Malayalam Parser for Dataset Creation

Design Presentation

Guided by:
Dr. Mary Priya Sebastian

Fathima Jennath
Gautham C Sudheer
Godwin Gino
Mohammed Basil

Contents

Introduction

- Importance of Natural Language Processing (NLP) in regional languages
- Focus on the specific relevance of Malayalam in the context of NLP applications.
- Challenges associated with the scarcity of annotated datasets.
- Analyze both the syntactic and semantic structures of Malayalam sentences
- Applications such as sentiment analysis, named entity recognition, etc.
- Potential impact on advancing research and applications specific to the Malayalam

Problem Definition

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.

Objectives

- Data Collection
  a) Gather text data from diverse sources in Malayalam language
  b) Aim for a sufficient volume of data to represent the language's usage patterns adequately
- Data Preprocessing
  a) Perform tokenization, normalization, and cleaning of the collected data
  b) Handle any inconsistencies or noise in the data to ensure quality

Objectives

- Manual Annotation
  a. Annotate a representative subset of the preprocessed data with grammatical and syntactic information
  b. Employ linguistic experts or proficient annotators to ensure accurate annotations.
- Parser Development
  a. Train the parser using the annotated dataset to understand Malayalam syntax and semantics

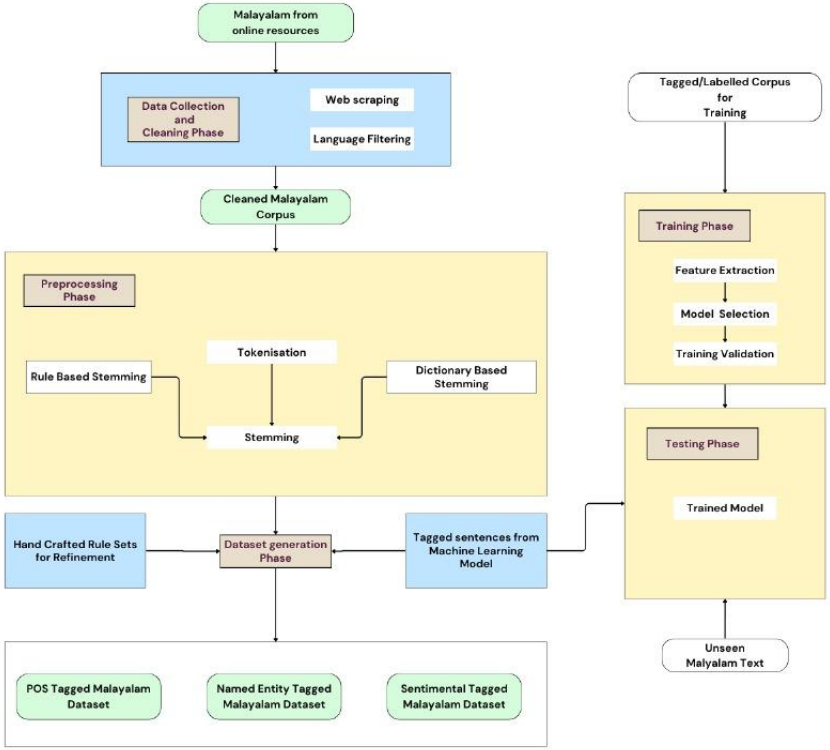
Functional Requirements

- Parse and analyze Malayalam language text to identify linguistic components such as words, phrases, and sentences.
- Determine grammatical structure, syntax, and semantics of Malayalam sentences to facilitate accurate linguistic analysis.
- Provide functionality for part-of-speech tagging, syntactic parsing, and semantic analysis tailored for the Malayalam language.

