Sentiment Classification of Movie Review Phrases Using NLP Techniques

Neysha L. Pagán Vargas  
IST 664 - Natural Language Processing

Dr. Preeti Mahaveer Jagadev

June 10, 2025

**Abstract**

*This project presents a sentiment classification system for movie review phrases using a Kaggle dataset originally derived from Rotten Tomatoes. The dataset includes over 156,000 phrases labeled on a five-point sentiment scale: negative, somewhat negative, neutral, somewhat positive, and positive. The objective was to develop a supervised machine learning model capable of classifying the sentiment of any given phrase.*

*A logistic regression model trained on TF-IDF vectorized text was used as a baseline. Additionally, a second model combined these TF-IDF features with handcrafted sentiment features derived from the MPQA Subjectivity Lexicon. Evaluation results showed that combining TF-IDF with lexicon-derived features led to modest but consistent improvements, especially in the underrepresented sentiment classes. Overall, the hybrid approach achieved a 63.14% accuracy with improved macro F1-score. Future improvements may include using deep learning models such as LSTM or BERT and applying cross-validation or class rebalancing techniques.*

Table of Contents

[Introduction 3](#_Toc200490838)

[Background and Motivation 3](#_Toc200490839)

[Problem Statement 3](#_Toc200490840)

[Objectives 3](#_Toc200490841)

[Scope and Limitations 3](#_Toc200490842)

[Methodology 4](#_Toc200490843)

[Dataset 4](#_Toc200490844)

[Tools Used 4](#_Toc200490845)

[Design and Implementation 4](#_Toc200490846)

[Models Used 4](#_Toc200490847)

[Experiments and Results 5](#_Toc200490848)

[Metrics Used 5](#_Toc200490849)

[TF-IDF Model Results 5](#_Toc200490850)

[Hybrid Model (TF-IDF + Lexicon) Results 5](#_Toc200490851)

[Interpretation 5](#_Toc200490852)

[Discussion 5](#_Toc200490853)

[Conclusion 6](#_Toc200490854)

[References 6](#_Toc200490855)

[Appendices 7](#_Toc200490856)

# Introduction

## Background and Motivation

Understanding sentiment in textual data is essential in natural language processing (NLP) for applications ranging from customer feedback analysis to social media monitoring. Movie reviews are an ideal testbed for sentiment classification due to their subjective and opinion-rich content.

## Problem Statement

Can we accurately classify movie review phrases into one of five sentiment categories?

## Objectives

* Preprocess and clean the dataset
* Extract features from text (TF-IDF, lexicon-based)
* Train classification models
* Evaluate model performance using classification metrics and visualization tools

## Scope and Limitations

This project is limited to the Kaggle-provided train.tsv dataset and applies logistic regression as the classification algorithm without leveraging deep learning methods. Lexicon-based features are extracted exclusively from the MPQA Subjectivity Lexicon. Additionally, performance was impacted by class imbalance, which constrained the model's ability to accurately classify underrepresented sentiment categories.

# Methodology

## Dataset

* Source: Kaggle competition
* Format: TSV file with PhraseId, SentenceId, Phrase, and Sentiment
* Total phrases: 156,060
* Sentiment labels: 0 = Negative, 1 = Somewhat Negative, 2 = Neutral, 3 = Somewhat Positive, 4 = Positive

## Tools Used

* Python (pandas, scikit-learn, matplotlib, seaborn, MPQA Subjectivity Lexicon)
* Jupyter Notebook

## Design and Implementation

* Step 1: Loaded and explored the dataset, including class distribution
* Step 2: Preprocessed data and split into training and validation sets
* Step 3: Transformed text data into TF-IDF feature vectors
* Step 4: Trained logistic regression model (baseline)
* Step 5: Parsed lexicon and extracted 4 features per phrase (strong, weak, pos, neg)
* Step 6: Combined TF-IDF and lexicon features; trained hybrid model

## Models Used

* Logistic Regression (TF-IDF)
* Logistic Regression (TF-IDF + Lexicon features)

# Experiments and Results

## Metrics Used

* Accuracy
* Precision
* Recall
* F1-Score
* Confusion Matrix

## TF-IDF Model Results

* Accuracy: 63.00%
* Macro F1-Score: 0.4641

## Hybrid Model (TF-IDF + Lexicon) Results

* Accuracy: 63.14%
* Macro F1-Score: 0.4683

## Interpretation

* Class 2 (Neutral) was easiest to classify (high support and accuracy)
* Class 0 (Negative) and 4 (Positive) had poor recall due to class imbalance
* Lexicon features improved precision and F1 in minority classes

# Discussion

The results suggest that TF-IDF features alone perform well for frequent classes but combining them with lexicon-derived features provides a small but meaningful performance boost. The hybrid model showed increased F1-scores for classes 0 and 4. However, class overlap and imbalanced distribution continued to limit performance. This indicates the need for advanced modeling strategies.

# Conclusion

This project demonstrated that a logistic regression classifier trained on TF-IDF features are effective for phrase-level sentiment analysis. By incorporating lexicon-based sentiment features, the model achieved slightly better performance across several metrics. This hybrid approach leverages both statistical and linguistic insights and can be extended with deeper semantic modeling in future iterations.

In future iterations, it may be valuable to explore deep learning architectures (e.g., BERT) and apply class balancing or ensemble methods to enhance minority class performance.

# References

* Socher et al. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. ACL.
* MPQA Subjectivity Lexicon: http://mpqa.cs.pitt.edu/lexicons/
* Kaggle Competition: https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews
* Scikit-learn documentation: https://scikit-learn.org

# Appendices

* Full classification report for both models
  + Shows precision, recall, and F1-score per class for both baseline and hybrid models.

**TF-IDF features**

A screenshot of a graph

AI-generated content may be incorrect.

**TF-IDF features with lexicon-based features from the MPQA subjectivity dictionary**

A screenshot of a computer screen

AI-generated content may be incorrect.

* Confusion matrix heatmaps
  + Visual comparison of prediction accuracy across sentiment classes.

A graph with numbers and a blue square

AI-generated content may be incorrect.

* Feature extraction code (lexicon parser)
  + Python function to extract polarity and subjectivity features using the MPQA dictionary.

A screenshot of a computer program

AI-generated content may be incorrect.

* Visualization of sentiment class distribution

A graph of a number of bars

AI-generated content may be incorrect.

**Note:** While this project used a single train-test split, implementing k-fold cross-validation is recommended for a more robust evaluation. This helps reduce variance from a single split and ensures generalization across datasets.