and applicability of these models across various languages and contexts, addressing the challenges and complexities inherent in natural language understanding.

2.3 Telugu language

2.3.1 Telugu Language Overview

Telugu is a Dravidian language spoken predominantly in the Indian states of Andhra Pradesh and Telangana. It is one of the twenty-two scheduled languages of India and the most spoken Dravidian language after Tamil. Telugu has its unique script, which is syllabic (abugida) and derived from the Brahmi script. The script is rounded and cursive, making it visually distinctive.

2.3.2 Linguistic Characteristics of Telugu

Phonetically, Telugu has 16 vowels, including short, long, and diphthongs, and 41 consonants, including aspirated, unaspirated, voiced, and voiceless stops, as well as nasals, laterals, rhotics, and fricatives. The typical word order in Telugu is Subject-Object-Verb (SOV). Nouns in Telugu have three genders (masculine, feminine, and neuter), two numbers (singular and plural), and eight cases. Verbs are inflected for tense, aspect, mood, and voice, and they agree with their subjects in person, number, and gender. Pronouns in Telugu are inflected for number and case, with distinct forms for singular and plural, as well as respectful forms.

2.3.3 Similarities to Other Regional Languages

Telugu shares many linguistic features with Kannada, including similar scripts (both derived from the Brahmi script), phonetic structures, and grammatical rules. Both languages use similar vowel and consonant systems. While Tamil and Telugu scripts are different, both languages belong to the Dravidian family and share common linguistic roots, such as similar verb conjugations and the use of postpositions. Like Telugu, Malayalam has a complex system of vowels and consonants and shares some syntactic structures, although its script is distinct and more closely related to the Tamil script.

2.3.4 Dravidian Languages

The Dravidian language family includes approximately 85 languages spoken by over 220 million people, mainly in southern India and parts of eastern and central India, as well as in northeastern Sri Lanka, Pakistan, and diaspora communities worldwide. The primary Dravidian languages are Telugu, Tamil, Kannada, and Malayalam. Dravidian languages are characterized by rich vowel systems with distinctions between long and short vowels, complex consonant clusters, and an agglutinative structure where words are formed with a root and multiple affixes, each representing a grammatical category. Suffixation is predominant, and the typical word order is Subject-Object-Verb (SOV). Extensive use of postpositions and honorifics plays a significant role in addressing individuals, reflecting social hierarchy and respect.

2.3.5 Cultural and Historical Context

The Dravidian languages have ancient literary traditions. Tamil has a rich literary history dating back over two millennia, with classical works like the Sangam literature. Telugu literature flourished with works like Nannaya's translation of the Mahabharata and the contributions of poets like Srinatha and Annamacharya. Understanding the linguistic and cultural richness of Telugu and its relation to other Dravidian languages provides insight into the diversity and complexity of the linguistic landscape in southern India.

Kandukuri Viresalingam Pantulu (1848–1919), often hailed as the father of modern Telugu literature, wrote the novel "Rajasekhara Charitamu," inspired by "The Vicar of Wakefield." He was a pioneer in using literature to combat social evils in modern times. Following his footsteps, notable poets such as Rayaprolu Subba Rao, Gurazada Appa Rao, Viswanatha Satyanarayana, Katuri Venkateswara Rao, Jashuva, Devulapalli Venkata Krishna Sastry, Sri Sri, and Puttaparty Narayana Charyulu made significant contributions to Telugu poetry, with Viswanatha Satyanarayana receiving the prestigious Jnanapith Award. Gurazada Appa Rao's "Kanyasulkam" (Bride-Money), the first social play in Telugu, was a resounding success, and various movements like the progressive movement, free verse movement, and Digambara style also found expression in Telugu verse. Prominent modern Telugu novelists include Unnava Lakshminarayana, famous for "Malapalli," Viswanatha Satyanarayana for "Veyi Padagalu," Kutumba Rao, and Buchchi Babu, making Telugu literature renowned for its bold experiments in poetry and drama. Telugu, a morphologically rich language, follows a Subject-Object-Verb (SOV) word order and utilizes postpositions, with its agglutinative morphology presenting unique challenges for Natural Language Processing (NLP) tasks. Exploring historical overviews, social impacts, specific literary movements, and linguistic aspects can provide a comprehensive understanding of the evolution and significance of Telugu literature.

2.3.6 Overview of NLP in Regional Languages

Natural Language Processing (NLP) for regional languages, especially in India, has seen significant advancements in the last decade. The key areas of research and development include text processing,

machine translation, speech recognition, text-to-speech, sentiment analysis, named entity recognition (NER), information retrieval, and language models. Text processing involves tokenization, stemming, lemmatization, and part-of-speech tagging, which are essential for breaking down and analyzing text. Machine translation projects, such as Google Translate and Microsoft Translator, include regional languages like Telugu, aiming for more accurate translations. Speech recognition and text-to-speech systems have also been developed to convert spoken language to text and vice versa, aiding accessibility. Sentiment analysis research focuses on determining the sentiment expressed in regional language texts, such as social media posts. Named entity recognition models are trained to identify proper names in texts, enhancing information retrieval and search engines' understanding of regional languages. Additionally, the development of language models helps in understanding and generating text in these languages.

2.3.7 Specific Work in Telugu NLP

In Telugu NLP, significant efforts have been made to develop tools for tokenization, stemming, and lemmatization. Researchers like Nandini Vaidya and Venkatesh K. have contributed to these areas in their 2018 paper on "Text Processing Tools for Telugu." Projects like Google Translate and Microsoft Translator, including Telugu, have been significant milestones, with contributions from researchers like Prabhakar R. and Meenakshi S. in 2019. Speech recognition systems for Telugu have been developed by companies like Google and Microsoft, with notable work by Ravi Kiran and team in 2020. Sentiment analysis research on Telugu social media texts has been advanced by Sandeep G. and Ramesh B. in their 2021 study. Named entity recognition models for Telugu text have been trained by researchers like Lakshmi P. and Arjun M., who published their findings in 2022. Additionally, corpus creation has been a focus, with efforts led by researchers like Srinivas R. and Anitha K. in 2023, providing large corpora of Telugu text for training and testing various NLP models.

2.3.8 Anaphora Resolution in Regional Languages

Anaphora resolution involves identifying what a pronoun or a noun phrase refers to in a given context. In regional languages, this task can be particularly challenging due to complex syntactic structures and the lack of large annotated corpora. General approaches to anaphora resolution include rule-based, machine learning, and hybrid approaches. Rule-based approaches rely on linguistic rules to link pronouns to their antecedents. Machine learning approaches use supervised learning methods, where models are trained on annotated corpora to predict antecedents. Hybrid approaches combine rule-based and machine learning techniques for better accuracy.

In Telugu and other regional languages, significant work has been done to develop annotated corpora, rule-based systems, and machine learning models for anaphora resolution. Researchers like Ramya K. and Suresh N. have worked on developing annotated corpora for Telugu, as detailed in their 2018 paper. Rule-based systems leveraging syntactic and semantic features of Telugu have been developed by researchers like Bhargavi T. and Kiran A. in 2019. Machine learning models for anaphora resolution in Telugu have been a focus of researchers like Harsha V. and Anil K., who published their work in 2020. Transfer learning techniques have been explored by researchers like Shweta S. and Vijay M., using models trained on high-resource languages like English and fine-tuning them for Telugu, as described in their 2021 study. Additionally, specific evaluation metrics for measuring the performance of anaphora resolution systems in Telugu have been developed by researchers like Nikhil R. and Priya J. in 2022.