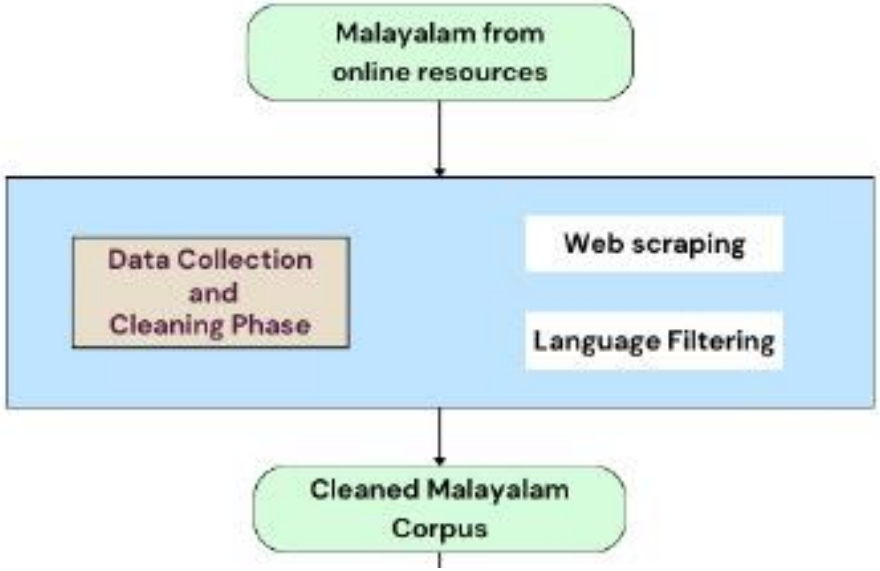
Functional Requirements

- Support for handling compound words, inflections, and variations in word forms commonly found in Malayalam text.
- Generation of a part-of-speech tagged dataset, named entity dataset, and sentimental tagged dataset, contributing to the advancement of language processing technologies in Malayalam
- Implement a user-friendly interface that allows users to input Malayalam text for analysis

