Sentiment Analysis: Detailed Documentation

Author: Aakarshan Khosla
Date: 21/03/2025

---

1. Introduction

1.1 Background:

Sentiment analysis, also known as opinion mining, is the computational task of determining and extracting the emotional tone from a piece of text. With applications ranging from customer feedback and social media monitoring to product reviews, sentiment analysis plays a crucial role in understanding public opinion and making informed decisions.

1.2 Objective:

The goal of this research is to develop a robust sentiment analysis system that accurately calculates sentiment scores for text inputs. The system incorporates:

1. Quantifiers and diminishers to adjust sentiment intensity.
2. Negation handling to correctly interpret negated sentiments.
3. Contextual analysis to detect sarcasm and other context-dependent sentences.

---

2. Definitions

2.1 Sentiment:

In this research, sentiment is defined as the emotional tone or attitude expressed in a piece of text. Sentiment will be represented as a numerical score ranging from -1 to 1, where:

- -1 indicates extremely negative sentiment
- 0 indicates neutral sentiment
- 1 indicates positive sentiment

This score is calculated based on the presence of positive and negative words, adjusted for negations, quantifiers, diminishers, and contextual factors.

2.2 Negations:

In this research, negations are defined as words or phrases that reverse or alter the sentiment of a sentence or phrase. Negations can turn a positive sentiment into a negative one, or vice versa. Examples of common negations include words like "not," "no," "never," "don't," "isn't," and similar terms that indicate negation.

The key challenge in sentiment analysis with respect to negations is accurately interpreting the sentiment in the context of these negated words. For example, the phrase "I like ice-cream" has a positive sentiment, but "I don't like ice-cream" introduces a negation that changes the sentiment to negative. To properly handle negations, we use a negation adjustment function (3.2) that modifies the base sentiment score depending on the presence and number of negations in a sentence.

2.3 Quantifiers:

In this research, quantifiers are defined as words that modify or intensify the sentiment of a sentence by indicating the degree or extent of a sentiment. Quantifiers are used to express intensity, frequency, or magnitude and can significantly affect the overall sentiment score. For example, words like "very," "extremely," "definitely," and "really" typically increase the intensity of the sentiment, making it stronger.

Quantifiers in positive contexts amplify the positivity, and in negative contexts, they strengthen the negativity. The key to handling quantifiers is to assign each quantifier a weight or multiplier that adjusts the base sentiment score. This allows us to scale the sentiment intensity appropriately. For example, if the sentence "I like ice cream" has a final score of 0.6 (say), then "I really like ice cream" must have a final score greater than 0.6

2.4 Diminishers:

In this research, diminishers are defined as words that reduce or lessen the intensity of sentiment. They serve to weaken the sentiment expressed in a sentence. Diminishers are commonly used to soften the tone or express a more

restrained opinion. Words such as “somewhat,” “slightly,” “mildly,” and “partially” are typical examples of diminishers.
Like quantifiers, diminishers also affect the sentiment score by applying a scaling factor. However, instead of intensifying sentiment, diminishers reduce the magnitude of the sentiment. This allows for more precise sentiment analysis, especially in cases where the sentiment isn't extreme. Consider the same example for quantifiers, but this time say the sentence was “I kind of like ice cream”. Here, the score of the sentiment must be less than 0.6

---

3. Approach (The Vegetarian Shark)

In this section, we present the approach used to calculate the sentiment score for a given sentence. The calculation process involves several key components, including base sentiment, negation handling, adjustments for quantifiers and diminishers, that we will investigate.

3.1 Base Sentiment

The base sentiment of a sentence is determined by subtracting the number of negative words from the number of positive words present. This base sentiment provides an initial estimate of whether the overall sentiment is positive or negative. The formula for base sentiment is:

Base Sentiment = Number of Positive Words – Number of Negative Words

Note that positive words and negative words can be specified by the user.

3.2 Negation Adjustment Function:

The presence of negations in a sentence (such as "not," "don’t," "never") can reverse or neutralize the sentiment expressed in the text. We adjust the sentiment score based on the number of negations present and the base sentiment.
The negation adjustment function is designed to handle sentences in a more nuanced way, specifically focusing on whether the number of negations is odd or even and how it affects the sentiment.

The adjustment logic follows these rules:

a. If there are no negations, the sentiment remains unchanged.
b. If there is an odd number of negations:
    i. Positive sentiment (base sentiment > 0) becomes negative.
    ii. Negative sentiment (base sentiment < 0) becomes neutral.
    iii. Neutral sentiment (base sentiment = 0) becomes negative if the number of negations is odd, or positive if even.
c. If there are no negations or an even number of negations, the sentiment remains unchanged.

Let's understand the negation adjustment function with a few examples. Consider the sentence "I like going to school". Note that this sentence has a positive base sentiment (bs > 0). Now, consider the negation of this sentence, "I don't like going to school". This clearly is a negative sentiment. Using the negation adjustment formula, we see that it satisfies criteria (i). So, simply multiplying the original base sentiment by -1 flips the sentiment from positive to negative.

