EVOLUTIONARY COMPUTATION PROJECT

INTENT DETECTION AND SLOT FILLING

Submitted by

Smriti Priyadarshani(23DP0098)
Aman Rawat(20JE0110)
Suryaghya Saha(20JE0998)
Tanush Garg(20JE1020)
Thamatam Pradeep(21JE0990)
Sai Pranav(21JE0279)
Cherukuri Madhulika(21JE0264)
Adari Sindhu Sree Sameera(21JE0032)
Gaddam Pallavi(21JE0338)
Aman Raj(21JE0087)

**Department of Computer Science and Engineering
Indian Institute Of Technology (Indian Schools of Mines), Dhanbad, Jharkhand - 826004, India**

PROBLEM STATEMENT

The objective of this project is to develop a system for intent detection and slot filling in Hindi language using translated datasets such as ATIS (Airline Travel Information System), MTOP (Movie Ticket Ordering Process), or SNIPO (Social Network Interaction Process). Additionally, the optimization of the system will be achieved by employing evolutionary algorithms such as Multi-Objective Genetic Algorithm (MOGA) or Genetic Algorithm (GA).

INTRODUCTION

Natural Language Understanding (NLU) plays a pivotal role in enabling human-computer interaction, automated customer service, and various other applications. One fundamental aspect of NLU is intent detection, which involves identifying the purpose or goal behind a user's input, and slot filling, which involves extracting specific pieces of information (slots) from the user's utterance. These tasks are essential for developing efficient and user-friendly conversational agents, chatbots, and virtual assistants.

While significant progress has been made in NLU for languages such as English, extending these advancements to languages like Hindi presents unique challenges. Hindi, being one of the most widely spoken languages in the world, demands robust NLU systems to cater to its diverse user base across various domains. However, the scarcity of labelled datasets and linguistic complexities pose significant obstacles in developing effective NLU systems for Hindi.

To address these challenges, this project focuses on developing an intent detection and slot filling system tailored for Hindi language. Leveraging translated datasets such as ATIS (Airline Travel Information System), MTOP (Movie Ticket Ordering Process), or SNIPO (Social Network Interaction Process), the project aims to create a robust framework for understanding user intents and extracting relevant information from their queries.

REQUIREMENTS

- Translated Hindi datasets (ATIS/MTOP/SNIPO)
- Natural Language Processing (NLP) tools for Hindi language processing
- Evolutionary algorithm library
- Machine learning models for intent detection and slot filling (e.g., LSTM, Transformer)
- Development environment (e.g., Python, Jupyter Notebook)
- Evaluation metrics (e.g., F1-score, accuracy)

OBJECTIVES

Data Acquisition and Translation:

Collect or translate datasets in Hindi (ATIS/MTOP/SNIPO) to facilitate training and evaluation.

Data Preprocessing and Feature Engineering:

Clean, preprocess, and engineer features for effective intent detection and slot filling, including word embeddings for RNNs and contextual embeddings for BERT.

Model Development:

Implement machine learning models tailored to Hindi for intent detection and slot filling tasks, including RNN-based models for intent detection and BERT-based models for slot filling.

Evolutionary Algorithm Optimization:

Integrate evolutionary algorithms (e.g., GA, MOGA) to optimize model parameters or hyperparameters for RNNs and BERT-based models.

Evaluation and Fine-Tuning:

Evaluate system performance using appropriate metrics and fine-tune RNNs and BERT-based models for enhanced accuracy and efficiency.

Documentation and Reporting:

Compile a comprehensive project report detailing methodologies, experiments, results, and conclusions, emphasizing the effectiveness of RNNs and BERT-based models in Hindi intent detection and slot filling.

PROJECT WORK FLOW

Data Collection and Translation	• Obtain the translated Hindi datasets (ATIS/MTOP/SNIPO) or translate them if necessary. • Ensure the datasets cover a wide range of intents and slot variations relevant to the application domain.
Data Preprocessing	• Clean the datasets by removing noise, special characters, and irrelevant information. • Tokenize the text and convert it into a suitable format for machine learning models. • Split the data into training, validation, and test sets.
Feature Engineering	• Extract features relevant to intent detection and slot filling tasks. • Utilize techniques such as word embeddings (e.g., Word2Vec, GloVe) or contextual embeddings (e.g., BERT, ELMo) to represent words in a vector space.
Model Development for Intent Detection	• Choose a machine learning model suitable for intent detection, such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), or Transformer-based models. • Train the model using the pre processed training data and validate its performance using the validation set. • Fine-tune the model hyperparameters to optimize performance.

