Interpreting Language Models for the Detection of Sexist Content on Social Media

Tancredi Bosi Francesco Farneti Giovanni Grotto

Abstract

Our work focuses on analyzing the decision-making processes of language models in the context of detecting sexist or gender-based offensive content. We used a corpus of comments collected from the social platform X (Twitter) as our study material, with the aim of clarifying the mechanisms through which these models attribute relevance to specific textual elements during the classification phase. The main challenge of this task lies in the intrinsic complexity of natural language, where words take on different meanings and nuances depending on the context of use. This characteristic of human language makes this task a particularly stimulating problem with great social relevance in the digital era. Our project aims to implement methodologies to understand the reasoning that guides language models in the contextual interpretation of sentences. In general, as model complexity increases, there is a corresponding increase in opacity in their decision-making processes, but today we can address this issue thanks to specialized explainability libraries.

1 Introduction

The increasing complexity of language models, particularly those based on transformer architectures, has created a significant interpretability challenge, which has become highly relevant both from what we might define as an ethical perspective and from a purely technical standpoint.

It is crucial for identifying and mitigating algorithmic bias, as demonstrated by the study of Ribeiro et al. (2016), which showed how apparently accurate models can be based on spurious correlations that do not reflect a real understanding of the data [1].

Moreover, from a regulatory perspective, explainability is becoming a legal requirement in many contexts; for example, the General Data Protection Regulation (GDPR) of the European Union establishes the ”right to explanation” for algorithmic decisions that significantly affect individuals [2].

To summarize, the importance of this topic is in fact highlighted by the increasing use of language models in critical and sensitive areas spanning diverse environments, ranging from judicial systems to job selection processes, where erroneous or biased decisions could generate significant consequences on people's lives. Finally, the ability to interpret the behavior of language models constitutes a valuable learning tool for researchers in the field, allowing them to identify recurring patterns, limitations, and potential areas for improvement.

We conduct our experiments on a labeled dataset of tweets, where each sample is annotated for presence or absence of sexist content. Our goal is:

- To assess which words or features most influence model predictions.
- To evaluate how consistent and complementary the different explainability techniques are in highlighting relevant aspects of the input.

By combining global and local explainability techniques with attention analysis, we aim to build a more comprehensive picture of how modern NLP models process and classify potentially harmful content.

2 Background

In recent years, deep learning models have achieved state-of-the-art results in natural language tasks, but they are also hard to understand. For detecting sexism in tweets, where language is often hidden or complex, it's very important to explain how the model works to ensure fairness and trust.

To tackle these challenges, various post-hoc explainability techniques have been proposed. These methods aim to provide human-interpretable insights into model behavior without altering the underlying architecture. In our project, we focus on three widely used approaches: LIME (Local Interpretable Model-Agnostic Explanations) [1], SHAP (SHapley Additive exPlanations)[4], and attention weight analysis for Transformer-based models [5].

LIME explains predictions by using a simple model (like linear regression) around one example. It changes the input (for example, by removing words) and sees how the output changes, giving importance scores to each word. But because it only looks locally and doesn't use the full model, it can give wrong explanations in NLP tasks [6].

SHAP assigns a fair contribution to each input feature by using Shapley values from game theory. It is compatible with any machine learning model and is especially useful for comparing the behavior of different classifiers in a consistent and interpretable way.

Another way is to look at attention weights in models like BERT or RoBERTa [7]. These weights show what the model is focusing on, but they are not always reliable [8]. Still, they can help understand how the model reacts to gendered or abusive words.

Past studies have explored explainability in various domains, including vision [9], medical NLP [10], and large language models [11], but relatively few works combine these methods for bias or fairness-related classification tasks. Our work extends this direction by applying and comparing multiple explainability techniques in the context of sexism detection, a nuanced subtask of abusive language identification.

3 Metodologies

Dataset

The dataset used in this project consists of tweets annotated for the presence of sexist content. The dataset is pre-split into three separate JSON files: `training.json`, `validation.json`, and `test.json`; each tweet of the three files is labeled by six independent annotators with a binary value (YES/NO). A hard label is computed by majority voting, and tweets with ties are discarded. Additionally, only English-language tweets are considered, based on the `lang` field.

The three final datasets (training, validation and test) include the following fields:

- `id_EXIST`: unique tweet identifier;
- `tweet`: raw tweet content;
- `hard_label_task1`: final binary label derived from majority vote.

During preprocessing, tweets are cleaned to remove mentions, hashtags, URLs, punctuation, and non-ASCII characters, and then converted to lowercase. These cleaned versions are then tokenized using the tokenizer associated with the chosen transformer model, preparing them for input to the classifier.

Architectures

For the classification task, we employed a transformer-based architecture specifically designed for hate speech detection on social media. In particular, we used the `cardiffnlp/twitter-roberta-base-hate`

model, which is a variant of RoBERTa pre-trained on Twitter data and fine-tuned for hate-related content classification.

The implementation leverages the HuggingFace Transformers library. Tweets are tokenized using the associated tokenizer and passed to the `AutoModelForSequenceClassification` module. The dataset is managed using the HuggingFace `datasets` library, with `DataCollatorWithPadding` used to dynamically pad inputs during batching.

The model was fine-tuned using the following configuration:

- Learning rate: 1×10^{-6};
- Batch size: 4 (training), 8 (evaluation);
- Weight decay: 0.2;
- Number of epochs: 4;
- Evaluation and checkpoint strategy: per epoch.

Training was performed using the HuggingFace `Trainer` API, and evaluation metrics included accuracy and macro-averaged F1 score. Fine-tuning the model on our dataset led to an increse in performance: both accuracy and F1 score increased by approximately 4%, reaching an overall accuracy of 87% on the validation set.

Explainability Tools

To interpret the model's predictions and understand which features contributed most to the classification decisions, we employed three explainability techniques: LIME, SHAP, and Attention Weights Analysis. These tools allowed both local (individual prediction) and global (dataset-wide) inspection of the model's behavior.