Now, consider the sentence "I hate the ideas that you suggest". Here, the sentence has a negative base sentiment due the word "hate" (bs < 0). Negating this sentence gives us "I don't hate the ideas that you suggest". Looking at this sentence closely, we realize that this is not a positive sentence, like we may have expected it to be. However, it seems much more neutral than positive. The key idea here is that negating negative sentiments don't make them positive but makes them neutral. The negation formula from above tells us that it satisfies criteria (ii), so multiplying our base sentiment with 0 makes the sentiment neutral.

Additionally, consider the sentence "I can't say I don't enjoy the food." Here, we can see that the base sentiment is positive. Again, from inspection, we can tell that the speaker is indirectly confirming enjoyment by negating the absence of enjoyment, so it should be an overall positive sentiment. Using the negation formula, we see it satisfies (c).

Finally, let's look at the sentence "I don't know if I can do this". Here, there are no negative or positive words, so our base sentiment is 0 or neutral. However, one can argue that this sentence is close to being a negative neutral (in other words, final score should be near -0.1 to -0.3). This is also accounted for by the negation adjustment by criteria (iii).

3.3 Quantifiers and Diminishers:

The goal here is to adjust the sentiment score based on the presence of specific words that either amplify or attenuate the sentiment.

To account for this, each word is assigned a weight. This weight is then multiplied by the base sentiment score to adjust the sentiment accordingly. The appropriate weights can be determined through empirical analysis, using sentiment datasets to observe how much each quantifier or diminisher typically alters the sentiment. For example, comparing the sentences "I like ice cream" and "I really like ice cream," it's clear that the second sentence should have a higher sentiment score due to the presence of the quantifier "really."

However, complications arise when negations interact with quantifiers and diminishers. Specifically, we need to handle two special cases:

- **d.** When a quantifier or diminisher precedes a negation.
- **e.** When a quantifier or diminisher follows a negation.

Let's illustrate these cases with examples:

Example 1 (Quantifier before negation): "I don't really like ice cream." In this sentence, instead of the quantifier increasing the intensity of the negation, it reduces the intensity of the negation. The word "really" behaves almost like a diminisher here, weakening the negative sentiment.

Example 2 (Quantifier after negation): "I really don't like ice cream." In this case, the quantifier correctly amplifies the negative sentiment, as expected, because it follows the negation. Similarly, a similar pattern applies to diminishers:

Example 3 (Diminisher before negation): "I don't kind of like ice cream" Here, the diminisher "kind of" behaves as a quantifier rather than a diminisher. Instead of further reducing the sentiment, it balances it out, softening the overall negative tone.

Example 4 (Diminisher after negation): "I kind of don't like ice cream" In this case, the diminisher "kind of" diminishes the negative sentiment as expected, reducing the overall sentiment score. To handle these interactions, we simply check whether a negation occurs before or after a quantifier or diminisher. If a negation precedes one of these words, we reverse the intended effect. Specifically, a quantifier becomes a diminisher, and vice versa. This process is called determining the "effective intensity," and we'll explore this method in greater detail later.

3.4 Sentence handling:

In our sentiment analysis approach, we account for sentence boundaries using punctuation marks such as periods (.), exclamation marks (!), and question marks (?). These punctuation marks help identify where one sentence ends, and another begins. Additionally, we handle conjunctions (such as "and" "but" or "because") by splitting complex sentences into smaller, meaningful units for more accurate sentiment analysis.

When dealing with conjunctions, we identify the parts of the sentence before and after the conjunction, treating them as separate entities. This allows us to apply the sentiment analysis formula to each part individually, which leads to more accurate sentiment classification.

In general, this covers with sentences like “I don’t like going to school because I hate studying”, “I don’t care about the news! It’s really boring”, “I love eating and enjoying playing basketball”, “I think the movie was good, but it was a bit boring”, etc. Here, simply “adding” the individual scores does not work for every sentence, as we can come up with more complicated sentences that lead to the wrong sentiment score.

Moreover, methods like simple average, exponential average, multiplicative interactions etc. cannot account for all cases. Let’s first look at some of these methods and see what they are and why they don’t work.

3.4.1 Basic Addition

Say we have a sentence with 2 or more parts, or there is either a conjunction, or a sentence boundary present. We want to calculate $S_{final.}$

The basic addition approach therefore results in the formula:

$S_{final} = S_1 + S_2 + S_3 + \dots$

However, we can easily give a counter example on why this fails. Consider the sentence “I really like my home. It’s pretty, beautiful, and comfortable. There is nothing better than it”.

Here, we can keep adding sentences and at some point, $S_{final} > 1$, since we are adding terms. Note that this is impossible since our value needs to be between -1 and 1. Moreover, even if we were to “normalize” S_{final} between -1 and 1.

Now, it could either be normalized back to 1, $1 – \varepsilon$ (where $\varepsilon > 0$). Note that both these cases are incorrect, since having a score of 1 would mean there is nothing more positive than this sentence, and if its slightly smaller than 1, then there can be 2 or more sentences with the same value but one would have this score due to it being normalized, and one would have it naturally without any normalization.