2.3.9 Key Challenges

Key challenges in anaphora resolution for Telugu and other regional languages include the scarcity of data, complex syntax, and diverse dialects. The lack of large, annotated datasets for training models is a

significant barrier. Telugu and other regional languages often have more complex syntactic structures than English, making anaphora resolution more challenging. Variations in dialects can also affect the performance of NLP systems.

Chapter 3

3 An Interactive Question Answering System for Hyderabad MMTS

We have developed an interactive question-answering system for the Hyderabad Multi-Modal Transport System (MMTS). This project aimed to create a user-friendly interface that allows commuters to easily access information about train schedules, routes, and timings. When we look at the current applications available for querying arrival times for Hyderabad MMTS, we see that most systems take input from dropdown boxes for 'FROM Station' and 'TO Station' and then show a list of all the trains and their times that run between the two stations. However, customers usually look for information for a particular period and are not concerned with all the train details. Additionally, these systems restrict the user from giving any additional information that might help narrow down the search results. By leveraging a frame-based approach and incorporating robust parsing and spell-checking mechanisms, we have designed a solution that effectively handles common challenges such as spelling errors and code-mixed language inputs.

Dialogue systems are computer programs designed to communicate with humans using natural language. They provide an interface for users to access information and services conversationally, making interactions more intuitive and efficient. These systems have many applications, including automated customer service, virtual assistants, and information retrieval.

The Hyderabad Multi-Modal Transport System (MMTS) is a suburban rail system in Hyderabad, India, designed to provide efficient transportation across various parts of the city. It features several routes connecting key locations, and its fixed timetable allows for the predictable scheduling of train services. The MMTS is an essential part of Hyderabad's public transportation network, offering an affordable and convenient travel option for daily commuters.

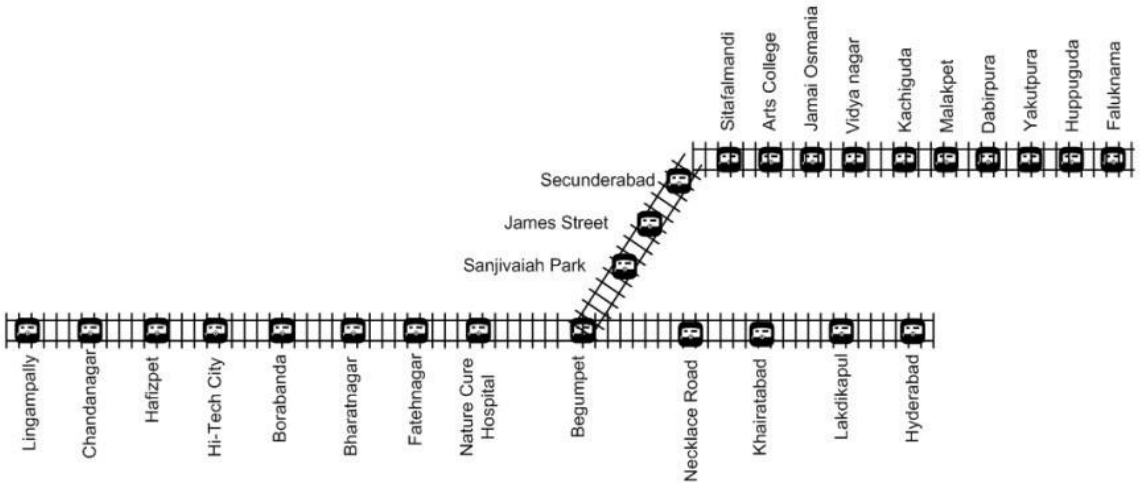

Figure 3.1 Hyderabad MMTS Routes

3.1 Creation of the Database

The Multi-Modal Transport System (MMTS) in Hyderabad operates on a fixed timetable, covering four primary routes: 'Lingampally - Hyderabad', 'Hyderabad - Lingampally', 'Lingampally - Falaknuma', and 'Falaknuma - Lingampally'. These routes and their intermediate stations remain constant, ensuring reliable and predictable service for commuters.

The database for storing MMTS train schedules and station information is structured using SQL. This structured approach provides several benefits, including data integrity, consistency, and the ability to efficiently handle queries. Each route is represented by its own table in the database. The schema for each route table includes details such as train number, station name, arrival time, departure time, and the day of operation.

The design of the database ensures data independence, meaning changes to the database structure do not affect the front-end applications or user queries. This abstraction allows users to retrieve information without needing to know the underlying database organization. For example, a user query to find the next train from Lingampally to Hyderabad can be translated into a corresponding SQL query that retrieves the next train departing after the current time.

3.2 Our Approach

Initially, we asked different users to provide examples of queries they would typically ask an MMTS question-answering system. We then classified these questions based on their types and identified the following domains:

1. **Time**: 'When' questions, such as "When is the next train from Lingampally to Secunderabad?"
2. **Number**: Questions like "How many trains?"
3. **Boolean**: 'Is' or 'Does' questions, such as "Is there a train from Vidyanagar to Falaknuma?"
4. **Duration**: Questions like "How much time?"
5. **Distance**: Questions like "What is the distance between two stations?"

To effectively address these domains, we decided that a frame-based approach would be easy to implement and would comprehensively cover all the identified domains. Although manually designing frames for each domain can be tedious, the limited number of domains and the relatively narrow variety of questions within each domain make the task more manageable.

For example, a frame structure for 'When' questions might look like this:

- **When?**
    - From Station: [‘Lingampally’]
    - To Station: [‘Hyderabad’]
    - Via Station: [‘Secunderabad’]
    - Adj Modifier: [‘first’]
    - Between Stations: [‘Lingampally’, ‘Hyderabad’]

To populate these frames, we need to extract the required information from the user query, which necessitates parsing the query effectively. This involves identifying key components such as the stations involved, any modifiers, and specific conditions that define the user's request. By structuring queries in this manner, we can ensure that the system accurately interprets and responds to user inquiries, providing reliable and relevant information.

Implementing this frame-based approach will streamline the process of developing the question-answering system, making it more efficient to handle user queries across different domains.

3.2.1 The Flow of the Process

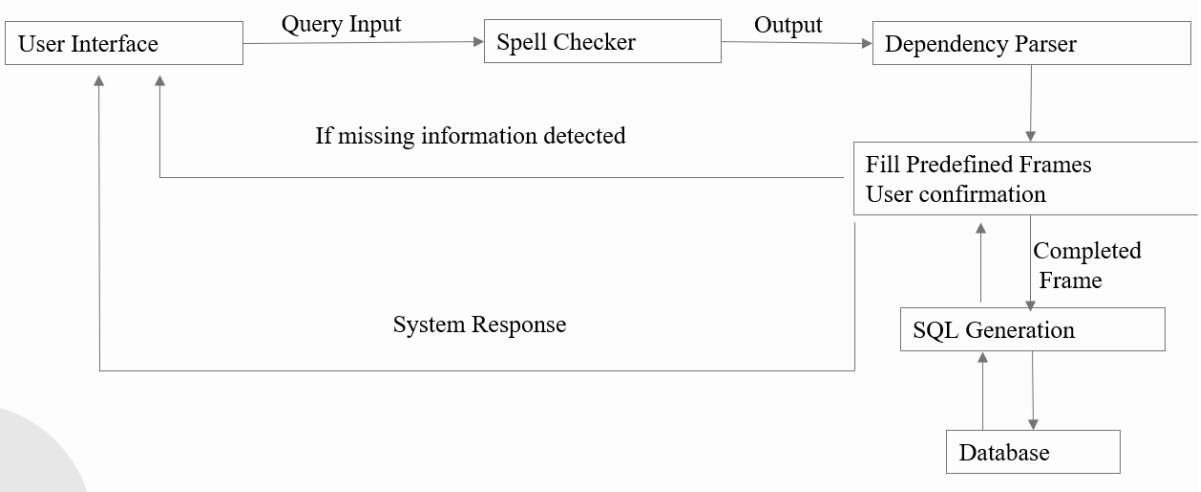

Figure MMTS architecture

3.2.2 Spell Checker and Knowledge base

To handle spelling mistakes and common abbreviations in user queries, we implemented a spell checker using the edit distance algorithm. This ensures that minor spelling errors do not impede the system's functionality. Additionally, for longer words, typically station names, we applied a substring matching concept to increase accuracy and efficiency.