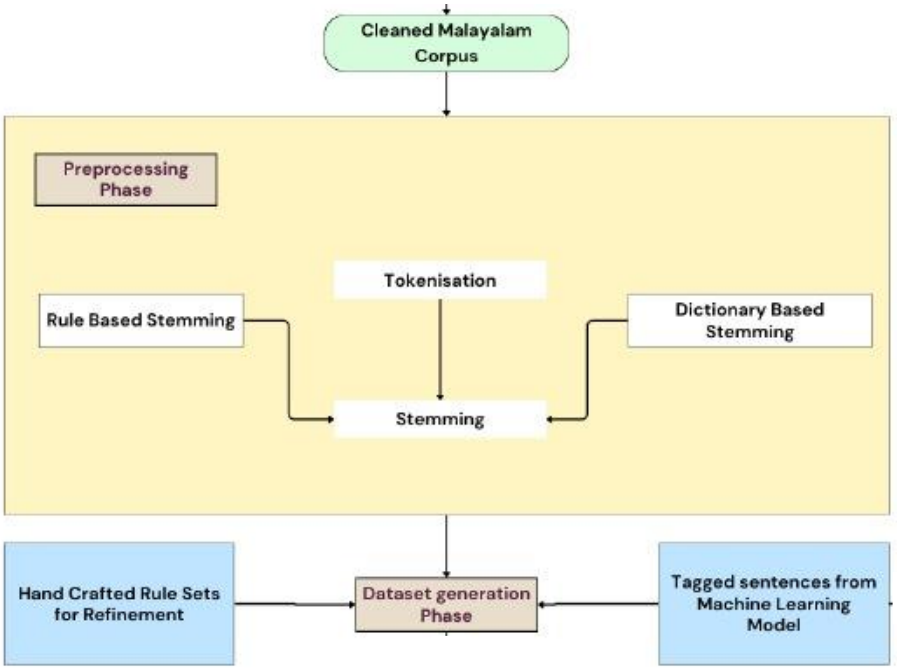
System Design



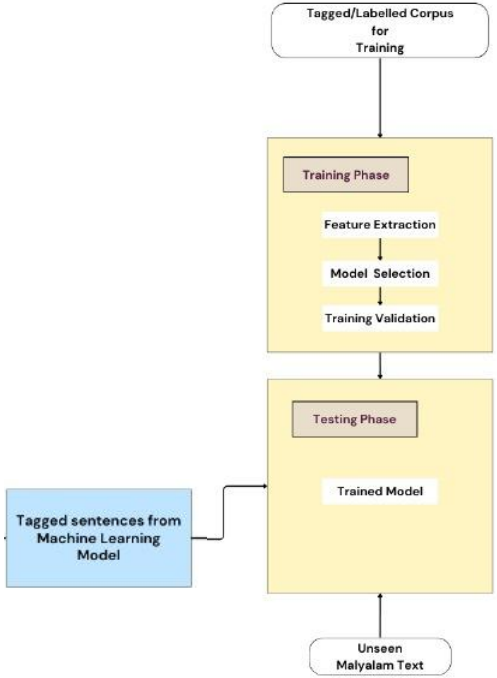
Data Collection and Cleaning Phase



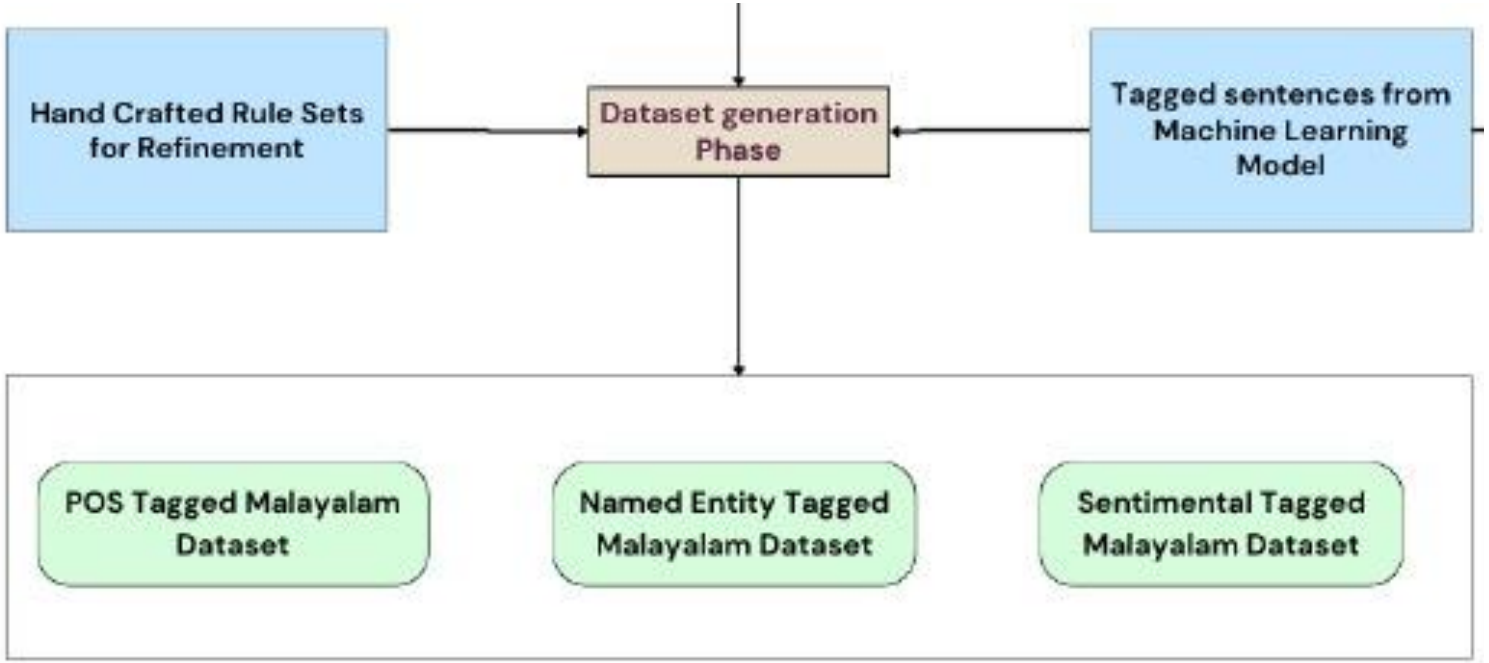
Data Preprocessing



Training and Testing



Output


Hand Crafted Rule Sets for Refinement
Dataset generation Phase
Tagged sentences from Machine Learning Model
POS Tagged Malayalam Dataset
Named Entity Tagged Malayalam Dataset
Sentimental Tagged Malayalam Dataset

Algorithm

1. Data Collection:
   Use web scraping techniques to gather Malayalam text data from online sources.
   Apply language filtering to ensure only Malayalam text is retained.
   Store the collected data in a structured format for further processing.

2. Data Cleaning:
   Remove irrelevant information or non-textual content.
   Eliminate errors and inconsistencies, such as misspellings or formatting issues.
   Filter out special characters or symbols that do not contribute to the linguistic content.

3. Preprocessing:
   Tokenization: Split the cleaned text into individual words or tokens.
   Stemming: Reduce inflected words to their base form to simplify the text for analysis.

Algorithm

4. Annotation:
    Manually annotate a subset of the preprocessed data with desired linguistic information
    Use linguistic expertise to ensure the accuracy and consistency of annotations.

5. Feature Extraction:
    Extract relevant features from the annotated data
    Word embeddings, syntactic features, or semantic features.
    Design feature representations that capture linguistic properties essential for the parsing.

