PROJECT REPORT:

INTENT DETECTION AND SLOT FILLING ON HINDI DATASET

15-02-2024

GROUP 2

~EVOLUTIONARY COMPUTATION

Prof. Dr.A.C.S. Rao

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 labeled 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 (e.g., DEAP for GA, pymoo for MOGA)
- 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 CNNs and contextual embeddings for BERT.

- Model Development:

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

- Evaluation and Fine-Tuning:

  Evaluate system performance using appropriate metrics and fine-tune CNNs 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 CNNs and BERT-based models in Hindi intent detection and slot filling.

PROJECT WORKFLOW

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 preprocessed 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 preprocessed training data and evaluate its performance using the validation set. ❖ Adjust model parameters and architecture as needed to improve performance.
Evolutionary Algorithm Integration	❖ 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 Optimization	❖ 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.
- Libraries: TensorFlow, PyTorch, scikit-learn, DEAP, pymoo.
- Development Environment: Jupyter Notebook.
- Data Processing: Pandas, NumPy.
- Evaluation: NLTK (Natural Language Toolkit), spaCy.
- Visualization: Matplotlib, seaborn.

CONCLUSION

As a team, we propose to develop an intent detection and slot filling system for Hindi using translated datasets and advanced methods like CNNs and BERT. By leveraging evolutionary algorithms, we aim to optimize model parameters for superior performance. Our project emphasizes rigorous evaluation and fine-tuning to ensure robustness and effectiveness. We aspire to contribute to Hindi natural language understanding, enabling more sophisticated and user-friendly applications. With our expertise and dedication, we are committed to delivering a high-quality solution that meets the needs of our stakeholders.

CONTRIBUTIONS & ACHIEVEMENTS TILL NOW

- **Data Collection and Preprocessing**: We have collected from standard source ATIS and used google Trans library for translation into Hindi.
    - Aman Raj (21je0087)
    - Sai Pranav (21je0279)

- **Data splitting for training and testing**:
    - Adari Sameera (21je0032)
    - Gaddam Pallavi (21je0338)

- **Intent classification using mBert**: Used a multilingual BERT-based model(mBERT) as the backbone for extracting contextual embeddings. BERT's transformer architecture is effective in capturing contextual nuances in text.

&&

- **Intent Classification Head**: Added a dense layer on top of the BERT output corresponding to the [CLS] token, as it aggregates sentence-level representation. This dense layer will perform the intent classification.
    - Thamatam Pradeep (21je0990)
    - Tanush Garg (20je1020)

- **Slot filling**: Used the output embeddings from BERT corresponding to each token in the input sequence. Pass these embeddings through a BiLSTM layer to capture dependencies in the input sequence, followed by a dense layer for slot prediction.
    - Suryarghya Saha (20je0998)
    - Cherukuri Madhulika (21je0264)

- **Optimization using Genetic Algorithm**: Our objective is to balance intent detection and slot filling accuracy. For example, like maximizing the F1 score for slot filling and maximizing accuracy for intent classification.
    - Aman Rawat (20je0110)
    - Smriti Priyadarshani (23dp0098)

- **Documentation and reporting each work:**
    - Team Work.

FUTURE WORK

- Final workings on the Hindi dataset prepared after translation and preprocessing.
- Python based implementation of Intent classification using BERT model and Slot filling method using LSTM/CNN along with the BERT embeddings using PyTorch and Transformers library.
- Implementation of Genetic Algorithm using DEAP library for optimization of the model where our objective will be to balance and improve intent classification and slot filling accuracy.
- Combine all pieces of codes and implementations and then deployment of the model so that it is available for use.
- Documentation and reporting of work.

THANK YOU