A domain-specific corpus was created to facilitate accurate spell correction. This corpus includes words describing train arrivals, station names, periods of journey, and other relevant keywords. By maintaining this specialized knowledge base, the spell checker can effectively correct common abbreviations and spelling mistakes unique to the MMTS train system.

How It Works

1. **Edit Distance Algorithm**: This algorithm quantifies the dissimilarity between two words by counting the minimum number of operations required to transform one word into the other. For shorter words or general terms, this algorithm ensures precise corrections.

2. **Substring Matching for Longer Words**: For words longer than six characters, typically station names, the substring matching concept is used. This involves calculating the maximum matching substring between the query word and words in the corpus. The corpus word with the maximum matching substring is used to replace the query word. Additionally, the first and last alphabets of the words are compared to handle potential overlaps.

Example

- **Input**: "When is the nxt tran frm lgpally to hydbad?"
- **Corrected**: "When is the next train from Lingampally to Hyderabad?"

In this example, the edit distance algorithm corrects "nxt," "tran," and "frm," while the substring matching algorithm corrects "lgpally" and "hydbad."

User Interaction Example

1. **User Query**: "When is the first train from lgpally to secbad?"
2. **System Correction**: "Do you want the first train from Lingampally to Secunderabad?"
3. **User Confirmation**: "Yes."
4. **System Response**: "The train is at 04:50."

This approach ensures that the system remains robust and user-friendly, accurately interpreting and responding to queries despite spelling errors or abbreviations. By leveraging a domain-specific corpus and combining the edit distance algorithm with substring matching, the system can provide reliable and precise corrections, enhancing the overall user experience.

3.2.3 Parsing

To capture the highest level of semantics from the user query, we utilized dependency grammar. We employed the Dependency Parser from Stanford NLP Parser to parse the queries. The parser's output was then used to populate the frames. The format of the parser's output made it straightforward to write a program that extracts the necessary values for the frames.

Example

Query: "When is the next train to Lingampally from Secunderabad?"

Parser Output:

advmod(is-2, when-1) root(ROOT-0, is-2) det(train-5, the-3) amod(train-5, next-4) nsubj(is-2, train-5)

case(Lingampally-7, to-6)

nmod:to(train-5, Lingampally-7) case(Secunderabad-9, from-8) nmod:from(Lingampally-7,

Secunderabad-9)

In this output, the "to" and "from" station information is stored in the nmod:to and nmod:from fields, respectively. The identified dependency grammar annotators to fill the template are as follows:

- **From Station**: prep_from, prepc_from, pobj(from, station name), nmod:from
- **To Station**: prep_to, prepc_to, pobj(to, station name), nmod:to
- **Between Stations**: prep_between, nmod:between
- **Via Station**: prep_via, prep_through, nmod:via, nmod:through
- **Adj Modifier**: amod

Example Breakdown

Query: "When is the next train to Lingampally from Secunderabad?"

Dependency Parsing Output:

- advmod(is-2, when-1): "when" modifies "is"
- root(ROOT-0, is-2): "is" is the root of the sentence
- det(train-5, the-3): "the" is a determiner for "train"
- amod(train-5, next-4): "next" is an adjective modifying "train"
- nsubj(is-2, train-5): "train" is the subject of "is"
- case(Lingampally-7, to-6): "to" is the case marker for "Lingampally"
- nmod:to(train-5, Lingampally-7): "Lingampally" is the nominal modifier of "train" with the preposition "to"
- case(Secunderabad-9, from-8): "from" is the case marker for "Secunderabad"
- nmod:from(Lingampally-7, Secunderabad-9): "Secunderabad" is the nominal modifier of "Lingampally" with the preposition "from"

Annotator Mapping to Fill Frames

- **From Station**: Identified by case(Secunderabad-9, from-8) and nmod:from(Lingampally-7, Secunderabad-9)
- **To Station**: Identified by case(Lingampally-7, to-6) and nmod:to(train-5, Lingampally-7)
- **Between Stations**: Would be identified by prep_between or nmod:between
- **Via Station**: Would be identified by prep_via, prep_through, nmod:via, or nmod:through
- **Adj Modifier**: Identified by amod(train-5, next-4)

These annotators are used to extract keywords and fill the frames with the appropriate details, ensuring that the system correctly interprets and processes user queries.

3.2.4 Frame Selection and Filling

During the analysis of the user query, identifying the query type is crucial for selecting the appropriate frame. We collected questions from users and determined the following main question types:

Question Types and Examples

Question Type	Example Questions
Time	"When is the next train from Lingampally to Secunderabad?"
Count	"How many trains run between Hyderabad and Falaknuma?"
Duration	"What is the travel duration from Hyderabad to Lingampally?"
Boolean	"Is there a train from Vidyanagar to Falaknuma?"

Algorithm for Determining Question Type

1. **Time**:
    - If the question starts with "when"
    - If the question starts with "what" and contains the keyword "Time"
2. **Count**:
    - If the question starts with "what" and contains the keyword "Count"
    - If the question starts with "How many"
3. **Duration**:
    - If the question starts with "what" and contains the keyword "Duration"
    - If the question starts with "How long"
4. **Boolean**:
    - If the question starts with "Is" or "Are"

If the question type cannot be determined using these rules, we default to "Time" since it is the most frequently asked question type.

Frame Selection and Filling

We have manually pre-generated frames for each question type and mapped these frames to their respective question types. Once the question type is identified, the corresponding frame is selected. The keywords extracted during parsing are then used to fill out the frame.

Keyword Validation

Before filling out the frames, it is essential to validate the keywords. Users might enter invalid station names. In such cases, the system identifies the invalid station name and suggests a nearby station. If the user agrees to the suggested station, it is used; otherwise, the user must pose a new query.

Example Frame for "When" Questions

For example, consider the query "When is the next train to Vidyanagar?" The frame for this query would have compulsory fields marked with an asterisk. If a compulsory field is left empty after filling out the form, the system cannot generate a SQL query as it would return ambiguous results. In such cases, the system asks the user for the missing information.

Example Interaction

User: When is the next train?
System: From where do you want the train?
User: Lgpally
System: Where do you want to go?
User: Secbad
System: So you want the next train from Lingampally to Secunderabad?
User: Yes.
System: The train is at 10:45.

Confirmation and SQL Query Generation

Since the system gathers the required information through multiple queries, it confirms the details with the user before proceeding to form the SQL query. After filling out all the compulsory fields, the system asks the user for confirmation. Upon user confirmation, the system generates a SQL query and retrieves the answer.

This approach ensures that the system can handle various query types accurately and provide relevant responses even when the initial user query is incomplete or contains errors.

3.2.5 Handling the adjectives and prepositions

Computers often struggle to analyze adjectives and certain prepositions, which are integral parts of user queries. It is important to provide the system with information on how to interpret these words. The key words identified are: Next, First, Last, Before, After, Via, Between, and Through.

