

# Sentiment Analysis Program

Nana Frimpong Poku

## 1 Overview

This project implements a sentiment analysis system in **C** using a VADER-style sentiment lexicon. The program reads a lexicon file containing word-score pairs and evaluates the sentiment of sentences provided in an input text file.

The project demonstrates fundamental concepts in text preprocessing, lexicon-based sentiment scoring, and modular program design in C.

## 2 Project Structure

- `SA.c` — Main driver file. Manages program execution and command-line arguments.
- `SAfunctions.h` — Header file containing function prototypes, constants, and shared data structures.
- `SAfunctions.c` — Implements helper functions for lexicon loading, text preprocessing, and sentiment scoring.
- `vader_lexicon.txt` — Sentiment lexicon mapping words to numerical sentiment scores.
- `validation.txt` — Example input file containing sentences to analyze.
- `Makefile` — Automates compilation and builds the executable.

## 3 Program Workflow

1. Load the sentiment lexicon into memory.
2. Read the input text file line by line.
3. Normalize and tokenize each sentence.
4. Match tokens against the lexicon.
5. Accumulate sentiment scores for each sentence.
6. Output sentiment results to standard output.

## 4 Key Functions

The core functionality is implemented in `SAfunctions.c` and declared in `SAfunctions.h`. These include:

- **Lexicon loading functions** Read and store word-score pairs from the lexicon file.
- **Text preprocessing functions** Convert text to lowercase and remove punctuation for consistent matching.
- **Tokenization functions** Split sentences into individual words.
- **Sentiment scoring functions** Compute sentiment scores by summing lexicon values of recognized tokens.
- **Utility functions** Handle file input/output, memory management, and error checking.

## 5 Building the Program

Compile the program by running:

`make`

This generates the executable `mySA`.

## 6 Running the Program

Run the executable with the following command:

`./mySA vader_lexicon.txt validation.txt`

### Arguments

- `vader_lexicon.txt` — sentiment lexicon file
- `validation.txt` — input text file containing sentences

## 7 Notes

- This program uses a lexicon-based approach rather than machine learning.
- Only words present in the lexicon contribute to sentiment scores.
- Additional input files can be analyzed by replacing `validation.txt`.