Deadline 26th of November

Task 4: Detecting Dementia Using Natural Language Processing

Mentor to be contacted: Shaima Ahmad Freja ***(sa.freja@stud.uis.no)***

Goal: This project aims to develop a predictive model for detecting dementia using Natural Language Processing (NLP). The model will utilize linguistic features extracted from text data to predict dementia diagnoses. You will be responsible for preprocessing the data, performing exploratory analysis, and building machine learning or deep learning models to classify transcripts into either dementia or control categories (binary classification). Students are encouraged to explore model optimization and fine-tuning techniques using frameworks like Scikit-learn, TensorFlow, Keras, Transformer or PyTorch.

Dataset: The dataset for this project is sourced from DementiaBank. The transcript files have been prepared in CSV format and divided into training and testing sets:

- Training Dataset: Includes two files, Dementia_db.csv (containing dementia patient transcripts) and Control_db.csv (containing healthy control transcripts).
- Testing Dataset: Contains one file, testing_db.csv, for evaluating the model's performance.

Link to dataset:

Control_db

Dementia_db

Testing_db

Instructions

1. **Data Preprocessing**:
    - Read transcription files from Training & Testing folders into a dataframe
    - In Testing folder we have transcription files and one text file which contains (label , MMSE) for each transcript file
    - Read lables for the testing dataset from Test_result_label.txt
    - Merging two dataframe (df_Testing & df_label) in one dataframe where filename in the df_testing equal ID in the df_label
        - df_testing: dataframe for texting which contains all the columns except label & mmse
        - df_label: dataframe for testing which contains ID = Filename and lable, mmse columns
    - cleaning the data (lowercase all the texts, remove punctuation marks, all the symbol, ...)
    - Add filter to select only 'ProbableAD', 'PossibleAD' from Dementia data

- Remove punctuation mark
- lemmatization, we will activate removing the stopword here
- Convert the categorical columns like gender to numerical where (female :0, male: 1)
- Save the preprocessing Dataset into csv file with

2. **Exploratory Analysis**
    - To begin this exploratory analysis, first use matplotlib to import libraries and define functions for plotting the data. Depending on the data, not all plots will be made.
3. **Feature Extraction**
    - You can either use one of the methods to convert the text data into numerical features that a model can process.
    - Bag-of-Words (BoW)
    - TF-IDF (Term Frequency-Inverse Document Frequency)
    - Word Embeddings (Word2Vec, GloVe, FastText
    - Pretrained transformer Transformers (e.g., BERT, GPT)
4. **Model selection:**

You can explore traditional machine learning models, deep learning models, or transformer models (optional):

- To Explore traditional machine learning models for text classification such as (Logistic Regression, Support Vector Machine (SVM), Random Forest,...)
- You may also explore deep learning models such as Long Short-Term Memory Networks (LSTM) or Gated Recurrent Units (GRUs)
- Transformer-based models like BERT (Bidirectional Encoder Representations from Transformers) or RoBERTa.

5. **Evaluation Metrics:**
    - Confusion matrix
    - Precision, Recall, and F1-score for a more detailed analysis of the model performance, especially since this is a binary classification task.
6. **Model Training:**
    - Train the model on the training split and validate on the development test split.
    - Plot the training and validation accuracy/loss curves over the training epochs.

Experimenting with different models like BERT or SVMs to see which performs best on your dataset.