RSET
RAJAGIRI SCHOOL OF
ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

Mini Project Report On

Malayalam Parser for Dataset Creation

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science & Engineering

By

Fathima Jennath N.K (U2103089)

Gautham C Sudheer (U2103092)

Godwin Gino (U2103096)

Mohammed Basil (U2103139)

Under the guidance of

Dr.Mary Priya Sebastian

Department of Computer Science & Engineering
Rajagiri School of Engineering & Technology (Autonomous)
(Affiliated to APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039
May 2024

CERTIFICATE

This is to certify that the mini project report entitled ***"Malayalam Parser for Dataset Creation"*** *is a bonafide record of the work done by* ***Fathima Jennath N.K (U2103089)***, ***Gautham C Sudheer (U2103092)***, ***Godwin Gino (U2103096)***, ***Mohammed Basil (U2103139)***, *submitted to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2023-2024.*

Dr.Mary Priya Sebastian
Associate Professor
Dept. of CSE
RSET

Dr.Saritha S
Professor
Dept. of CSE
RSET

Dr.Preetha K.G
Head of the Department
Dept. of CSE
RSET

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude towards Dr P. S. Sreejith, Principal of RSET, and Dr. Preetha K.G., Head of the Department of Computer Science and Engineering for providing me with the opportunity to undertake my mini project, "Malayalam Parser for Dataset Creation".

I am highly indebted to my project coordinators, **Dr.Saritha S**, Professor, Department of Computer Science and Engineering for their valuable support.

It is indeed my pleasure and a moment of satisfaction for me to express my sincere gratitude to my project guide **Dr.Mary Priya Sebastian** for her patience and all the priceless advice and wisdom she has shared with me.

Last but not the least, I would like to express my sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Fathima Jennath NK

Gautham C Sudheer

Godwin Gino

Mohammed Basil

Abstract

The "Malayalam Parser for Dataset Creation" project aims to address the scarcity of annotated datasets in the Malayalam language for Natural Language Processing (NLP) applications. The primary objective is to develop a robust Malayalam parser capable of analyzing the syntactic and semantic structures of Malayalam sentences. The creation of this parser involves several key steps, including data collection from diverse sources, preprocessing to ensure data quality, and man-ual annotation of a representative subset of the data with grammatical and syntactic information. The parser development process encompasses the selection of an appropriate parsing approach,whether rule-based, statistical, or machine learning-based. The model is trained using the annotated Malayalam dataset, focusing on capturing the unique linguistic nuances of the Malayalam language. Evaluation metrics are employed to assess the parser's performance on a separate test set, guiding iterative refinement and enhancement. The resulting Malayalam parser serves as a valuable tool for the analysis of grammatical structures in new Malayalam text data. Its application contributes to the creation of high-quality Malayalam datasets, crucial for advancing NLP research and applications in the Malayalam language. This project encourages collaboration with linguists,researchers, and the Malayalam-speaking community to ensure linguistic accuracy and relevance in the development of the parser. The "Malayalam Parser for Dataset Creation" project aligns with the broader goal of promoting linguistic diversity in NLP, addressing the challenges posed by the scarcity of resources for underrepresented languages. Through the development of this parser, the project aims to facilitate further research and innovation in Malayalam NLP, opening avenues for the exploration of various language-related tasks and applications.

Contents

List of Figures

List of Tables

List of Abbreviations

- NLP - Natural Language Processing
- NER - Named Entity Recognition
- POS - Part-of-Speech

Chapter 1

Introduction

1.1 Background

Malayalam is spoken widely in Kerala and neighboring areas but has not received as much attention in the tech world as bigger languages like English. This lack of attention has led to a scarcity of tools for analyzing Malayalam text, despite its complex grammar and word forms.

Our project aims to address this issue by creating a specialized system for Malayalam that can understand and process Malayalam text more effectively than current tools. This system will facilitate tasks such as sentiment analysis, translation, and summarization, benefiting areas such as education and business.