- **LIME**: Local Interpretable Model-agnostic Explanations was used to perturb tweet inputs and fit a linear surrogate model locally around each prediction. The model's probability outputs were obtained through a custom `predict_proba()` function. LIME was configured with 500 samples and 10 top features, and results were visualized using the `show_in_notebook()` function provided by the library.
- **SHAP**: SHapley Additive exPlanations was applied using SHAP's deep learning framework. It computes token-level importance values based on game-theoretic principles. This method allowed us to obtain more consistent and interpretable insights into the contribution of each word across the entire dataset.
- **Attention Weights Analysis**: As a built-in mechanism of the RoBERTa architecture, attention weights were extracted directly from the transformer. By averaging attention scores across heads and layers, we visualized which tokens the model focused on the most during prediction.

The combination of these techniques enabled a robust analysis of model behavior, highlighting not only what the model predicted but also why it did so.

4 Attention Weights Analysis

Attention weight analysis is an interpretability technique that examines the attention scores generated by transformers models to shed light on their decision-making processes. These weights quantify how much importance the model assigns to different parts of the input during prediction.

The primary aim of this analysis is to uncover which input tokens the model consider most relevant for a given output. By looking at these weights, we can explore questions such as: Which words influenced the classification decision the most? or Does the model focus on semantically or syntactically important elements?

One of the key advantages of attention weight analysis is its model-intrinsic nature. It utilizes information already computed during the forward pass of attention-based models, requiring no external perturbations or approximations.Furthermore, attention scores can sometimes provide human-interpretable explanations that align with linguistic intuition—for instance, emphasizing sentiment-laden words in sentiment analysis tasks.

Nevertheless, this approach has notable limitations. A central concern is whether attention truly constitutes an explanation. High attention scores do not necessarily imply causal importance: a token might be heavily attended to without significantly impacting the final output. Additionally, attention patterns vary across layers and heads, complicating aggregation and interpretation. There's also a risk of over-interpretation, particularly when attention maps are taken at face value without additional validation.

Despite these caveats, attention weight analysis remains a useful tool for probing the inner workings of Transformer-based models. When applied thoughtfully and in tandem with other interpretability techniques, it can yield meaningful insights into how models process and prioritize input information.

4.1 Results

We performed two complementary analyses:

- **Layer and Head-Level Specialization**: examining how attention behavior varies across different parts of the model.
- **Token-Level Aggregation Across the Dataset**: identifying broader patterns in attention distributions over the input tokens.

These approaches provide both micro-level (per-layer/per-head) and macro-level (dataset-wide) perspectives on the model's attention behavior and its alignment with human intuitions about identifying sexist language.

4.1.1 Layer and Head-Level Attention Specialization:

This first analysis focuses on how attention is distributed across different layers and heads within the model. Specifically, it aims to determine whether certain components specialize in capturing specific linguistic structures—such as syntactic roles or sentiment-laden words—and how these contribute to final predictions.

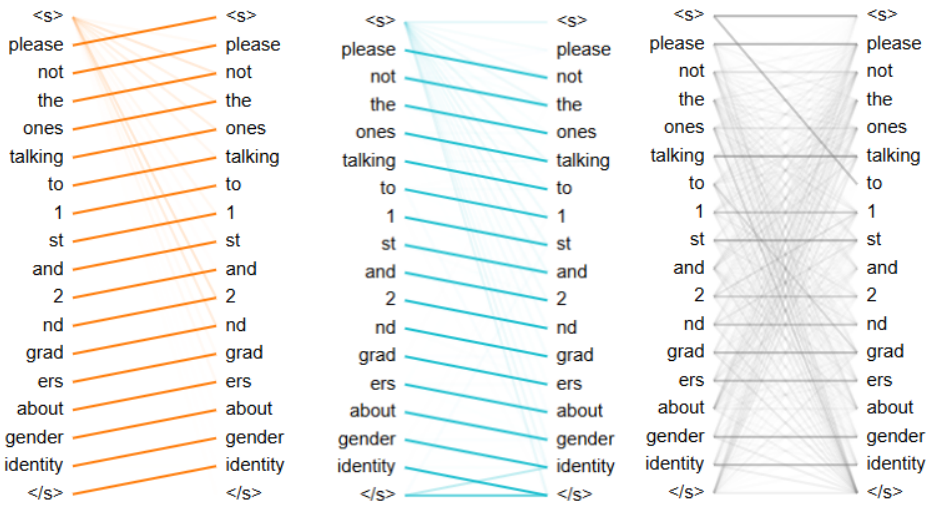

Figure 1: Attention patterns from heads 7, 9, and 11 of layer 1.

As shown in Figure 1, the attention heads in the **first layer** focus primarily on simple and local relationships. We observe distinct specialization among different heads:

- One head consistently attends to the **previous token**, suggesting it may help the model build up a sense of sequential order or preserve syntactic flow from left to right.
- Another head attends to the **next token**, possibly supporting the model in capturing right-to-left dependencies which are useful in understanding bidirectional context.
- A third head appears to attend mainly to the **token itself**, effectively reinforcing self-identity or token preservation early in the model's encoding.

These behaviors are likely crucial in encoding local syntax and continuity, forming the foundation upon which deeper semantic representations are built.