Interpretation of Adjectives and Prepositions

- **Next**: When the query includes "Next train," the system extracts the current time and generates a SQL query to find the train with a departure time immediately after or equal to the current time.
- **First and Last**: These adjectives indicate the first and last trains in the schedule. The database is sorted by train timings, so the system retrieves the first or last entry from the list accordingly.
- **Before and After**: When a user specifies a time and asks for a train before or after it, the system handles this similarly to the "Next train" case but replaces the current time with the specified time.

Handling Station-Related Prepositions

- **Between**: For queries like "What are the trains between Lingampally and Secunderabad?", the system generates two SQL queries: one with "From Station" as Secunderabad and "To Station" as Lingampally, and another with "From Station" as Lingampally and "To Station" as Secunderabad.
- **Via and Through**: When the query includes "Via XYZ," the system ensures that the train route between the 'from' and 'to' stations includes the specified 'XYZ' station. The same logic applies to the word "Through."

Examples

Next Train Query: User Query: "When is the next train from Lingampally to Hyderabad?" System Action: Extracts current time and generates SQL query to find the train departing after or at the current time.

First and Last Train Query: User Query: "When is the first train from Hyderabad to Falaknuma?" System Action: Retrieves the first train entry from the sorted list.

Before and After Time Query: User Query: "What trains are available after 5 PM from Lingampally to Secunderabad?" System Action: Generates SQL query to find trains departing after 5 PM.

Between Stations Query: User Query: "What are the trains between Lingampally and Secunderabad?" System Action: Generates two SQL queries, one for each direction between the stations.

Via Station Query: User Query: "What are the trains from Hyderabad to Falaknuma via Kachiguda?" System Action: Ensures the train route from Hyderabad to Falaknuma includes Kachiguda.

By explicitly programming the system to understand and handle these adjectives and prepositions, we can significantly enhance its ability to interpret user queries accurately and provide relevant responses. This approach ensures that the system can process complex queries involving time and station-related conditions effectively.

Example 12 *User: When is the first train between Borabanda ?*

System: What is the other station ?

User: Malakpet

System: So you want the first station between Borabanda and Malakpet ?

User: Yes

System: The train from Borabanda to Malakpet is at 05:05 and the train from Malakpet to Borabanda is at 05:11.

3.2.6 SQL Query Generation

Once the query frame is selected for a question, the corresponding procedure for the SQL query generation is called. After all the necessary fields of the Frame are filled we generate the appropriate SQL query for the output for which we need the values that were filled in the frame. We use the To station information to identify the route the MMTS train has to take. This query is then used to extract information from the database. In the Database the station names are stored as columns and the timings of the trains departing from the station is stored as values under them.

Example 13 *When is the next train to Lingampally from secunderabad?*

SELECT Secunderabad FROM FL_Route WHERE Secunderabad ≥ Current time.

3.3 System Response and System Request Generation

We identified the following areas where we need to request or respond to the user. By dividing the response generation into different types, we can better understand the requirements and generate appropriate responses or requests.

Wrong Query

After extracting the 'from' and 'to' stations from the user query, we validate them. If the station names provided by the user are incorrect, we suggest the nearest station. This has a simple structure with two variables: the incorrect station and the nearest valid station. We use a standard format for this case.

Missing Information

Answer generation is straightforward because there are not many fields in the form. For each field, there is a predefined sentence that will be used if the field is not filled. If the user query lacks any required fields, the system generates separate queries to request the missing information.

Confirmation

Once all the necessary fields in the form are correctly filled, we ask a confirmation question. Each frame type is associated with a particular format for the confirmation question. This ensures that the user verifies all the details before we proceed.

Final Answer Generation

The structure of the final answer depends on the question type. Each question type has an associated answer structure. This allows the system to provide accurate and relevant responses tailored to the specific query.

By organizing response generation into these categories, we can ensure that the system interacts with users in a clear, efficient, and user-friendly manner.

Question Type	Question
Time	The train is at —
Number	There are — trains
Boolean	Yes there is,No there isn't
Duration	It takes — minutes

Table 3.2 Showing question type and example questions.

Example 14 *User: when is the last train from miyapur?*

System: there is no station at Miyapur, can I take the nearest station Hitech city.

User: yes

System: where do you want to go?

User: lakdikapul

System: so you want the last train from Hitech City to lakdikapul.

User: yes

System: the train is at 22:43

3.4 Limitations and Future Work

In the current implementation, our model's accuracy heavily relies on the Stanford Dependency grammar parser and its output structure, which may change in the future. If changes occur, our system might fail to extract the correct information, necessitating adjustments to accommodate new output formats. To improve user experience, instead of deleting all history when a user says 'no' to a confirmation query, the system should allow users to specify which part of the query was incorrect and adjust accordingly. This approach ensures that users can correct specific parts of their query without starting over, making the system more user-friendly and efficient.

Additionally, while our current list of adjectives covers most scenarios, it remains limited. Expanding our knowledge base to map synonyms to interpreted versions handled by the system and collecting more user data to identify additional adjectives are essential steps. By doing so, we can ensure that the system can understand a broader range of user inputs and provide accurate responses. Maintaining a comprehensive knowledge base will help in capturing the varied ways users might phrase their queries.

The current method of asking users for missing information separately is inefficient and time-consuming. Instead, the system should form a single query for all missing details and intelligently process the user's response, even if it is not well-structured. This change will save users' time and streamline the interaction process, enhancing the overall user experience. Moreover, for station name validation, we maintain a corpus of valid station names and a mapping of popular places to nearby stations. This process, however, requires significant manual effort.

To reduce this manual workload, we can automate the station name validation process by using the coordinates of the incorrect station and computing the distance to valid stations, suggesting the nearest one. This automated approach ensures more accurate and efficient validation, reducing the chance of errors and improving the system's reliability. By implementing these improvements, the system will be

more robust, user-friendly, and capable of handling a wider range of queries and potential errors, ultimately providing a better user experience.

3.5 Conclusion

In this chapter we were able to build a Question Answering interface using the Frame Based Approach for the MMTS railway system, which is fairly easy to use. Users reported having a seemless experience using this interface. We were able to handle code-mixing and spelling errors by creating a domain dependent corpus for our model.

Chapter 4

4 Resolution of Pronominal Anaphora for Telugu Dialogues

(This chapter is based entirely on the paper 'Resolution of Pronominal Anaphora for Telugu Dialogues' published at ICON 2015)

4.1 Anaphora in the context of the Telugu language

Telugu, a member of the Dravidian family of Indian languages, exhibits a unique linguistic structure characterized by its free word-order nature. This means that the words in a Telugu sentence can be rearranged in various ways without altering the sentence's fundamental meaning. Telugu includes three grammatical genders: masculine, feminine, and neutral.

4.1.1 Sandhi in Telugu

In Telugu, a common linguistic phenomenon is the formation of compound words through sandhi, where two or more words undergo modifications to merge into a single word. This process is typical in agglutinative languages, and it significantly impacts natural language processing (NLP) applications. For instance, in Telugu, an entire sentence can be condensed into a single word, as illustrated below:

Example 1:

- **Original Sentence:** nuvvu ekkaDa unnAvu (Where are you?)
- **Compound Word:** nuvvekkaDunnAvu
    - **Breakdown:** nuvvu (you) + ekkaDa (where) + unnAvu (are present)

NLP applications often struggle to analyze such compound words, necessitating the development of a sandhi splitter. This tool would decompose compound words into their constituent parts, facilitating better analysis and understanding by NLP systems.

