Assignment - 1

NLP CS60075
Autumn Semester 2023
IIT Kharagpur

Enhancing Sentiment Analysis Using POS Tagging

Assignment Beginning: 9:30 AM, 29th August 2023

Assignment Deadline: 2:00 PM, 16th September 2023

OVERVIEW

The objective of this assignment is to deepen your understanding of Natural Language Tagging or more specifically Part-of-Speech (POS) tagging which involves assigning grammatical categories (such as nouns, verbs, and adjectives) to words in a sentence.
One of the most pivotal NLP tasks is Sentiment Analysis which involves determining the emotional tone or polarity of a piece of text, whether it's positive, negative, or neutral.

While POS tagging might seem like a primary linguistic task, its impact extends far beyond syntax. By accurately identifying the grammatical roles of words in a sentence, POS tagging contributes to understanding the context, semantics, and even sentiments conveyed by the text. This is where the concept of this assignment emerges – the exploration of the integration of POS tagging and sentiment analysis.

We have seen that some words can be used as multiple parts of speech in the English language. For example:

*"Kevin has dark hair and **fair** skin."*	*(ADJECTIVE)*
*"The new **fair** is boring."*	*(NOUN)*

The addition of POS tags introduces a new dimension of linguistic insight that may significantly enhance sentiment classification accuracy and depth.

You will implement a POS tagging system, apply it to a Classical Sentiment Analyzer, and evaluate its impact on sentiment classification accuracy.

TASKS

1. **POS Tagger** Implementation (from scratch). **[40% of TOTAL]**
    a. Use the treebank corpus (from nltk) for training data. [nltk downloader]
    b. Implement the Viterbi Algorithm (dynamic programming) for POS Tagging.
    c. Keep all the POS tags for this task as that will give you proper transition and emission probabilities.
2. **Vanilla Sentiment Analyzer** **[15% of TOTAL]**
    a. Use the movie_reviews corpus (from nltk) for data. [Split it into train-val-test]
    b. You may use Tfidf, Count, or any other vectorizer (Word2Vec, Transformers, etc.) for creating sentence embeddings.
    c. Train a classical Classifier (Naive Bayes or SVM from sklearn) for sentiment classification using the above features.
3. **Improved Sentiment Analyzer** **[25% of TOTAL]**
    a. Use the POS Tagger in Task 1 for POS tagging the dataset.
    b. Implement a pipeline to integrate the POS tag features along with the sentence embeddings. (Use your own creativity and **mention your strategy in the report**)
    c. Train the same Classifier (as chosen in Task 2) again for sentiment classification using the new features.
4. **Report** **[20% of TOTAL]**
    a. Add your observations to a report (submit in pdf format).
    b. Compare the performance of your POS-tag-enhanced model with a baseline model that doesn't use POS tags. [Don't worry about scores]
    c. Make sure to include the classification reports of both models in the report.
    d. Make sure to highlight any advanced modifications that you've done in your report

Learning Outcomes

- POS Tagging
- Sentiment Analysis
- Embeddings
- Custom Pipeline with Additional Features
- Work Presentation

Submission Guidelines

- You may use Jupyter Notebooks, Google Colab, or Python files for coding.
- Include Documentation of the whole codebase. (In case you're using Python files)
- Properly add comments on your code.
- Of course, **Plagiarism in any form is strictly prohibited**.
- Submit all files in **.zip format** on **CSE Moodle**.

HAPPY LEARNING