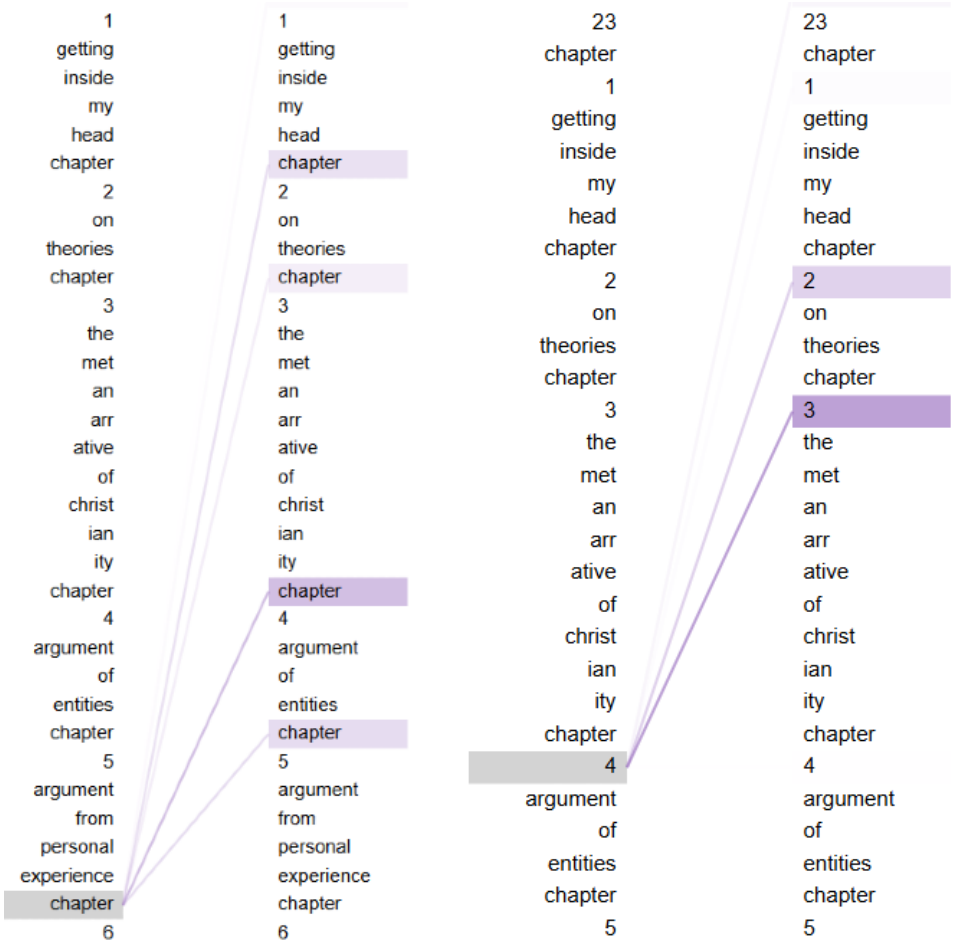

Figure 2: Attention pattern from head 4 of layer 3.

In the **third layer**, as seen in Figure 2, more complex and abstract attention patterns begin to form. Specifically, head 4 exhibits:

- Sensitivity to the **repetition of previous tokens**, where repeated words receive higher attention weights.
- Recognition of **numeric sequences**, with tokens attending back to previous numbers in a gradually decreasing manner, suggesting a decaying memory of previous numerical elements.

This behavior implies that the model is developing a mechanism similar to recurrence or counting, which may support numerical reasoning or pattern matching tasks.

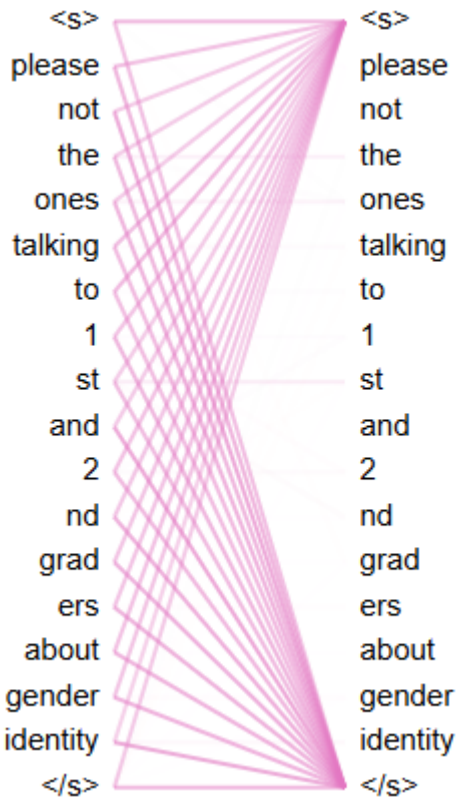

Figure 3: Attention pattern from head 6 of layer 5.

Figure 3 displays a less interpretable, yet intriguing, attention pattern where all tokens in the sequence primarily attend to the **[CLS]** and **[SEP]** tokens. This may serve to inject global contextual information throughout the sequence. By linking to [CLS], which typically aggregates global meaning, and [SEP], which denotes boundaries, the model may be anchoring the tokens to sentence-level semantics or task-relevant signal.

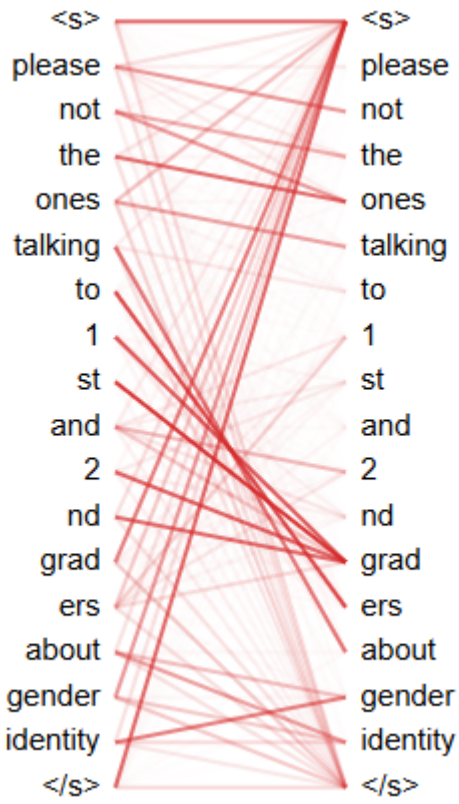

Figure 4: Attention pattern from head 3 of layer 6.

In deeper layers, notably in Figure 4, attention heads begin to capture more **semantic relationships and roles**. Examples from the head include:

- Pairs like *the* $\rightarrow$ *ones* and *ones* $\rightarrow$ *talking*, hinting at the emergence of **grammatical subject-object structures**.
- Relations such as *talking* $\rightarrow$ *about*, showing the model is able to link **verbs to their prepositional objects**, possibly facilitating predicate-argument structure resolution.
- In sequences such as *"to 1st and 2nd"*, we see multiple tokens attending to a shared referent like *"graders"* while the **"and"** in the middle of the sentence is not attending to it, suggesting **co-reference tracking** or hierarchical phrase understanding.

These observations suggest that the model is constructing more abstract semantic roles, a key mechanism for understanding sentence meaning and structure.

