Milestone 2 Progress Report

Lexicon-Based Sentiment Analysis of Movie Reviews

CPE-593

Team Members:
Jude Eschete
Ella Disanti
Raymond Donkemezuo

April 30, 2025

Milestone 2 Status Summary

Research Completed

The research phase is complete. We compared several popular sentiment lexicons including VADER, SentiWordNet, Bing Liu's lexicon, AFINN, and the NRC Emotion Lexicon. Each was evaluated based on domain relevance, scoring methods, and language coverage. For this milestone, we have implemented a custom lexicon-driven framework with extensibility to test additional lexicons in future phases. We also explored preprocessing and negation-handling techniques suitable for review-based sentiment classification.

Algorithm/Solution Formed

We have developed a complete lexicon-based sentiment classification system that uses token-level scoring with contextual negation awareness. Our class, `MovieSentimentAnalyzer`, includes:

- Text cleaning (lowercasing, punctuation removal)
- Tokenization and stop word removal using NLTK
- Porter stemming for base form normalization
- Rule-based negation detection over a configurable window
- Score aggregation via lexicon lookup with polarity inversion when needed
- Polarity classification as *positive*, *negative*, or *neutral*

The algorithm is designed to be lexicon-agnostic and modular, supporting drop-in replacement of the sentiment dictionary.

Major Functions Implemented

All core functions are implemented in `movie_sentiment_analyzer.py`. Major methods include:

- `__init__()`: Initializes the analyzer with a sentiment lexicon and configures stop words, stemmer, and negation words.
- `load_kaggle_data(path)`: Loads and validates CSV-formatted review data.
- `preprocess_text(text)`: Cleans text by removing punctuation, lowercasing, removing stop words, and applying stemming.
- `apply_negation_handling(tokens)`: Tags tokens with negation status based on a predefined word list and scope.
- `compute_sentiment_score(tokens_with_negation)`: Aggregates sentiment scores from the lexicon while inverting for negated terms.
- `classify_sentiment(score)`: Maps the final numerical score to a polarity class.

- `analyze_review(text)`: Complete end-to-end review analysis returning both sentiment label and score.

These functions are integrated, thoroughly commented, and are prepped for testing.