We are designing our system to be adaptable and scalable, ensuring that it can evolve to meet diverse needs. Our ultimate goal is to establish a strong foundation for Malayalam language technology, paving the way for future improvements and innovations.

In short, our project focuses on using technology to make working with Malayalam text more efficient, enabling individuals to achieve more in various fields.

1.2 Problem Definition

The aim of the project is to create a Malayalam Parser for Dataset Creation, involving data collection, preprocessing, manual annotation, and training using various parsing approaches to address the scarcity of annotated datasets in Malayalam for NLP applications.

1.3 Scope and Motivation

Scope:

The Malayalam Parser project aims to develop an advanced tool capable of understanding

and processing Malayalam text efficiently. It encompasses essential tasks such as tokenization, part-of-speech tagging, parsing, and semantic analysis, providing a comprehensive breakdown of Malayalam sentences. The system's scope extends to facilitating tasks such as identifying different parts of speech and extracting meaningful insights from text. Additionally, the parser's design includes provisions for future expansion, allowing for the incorporation of domain-specific or specialized parsing tasks. This flexibility ensures that the parser can adapt to evolving needs and requirements, making it applicable across various domains and applications.

Motivation:

The motivation behind the Malayalam Parser project stems from the necessity for effective tools to process Malayalam text, essential for informed decision-making and strategic planning. By implementing parsing tasks such as tokenization, part-of- speech tagging, parsing, and semantic analysis, the system enables data-driven decision-making processes, enhancing strategic planning and execution.Furthermore, the project's innovation lies in the creation of annotated datasets,crucial for training and evaluating models for sentiment analysis, machine translation,question answering, and domain-specific parsing. Through these efforts, the project aims to advance natural language processing capabilities in Malayalam, contributing to the improvement of Malayalam language processing technologies and fostering innovation in linguistic research and technology development.

Parsing Tasks	**Description**
Tokenization	Breaking down sentences into individual words or tokens.
Part-of-Speech Tagging	Assigning grammatical tags to each word in a sentence.
Parsing	Analyzing the structure of sentences to understand their grammatical relationships.
Semantic Analysis	Extracting the meaning and context from sentences.

Table 1.1: Description of parsing tasks

1.4 Objectives

- Develop parsing algorithms to accurately identify linguistic components such as words, phrases, and sentences in Malayalam text, laying the foundation for compre-

hensive analysis.

- Implement functionalities to determine the grammatical structure, syntax, and semantics of Malayalam sentences, facilitating precise linguistic analysis and comprehension.

- Incorporate part-of-speech tagging, syntactic parsing, and semantic analysis capabilities tailored specifically for the Malayalam language, enabling detailed linguistic processing.

- Enhance the Malayalam Parser system to effectively handle compound words, inflections, and variations in word forms commonly encountered in Malayalam text, ensuring robust parsing capabilities.

- Ensure compatibility and interoperability with existing linguistic analysis frameworks, facilitating seamless integration and utilization of the Malayalam Parser within broader NLP applications.

- Continuously refine and optimize the Malayalam Parser system to improve efficiency, accuracy, and adaptability in analyzing and processing Malayalam text data.

1.5 Challenges

1. Morphological Complexity: Malayalam words can change a lot by adding suffixes, making it hard for the parser to figure out their basic forms and parts of speech. This makes it tough to understand the meaning and grammar of sentences.

2. Limited Resources: The scarcity of Malayalam-specific NLP tools and datasets complicates the development and training process. Adapting existing tools from other languages or creating new ones becomes necessary, potentially slowing down progress and hindering the effectiveness of the parser.

3. Syntactic Freedom: Malayalam's relatively flexible sentence structure allows for varied word orders, challenging parsing algorithms in determining precise word relationships. This freedom introduces complexity in identifying grammatical elements like subjects, objects, and verbs, especially when word order isn't a definitive indicator.