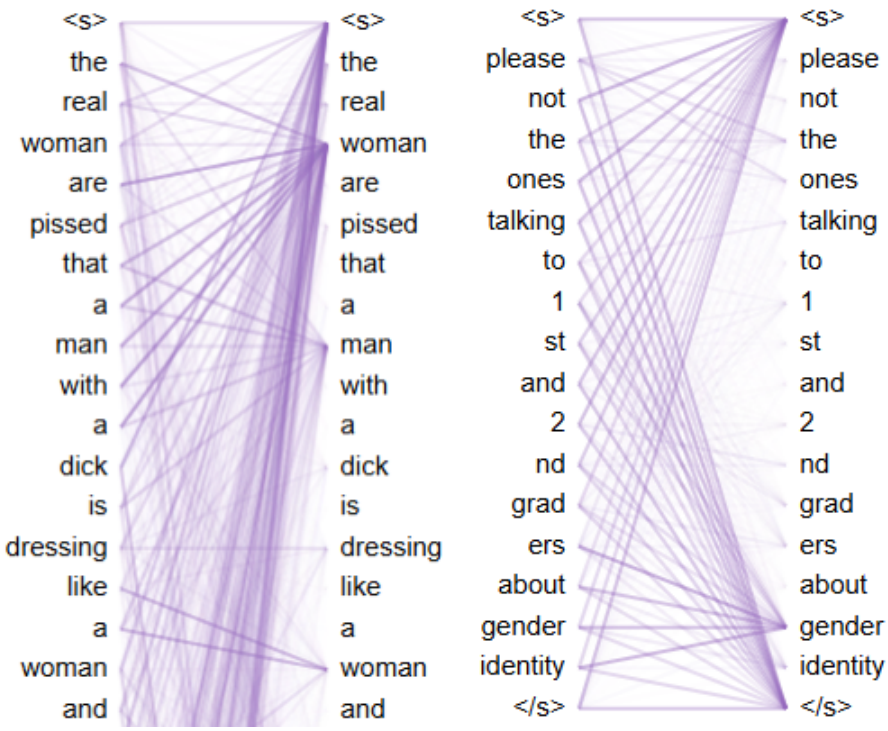

Figure 5: Attention pattern from head 4 of layer 9.

In the very deep layers, such as **layer 9** in Figure 5, attention heads start to encode features that are **directly relevant to the downstream task**, such as sexism detection. We observe that:

- Tokens related to **gender**, such as *woman*, *man*, and *gender*, receive the highest attention scores.
- These tokens dominate the attention distribution, suggesting they are being used as key indicators for the task at hand.

This indicates that the model has developed heads that function almost like **semantic filters**, isolating information relevant to classification. Such heads may be critical in driving final-layer predictions.

Our analysis shows that RoBERTa's attention heads evolve across layers: early layers capture local and positional relationships, middle layers identify patterns like repetition and numerical sequences, and deeper layers focus on semantic roles and task-specific features. This layered progression reflects how the model builds from syntax to meaning to support downstream tasks..

These findings support the view that attention heads are not uniform or arbitrary, but rather **specialize in distinct linguistic and semantic functions** as depth increases. While not all heads are immediately interpretable, a significant number demonstrate clear and consistent behaviors that can be linked to known linguistic structures.

4.1.2 Last Layer Attention Score Analysis:

The second analysis focuses on the final layer of attention, particularly the attention scores from the [CLS] token to all other tokens. Since [CLS] typically aggregates information for classification, analyzing its attention distribution can reveal which tokens were most influential in the final decision.

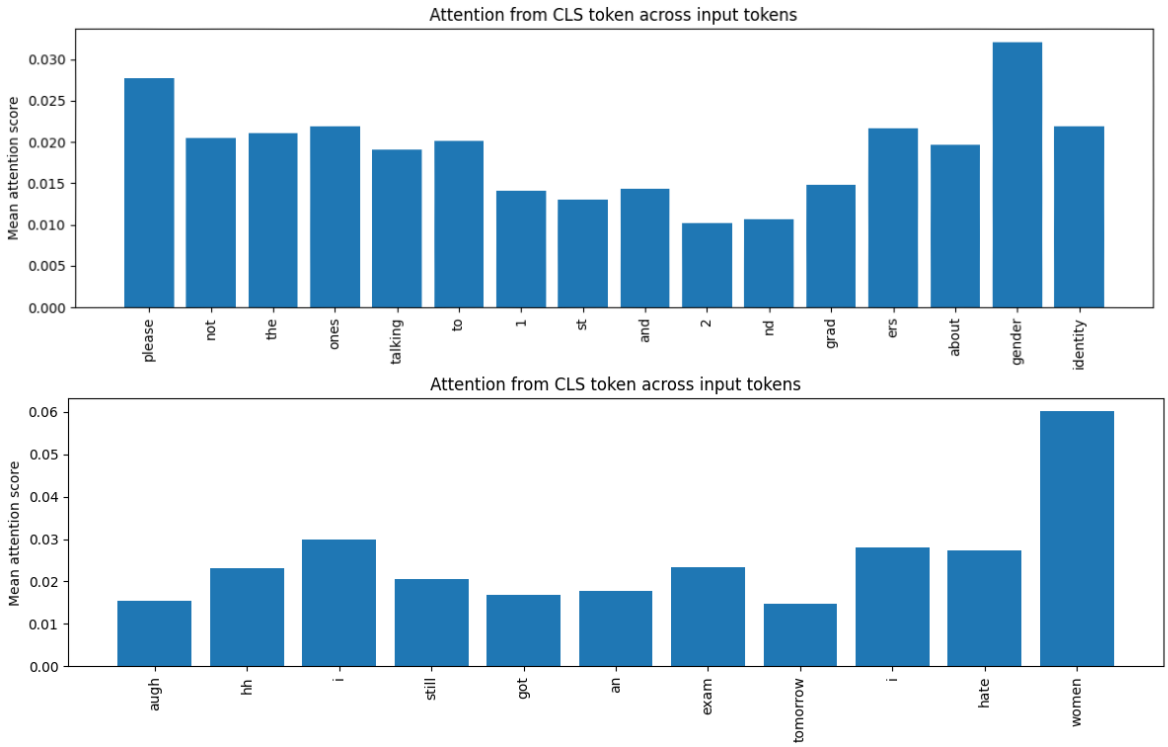

Figure 6: Attention from CLS token across input tokens for two different tweets.