6. Model Selection:
    Rule-set generation
    Rule-based parsing and machine learning-based parsing.

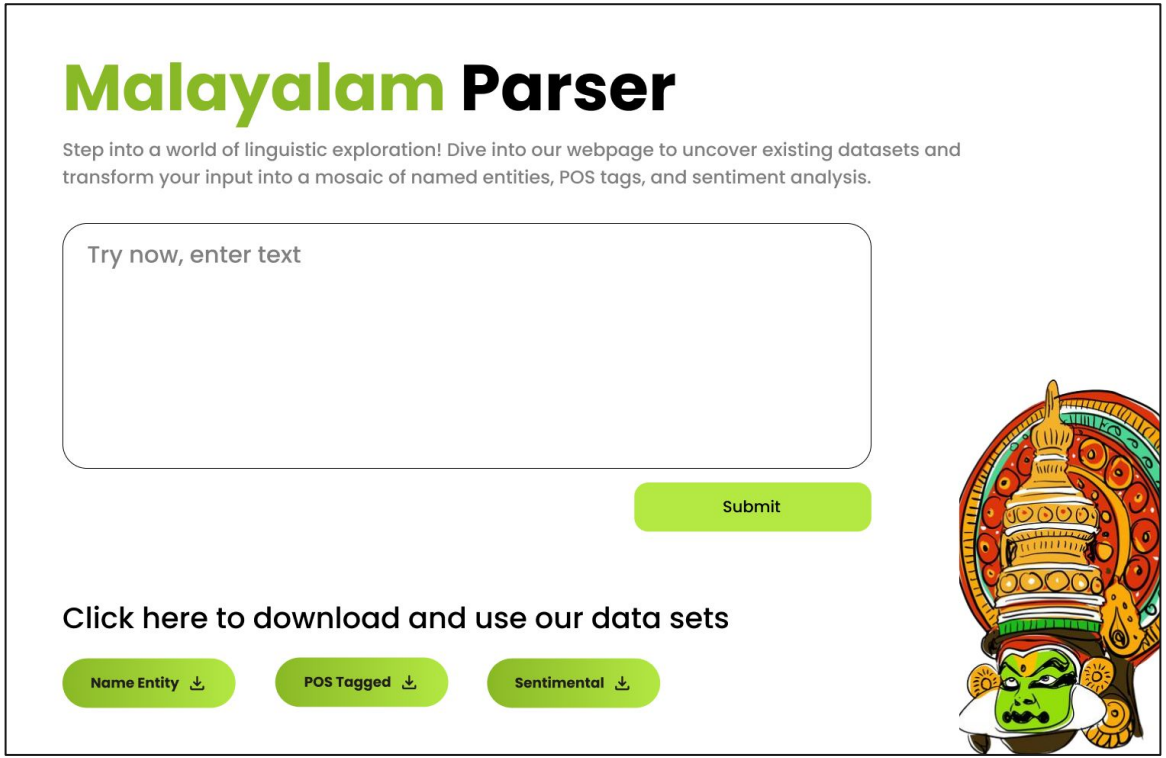
Algorithm

7. Model Training:
    Train the selected parsing model using the annotated data and extracted features.
    Optimize model parameters and hyperparameters to improve performance.
    Validate the model using cross-validation techniques to ensure generalization

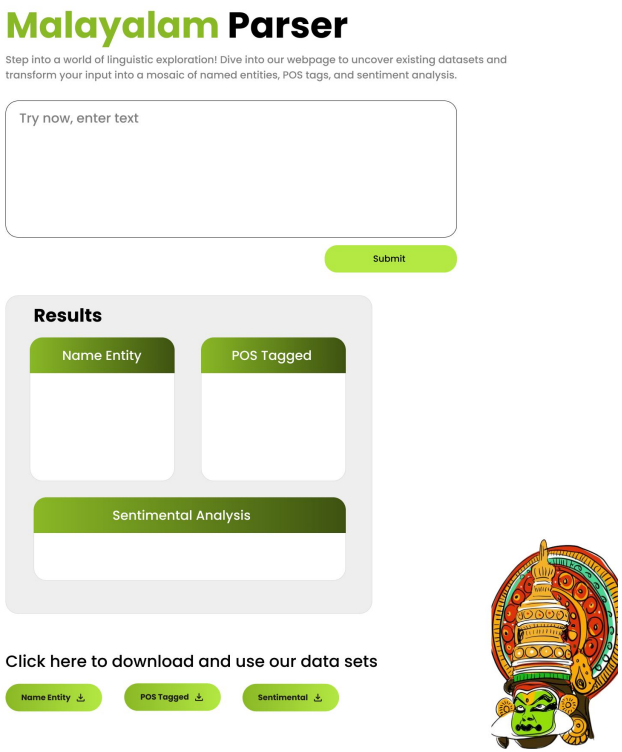
8. Evaluation:
    Evaluate the trained model's performance on a separate test set
    Analyze the model's strengths and weaknesses to identify areas for improvement.