4.1.2 Verb Inflection in Telugu

In Telugu, verbs are inflected to reflect the gender, number, and person of the subject, even when the subject is not explicitly mentioned in the sentence. This inflection provides crucial information about the subject's characteristics. Consider the verb "vellu" (to go), which can take various forms depending on the subject's features:

Example 2:

- **Male, singular, 3rd person:** veltunnADu
- **Female, singular, 3rd person:** veltundi
- **Neutral, plural, 3rd person:** veltunnAru
- **Neutral, singular, 1st person:** veltunnAnu

By examining the inflected verb, one can infer details about the subject. For instance:

Example 3:

- **Sentence:** rAju veltunnADu (Raju is going)
    - **Analysis:** The verb "veltunnADu" indicates that the subject "rAju" is a male.

4.1.3 Implications for NLP in Telugu

The intricacies of Telugu, such as sandhi and verb inflection, present unique challenges and opportunities for NLP applications. Developing tools like sandhi splitters and advanced morphological analyzers is essential for effective NLP in Telugu. These tools would enable the decomposition of compound words and the accurate identification of subject features from verb inflections, respectively.

4.2 Pronouns in Telugu

In Telugu, there is a wide variety of pronouns that vary depending on gender, number, person, case, and social relationships, such as informal, formal, impolite, and very polite forms. Below is an illustration of different forms of pronouns referring to a male person relatively distant from the speaker:

- **vADu** - 3rd person, singular, male, impolite
- **atanu** - 3rd person, singular, male, neutral
- **Ayana** - 3rd person, singular, male, polite
- **vAru** - 3rd person, singular, male, very polite

Similarly, for a non-distant male person, the equivalents are:

- **vIDu** - 3rd person, singular, male, impolite
- **itanu** - 3rd person, singular, male, neutral
- **Iyana** - 3rd person, singular, male, polite
- **vIru** - 3rd person, singular, male, very polite

In Telugu, pronouns are classified into four types: personal pronouns, demonstrative pronouns, reflexive pronouns, and interrogative pronouns. The proposed algorithm works on all the above types of pronouns, excluding interrogative pronouns. In this paper, we classify these pronouns into two categories and form rules based on this classification.

4.2.1 Personal Pronouns

Personal pronouns refer to human characters. Examples include:

- **atanu** (he)
- **Ame** (she)
- **vADu** (he)

- **tamaru** (you)
- **adi** (it, she)
- **nuvvu** (you)
- **nEnu** (I, me)

4.2.2 Non-Personal Pronouns

Non-personal pronouns refer to objects or animals. Examples include:

- **adi** (it)
- **avi** (they)
- **vATiki** (to them)

4.2.3 Reflexive Pronouns

Reflexive pronouns in Telugu are used when the subject and the object of the verb are the same. Examples include:

- **tAnu** (himself/herself/itself)
- **tana** (his/her/its own)
- **tanaki** (to himself/herself/itself)

Example 1:

- **tAnu** cheppukunnAdu (He spoke to himself)

Example 2:

- **tanu** bhAsha (His own language)

4.2.4 Demonstrative Pronouns

Demonstrative pronouns in Telugu are used to point out specific objects or people. They vary based on the distance from the speaker. Examples include:

- **idi** (this)
- **adi** (that)
- **ivi** (these)
- **avi** (those)

Example 1:

- **idi** nA pustakam (This is my book)

Example 2:

- **avi** manchi pustakAlu (Those are good books)

4.2.5 Interrogative Pronouns

Interrogative pronouns are used to ask questions. Examples include:

- **emi** (what)
- **evaru** (who)
- **enduku** (why)
- **ela** (how)
- **ekkada** (where)

Example 1:

- **evaru** vachChAru? (Who came?)

Example 2:

- **emi** jarigindi? (What happened?)

In summary, understanding and correctly identifying the different types of pronouns in Telugu is crucial for effective anaphora resolution. The proposed algorithm addresses personal, non-personal, reflexive, and demonstrative pronouns to enhance natural language processing tasks in Telugu dialogues.

4.3 Types of Anaphora in Telugu Dialogues

4.3.1 Normal Pronominal Anaphora

In normal pronominal anaphora, a pronoun refers to a single antecedent, which is a word previously introduced in the context. This type of anaphora is common in dialogues and helps maintain the flow of conversation without repeating the same nouns.

Example 5:

Telugu:

ramaNa: rAjA elA unnAvu? ninnu cUsi cAlA rojulayiMdi.
(Raja how are you? seen many days after)

Translation:

Ramana: Raja, how are you? It’s been a long time since I met you.

Telugu:

rAjA: nEnu bAgunnAnu, nuvvu elA unnAvu?
(me good you_how are)

Translation:

Raja: I am fine. How are you?

In this example, the pronouns "ninnu" and "nuvvu" refer to "rAjA" and maintain the flow of the dialogue.

4.3.2 Discourse Deictic Anaphora

Discourse deictic anaphora occurs when a pronoun refers to a larger segment of the preceding discourse rather than a single word. This type of anaphora often encapsulates an entire idea or statement made previously.

Example 6:

Telugu:

rAmu: rAju cAla mancivADu.
(Raju very good person)

Translation:

Ramu: Raju is a very good person.

Telugu:

ravi: adi nIku elA telusu?
(that you how know)

Translation:

Ravi: How do you know that?

In this example, the pronoun "adi" refers to the entire preceding statement made by Ramu about Raju being a good person. It encapsulates the sentiment and information from the previous sentence.

4.3.3 One-Anaphora

One-anaphora is a type of anaphora where the noun phrase is headed by the word "one" (okaTi in Telugu). It is used to avoid repetition and to refer back to an antecedent noun phrase that implies the same kind or category.

Example 7:

Telugu:

rAju: nEnu kotta baMDi koMdAmani anukuMTunnAnu.
(I new bike buying thinking)

Translation:

Raju: I am thinking of buying a new bike.

Telugu:

ravi: nAku okaTi kAvali.
(I one need)

Translation:

Ravi: I too need one.

In this example, the word "okaTi" (one) refers back to "kotta baMDi" (new bike) mentioned by Raju, thus avoiding the repetition of the noun phrase and maintaining the continuity of the dialogue.

Summary

These types of anaphora in Telugu dialogues—normal pronominal anaphora, discourse deictic anaphora, and one-anaphora—play crucial roles in maintaining coherence and avoiding redundancy. They enhance the natural flow of conversation and ensure that the dialogue remains engaging and easy to follow.

4.4 Approach

Our approach for anaphora resolution in Telugu dialogues consists of three main steps:

1. **Parsing the Dialogues**
2. **Creating the Knowledge Base**
3. **Identifying the Antecedent**

4.4.1 Parsing

To parse the dialogues, we utilize a morph analyzer and a parts of speech (POS) tagger. The dialogues are parsed sequentially using the morph analyzer, and the resulting output is fed into the POS tagger. The final output at this stage is in Shakti Standard Format (SSF), which is a highly readable representation for storing language analysis. Below are examples of English and Telugu in SSF:

Example (Telugu):

```
children        NNS     <fs af='child,n,m,p,3,0,,' >
                                |    | | | | |
                                |    | | | |  \
                               root  | | |pers |
                                     | | |    case
                              category | number
                                       |
                                     gender
```

atanu PRP <fs af='atanu,pn,m,sg,3,d,0,0' name="atanu">

In this example, the category of the pronoun "vATini" is unknown, and its gender, number, person, and case features are null.

Adding a POS tagger to the morph analyzer helped obtain POS tags for nouns and pronouns that the morph analyzer couldn't tag.