3.4.2 Simple Average

The simple average approach gives us the
$S_{final} = (S_1 + S_2 + S_3 + \ldots S_n) / n$

Again, it is easy to come up with an example that fails. Consider the sentence "I like oranges, and I also like mangoes". The sentence would be split up as:
I like oranges   I also like mangoes
Say that S_1, S_2 in this case are equal to x, y respectively. The problem with taking average is that it will always bound our answer. $(S_1 + S_2) / 2$ will always have a value less than (or equal to) x or y. Note that this should not be our final answer as this is very inconclusive, since we wanted a sentiment greater than both x and y.

3.4.3 Exponential Average

The exponential average approach calculates the final sentiment score as
$S_{final} = \alpha(S_{current}) + (1-\alpha) S_{previous})$

The Exponential Average method is designed to capture the dynamic nature of sentiment over time within a sentence. By adjusting the value of α, we can assign more weight to either the more recent or older parts of the sentence. This approach is straightforward to implement and appears realistic, as it mirrors the process of adjusting sentiment based on preceding or subsequent content. However, it shares the same limitations as the basic addition method, as it can produce sentiment scores that exceed the desired range of -1 to 1.
Moreover, selecting an optimal alpha value is challenging, as it requires a careful balance between giving appropriate weight to recent or earlier segments of the sentence. Misjudging this balance can result in skewed sentiment scores, either overemphasizing or underemphasizing portions of the text. From my own experience testing this formula with sentences of varying complexity, I found that it fails to provide accurate sentiment

analysis, especially when dealing with conjunctions and complex sentence structures.

3.4.4 Multiplicative Interaction

Multiplicative Interaction tries to calculate S_{final} as

$S_{final} = S_1 * f_1 (S_2) * f_2 (S_3) * f_3 (S_4) * \ldots$

where f_i is a function that softens or amplifies the effect of a sentiment

The core idea behind this method is to account for shifts in sentiment intensity, providing a more nuanced calculation of the overall sentiment. By multiplying individual sentiment components in this manner, the approach can capture complex interactions between different parts of a sentence or text. However, determining the optimal form of f_i for each component is a highly labor-intensive process.

Additionally, this approach presents challenges such as the potential for exponential growth in the value of the sentiment score, which can complicate normalization. The method is also highly sensitive to small changes in individual sentiment components—particularly, a single negative term could disproportionately influence the final score, potentially altering the sentiment in a way that was not intended.

While this issue could theoretically be mitigated by carefully optimizing the function f_i, the process itself is paradoxical: choosing an optimal function becomes increasingly complex as the number of sentiment components grows.

3.4.5 Vector-Based Addition

This approach attempts to calculate S_{final} as

$\boldsymbol{S}_{final} = \boldsymbol{S_1} + \boldsymbol{S_2} + \boldsymbol{S_3} + \ldots$

Instead of treating our score as a number between -1 and 1, this method attempts to represent it as a vector of n-dimension. This is the solution to effectively add two or more sentiments. Vector addition can preserve sentiments and their shifts. It adds them in a meaningful way as it preserves each component. Choosing our dimensions carefully can give us a way to deal with any sentence of any complexity, given than we can decompose a scalar value between -1 and 1 into an n-dimension vector. Then we can come up with rules to combine two vectors based on their components. The only

downside is that the more dimensions we add, the more complicated the combination process gets.
The code written to analyze sentiments involves treating sentiments as a 3-Dimensional vector. It has the following components: -

1. Magnitude (How strong the sentiment is)
2. Polarity (Negative, Neutral or Positive)
3. Intensity (Degree of emphasis)

We can add more dimensions and change our combination process to account for more complex sentiments, but it's sufficient to get good approximations for sentiments with just these 3 components. More on this will be discussed later.

3.5 V-Scaled Arctan Normalization:

After applying all the necessary adjustments to the sentiment (such as negation, quantifiers, and conjunctions), we may notice that the resulting sentiment score might not always fall within the desired range of -1 to 1. To address this, we apply a normalization technique known as *V-Scaled Arctan Normalization.*
To ensure the sentiment score remains within the desired (-1, 1) range, we apply a scaling factor (V-scaling). The scaling factor adjusts the arctan output to map the results directly to our desired output range. This step ensures that the final sentiment score is bounded within the proper limits. This method ensures that any sentiment score, regardless of how extreme the adjustments were, will fall within the expected range of -1 (strongly negative) to 1 (strongly positive).

Mathematically, the *V-Scaled Arctan Normalization* looks like: -

sentiment = base sentiment * negation * quantifier/diminisher adjustment
final score = v * arctan(sentiment), where v $\cong 0.636$

By considering the interaction of these elements and applying them in a structured manner, our system can provide highly accurate sentiment analysis for complex sentences. The use of vector-based addition, which allows for the decomposition of sentiment into multiple dimensions, ensures that even the most intricate sentiment expressions can be captured effectively.

This approach overcomes the limitations of traditional sentiment analysis methods, making it well-suited for real-world applications like social media monitoring, customer feedback analysis, and product reviews, where context and subtlety are key.