Explaining Sentiment Predictions:
A Comparison of Attention Maps and SHAP Attributions

Ali Lowni — s1097796

29 March 2025

Abstract

Language models can predict the overall sentiment of a sentence with high accuracy, but we still do not know exactly *why* they choose one label over another.

This report compares two popular ways to explain a model's prediction. The first is the technology behind many NLP systems today called attention heads, this is how the model connects the tokens. The second method is a post-hoc analysis using SHAP, a tool that asks: which words contributed to the model's final decision?

The goal of this report is not to explain the model's decision itself, but to compare how well these two explanation methods align with each other. Do they highlight the same parts of the input and Can we trust attention to tell us what mattered?

In our analysis, we find that there are many cases where attention and SHAP do not agree. So attention should not be interpreted as a built-in explanation on its own.

1 Introduction

Large language models (LLMs) like GPT are almost everywhere in our lives now, from simple chatbots to more advanced LLMs used to design proteins and molecules for drug discovery like 310.ai. These NLP systems are built using an architecture known as attention [Vaswani et al., 2017], which is one of the most important foundations behind the AI tools we use today.

Even though the foundational paper behind attention came out nearly a decade ago and the core technology is open source, people still refer to these models as "black boxes." We believe the main reason for that is the complexity of these systems. Most models have billions of parameters, operate in high dimensional spaces, and are trained over months on massive GPUs. This makes their reasoning process difficult to follow and usually it's not intuitive to humans why a model gives a certain output.

In this report, we're focusing on a specific type of AI models called, sentiment models. These models are used to label sentences such as comments or reviews as either negative or positive. They have many real-world applications: from content filtering to analyzing customer feedback.

But sentiment analysis is more than just getting a label and knowing why the model predicted something is just as important as the prediction itself. For example, if we run sentiment analysis on course evaluations for University, a simple label like positive or negative might not be helpful unless we also understand what part of the feedback influenced that label.

2 Related Works

2.1 Attention as Explanation

In transformer-based models, every prediction relies on attention scores, a score that indicates how much each word in the input is "looking at" other words. These attention weights are the foundation of the model's ability to understand context.

Because attention scores are already computed during prediction, they are often visualized as heatmaps. This has led many to believe that attention can be used as a natural built-in explanation mechanism. Since we can use them to see what the model is focusing on. These maps are easy to generate and often look convincing to humans. But the key question is: do attention weights actually tell us why the model made a certain decision?

In their well-known paper, Jain and Wallace [2019] tested whether attention maps truly reflect important features. They manipulated the attention weights (changing them or replacing them with entirely different distributions) and found that the model's output often didn't change! This raises a big red flag. If the explanation can change drastically without affecting the outcome, it probably wasn't a reliable explanation in the first place.

In another reaserch Serrano and Smith [2019] proposed a method to test the impact of high attention tokens. Their work suggests that attention may highlight correlated tokens rather than causal ones. However, they also acknowledged that attention can still offer useful signals, especially when combined with other techniques like input gradients or layer propagation.

2.2 Post-Hoc Explanation: SHAP and LIME

One of the central questions in explainability is: what exactly made the model choose this label? Post-hoc explanation methods attempt to answer this by analyzing the model's output and working backward to figure out which parts of the input mattered most. This is a bit like reverse engineering the decision, not from the inside of the model, but from the effect it produced.

One of the most powerful and widely used post-hoc methods is SHAP, which stands for SHapley Additive exPlanations [Lundberg and Lee, 2017]. The core idea behind SHAP comes from game theory. Think of each word in a sentence like a player in a game, and the model's output is the total score. SHAP tries to divide the credit among the words based on how much each one contributed to the final prediction. This is done by looking at all possible combinations of words and asking how the prediction changes when we include or exclude each one.

The SHAP paper [Lundberg and Lee, 2017] shows that SHAP actually unifies several earlier methods, including LIME, into one consistent framework. It also introduces new algorithms like Kernel SHAP and Deep SHAP (for deep neural nets), which make SHAP practical to use on real world systems.

Another strength of SHAP is that it's model agnostic. Unlike attention scores that are part of the model's internal architecture, SHAP can be used with any kind of model. Such as decision trees, transformers, or CNNs, and still produce explanations at the word or feature level.