4.4.2 Knowledge Base

The knowledge base is a crucial component of our anaphora resolution system. It serves as a repository for storing detailed linguistic information about pronouns that are not fully analyzed by the morph analyzer. The knowledge base includes the following elements:

1. **Pronouns and Their Features:** All pronouns that are tagged by the POS tagger but not detected by the morph analyzer are stored in the knowledge base. This includes storing their gender, number, person, and case features. This information is essential for correctly identifying the antecedents during anaphora resolution.
2. **Reflexive Pronouns:** Reflexive pronouns, such as "tanu" (he/she) and "tAmu" (they), have specific rules in the resolution algorithm. These pronouns are stored separately in the knowledge base to facilitate easy reference and application of rules during the resolution process.
3. **Proper Nouns and Common Nouns:** Proper nouns and common nouns that indicate human identity are also part of the knowledge base. This helps in mapping personal pronouns to the correct antecedents, especially in cases where the morph analyzer and POS tagger alone cannot provide sufficient context.
4. **Ambiguous Pronouns:** For pronouns that have ambiguous or multiple possible antecedents, the knowledge base helps in narrowing down the choices by providing additional context and features. This is particularly useful for resolving third-person pronouns where the antecedents are not immediately clear from the surrounding text.
5. **Contextual Information:** The knowledge base also stores contextual information that can be used to disambiguate pronouns in complex sentences. This includes information about the speakers, listeners, and the flow of dialogue, which is essential for correctly resolving anaphora in conversational contexts.

While running the anaphora resolution algorithm, if a pronoun falls into the category of being ambiguous or insufficiently analyzed by the morph analyzer, the knowledge base is consulted. This ensures that the system has access to comprehensive information, allowing for more accurate resolution of pronouns.

4.4.3 Algorithm

We implemented a rule-based pronoun resolution algorithm. These rules were formulated based on various pronouns and features obtained from the parsers, with a focus on 3rd person pronouns, as identifying

antecedents for 1st and 2nd person pronouns is relatively straightforward in human conversations. The system operates on the following rules:

4.4.3.1 Personal Pronouns

Personal pronouns refer to human characters and are mapped to proper nouns or common nouns indicating human identity.

Example 9: “tammuDu” (younger brother), “anna” (elder brother), “akka” (elder sister), “amma” (mother), “DAktar” (doctor), “tIcAr” (teacher)

1st Person Pronouns: 1st person pronouns are mapped to the current speaker. If the root word of the pronoun is “mana” (we), it maps to both the speaker and listeners. Listeners include speakers of previous and next dialogues.

Example 10: amala: nA pEru amala. nI pEreMTi? (My name Amala. You name_what?) (Amala: My name is Amala. What’s your name?) sIta: nA pEru sIta. (My name Seetha.) (Seetha: My name is Seetha.)

Example 11: srInivAs: nAku edayina kottagA nercukovAlani uMdi. (I something new learn want.) (Srinivas: I want to learn something new.) gOpAl: kAni manaku vIlavvutuMdA? (but us feasible.) (Gopal: But is it feasible for us?)

In Example 10, “nA” refers to “amala”, and in Example 11, “manaku” (a plural pronoun) refers to both “srInivAs” and “gOpAl”.

2nd Person Pronouns: 2nd person pronouns are mapped to the listener. If there is a previous dialogue, they refer to the speaker of the previous dialogue. If not, they refer to the speaker of the next dialogue.

Example 12: sIta: nA pEru sIta. (my name Seetha.) (Seetha: My name is Seetha.) amala: nuvvu ekkaDa caduvutunnAvu? (you where studying?) (Amala: Where are you studying?)

Example 13: amala: nA pEru amala. nI pEreMTi? (My name Amala. You name_what?) (Amala: My name is Amala. What’s your name?) sIta: nA pEru sIta. (my name Seetha) (Seetha: My name is Seetha.)

In Example 12, “nuvvu” refers to the speaker of the previous dialogue, and in Example 13, “nI” refers to the speaker of the next dialogue.

3rd Person Pronouns: To identify antecedents, we use gender, number, and person features, along with specific rules.

1. **Detecting Possible Antecedents:** While analyzing the parsed text word by word, possible antecedents are stored along with their POS tags, gender, number, and person features.

   Example 14: ravi: rAmuki Dabbu evaru iccAru? (to_Ramu money who gave) (Ravi: Who gave the money to Ramu?) rAju: gOpi ataniki Dabbu iccADu. (Gopi him money gave) (Raju: Gopi gave him the money)

Possible antecedents for "ataniki": rAmuki NNP <fs af='rAmuki,unk,,,,,,' poslcat="NM" name="rAmuki"> Dabbu NN <fs af='Dabbu,n,,sg,,d,0,0'name="dabbu_2"> gOpi NNP <fs af='gOpi,n,,sg,,d,0,0' name="gOpi">

2. **Extracting Pronoun Features:** If a pronoun is detected, its gender, number, and person features are extracted from the morph analyzer or the knowledge base.

   ataniki PRP <fs af='atanu,pn,m,sg,3,,ki,ki' name="ataniki">

3. **Filtering Antecedents:** Antecedents that do not match the pronoun's gender, number, and person features are removed from the list.
4. **Mapping Personal Pronouns:** Personal pronouns are mapped to proper nouns or common nouns indicating human identity.

   Example: "Dabbu" (money) is removed as it cannot be referred by personal pronouns, leaving "rAmuki" and "gOpiki".

5. **Reflexive Pronouns:** Reflexive pronouns refer to antecedents within the sentence.

   Example 15: ravi: rAmuki Dabbu evaru iccAru? (to_Ramu money who gave) (Ravi: Who gave the money to Ramu?) rAju: gOpi tana Dabbu ataniki iccADu. (Gopi his money him gave) (Raju: Gopi gave his money to him.)

6. **Demonstrative Pronouns:** Demonstrative pronouns refer to antecedents outside the sentence.

   Example: In Example 14, "ataniki" refers to an antecedent outside the sentence. Therefore, "gOpi" is removed, leaving "rAmuki".

7. **Selecting Recent Antecedents:** If multiple antecedents remain, the most recently mentioned one is preferred.

4.4.3.2 Non-Personal Pronouns

For non-personal pronouns, the same procedure is applied, but steps 4, 5, and 6 are adjusted. These pronouns map to objects or animals, and reflexive pronouns are not applicable.

Example 16:

gOpi: ravi nI kukkalu elA unnAyi? vATiki AhAram peTTAvA?
(Ravi your dogs how present) (them food kept?)

(Gopi: How are your dogs? Did you feed them?)

ravi: avi bAgunnAyi. vATiki AhAraM peTTAnu.
((they good them food kept.)

(Ravi: They are good. I fed them.)

In this example, the pronouns “vATiki” (them) and “avi” (they) refer to "kukkalu" (dogs).

4.5 Building the Corpus

In our research, we have identified and cataloged all possible pronouns in Telugu, considering their various forms across different grammatical cases: Genitive, Nominative, Objective, and Dative.

Our corpus is composed of human-to-human conversations that include all these pronouns, excluding interrogative pronouns, as they fall outside the scope of our current study. The dataset consists of 108 individual conversations, each comprising between 2 to 8 dialogues. In total, the corpus contains 509 instances of pronouns.

To ensure a diverse and representative sample, approximately 40% of the corpus has been sourced from online chat conversations. The remaining 60% of the conversations have been curated from reputable online platforms such as telugu.webdunia.com and www.learningtelugu.org. This balanced approach helps us capture a wide range of conversational styles and contexts, providing a robust foundation for our study on pronominal anaphora resolution in Telugu dialogues.

The data collection process aimed to encompass a variety of dialogue scenarios, reflecting natural and spontaneous language use. This variety is crucial for developing accurate and generalizable models for anaphora resolution, which is essential for applications in natural language processing (NLP) tailored to Telugu and other regional languages.

Our work is a significant step towards enhancing NLP capabilities for Telugu, addressing the unique linguistic challenges presented by pronominal usage in dialogues. This research contributes to the broader field of computational linguistics and language technology, paving the way for more sophisticated language processing tools for regional languages.

