Group Project: Twitter Sentiment Classification with NLP

Overview

This group project challenges students to apply Natural Language Processing (NLP) methods to build and evaluate a sentiment classifier for Twitter data. Teams will explore and compare a range of linguistic and structural features from tweets, implement appropriate text pre-processing and feature selection techniques, and train machine learning classifiers for sentiment prediction.

Objectives

The main goals of the project are to:

- Apply text pre-processing and tokenization techniques specific to Twitter data.
  Identify and extract relevant lexical, syntactic, and semantic features for sentiment analysis.
- Train, evaluate, and compare sentiment classifiers using different feature sets.
- Understand the effect of feature selection and feature engineering on model performance.
- Collaboratively manage code, results, and documentation using GitHub.

Dataset

Teams will use the TweetEval Sentiment Dataset [1], which contains short English tweets labeled as positive or negative. For the project, you will use only the `text` and `label` fields.

Twitter Tokenization

Tweets are linguistically distinct from regular text — they contain user mentions, hashtags, emojis, and links.

Students should use specialized tokenizers such as twokenize or ekphrasis, which are designed to handle these cases.

For example:

```
from twokenize import tokenizeRawTweetText

tweet = "Amazing movie!! 😂🔥 Check it out at https://t.co/example
@user #BestFilm"
tokens = tokenizeRawTweetText(tweet)
print(tokens)
```

Output:

```
['Amazing', 'movie', '!!', '😂', '🔥', 'Check', 'it', 'out',
'at', 'https://t.co/example', '@user', '#BestFilm']
```

This tokenizer keeps emojis, mentions, hashtags, and URLs intact, essential for capturing meaningful Twitter features.

Feature Engineering and Selection

Each team must identify and extract multiple feature types for classification.

Examples of Features:

Lexical / Textual Features

- Bag of Words (BoW)
- TF-IDF vectors
- Word2Vec or GloVe embeddings
- Sentiment polarity and subjectivity
- Lexicon-based counts (e.g., number of positive/negative words)

Structural / Behavioral Features

- Number of user mentions (@username)
- Number of URLs
- Number of hashtags
- Number of profanity words
- Tweet length (in tokens or characters)
- Number of emojis or emoticons
- Capitalization ratio (percentage of capitalized words)
- Number of exclamation/question marks
- Number of negation words (e.g., “not”, “never”)
- Sentiment score and subjectivity (as numeric features)

You are encouraged to design additional features that may improve classification performance.

After feature extraction, apply a feature selection method to identify the most informative features for classification.

Model Training

Train and evaluate at least one sentiment classifier for binary classification (positive vs. negative).

Optional: Perform hyperparameter tuning using optuna or grid search.

Any advanced or specialized methods (e.g., transformer-based models like BERT) must first be discussed and approved by the instructor.

Evaluation and In-Class Competition

Each team must evaluate their models using internal validation (e.g., train/test split or cross-validation). In addition, a held-out test set, kept hidden by the instructor, will be used to assess the final performance of all submitted models as part of an in-class competition.

Competition Details

- The instructor will evaluate all submitted models using the held-out dataset.
- Performance on this hidden test set will determine the final leaderboard ranking.
- The three best-performing teams will receive a surprise award 🎁.

Model Submission

Along with your Jupyter Notebook, and the rest of the necessary files, you must also provide:

- Your best-performing classifier,
- The feature extraction functions required to generate the input features, and
- The saved model file (model.pkl), using the following format:

```
import pickle

# Save the model
with open('model.pkl', 'wb') as f:
    pickle.dump(clf, f)

# Load the model
with open('model.pkl', 'rb') as f:
    clf = pickle.load(f)
```

Repository and Collaboration

Each team must create a private GitHub repository named: `epl499_group_project_<TEAM NAME>`

Invite **dpasch01** as a collaborator.

Your repository should include:

- Source code and Jupyter notebooks
- A README.md describing:
    - Team members
    - Project overview
    - Feature types used
    - Model(s) implemented
    - Key results and findings

Grading

Component	**Description**	**Weight**
Methodology	Logical workflow for data processing, feature extraction, selection, and classification	30%
Justification of Approach	Rationale for feature choices (prior to feature selection).	20%
Performance Results	Accuracy and robustness of classifier.	10%
Code Quality	Clarity, Structure, documentation, and readability.	20%
Collaboration	Contribution of individual team members through Github.	20%

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.