3 Methods

3.1 Model

We wanted to keep this project simple and efficient for the CPU so this project uses the distilbert-base-uncased-finetuned-sst-2-english model from HuggingFace. It's a lighter version of BERT trained specifically for sentiment analysis, and fast, yet complex enough to demonstrate realistic behavior. Most importantly, it gives us access to attention weights, which we need to compare with SHAP explanations.

I used the HuggingFace pipeline API to create a sentiment classifier. This setup not only gives us prediction probabilities for both positive and negative sentiment, but also integrates with SHAP's TransformersPipeline wrapper. That way, I could extract both attention-based and SHAP-based explanations using the same input and model.

3.2 Explanation Methods

For every sentence in the test set, we extracted two types of explanations:

- **Attention Scores:** From the final attention layer of the model. I averaged across all attention heads, then extracted the attention from the [CLS] token to each input token. The raw attention scores were normalized between 0 and 1 for easier comparison across examples.
- **SHAP Scores:** Using SHAP's 'Explainer', I computed the contribution of each token to the final predicted class. The SHAP values show how much each token increased or decreased the logit score for the predicted sentiment (positive or negative).

While this report includes the main findings and visuals, a more extensive analysis is available in the Jupyter notebook. It contains additional plots, and all the evaluation code used in this project.

4 Results

4.1 Correlation Analysis: Across Sentences

We compared attention and SHAP values for all 50 balanced sentences in the dataset. For each sentence, I computed the Pearson correlation between the attention scores and the absolute SHAP values for each token. These correlations give a sense of how often the two methods agree on what matters.

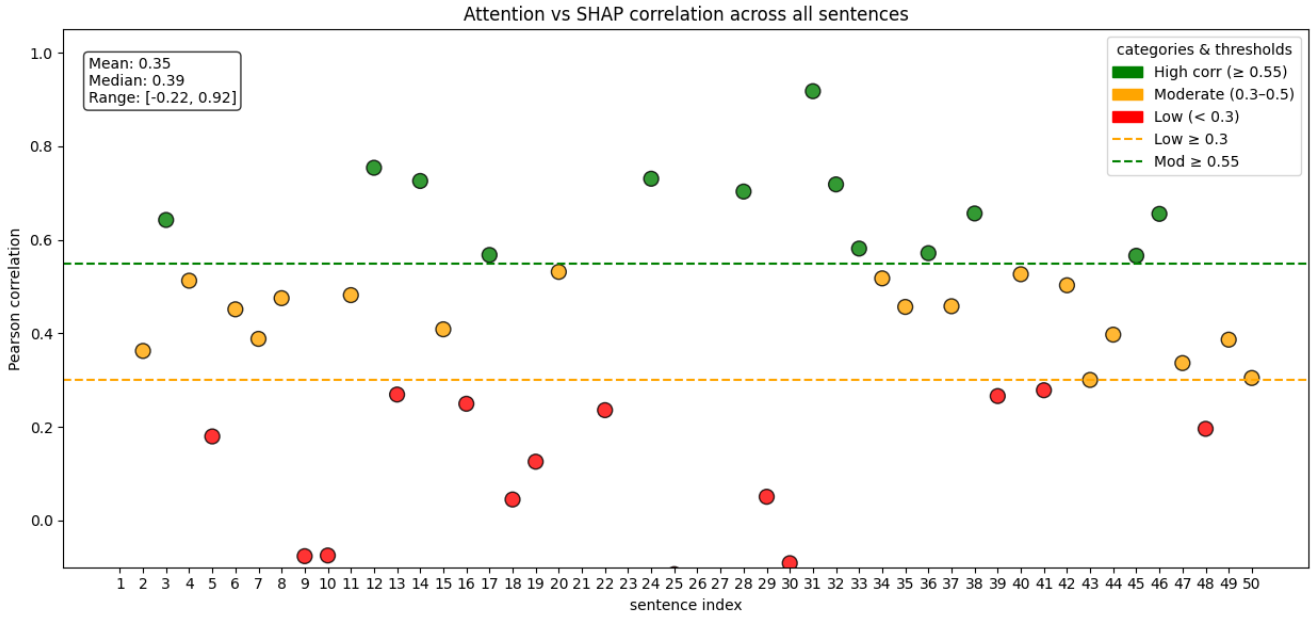

Figure 1: Correlation between attention weights and SHAP scores across 50 sentences.