Model development for Slot filling	• Select a model architecture appropriate for slot filling tasks, such as Conditional Random Fields (CRFs), LSTM-CRFs, or Transformer-based models. • Train the slot filling model on the pre processed training data and evaluate its performance using the validation set. • Adjust model parameters and architecture as needed to improve performance.
Evolutionary Algorithm Implementation	• Choose an evolutionary algorithm (e.g., Genetic Algorithm or Multi-Objective Genetic Algorithm) for optimization. • Define the objective function(s) to be optimized, considering factors such as model accuracy, efficiency, and resource usage. • Incorporate the evolutionary algorithm into the training pipeline to optimize model parameters or hyperparameters.
Model Optimisation	• Execute the evolutionary algorithm to search for optimal model configurations. • Evaluate the performance of each candidate solution using the validation set. • Select the best-performing solutions based on predefined criteria or Pareto optimality in the case of multi-objective optimization.
Evaluation and Fine Tuning	• Assess the final model's performance on the test set using appropriate evaluation metrics such as accuracy, F1-score, or precision-recall curves. • Analyze the strengths and weaknesses of the system and identify areas for improvement. • Fine-tune the model based on evaluation results, such as adjusting thresholds, refining features, or retraining with additional data if available.
Documentation and Reporting	• Compile a comprehensive project report documenting the entire workflow, including data collection, preprocessing, model development, optimization, evaluation metrics, and results. • Provide insights into the effectiveness of the evolutionary algorithm in optimizing model performance. • Discuss any limitations or challenges encountered during the project and propose future directions for improvement.

TECH STACK

- Programming Language: Python
- Machine Learning Frameworks: PyTorch, Transformers
- Data Processing and Analysis: Pandas/Numpy, Scikit-learn, Matplotlib

DATA COLLECTION AND PREPROCESSING:

Work done by:

- Aman Raj (21je0087)
- Sai Pranav (21je0279)

Output after preprocessing:

```
Preprocessed sample
{'utterance': 'आज मेरी उड़ान क्यों है?', 'intent': 'Flight Status Inquiry', 'slots': {'तारीख': 'आज'}}
```

DATA SPLITTING FOR TRAINING AND TESTING:

Work done by-

o Adari Sameera (21je0032)

o Gaddam Pallavi (21je0338)

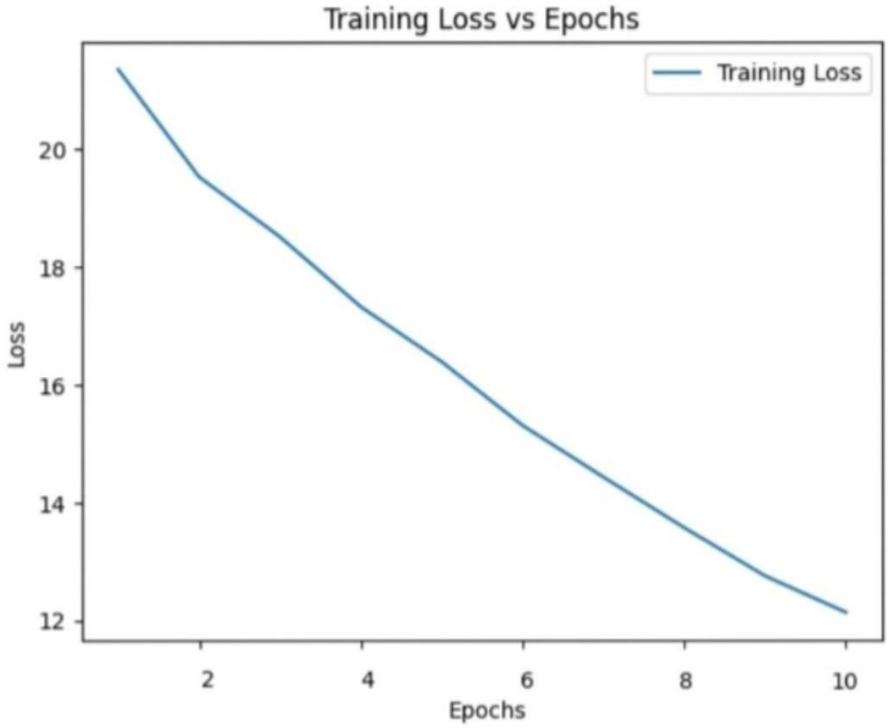
Output after splitting data for training, testing and cross-validation:

```
Size of Training Dataset:
94
Size of Testing Dataset:
27
Size of Validation Dataset:
11
```

INTENT CLASSIFICATION USING BERT:

Work Done By-
o Thamatam Pradeep (21je0990)
o Tanush Garg (20je1020)

```
Epoch 1/10, Loss: 21.34467101097107
Epoch 2/10, Loss: 19.522290468215942
Epoch 3/10, Loss: 18.50565791130066
Epoch 4/10, Loss: 17.318867921829224
Epoch 5/10, Loss: 16.385982751846313
Epoch 6/10, Loss: 15.306236267089844
Epoch 7/10, Loss: 14.429723381996155
Epoch 8/10, Loss: 13.570236682891846
Epoch 9/10, Loss: 12.751965999603271
Epoch 10/10, Loss: 12.13326370716095
```

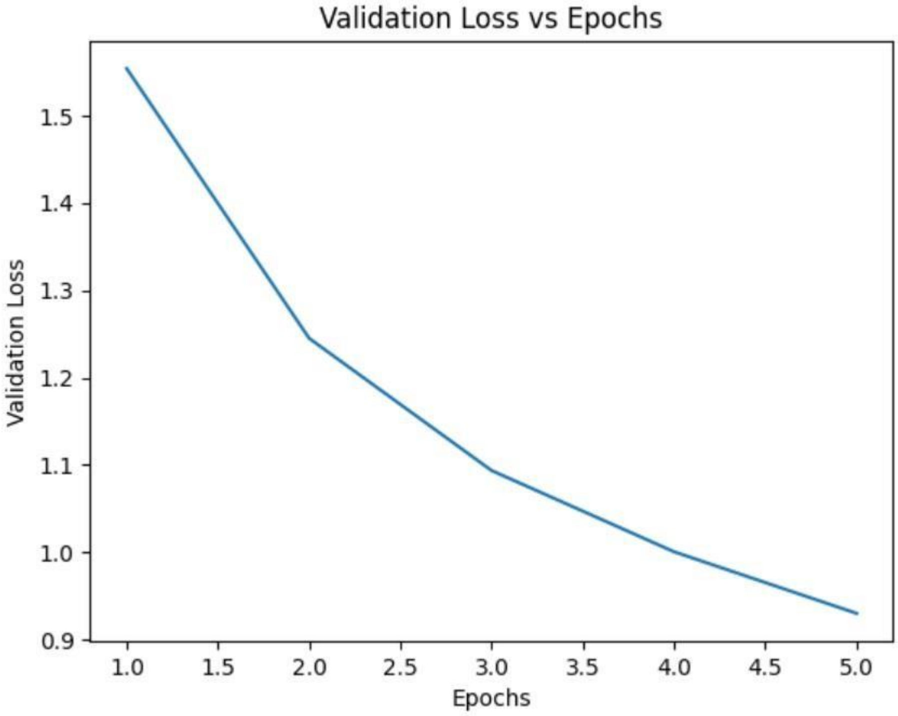



```
Validation Accuracy: 0.5454545454545454
```

SLOT FILLING:

Work Done By-
o Suryarghya Saha (20je0998)
o Cherukuri Madhulika (21je0264)

```
Epoch 1/5, Loss: 11.541724443435669, Validation Loss: 1.5536105632781982
Epoch 2/5, Loss: 8.57704508304596, Validation Loss: 1.2450419664382935
Epoch 3/5, Loss: 7.176114201545715, Validation Loss: 1.093740463256836
Epoch 4/5, Loss: 6.432607412338257, Validation Loss: 1.0008349418640137
Epoch 5/5, Loss: 5.922153055667877, Validation Loss: 0.9303888082504272
```



OPTIMISATION USING GENETIC ALGORITHM:

Work Done by-
o Smriti Priyadarshani (23dp0098)
o Aman Rawat (20je0110)

```
Generation 1
Best individual: [0.018906659323883843, 0.3487458339934118], Fitness: (0.9303888082504272,)
Generation 2
Best individual: [0.03466288223577364, -0.030782426955571256], Fitness: (0.9303888082504272,)
Generation 3
Best individual: [0.06528787912287992, -0.042551030467792794], Fitness: (0.9303888082504272,)
Generation 4
Best individual: [0.036872615515614486, 0.4225373193187302], Fitness: (0.9303888082504272,)
Generation 5
Best individual: [0.03466288223577364, -0.030782426955571256], Fitness: (0.9303888082504272,)
Best Parameters after GA Optimization:
[0.03466288223577364, -0.030782426955571256]
```

CONCLUSION

We, as a team, plan to create a system for detecting intents and filling slots in Hindi by utilizing translated datasets and advanced techniques such as RNNs and BERT. Through evolutionary algorithms, we hope to fine-tune model parameters for better performance. Our project focuses on thorough evaluation and adjustments to ensure reliability and efficiency. We aim to enhance Hindi natural language processing, allowing for more advanced and user-friendly applications. With our skills and commitment, we are dedicated to providing a top-notch solution that satisfies our stakeholders' requirements.