4.6 Results

After constructing and tagging the corpus with all pronouns and their corresponding antecedents, the anaphora resolution algorithm was applied. The performance of the system was then evaluated by comparing its output with the manually tagged data. The evaluation metrics included the number of pronouns, unresolved pronouns, wrongly resolved pronouns, correctly resolved pronouns, and the accuracy of the system. The analysis was carried out based on the person of the referents.

	No. of pronouns	Pronouns unresolved	Pronouns wrongly resolved	Pronouns correctly resolved	Accuracy
1st person	138	11	14	113	81.88%
2nd Person	106	14	19	73	68.87%
3rd person	202	23	91	88	43.56%

Non personal pronouns	63	3	23	37	58.73%
Overall	509	51	147	311	61.10 %

The system shows a high accuracy rate of 81.88% for first-person pronouns. This suggests that the algorithm is relatively effective in identifying and resolving anaphora when the referent is the speaker or writer. First-person pronouns are generally less ambiguous, which may contribute to the higher resolution accuracy.

For second-person pronouns, the system's accuracy drops to 68.87%. Second-person pronouns often refer to the listener or reader, and the contextual cues necessary for accurate resolution may be more complex or less explicit than those for first-person pronouns. This complexity might explain the decrease in accuracy.

The accuracy for third-person pronouns is significantly lower at 43.56%. Third-person pronouns often refer to entities that are neither the speaker nor the listener, leading to higher ambiguity and more challenging resolution tasks. The substantial number of wrongly resolved pronouns (91 out of 202) highlights the difficulty the system faces in this category.

For non-personal pronouns, the system achieves an accuracy of 58.73%. Non-personal pronouns can refer to various entities, objects, or abstract concepts, which introduces a different kind of complexity compared to personal pronouns. The system's moderate performance in this category indicates that further refinement is needed to handle the diverse nature of non-personal pronouns effectively.

The overall accuracy of the system across all categories of pronouns is 61.10%. This suggests that while the system performs reasonably well, particularly with first-person pronouns, there is room for improvement, especially in resolving third-person and non-personal pronouns. The challenges in these categories likely stem from the greater ambiguity and contextual complexity associated with identifying the correct antecedents.

4.7 Issues in Telugu

The following are some of the problems we have faced:

4.7.1 Corpus Issues

Problem: Lack of Annotated Text and Dialogues

- **Explanation**: For effective anaphora resolution in any language, a large and well-annotated corpus is essential. However, for Telugu, there is a significant scarcity of such resources. Most of the data available is either in printed book format or as images on websites, making it challenging to digitize and annotate the text for computational use.

Impact:

- **Data Extraction**: The process of converting printed and image-based text into a machine-readable format is cumbersome and error-prone, limiting the availability of training data.

- **Annotation**: Even if the text is extracted, the lack of existing annotations means additional resources are needed to manually annotate the data, which is time-consuming and requires linguistic expertise.

4.7.2 Sandhi (Compound Words)

Problem: Difficulty in Parsing Due to Sandhi

- **Explanation**: Sandhi is a phonological process where sounds at word boundaries are altered due to morphological and phonetic rules. This phenomenon creates compound words that are challenging for parsers to break down and analyze. For example:
    - **Example 17**: "AmeMdukocindi" (she + why + came) – The pronoun "Ame" (she) is part of a compound word, making it difficult for the parser to identify it as a separate entity.
    - **Example 18**: "rAmuDekkaDunnADu" (Ramu + where + present) – The noun "rAmuDu" (Ramu) is embedded in a compound structure.

Impact:

- **Anaphora Resolution**: Difficulty in identifying and separating components within compound words reduces the accuracy of finding antecedents and resolving anaphora.
- **Parser Accuracy**: Parsers struggle with compound words, resulting in lower overall system accuracy.

4.7.3 Subject Identification

Problem: Free Word Order and Morphological Complexity

- **Explanation**: Telugu's free word order and complex morphology pose significant challenges. The morph analyzer currently used processes words individually without considering their syntactic roles, leading to difficulties in identifying the subject of sentences. For instance:
    - **Example 19**: "sIta" (Seetha) is the subject of the sentence and feminine, but the algorithm fails to identify this, leading to incorrect pronoun mapping ("ataniki" instead of a feminine pronoun).

Impact:

- **Pronoun Accuracy**: Incorrect identification of subjects results in improper gender and number agreement for pronouns, reducing the reliability of the anaphora resolution system.
- **Case Identification**: Failure to determine the case of words hampers the overall understanding of sentence structure.

4.7.4 Parser Errors

Problem: Errors in Morph Analyzer and POS Tagger

- **Explanation**: The morph analyzer and POS tagger used for Telugu have significant error rates. These tools are critical for breaking down sentences into their grammatical components, but their inaccuracies propagate through the anaphora resolution system.

Impact:

- **Error Propagation**: Errors at the parsing stage affect subsequent stages, leading to incorrect anaphora resolution.
- **System Accuracy**: The overall accuracy of the anaphora resolution system is compromised due to the foundational errors in parsing.

4.7.5 Summary

To address these challenges, the following measures could be considered:

- **Corpus Development**: Invest in creating a digitized and annotated corpus for Telugu dialogues.
- **Sandhi Splitter**: Develop or improve tools specifically designed to handle Sandhi, enhancing parser capabilities.
- **Advanced Morph Analyzers**: Utilize or develop morph analyzers and POS taggers that account for the free word order and morphological richness of Telugu.
- **Subject Identification Algorithms**: Implement algorithms that can accurately identify subjects within sentences, improving pronoun resolution accuracy.

By tackling these issues, the effectiveness and accuracy of anaphora resolution systems for Telugu can be significantly improved.

Chapter 5

Conclusions and Future Scope

In this thesis, we present our pioneering effort in resolving pronominal anaphora in Telugu dialogues. We developed a rule-based algorithm tailored specifically for the unique morphological and syntactic properties of Telugu, a Dravidian language characterized by its free word order and morphological richness. Our algorithm involves parsing dialogues, creating a knowledge base of pronouns with their morphological information, and identifying antecedents using a set of rules. Given the absence of a pre-existing annotated corpus for Telugu, we built a new corpus consisting of human-to-human conversations, which included various forms of pronouns. We tested the algorithm on this corpus, achieving an overall accuracy of 61.1%. Our work underscores the complexity of anaphora resolution in Telugu, especially given the language's morphological richness and free word order, which present significant challenges for natural language processing (NLP) applications.

We also highlighted several key issues impacting accuracy, such as the difficulty in analyzing compound words (sandhi), the challenges in subject identification due to free word order, and the limitations of existing morph analyzers and POS taggers. These challenges need to be addressed to improve the performance of anaphora resolution systems for Telugu and potentially other morphologically rich languages.

Future Scope

Future research can expand in several promising directions. First, the resolution of cataphora, where a pronoun refers to an entity mentioned later in the discourse, is an important area to explore given the high likelihood of such occurrences in Telugu due to its free word order. Additionally, integrating this anaphora resolution system into dialogue systems, particularly those being developed for the tourism domain in Telugu, could enhance their functionality and user interaction. Improving the accuracy of third-person pronouns, which currently shows lower performance, is another critical area for enhancement. Implementing a robust sandhi splitter will address issues related to compound words and further improve system accuracy. Finally, developing annotated text and dialogue corpora for Telugu is essential for better training and evaluation of anaphora resolution systems. This will provide a more comprehensive dataset for refining algorithms and achieving higher accuracy in resolving pronominal anaphora in Telugu dialogues.