4. Data Annotation: The process of manually annotating data for parts of speech (POS), named entities, and intents is meticulous and time-consuming, requiring expertise in Malayalam grammar. It's crucial to ensure the quality and comprehensiveness of annotated training data for the parser's success, although this can be resource-intensive.

5. Model Selection and Training: Optimal performance of POS tagging, named entity recognition, and intent recognition tasks relies on choosing suitable algorithms and training them effectively. However, training on potentially limited or imbalanced datasets necessitates careful optimization, such as data augmentation and hyperparameter tuning, to mitigate any shortcomings.

1.6 Assumptions

1. Availability of Training Data: The project assumes a certain amount of high-quality Malayalam text data will be available for annotation and training the parser for POS tagging, NER, and intent recognition.

2. Effectiveness of Algorithms: The project assumes that the selected algorithms for POS tagging, NER, and intent recognition will be capable of accurately handling the linguistic complexities inherent in the Malayalam language.

3. Computational Resources: Adequate computational resources are assumed to be accessible for training the parser models, as this process can be computationally demanding.

4. Annotation Quality: The project assumes the quality of the annotations in the training data, including accuracy and consistency, as these factors greatly influence the performance of the parser models.

1.7 Societal / Industrial Relevance

The project aims to preserve and promote Malayalam, enhancing understanding and analysis of Malayalam text, providing datasets for parts-of-speech tagging, named entity recognition, and intent recognition. It supports research and learning in NLP and

Malayalam linguistics, improving access to information for Malayalam speakers online. The project also supports the development of local language technologies for industries in Malayalam-speaking regions, expanding market opportunities in e-commerce and social media.

1.8 Organization of the Report

The organization of the report are as follows:

- **Chapter 1-Introduction:**The introduction covers the background of the project, the problem definition, the scope and motivation, the objectives,the societal and industrial relevance, the assumptions and the challenges faced by the project.
- **Chapter 2-Software Requirements Specification:** This chapter outlines the functional and nonfunctional requirements of the NLP tools for Malayalam. It defines the overall description of the software, external interface requirements, system features, and other nonfunctional requirements necessary for the development and deployment of the tools.
- **Chapter 3-System Architecture and Design:** The system architecture and design chapter provides an overview of the project's technical framework. It includes discussions on the system overview, architectural design, identified datasets, proposed algorithms, implementation strategies, module division, and a work schedule presented as a Gantt chart for project planning and management.

Chapter 2

Software Requirements Specification

Insert your SRS document here.

2.1 Introduction

Figure 2.1: Insert your images here, and provide necessary captions

2.2 Overall Description

2.3 External Interface Requirements

You can insert equations into your file using the below code:

$$a = b + c \tag{2.1}$$

$$= y - z \tag{2.2}$$

2.4 System Features

2.5 Other Nonfunctional Requirements

Chapter 3

System Architecture and Design

3.1 System Overview

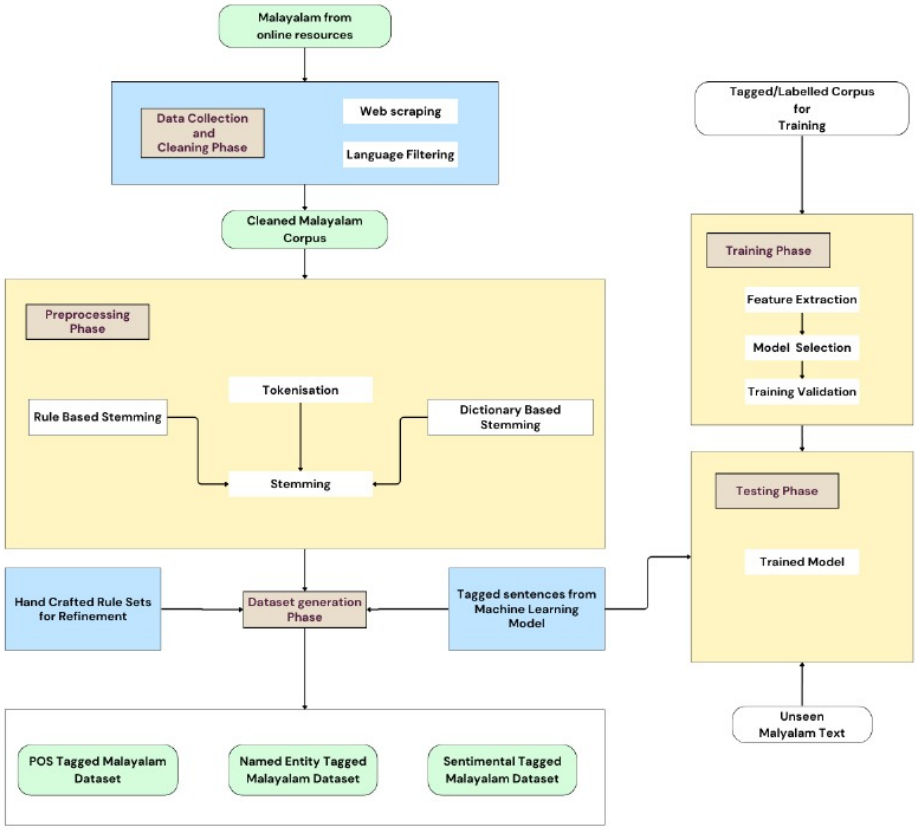

Figure 3.1: Architecture diagram

The process of developing a Malayalam parser involves several key stages to effectively process Malayalam text for tasks such as sentiment analysis, named entity recognition, and part-of-speech (POS) tagging. It begins with data collection, where Malayalam text is gathered from online sources using web scraping techniques. This collected text is then cleaned to remove any irrelevant information, errors, or special characters. In the cleaning

phase, language filtering is applied to ensure the text is in Malayalam. Following data cleaning, the text undergoes preprocessing, which include tokenization to break the text into individual words or tokens. Stemming is then applied to reduce inflected words to their base form, simplifying the text for analysis. A crucial step is building a training corpus of preprocessed Malayalam text labeled with the desired information, such as sentiment labels, named entities, and part-of-speech tags. Features are extracted from the preprocessed text and serve as inputs to the machine learning model. A suitable machine learning model is selected and trained on the extracted features and labeled training data. The model's performance is evaluated on a separate set of data to assess its accuracy and effectiveness to new, unseen data. Finally, the trained model is used to process new Malayalam text, tagging it with sentiment labels, named entities, part-of-speech tags, or other relevant information. Linguists can add custom rules to refine the model's output for better accuracy, particularly in cases where the model may not perform optimally. In summary, the development of a Malayalam parser involves collecting, cleaning, and preprocessing text, building a training corpus, extracting features, training and evaluating a machine learning model, and processing new text. This process enables the effective analysis and processing of Malayalam text for various natural language processing tasks, contributing to the advancement of language technology in the Malayalam language.

3.2 Dataset identified

This section describes the data source used in the project. Brief its properties and refer it to the appropriate location. Sample subsets of the dataset can be highlighted.

3.3 Proposed Methodology/Algorithms

This section describes in detail the methodologies or algorithms associated with your work. Algorithms should be written in appropriate format.

3.4 User Interface Design

The user interface design (wireframe designs) can be highlighted in this section. The figures titles should be in a chronological order and self explanatory.

3.5 Database Design

The detailed database design and its schema is expected in this section. The database used in the work can be mentioned here. The reason for choosing the database can be substantiated in this section.