In Figure e 6, we show attention bar plots for two example tweets:

- **Top Plot (Non-Sexist):** The highest attention scores are assigned to the tokens *please* and *gender*, with *identity* and *ones* also receiving relatively high attention. This suggests that the model considered the phrase *gender identity* and the polite request *please* particularly important in interpreting the tweet. In contrast, tokens such as *1st*, *2nd*, and common function words (e.g., *to*, *and*) received lower attention, which aligns with expectations.
- **Bottom Plot (Sexist):** The token *women* receives the highest attention score, followed by *i* and *hate*. This distribution suggests that the model flagged potentially harmful or offensive content, particularly focusing on the phrase *i hate women*. Such attention dynamics may play a role in tasks like toxicity detection or content moderation.

Overall, the attention distributions reflect human-like intuitions about the semantic and emotional weight of different tokens. Although not a definitive explanation, attention patterns offer a reasonable and interpretable approximation of the model's decision-making process.

To extend this analysis, we aggregated the tokens with the highest average attention scores across the validation and test sets. Table 1 lists the most frequent high-attention tokens for tweets classified as sexist vs. non-sexist.

Sexist Tweets	Non-Sexist Tweets
bald	witches
they	tits
que	sex
bathing	congratulations
you	blonde
took	sexism
gal	<s>
nails	</s>
school	stroke
skirt	lady
room	taxes
anking	harm
ush	ika
misogyny	ude
whore	mith
woman	fascists
fem	question
calling	porn
ulation	birds
pop	coins

Table 1: Most frequent high-attention tokens in sexist vs. non-sexist tweets across validation and test sets.

Some tokens in the sexist category (*misogyny, whore, woman*) appear semantically relevant, while others (*bald, nails, pop*) do not clearly connect to the classification objective. Likewise, several non-sexist tweets feature words like sex, tits, and porn, which could be controversial or ambiguous out of context.

Moreover, the presence of malformed or partial tokens (*e.g., que, ush, anking*)—likely due to subword tokenization—limits interpretability. These inconsistencies emphasize a critical point: attention alone is insufficient for reliable explanation.

These findings reinforce the argument that while attention can provide some insight into what the model may be focusing on, it should not be used in isolation as a faithful explanation of model behavior. Other methods—such as gradient-based attribution, perturbation analysis, or integrated gradients—should be considered to obtain a more robust understanding of model decision-making.

5 LIME

LIME (Local Interpretable Model-Agnostic Explanations) explains individual predictions of black-box models by approximating them locally with interpretable models, typically linear.

1. **Select an instance:** Choose a data point to explain.
2. **Generate perturbations:** Slightly modify the instance to create similar samples (e.g., by removing words or tokens).
3. **Predict outcomes:** Use the complex model (e.g., a transformer) to get predictions for perturbed samples.

4. **Weight samples:** Assign higher weights to samples more similar to the original.
5. **Fit interpretable model:** Train a simple model (e.g., linear regression) on the weighted data.
6. **Interpret coefficients:** The learned coefficients indicate which features most influence the prediction:
    - Positive = support the predicted class.
    - Negative = oppose the predicted class.
    - Larger magnitude = stronger influence.

For models like transformers, perturbations are done at the token level, and predictions are based on masked or modified token inputs.

5.1 Results

The LIME explainer was applied to analyze both individual predictions (local explanations) and aggregate feature importance across multiple predictions. The results are divided into two parts: local explanations for selected tweets and aggregated feature importance across 50 test tweets.

5.1.1 Local Explanations

Four representative local explanations for individual tweets are shown below — two classified as non-sexist and two as sexist. The choosen tweets are the following:
a) ”please not the ones talking to 1st and 2nd graders about gender identity” (**non sexist**)
b) ”chapter 1 getting inside my head, chapter 2 on theories, chapter 3 the metanarrative of christianity, chapter 4 argument of entities, chapter 5 argument from personal experience, chapter 6 human freedom, chapter 7 gods providence” (**non sexist**)
c) ”yup i hate when men rape and kill women” (**non sexist**)
d) ”aughhh i still got an exam tomorrow i hate women” (**sexist**)

Note: as defined above, the values shown in the graphs indicate how much a given token contributed consistently with the predicted class if positive. Viceversa, they indicate how much a token was ambiguous in contributing to the prediction, or in other words, how much that token “opposed” the generated prediction. This criterion for interpreting the values remains the same for SHAP as well.