9. Refinement:
    Refine the parsing model based on the evaluation results and feedback
    Iteratively improve the model's accuracy and robustness

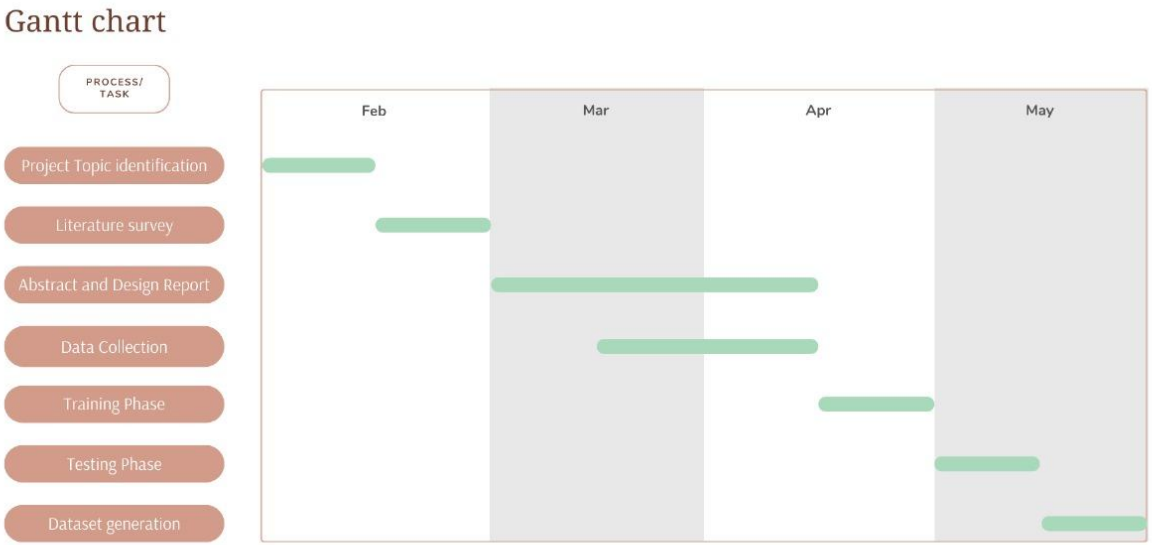
UI Design


Malayalam Parser
Step into a world of linguistic exploration! Dive into our webpage to uncover existing datasets and transform your input into a mosaic of named entities, POS tags, and sentiment analysis.
Try now, enter text
Submit
Click here to download and use our data sets
Name Entity
POS Tagged
Sentimental

UI Design


Malayalam Parser
Step into a world of linguistic exploration! Dive into our webpage to uncover existing datasets and transform your input into a mosaic of named entities, POS tags, and sentiment analysis.
Try now, enter text
Submit
Results
Name Entity
POS Tagged
Sentimental Analysis
Click here to download and use our data sets
Name Entity
POS Tagged
Sentimental

Work Division



Software / Hardware Requirements

- Windows 10 or later
- MacOS 10.13 High Sierra or later
- Ubuntu 18.04 LTS or later
- A modern processor (e.g., Intel Core i5 or equivalent)
- Sufficient RAM (at least 4GB)
- Available storage space for software installation
- Python (version 3.6 or later)
- Other programming languages and frameworks suitable for NLP development like NLTK, spaCy, scikit-learn, TensorFlow, etc. may be necessary

Conclusion

A comprehensive Malayalam language processing tool facilitating accurate linguistic analysis and dataset generation for NLP applications.

- Parsing and analysis of Malayalam text, enabling identification of linguistic components and determination of grammatical structure, syntax, and semantics
- Generates part-of-speech tagged, named entity, and sentiment-tagged datasets
- Contribute significantly to the advancement of language processing technologies in Malayalam.

References

- Asopa, S., and Sharma, N. (2021) A Hybrid Parser Model for Hindi Language. Indian Journal of Computer Science and Engineering (IJCSE), Vol. 12(1).
- 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).
- Nair, L. R. (2013). Language Parsing and Syntax of Malayalam Language. 2nd International Symposium on Computer, Communication, Control and Automation (3CA 2013).
- 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).
- 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).

References

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

References

- 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)
- 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).
- 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).
- 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).

References

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

Thank you