Automated Text Classification

Group: SOLO

Gayathri

Kodakandla

a1939114

The University of Adelaide

4533_COMP_SCI_7417_7717 Applied Natural Language Processing

Lecturer: Dr. Orvila Sarker

1. Abstract

This project aims to automate the classification of Stack Overflow posts related to Natural Language Processing (NLP) using a combination of rule-based keyword categorization and machine learning models. A dataset of 16,005 posts was collected using the Stack Exchange API, pre-processed with standard NLP techniques, and analyzed to assign categories and sentiment labels. The classification model achieved an accuracy of 73.26%, and sentiment analysis using VADER showed an ensemble agreement of 82.4%. This work demonstrates a pipeline for text preprocessing, categorization, and sentiment extraction from community-generated technical content**.**

---

2. Introduction

The increasing volume of developer discussions on platforms like Stack Overflow makes it challenging to filter and categorize posts effectively. Automating this process can help identify key NLP subtopics and sentiment trends within the community. This project explores a classification framework for NLP-tagged questions using traditional machine learning and lexicon-based sentiment analysis.

2.1 Data Collection Process

Posts tagged with [nlp] were fetched using the Stack Exchange API. A total of 16,005 unique questions were collected and saved in CSV format. The collected fields included question_id, title, body, and tags.

2.2 Summarisation Methods

Extractive Summarisation: This method selects and combines the most important sentences or phrases from the original text to form a summary.

Abstractive Summarisation: This method generates a summary by interpreting and paraphrasing the original content, possibly including new words or phrasing not present in the source.

---

3. Preprocessing

3.1 Text Cleaning

Each post's title and body were concatenated into a single text field. The following preprocessing steps were applied:

- Lowercasing
- Removal of non-alphanumeric characters
- Tokenization

- Stop word removal
- Lemmatization (using WordNetLemmatizer)

3.2 Feature Extraction

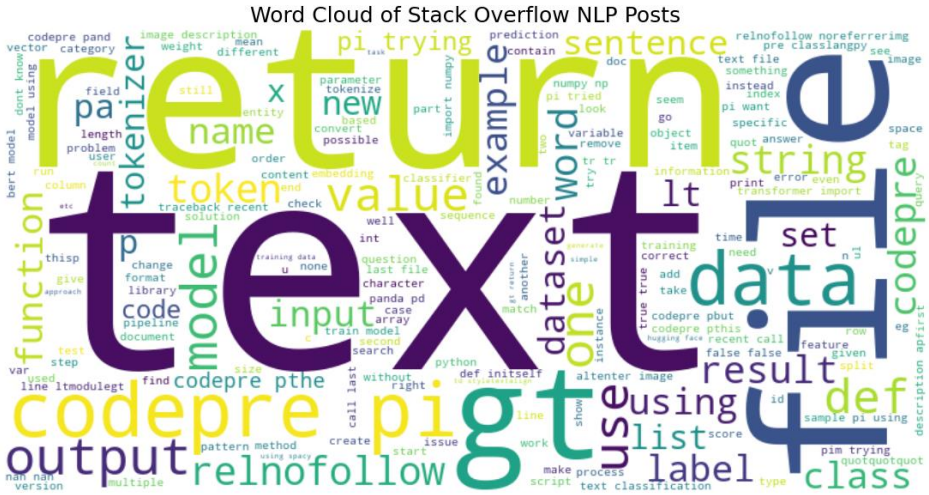
TF-IDF vectorization was used to transform the cleaned text into a numerical feature matrix suitable for model training.

