

## **ALGORITHM**

### 1. **Input:**

- Prompt the user to enter a sentence.
- Read the input sentence.

### 2. **Read Positive Words:**

- Open the file "goodwords.txt" for reading.
- Read each line from the file and store positive words in a vector (poswords).
- Close the file.

### 3. **Read Negative Words:**

- Open the file "badwords.txt" for reading.
- Read each line from the file and store negative words in a vector (negwords).
- Close the file.

### 4. **Read But Words:**

- Open the file "buts.txt" for reading.
- Read each line from the file and store but words in a vector (butcond).
- Close the file.

### 5. **Read Not Words:**

- Create a vector (notcond) with the words "not," "NOT," and "Not."

### 6. **Convert to Lowercase:**

- Convert the entire input sentence to lowercase.

### 7. **Tokenize Sentence:**

- Tokenize the lowercase sentence into individual words, ignoring spaces.
- Store the words in a vector (words) using the `wordcheck` function.

### 8. **Initialize Counters:**

- Initialize counters: `poscount = 0`, `negcount = 0`, `containsBut = false`, `containsNot = false`.

### 9. **Loop Through Words:**

- For each word in the vector of words:
  - Check if the word is in the positive words vector (poswords). If true, increment `poscount`.
  - Check if the word is in the negative words vector (negwords). If true, increment `negcount`.
  - Check if the word is in the but words vector (butcond). If true, set `containsBut = true`.
  - Check if the word is in the not words vector (notcond). If true, set `containsNot = true`.

### 10. **Original Sentiment Checking:**

- If `containsBut` is true:
  - Initialize variables: `butpos`, `wordbefore`, `wordafter`, `wb = 0`, `wa = 0`.
  - Loop through each "but" word in the but words vector:

- Find the position of the "but" word in the sentence.
- If found:
  - Extract words before and after "but."
  - Tokenize both parts into vectors (wordbef, wordaft) using `wordcheck`.
  - Loop through wordbef, incrementing `wb` for each positive word found.
  - Loop through wordaft, incrementing `wa` for each positive word found.
  - Update `butpos` to continue searching for subsequent occurrences.
- Adjust sentiment based on "but":
  - If `wb < wa`, output "the sentiment is positive" and exit.
  - If `wb > wa`, output "the sentiment is negative" and exit.

#### 11. Reverse Sentiment If "Not" is Present and "But" is Not:

- If `containsNot` is true:
  - If `poscount > negcount`, output "the sentiment is negative" and exit.
  - If `poscount < negcount`, output "the sentiment is positive" and exit.

#### 12. Output Final Sentiment:

- If `poscount > negcount`, output "the sentiment is positive."
- If `poscount < negcount`, output "the sentiment is negative."
- If both are equal, output "neutral."

#### 13. End:

- End the program.