3.6 Description of Implementation Strategies

The implementation strategies for the project are as follows:

- Data Acquisition and Cleaning:
    1. Web Scraping:
        - Utilize libraries like BeautifulSoup or Scrapy (Python) to scrape relevant Malayalam websites.
        - Define target URLs and develop scraping logic for text extraction.
        - Implement techniques to handle pagination or dynamic content loading (if applicable).
    2. Language Filtering:
        - Implement language detection using libraries like langdetect or textblob (Python) to identify non-Malayalam text.
        - Set a threshold or confidence score to filter out content below a certain level of Malayalam probability.
- Data Preprocessing:
    1. Tokenization: Choose a suitable tokenization method (word-based, sentence-based, etc.) using libraries like NLTK or spaCy (Python).
    2. Handling Non-Textual Elements: Develop logic for managing emojis or other non-textual elements (e.g., removal, special token representation).
    3. Stop Word Removal: Implement stop word removal based on a created or located Malayalam stop word list.
    4. Stemming or Lemmatization: Use libraries like NLTK or spaCy for stemming (reducing words to root forms) or lemmatization (finding dictionary base

forms). Explore specific stemming/lemmatization algorithms if necessary for Malayalam.

- Sentiment Analysis:

    1. Model Selection (if applicable): Consider factors like data size, desired accuracy, and computational resources when choosing a model (e.g., Logistic Regression, Naive Bayes, SVM, RNNs).
    2. Data Splitting (if applicable): Split the preprocessed data into training, validation, and testing sets for model development.
    3. Model Training and Tuning (if applicable): Implement hyperparameter tuning to optimize model performance during training.
    4. Evaluation (if applicable): Evaluate the trained sentiment analysis model's performance on the unseen testing data set. Analyze the results and identify areas for improvement in data quality or model architecture.

- Additional Considerations:

    1. Data Storage and Management: Consider implementing data loading/saving functions for various formats (text files, CSV, etc.) if needed for further analysis.
    2. Version Control: Utilize a version control system (e.g., Git) to track code changes and facilitate collaboration.
    3. Testing: Implement unit tests for critical functions and integration tests to ensure the entire NLP pipeline functions as expected.
    4. Logging: Implement logging functionalities to track code execution progress and identify potential issues.

3.7 Module Division

This section describes the different modules involved in this project and a small description of the same is expected. This section ends with the information of which module is assigned to each project member.