As we can see in Figure 1, we observed:

- The average correlation was **0.35** , 38% of them had low correlation (< 0.3).

This means that in many examples, attention and SHAP were not just different, they were quite unrelated. SHAP might highlight key words like "cold" or "terrible" while attention might be spread across neutral or structural tokens like "was" or "the".

4.2 Correlation Analysis: Attention vs SHAP Across All Tokens

To get a more global view of the relationship between attention and SHAP scores, I combined all token scores from every sentence into one scatter plot. Each point in Figure 2 represents a single token from one sentence, with its normalized attention score on the x-axis and normalized SHAP value on the y-axis.

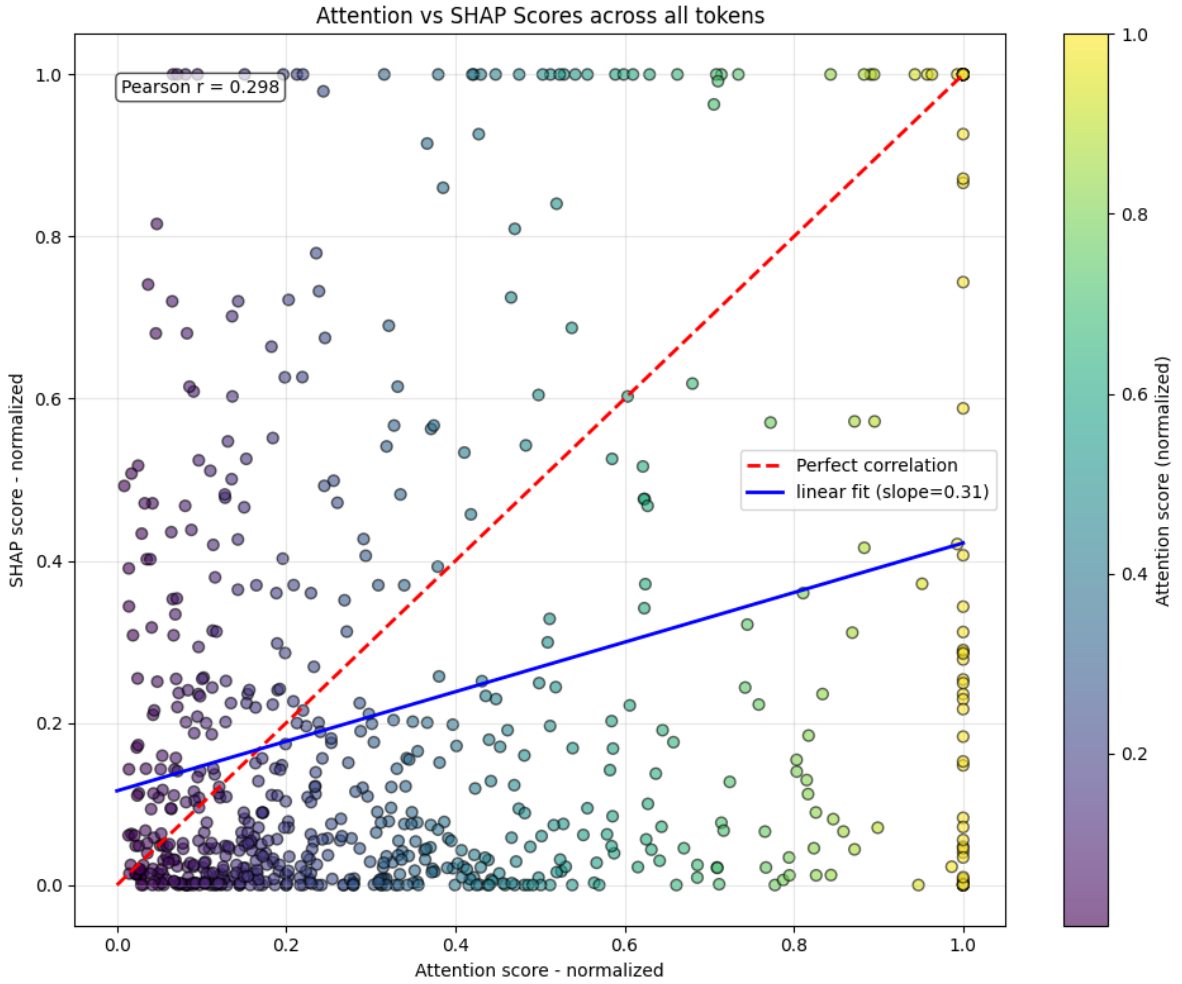

Figure 2: Scatter plot of all tokens across all sentences. Each dot is a token, colored by attention intensity. A red dashed line shows perfect correlation; the blue line is the actual linear fit.