---

4. Data Visualisation

4.1 Word Cloud

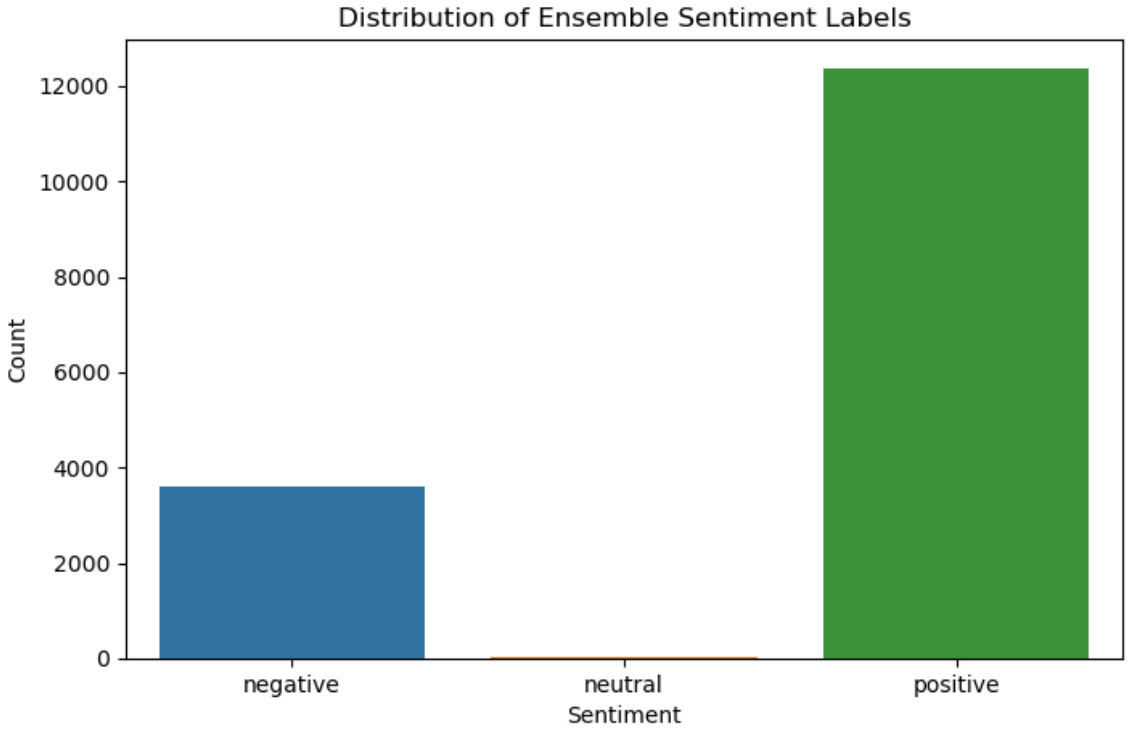
A word cloud was generated from the cleaned text, highlighting frequent terms such as "token", "model", "text", and "bert".



4.2 Sentiment Distribution

VADER sentiment scores showed that:

- Positive: 77%
- Negative: 22%
- Neutral: 1%

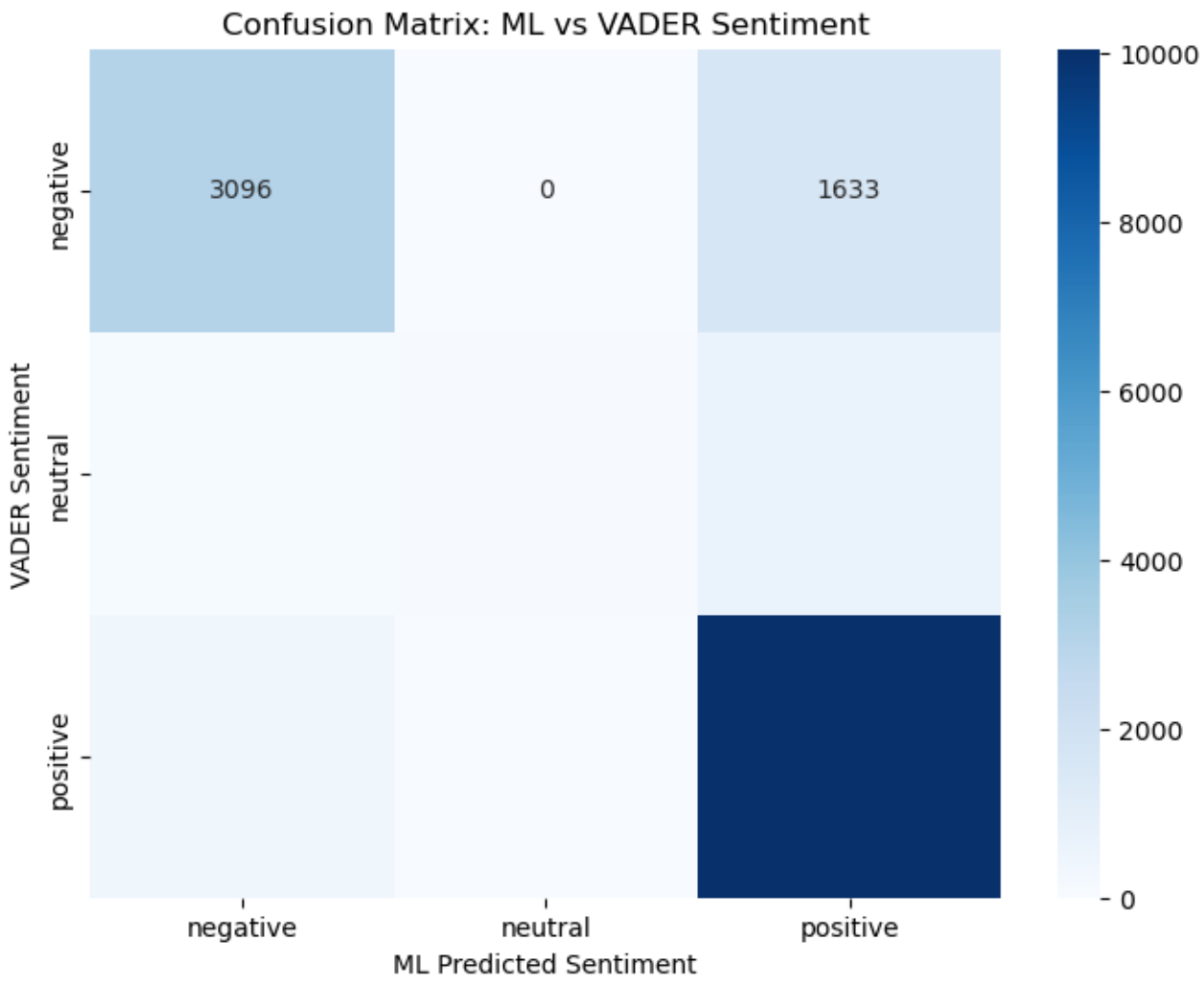


4.3 Topic Distribution

A bar chart was plotted to show the number of posts under each predefined NLP category.

4.4 ML vs VADER Confusion Matrix

A confusion matrix was plotted to compare the sentiment predictions made by the ML model and VADER.



---

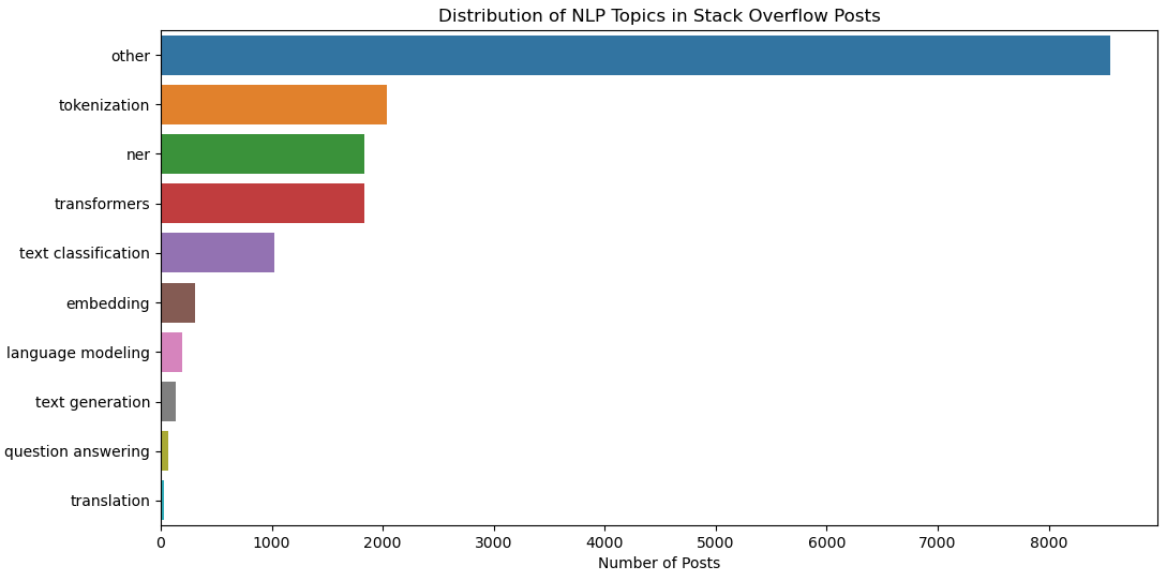
5. Data Categorization

5.1 Definition of Categories

Categories were manually defined based on keyword matches in the post titles:

- tokenization
- transformers
- ner
- language modeling
- text classification
- text generation
- translation
- question answering

- embedding
- other



-

5.2 ML Classification Results

- Accuracy: 73.26%
- Strongest classes: ner, tokenization, other
- Weakest classes (low support): translation, text generation

---

6. Sentiment Analysis

6.1 VADER vs ML Agreement

- Ensemble agreement rate: **82.4%**
- VADER Sentiment Distribution:
    - Positive: 10,448
    - Negative: 4,729
    - Neutral: 828

6.2 Classification Report (ML vs VADER)

- Positive: F1-score = 0.84
- Negative: F1-score = 0.63
- Neutral: F1-score = 0.02 (sparse)

---

7. Conclusion

This mini-project successfully implemented an end-to-end NLP pipeline to classify Stack Overflow posts by subfield and sentiment. While keyword-based labeling helped with bootstrapping categories, the classification model showed strong performance despite class imbalance. Sentiment analysis revealed a largely positive tone in the posts. Future improvements may include using BERT embeddings, class balancing techniques, and hierarchical classification.

---

8. References

- Jurafsky, D., & Martin, J. H. (2023). *Speech and Language Processing*. Stanford University.
- NLTK Documentation: https://www.nltk.org/
- Stack Exchange API: https://api.stackexchange.com/
- scikit-learn Documentation: https://scikit-learn.org/
- Hutto, C., & Gilbert, E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis.

GITHUB REPOSITORY LINK:

Please use the below link for redirecting to my github repo which has all the files

https://github.com/Gayathri224/NPL_STACKOVERFLOW.git