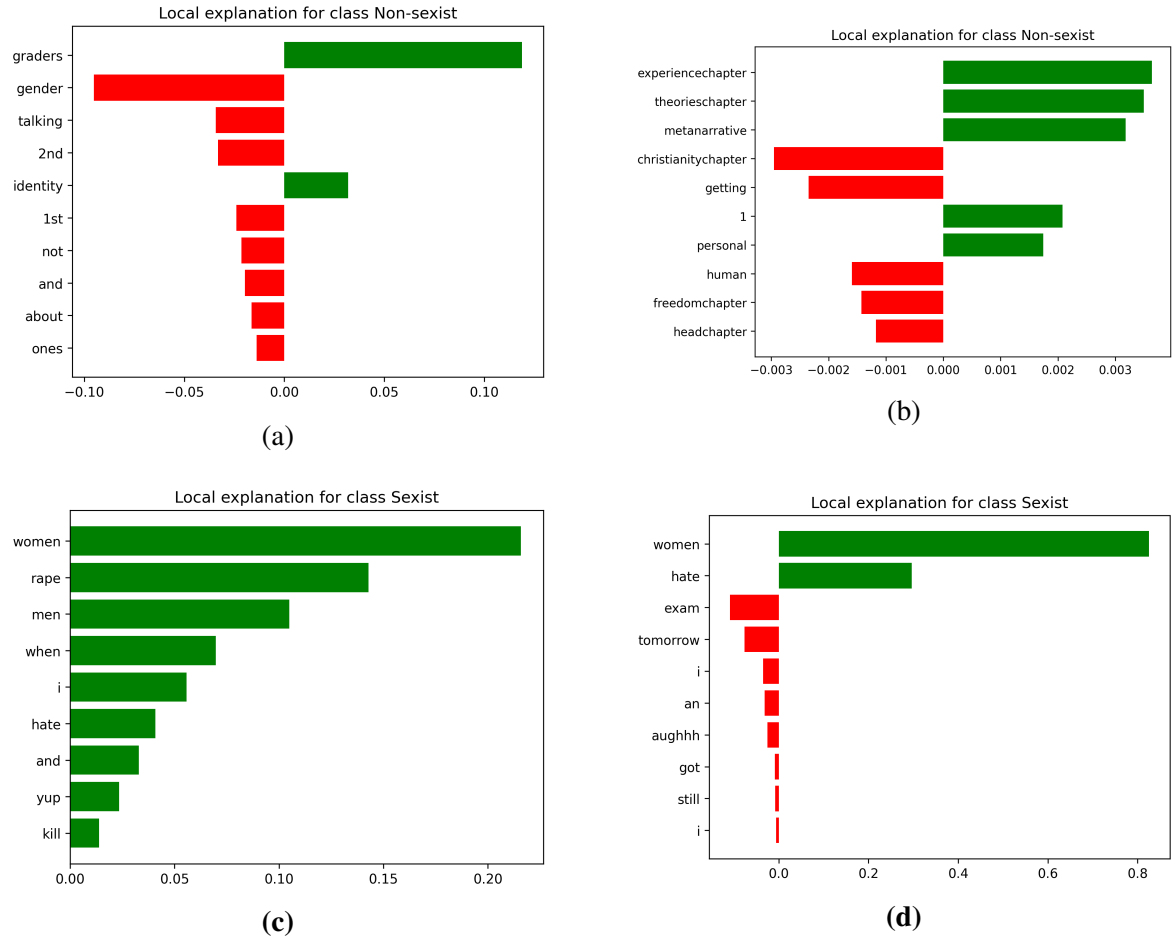

Figure 7: Local LIME explanations for tweets classified as non-sexist ((a), (b)) and sexist ((c), (d)). Feature weights indicate how much each word contributes to the classification decision.

Key observations from the local explanations:

- Non-sexist classifications often rely on the absence of strong sexist indicators rather than presence of specific non-sexist terms.
- Sexist classifications are strongly influenced by overtly hostile words (”rape”, ”hate”, ”kill”) and gendered terms used in negative contexts.
- The magnitude of feature weights is consistently higher for sexist classifications, suggesting the model has clearer markers for this class.
- The tweet *”yup i hate when men rape and kill women”*, although expressing condemnation of violence and therefore non-sexist in intent, is misclassified as sexist by the model. This likely occurs because it contains multiple strongly charged words—such as ”rape,” ”hate,” and ”kill”—which frequently appear in sexist contexts. The presence of these terms alone can mislead the classifier, highlighting a limitation where the model struggles to fully grasp the meaning behind phrases and instead relies heavily on the presence of certain keywords to make its prediction.

5.1.2 Aggregated Feature Importance

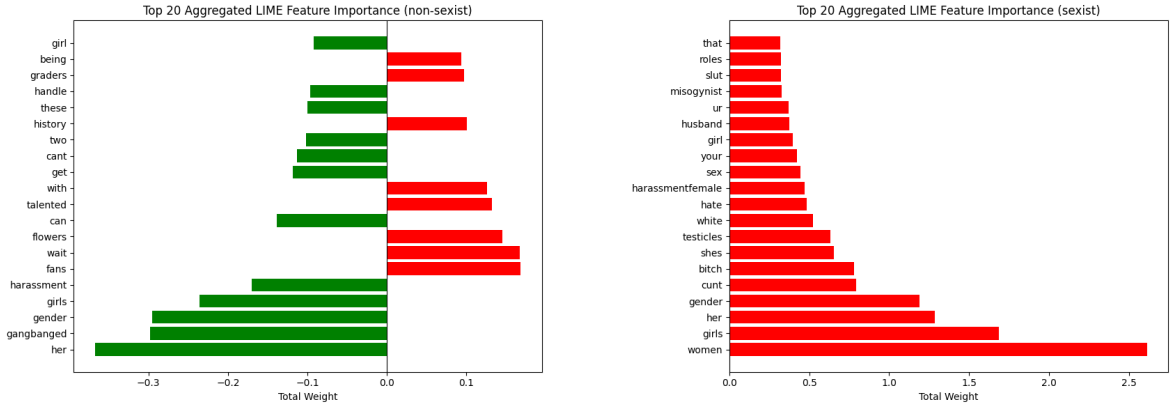

Figure 8: Aggregated LIME feature importance scores for (a) non-sexist and (b) sexist classifications.

Key findings from the aggregated analysis:

- Sexist classifications are driven by overtly misogynistic language, with feature weights an order of magnitude larger than non-sexist features.
- Non-sexist features show more variability, including some terms that might appear in either context (”girl”, ”gender”), with classification depending on usage.
- Multi-word phrases emerge as important indicators for sexist content, particularly combinations of gender terms with violent or demeaning language.

The combined results demonstrate that while the model identifies clear markers for sexist content, non-sexist classification often relies on more subtle patterns and the absence of hostile language.

6 SHAP

In this section, we describe how SHAP (SHapley Additive exPlanations) works for analyzing the importance of input tokens in Transformer-based models for classification tasks [3]. The goal is to assign a contribution score to each token in an input sequence, explaining its role in the model's prediction.

SHAP is based on cooperative game theory, and computes feature attributions using the Shapley value concept. The fundamental formula is:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \left[f(S \cup \{i\}) - f(S) \right] \tag{1}$$

Explanation of the Notation

- ϕ_i: the SHAP value for token i, representing its contribution to the model's output.
- N: the set of all tokens in the input.
- $S \subseteq N \setminus \{i\}$: any subset of tokens that does not include i.
- $f(S)$: the model output when only the tokens in subset S are present.
- $f(S \cup \{i\})$: the model output when token i is added to the subset S.
- $\frac{|S|!(|N|-|S|-1)!}{|N|!}$: the weight associated with the marginal contribution of token i with respect to subset S, based on all possible permutations.