The Pearson correlation across all tokens was r = 0.298, which confirms what we saw in previous sections. Attention weights and SHAP values often diverge. Some tokens received high attention but contributed little to the prediction (low SHAP), while others had high SHAP importance but were barely attended to. This suggests the two explanation methods are capturing different aspects of what the model cares about.

4.3 Heatmaps: Importance Patterns

Finally, I created heatmaps to show patterns across all 50 test sentences.

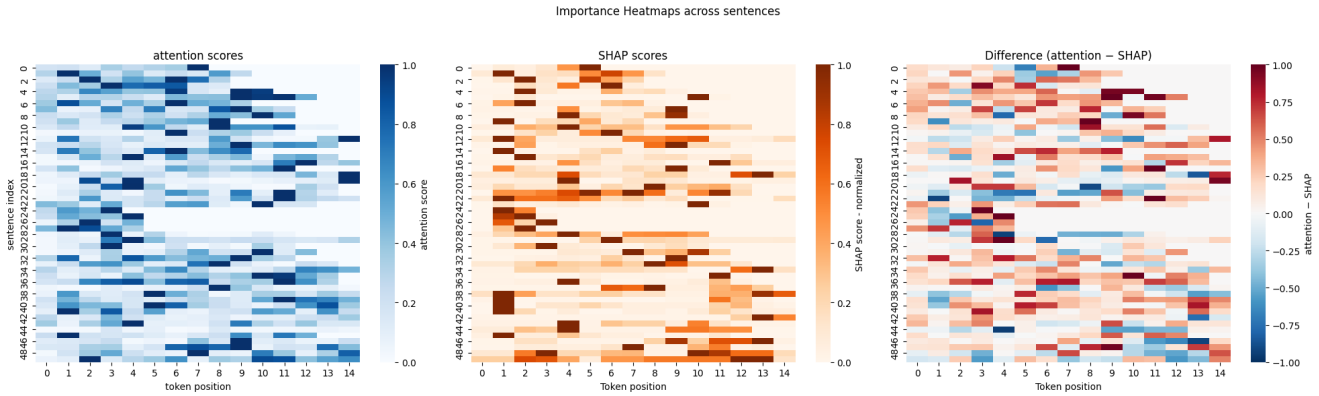

Figure 3: Importance heatmaps for all 50 sentences (trimmed to 15 tokens). Each row is a sentence, and each column is a token position. Blue: attention, orange: SHAP and the right heatmap shows where the methods disagree.

- Attention tends to be more evenly spread across tokens.
- SHAP scores are more "spiky", they focus on just a few important words.
- The difference heatmap, shows strong local mismatches. Red patches indicate where SHAP saw high importance but attention did not; blue indicates the reverse.

So while attention heatmaps may seem intuitive, they often highlight structurally or syntactically important tokens. In contrast, SHAP consistently picks out semantically meaningful tokens, the words that actually drive the prediction according to the model's output behavior.

5 Discussion

There were some cases where both methods aligned well, usually when the sentence was very short or extremely obvious in sentiment. But most of the time, the two explanations focused on different tokens. This supports the view that attention is not a reliable explanation by itself. It may be helpful as a visual guide, but it shouldn't be trusted as the true reasoning behind a prediction.

An interesting future direction would be to explore whether these differences also appear in models that use chain-of-thoughts structure. Do models attend to the same tokens that SHAP considers important in multi-step reasoning? This could open up a new path of research into how language models generate the next tokens.

6 Conclusion

In this project, we compared two explanation methods for transformer-based sentiment models: attention maps and SHAP values. Our goal was to understand how these two tools align and whether we can trust either of them as explanations.

We observed that while attention scores are useful for visualizing where the model looks, they do not always reflect what actually drove the decision. SHAP provided more consistent, human aligned explanations, but at a higher computational cost.

References

Sarthak Jain and Byron C Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Sofia Serrano and Noah A Smith. Is attention interpretable? *arXiv preprint arXiv:1906.03731*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.