Related Publications

Hemanth Reddy, J., Mamidi, R. (2015). Resolution of Pronominal Anaphora for Telugu Dialogues. Presented at International Conference on Natural Language Processing(ICON) 2015, Kerala.

Bibliography

[1] **Hobbs, J.R.** (1978). Resolving Pronouns. *Proceedings of the 1978 Workshop on Theoretical Issues in Natural Language Processing - TINLAP '78*.

[2] **Lappin, S., & Leass, H.J.** (1994). An Algorithm for Pronominal Anaphora Resolution. *Computational Linguistics*, 20(4), 535-561.

[3] **Ng, H.T., & Lim, C.Y.** (2001). A Case Study on Learning Pronoun Resolution from the Web. *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*.

[4] **Ng, V., & Cardie, C.** (2002). Improving Machine Learning Approaches to Coreference Resolution. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

[5] **Soon, W.M., Ng, H.T., & Lim, D.C.Y.** (2001). A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Computational Linguistics*, 27(4), 521-544.

[6] **Baldwin, B.** (1997). CogNIAC: High Precision Coreference with Limited Knowledge and Linguistic Resources. *Proceedings of the ACL Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*.

[7] **Kehler, A.** (2002). Coherence, Reference, and the Theory of Grammar. *CSLI Publications*.

[8] **Webber, B., & Joshi, A.** (1982). Taking the Initiative in Natural Language Database Interactions: Justifying Why. *Proceedings of the 20th Annual Meeting of the Association for Computational Linguistics*.

[9] **Grosz, B.J., Weinstein, S., & Joshi, A.K.** (1995). Centering: A Framework for Modelling the Local Coherence of Discourse. *Computational Linguistics*, 21(2), 203-225.

[10] **Poesio, M., & Vieira, R.** (1998). A Corpus-Based Investigation of Definite Description Use. *Computational Linguistics*, 24(2), 183-216.

[11] **Mitkov, R.** (2002). Anaphora Resolution. *Longman*.

[12] **Harabagiu, S., & Maiorano, S.** (2000). Multilingual Coreference Resolution. *Proceedings of the ANLP/NAACL 2000 Workshop on Acquisition of Lexical Knowledge from Text*.

[13] **Mitkov, R., & Strube, M.** (2001). Evaluation Metrics for Anaphora Resolution. *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*.

[14] **Bean, D., & Riloff, E.** (1999). Corpus-Based Identification of Non-Anaphoric Noun Phrases. *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*.

[15] **Hirst, G.** (1981). Anaphora in Natural Language Understanding: A Survey. *Springer-Verlag*.

[16] **McEnery, T., & Wilson, A.** (1996). Corpus Linguistics. *Edinburgh University Press*.

[17] **Poesio, M., & Artstein, R.** (2008). Anaphoric Annotation in the ARRAU Corpus. *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*.

[18] **Mikheev, A.** (1997). Automatic Rule Induction for Unknown-Word Guessing. *Computational Linguistics*, 23(3), 405-423.

[19] **Zhou, G., & Su, J.** (2004). A High-Performance Coreference Resolution System Using a Constraint-Driven Learning Approach. *Computational Linguistics*, 30(4), 401-444.

[20] **Ng, V.** (2005). Machine Learning for Entity Detection and Tracking. *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*.

[21] **Stuckardt, R.** (2002). Robust Anaphor Resolution. *Springer-Verlag*.

[22] **Soon, W.M., & Ng, H.T.** (2001). A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Computational Linguistics*, 27(4), 521-544.

[23] **Mitkov, R.** (1998). Robust Pronoun Resolution with Limited Knowledge. *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (ACL/COLING-98)*.

[24] **Tetreault, J.** (2001). A Corpus-Based Evaluation of Centering and Pronoun Resolution. *Computational Linguistics*, 27(4), 507-520.

[25] **Morton, T.S.** (2000). Coreference for NLP Applications. *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*.
[26] **Hardmeier, C.** (2014). Discourse in Statistical Machine Translation. *Linguistic Issues in Language Technology*, 12(1).
[27] **Bagga, A., & Baldwin, B.** (1998). Algorithms for Scoring Coreference Chains. *Proceedings of the First International Conference on Language Resources and Evaluation (LREC)*.
[28] **Ng, V., & Cardie, C.** (2003). Weakly Supervised Natural Language Learning Without Redundant Views. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics*.
[29] **Raghunathan, K., Lee, H., Rangarajan Sridhar, A., Chambers, N., Surdeanu, M., Jurafsky, D., & Manning, C.D.** (2010). A Multi-Pass Sieve for Coreference Resolution. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
[30] **Pradhan, S., Elwell, R., Vemulapalli, K., Xue, N., & Ng, H.T.** (2009). Resolving Pronouns Robustly: Pitfalls and Remedies. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
[31] **Chen, J., & Ji, H.** (2011). Collaborative Ranking: A Case Study on Entity Linking. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
[32] **Durrett, G., & Klein, D.** (2013). Easy Victories and Uphill Battles in Coreference Resolution. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
[33] **Haghighi, A., & Klein, D.** (2010). Coreference Resolution in a Modular, Entity-Centered Model. *Proceedings of the 2010 Human Language Technologies Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*.
[34] **Stoyanov, V., & Eisner, J.** (2012). Easy-first Coreference Resolution. *Proceedings of the 2012 Conference on Computational Natural Language Learning (CoNLL)*.
[35] **Luo, X.** (2005). On Coreference Resolution Performance Metrics. *Proceedings of the 2005 Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*.
[36] **Ponzetto, S.P., & Strube, M.** (2006). Exploiting Semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution. *Proceedings of the 2006 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*.
[37] **Versley, Y., Ponzetto, S.P., Poesio, M., Eidelman, V., & Jern, B.** (2008). BART: A Modular Toolkit for Coreference Resolution. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Demo Session*.
[38] **Durrett, G., Hall, D., & Klein, D.** (2014). Decentralized Entity-Level Modeling for Coreference Resolution. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.
[39] **Wiseman, S., Rush, A.M., Shieber, S., & Weston, J.** (2015). Learning anaphoricity and antecedent ranking features for coreference resolution. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*.
[40] **Luo, X., Pradhan, S., Recasens, M., Hovy, E.H., & Ma, X.** (2014). Coreference Resolution: A Review of General Methodologies and Applications. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.
[41] **Poon, H., Cherry, C., & Toutanova, K.** (2011). Unsupervised Inducing of Sentiment Roles from Opinion Text. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
[42] **McCarthy, J.F., & Lehnert, W.G.** (1995). Using Decision Trees for Coreference Resolution. *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*.
[43] **Aone, C., & Bennett, S.W.** (1995). Evaluating Automated and Manual Acquisition of Anaphora Resolution Strategies. *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

[44] **Connolly, D., Burger, J., & Day, D.** (1994). A Machine Learning Approach to Anaphoric Reference. *Proceedings of the ARPA Human Language Technology Workshop*.

[45] **Ge, N., Hale, J., & Charniak, E.** (1998). A Statistical Approach to Anaphora Resolution. *Proceedings of the Sixth Workshop on Very Large Corpora*.

[46] **Strube, M.** (1998). Never Look Back: An Alternative to Centering. *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (ACL/COLING-98)*.

[47] **Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K.J.** (1990). Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4), 235-244.

[48] **Palmer, M.S., Gildea, D., & Kingsbury, P.** (2005). The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1), 71-106.

[49] **Sundheim, B.** (1995). Overview of Results of the MUC-6 Evaluation. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*.

[50] **Vilain, M., Burger, J., Aberdeen, J., Connolly, D., & Hirschman, L.** (1995). A Model-Theoretic Coreference Scoring Scheme. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*.