Assignment 2: Twitter Sentiment Classification with NLP

Assignment Description

The purpose of this assignment is to challenge students to apply various NLP techniques learned in class to train a sentiment analysis classifier for Twitter data.

To achieve this, students will perform text pre-processing and feature extraction using **at least three different feature types**. Examples of such are:

- Bag of Words (BoW)
- TF-IDF
- Word2Vec embeddings
- Sentiment lexicon-based features

Train a sentiment analysis classifier for two classification tasks:

- **Binary classification**: Positive vs. Negative
- **Three-class classification**: Positive vs. Negative vs. Neutral

Students can choose any machine learning classifier (e.g., logistic regression, SVM, random forest, neural networks etc.) and may perform hyperparameter tuning if desired. The focus is on NLP-based feature engineering and evaluating how different feature representations impact model performance.

Dataset

The students will work with the *tweet_eval* [1] sentiment dataset that contains tweets with positive and negative sentiments.

Instructions

Students are required to:

- Perform pre-processing on the Twitter dataset. They may reuse and modify their code from previous exercises.

- Engineer at least three different NLP-based features (e.g., BoW, TF-IDF, Word2Vec embeddings, sentiment lexicons, etc). The use of these features is mandatory—students cannot use text-only classification models like BERT for their main submission. They can only do so to use as a baseline for comparison.
- Train sentiment classifiers for both binary and three-class classification tasks.
- Experiment with different feature combinations to analyze their impact on classifier performance, training time, and resource usage.
- Use Google Colab to run and share their code with the instructor.

Optional: In-Class Competition

Students who wish to participate in an optional competition for the best-performing sentiment classifier can submit their results for ranking. To join, they must:

- Send a pseudonym to the instructor for anonymity in the rankings.
- Compete based on classifier performance, with results listed on a leaderboard.

Participation in the competition is **not mandatory** and does not affect assignment grading.

Useful Tools

Sentiment Lexicons:
- https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon
- http://www2.imm.dtu.dk/pubdb/pubs/6010-full.html

Libraries:
- https://textblob.readthedocs.io/en/dev/
- https://github.com/cbaziotis/ekphrasis

Hyper-parameter Tuning:
- https://github.com/optuna/optuna

Pre-trained Word Embeddings:
- https://nlp.stanford.edu/projects/glove/

Deliverables

Students must submit the following:

1. **Jupyter Notebook**
    - Contains all code used for pre-processing, feature engineering, model training, and evaluation. Note: For evaluation you should use sklearn's `classification_report`.

    - Clearly structured with explanations and comments.

2. **Results and Findings**
    - A brief section in the notebook summarizing key insights, including:
        - Performance results of different feature sets.
        - Analysis of feature impact on model accuracy and efficiency.
        - Discussion of any challenges faced and potential improvements.

Grading

The grading will be based on:

- **Methodology:** Logical steps taken for data processing, feature extraction, and classification.
- **Justification of Approach:** Explanation of choices made for feature selection, model selection, and evaluation.
- **Performance Results:** Accuracy and effectiveness of the classifier.
- **Code Quality:** Clarity, structure, and readability of the Jupyter Notebook.

Deadline

The deadline for this assignment will be communicated by the instructor.

[1] Barbieri, Francesco, et al. 2020 "*TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification.*" Findings of the Association for Computational Linguistics: EMNLP 2020.