6.1 Intuition

This formula measures how much, on average, the addition of token i to various contexts (subsets S) changes the model's prediction. It is essentially a **weighted average of marginal effects**, following the fairness principles of Shapley values.

Simplified Example

Consider the sentence `I didn't feel humiliated`, which may be tokenized as:

[I, didn't, feel, humiliated]

To compute the SHAP value for the token `humiliated`, we evaluate the model output for all subsets of tokens that exclude it (e.g., {I}, {didn't, feel}, etc.), compute the output difference when adding `humiliated`, and average these differences using the appropriate weights. For Transformer models, masked or replaced versions of input tokens (e.g., using `[MASK]` or `[PAD]`) are used to simulate their absence, in order to compute the model's output on subsets of tokens.

6.2 The Role of the Combinatorial Coefficient

$$\frac{|S|!(|N| - |S| - 1)!}{|N|!}$$

This coefficient expresses the probability that, in a random permutation of all tokens, subset S precedes token i. It ensures fairness by treating all token orderings equally and aggregating contributions across all possible contexts.

6.3 Results

The SHAP analysis was conducted in the same way as the previously explained LIME analysis. Therefore, the same tweets were used for both the analysis of individual predictions and the analysis of the 50 test tweets.

6.3.1 Local Explanations

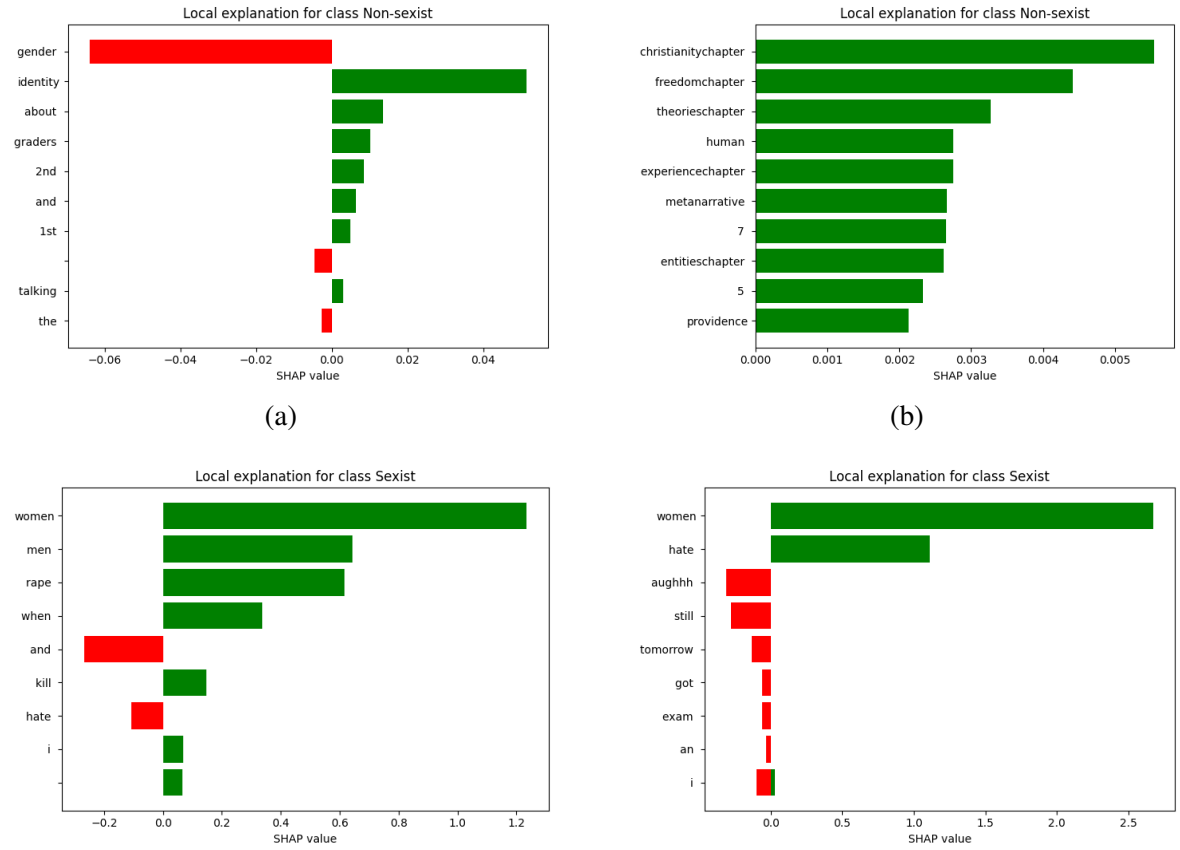

Figure 9: Local SHAP explanations for tweets classified as non-sexist ((a), (b)) and sexist ((c), (d)).

The SHAP analyses in Figure 9 highlight the most influential words for tweet classification into ”sexist” and ”non-sexist” categories. For non-sexist tweets (a, b), terms like “identity” and “freedomchapter” positively contribute to the prediction, but with very small weights, while words like “gender” can detract from it. In sexist tweets (c, d), terms such as “women,” “men,” and “rape” heavily influence the classification, suggesting their strong association with sexist content.

Note: The observations derived from the analysis conducted with LIME are also relevant to the analysis carried out with SHAP. Although the analytical methods are structured differently, the conclusions reached are fundamentally very similar.