3.8 Work Schedule - Gantt Chart

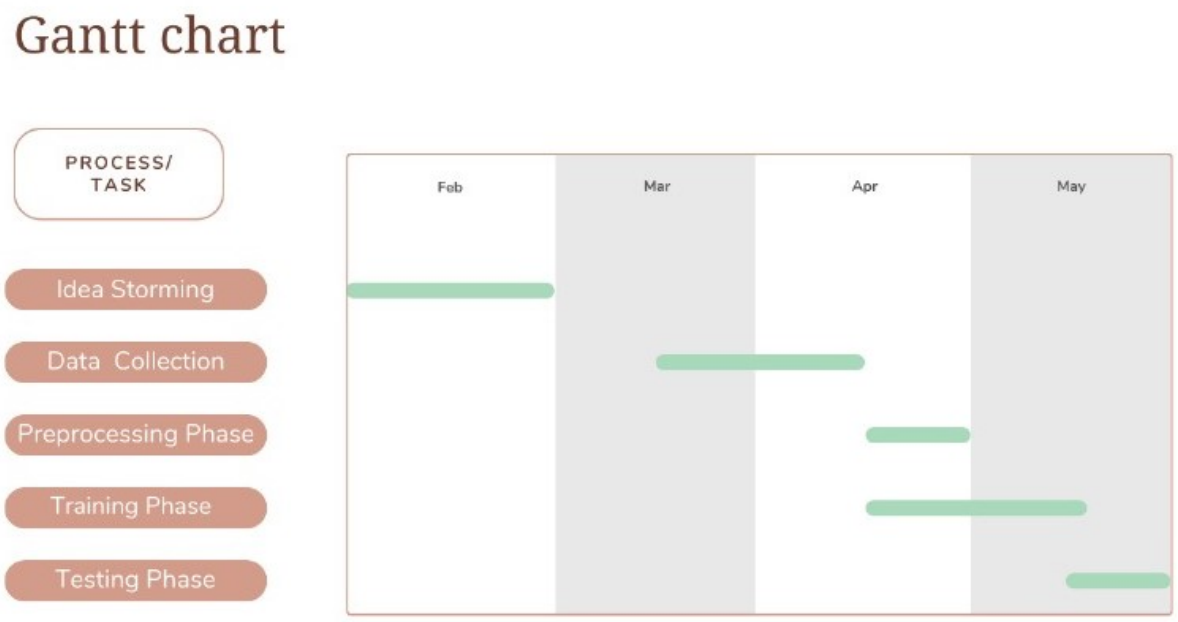

Figure 3.2: Gantt chart

Bibliography

[1] Asopa, S., and Sharma, N. (2021) A Hybrid Parser Model for Hindi Language. Indian Journal of Computer Science and Engineering (IJCSE), Vol. 12(1).

[2] Chen, D., and Manning, C. D. (2014). A Fast and Accurate Dependency Parser using Neural Networks. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).

[3] Nair, L. R. (2013). Language Parsing and Syntax of Malayalam Language. 2nd International Symposium on Computer, Communication, Control and Automation (3CA 2013).

[4] Berger, A. L., Della Pietra, V. J., and Della Pietra, S. A. (1996). A Maximum Entropy Approach to Natural Language Processing. Association for Computational Linguistics, Vol 22(1).

[5] Mestry, A., Shende, S., Mahadik, A., and Virnodkar, S. (2014). A Parser: Simple English Sentence Detector and Correction. International Journal of Engineering Research and Technology (IJERT).

[6] Sethi, N., Agrawal, P., Madaan, V., and Singh, S. K. (2016). A Novel Approach to Paraphrase Hindi Sentences using Natural Language Processing. Indian Journal of Science and Technology, Vol 9(28).

[7] Smith, D. A., and Eisner, J. (2008). Dependency Parsing by Belief Propagation. Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Page 145- 156.

[8] Bharati, A., Kulkarni, A., and Chaudhury, S. (2007). English Parsers: Some Information- based Observations.

[9] Jayan, J. P., and R, R. (2009). A Morphological Analyzer for Malayalam - A Comparison of Different Approaches. International Journal of Computer Science and Information Technology. Vol 2(2), Page 155-160.

[10] Vaidya, A., Choi, J. D., Palmer, M., and Narasimhan, B. (2011). Analysis of the Hindi Proposition Bank using Dependency Structure. Proceedings of the Fifth Law Workshop (LAW V), Page 21-29.

[11] Rajan, M., T.S, R., and Bhojane, V. (2014). Information Retrieval in Malayalam Using Natural Language Processing. International Journal of Scientific and Engineering Research, Vol 5(6)

[12] Rajan, M., Thirumalai, R., and Kumar, V. (2006). Development of a Tamil Parser using Natural Language Processing Techniques. A survey of the state of the art in tamil language technology Vol 6(10).

[13] Venkatesh, R., Kumar, S., and Arumugam, P. (2014). Building a Lexical Analyzer for Tamil Texts using NLP Approaches. 2014 International Conference on Advances in ICT for Emerging Regions (ICTer).

[14] Thavareesan, S., and Mahesan, S. (2019). Sentiment Analysis in Tamil Texts: A Study on Machine Learning Techniques and Feature Representation.2019 IEEE 14th Conference on Industrial and Information Systems (ICIIS).

[15] Pai, T. V., Devi, J. A., and Aithal, P. S. (2020). A Systematic Literature Review of Lexical Analyzer Implementation Techniques in Compiler Design. International Journal of Applied Engineering and Management Letters (IJAEML), Vol 4(2), Page 285-301.

[16] Simmons, R. F., and Burger, J. F. (1968). A Semantic Analyzer for English Sentences. Mechanical Translation and Computational Linguistics, Vol 11.