6.3.2 Aggregated Feature Importance

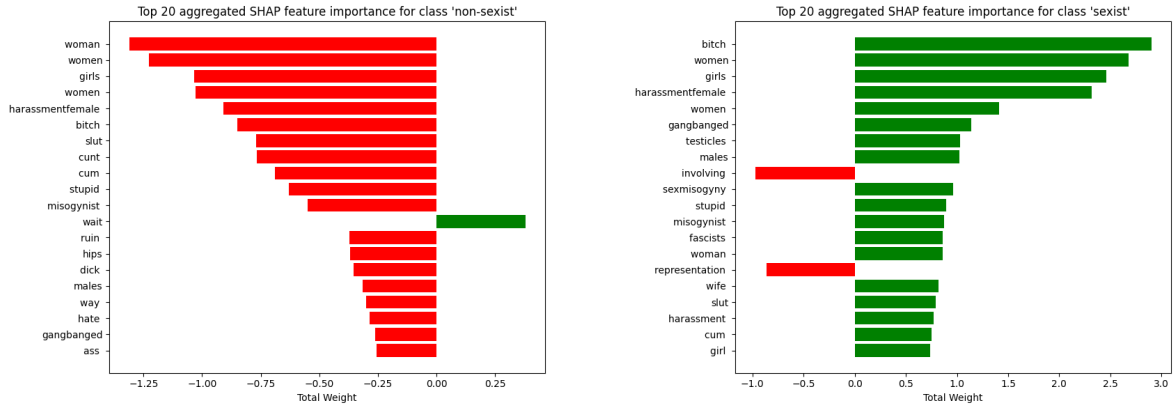

Figure 10: Aggregated SHAP feature importance scores for non-sexist and sexist classifications.

Figure 10 shows the aggregated SHAP feature importance for classifying tweets as non-sexist (left) or sexist (right). Words like ”woman,” ”bitch,” and ”girls” strongly push predictions toward the sexist class, as shown by their high positive SHAP values on the right. Conversely, these same terms contribute negatively to non-sexist classification, reinforcing their association with sexist content when used in tweets.

7 Conclusion

In this study, we applied and evaluated three interpretability methods — **Attention Weight Analysis**, **LIME**, and **SHAP**. Each method offers distinct strengths and limitations, which we compare and contrast in this chapter.

6.1 Attention Weight Analysis

Pros:

- *Model-intrinsic:* Leverages data already produced during the model's forward pass without requiring external approximations, making it computationally efficient.
- *Layer-wise interpretability:* Reveals that attention evolves meaningfully across layers — from syntactic to semantic to task-specific signals.
- *Visualization-friendly:* Visualizations provide intuitive insights into token relevance.

Cons:

- *Lack of causal grounding:* High attention does not imply causal influence.
- *Aggregation complexity:* Multiple heads and layers complicate interpretation.
- *Tokenization noise:* Subword fragments reduce interpretability.

Summary: A useful first step for internal model introspection, but limited as a standalone explanation.

6.2 LIME (Local Interpretable Model-Agnostic Explanations)

Pros:

- *Model-agnostic:* Applicable to any black-box model.
- *Local fidelity:* Explains individual predictions via local surrogate models.
- *Clear attribution:* Highlights supporting and opposing evidence per token.

Cons:

- *Stability issues:* Random perturbations can yield variable results.
- *High cost:* Requires generating and evaluating many perturbed inputs.
- *Perturbation artifacts:* Modified inputs may be syntactically unnatural.

Summary: Offers flexible, interpretable local explanations but sensitive to perturbation noise.

6.3 SHAP (SHapley Additive exPlanations)

Pros:

- *Theoretical rigor:* Based on Shapley values, ensuring fair and consistent attributions.
- *Local and global:* Can explain both individual and aggregate model behavior.
- *Consistency:* Guarantees meaningful ranking of features.

Cons:

- *Computationally expensive:* Especially burdensome for deep NLP models.
- *Independence assumption:* Assumes token independence, often invalid in language data.
- *Interpretability noise:* Some tokens receive high attributions despite minimal relevance.

Summary: A principled approach with robust attributions but computationally intensive and assumption-heavy.

6.4 Comparative Summary

Table 2: Comparison of Interpretability Methods

Criterion	**Attention Weights**	**LIME**	**SHAP**
Model Dependence	Intrinsic	Model-agnostic	Model-agnostic
Explanation Type	Internal Signal	Local Surrogate	Shapley Value-Based
Causality	Weakly implied	Approximate causal effect	Theoretically grounded
Computation Cost	Low	Medium to High	Very High

6.5 Final Remarks

No single method offers a complete solution to the interpretability challenge in NLP models. **Attention weight analysis** is fast and intuitive but limited in explanatory rigor. **LIME** excels in local fidelity but suffers from instability and perturbation artifacts. **SHAP** is theoretically grounded and consistent, but often computationally prohibitive.

Best practice involves using these methods in a *complementary fashion*. Attention can guide exploration, LIME can clarify specific decisions, and SHAP can validate attributions with theoretical rigor. When combined, they provide a more comprehensive and trustworthy view of model behavior—an essential asset in critical applications like content moderation and bias detection.

References

[1] Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. *Why should I trust you?: Explaining the predictions of any classifier*. Proceedings of the 22nd ACM SIGKDD, 2016. Available at: https://arxiv.org/abs/1602.04938

[2] Goodman, Bryce and Seth Flaxman. *European Union Regulations on Algorithmic Decision Making and a "Right to Explanation"*. AI Magazine, vol. 38, no. 3, pp. 50–57, 2017. Available at: http://dx.doi.org/10.1609/aimag.v38i3.2741

[3] Scott M. Lundberg. *SHAP Python Package Documentation*, 2024. Available at: https://arxiv.org/abs/1705.07874

[4] Lundberg, Scott M. and Su-In Lee. *A unified approach to interpreting model predictions*. Available at: https://shap.readthedocs.io

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. *Attention Is All You Need*. Available at: https://arxiv.org/abs/1706.03762

[6] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, Himabindu Lakkaraju. *Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods*. Available at: https://arxiv.org/abs/1911.02508

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Available at: https://arxiv.org/abs/1810.04805

[8] Sarthak Jain, Byron C. Wallace. *Attention is not Explanation*. Available at: https://arxiv.org/abs/1902.10186

[9] Finale Doshi-Velez, Been Kim. *Towards A Rigorous Science of Interpretable Machine Learning*. Available at: https://arxiv.org/abs/1702.08608

[10] Andreas Holzinger, Chris Biemann, Constantinos S. Pattichis, Douglas B. Kell. *What do we need to build explainable AI systems for the medical domain?*. Available at: https://arxiv.org/abs/1712.09923

[11] Rishi Bommasani, Drew A. Hudson. *On the Opportunities and Risks of Foundation Models*. Available at: https://arxiv.org/abs/2108.07258