Explaining Explanations using Large Language Models

Vlad Lungu
National University of Science and Technology Politehnica Bucharest
313 Splaiul Independentei
Bucharest, Romania
vlad.lungu3009@stud.acs.upb.ro

Ștefan Trăușan-Matu
National University of Science and Technology Politehnica Bucharest
313 Splaiul Independentei
Bucharest, Romania
and
Research Institute for Artificial Intelligence "Mihai Draganescu"
and
Academy of Romanian Scientists
313 Spaiul Independentei, Bucharest, Romania
stefan.trausan@cs.pub.ro

ABSTRACT
With deep neural networks being steadily adopted in various sectors such as healthcare, finance or tech, the need for interpretability of their results becomes paramount. In this context, not only is it difficult to produce reliable explanations of such complex models, but even more importantly, non-expert users - the main beneficiaries of real-world applications of artificial intelligence (AI) systems - may not have the capacity to comprehend and make use of these explanations. The contributions related to this paper are twofold. Firstly, we extend the work of Zytek et al. [19] by generating explanations for 3 different explainability techniques on the IMDB dataset [9], using different prompts. Secondly, we analyze the explanations produced to draw insights about the behavior and potential biases of the underlying sentiment classification model.

Author Keywords
explainable artificial intelligence; natural language processing; deep learning; transformers; prompt engineering

ACM Classification Keywords
I.2.7 Natural Language Processing

DOI: 10.37789/icusi.2025.16

INTRODUCTION
The ability of deep neural networks (DNNs) to match or even surpass human performance on certain tasks underscores their transformative potential. However, their widespread adoption is hindered by a significant challenge: the lack of transparency in their decision-making processes. For users to trust these systems, they must not only perform accurately, but also offer clear explanations of their behavior. Explainable AI (XAI) seeks to address this need by developing methods that make machine learning (ML) models more interpretable and their predictions more understandable. By demystifying how models operate, XAI fosters trust, empowers users to evaluate predictions critically, and enables system designers to refine their models effectively.

Widely adopted explanation techniques such as Local Interpretable Model-Agnostic Explanations (LIME) [12], SHapley Additive exPlanations (SHAP) [8] or Anchors [13] display their results in formats that are not intuitive, nor user-friendly and lack clarity and readability for larger audiences [6, 11]. A natural question that arises is whether we can make use of existing systems, like Large Language Models (LLMs) to further improve the field of XAI.

There are many strategies that can be employed to have LLMs improve XAI frameworks [18]. Kroeger et al. [7] assessed how GPT-3.5 and GPT-4 can work as post-hoc explainers, focusing on their ability to extract the most important features contributing to another model's decision. Four different strategies were used, but they all relied on sampling additional values from the local neighbourhood of the input being explained, with the variations being in the prompt given to the LLM. Their results suggest that the resulting explanations are faithful and can even exceed the results of SHAP in some cases. Alternatively, Bhattacharjee et al. [3], while still depending on LLMs' ability to extract relevant features, focused on providing causal explanations. To do so, they relied on the LLMs being able to discover the latent features and make minimal changes to them in order to flip the decision of the model, thus obtaining a counterfactual example.

In a similar direction, Slack et al. [15] defined TalkToModel, a dialogue-based system via which users are able to ask natural language questions like "Why was I denied a loan?" and receive the relevant answers, by having the model apply XAI techniques under the hood. Their results showed that the model was able to have an excellent understanding of the users' requests and that it was able to provide explanations for the results of other ML models in a way that would have increased the users' trust in the model, while at the same time

outperforming other existing systems. Similarly, Nguyen et al. [10] investigated how XAI can be included in a conversational agent, having natural, human-style communication as a starting point. Their work built a XAI question phrase bank, which defined general XAI questions the model could receive (e.g., "How does a specific feature impact the result?", "What is a required feature that would ensure a specific prediction") and the explainability technique (SHAP, LIME, Anchors, etc.) that would be applied under the hood.

In a somewhat different direction, but still relying on LLMs, Zytek et al. [19] investigated how LLMs can be employed to transform the outputs of existing XAI techniques like SHAP into natural language explanations that could be easily understood by the general public. Different zero-shot prompts were used for GPT-3.5 and GPT-4 models and their results were evaluated against various metrics on the student performance dataset [4] and the Ames housing dataset [5]. An extensive user analysis was conducted, showing promising results, as users generally agreed the explanations contained the sufficient amount of details, while conserving their usefulness and clarity.

Building on these core contributions, we present a prompting framework tailored to LLMs, enabling them to translate complex explainability outputs into concise and accessible narratives. This is especially valuable for long-form text inputs, where the raw outputs of explainability techniques are difficult to interpret, even for technical users. Our empirical results show that, when guided with appropriate instructions, LLMs like ChatGPT [1] and LLaMA [17] may effectively summarise and contextualise explanation data in a way that supports broader interpretability.

We also highlight the critical need for faithful explanations. While LLMs improve accessibility, we observed that they occasionally introduce interpretations not grounded in the underlying model's behavior, undermining the purpose of post-hoc explanation. This calls for further work to ensure alignment and fidelity between model predictions, explanation techniques, and LLM-generated summaries. Lastly, by examining patterns in the explanations across different inputs, we uncover evidence of bias in the sentiment analysis model used. Specifically, we note a tendency for short reviews containing minimally positive language to be misclassified as positive, suggesting over-sensitivity to certain features. Moreover, our method surfaced instances of potentially mislabeled examples in the dataset, demonstrating that this approach can aid not only in explainability but also in data validation.

PRELIMINARIES

In this work, we have used the IMDB dataset[1], a popular dataset used for sentiment analysis and Natural Language Processing (NLP). The dataset contains plain text movie reviews labeled with "positive" or "negative", representing the sentiment that is conveyed by the review. This is a balanced dataset of 50000 reviews, out of which 25000 are labeled as "positive" and 25000 are labeled as "negative". Furthermore, we use a commonly used heuristic: we split our dataset into four categories based on the review length: shorter than 50 words - very short, between 51 and 150 words - short, between 151 and 300 words - medium, longer than 300 words - long.

Predictions on this dataset consist of "positive" or "negative" labels, indicating the sentiment of a specific review. To obtain predictions, we have used the "distilbert-base-uncased-finetuned-sst-2-english", a fine-tuned version of DistilBERT [14] specifically trained on the Stanford Sentiment Treebank (SST-2) dataset [16] for sentiment analysis tasks. DistilBERT is simply a smaller, but faster pre-trained version of BERT that has been optimized to achieve faster inference and lower computational cost, while still retaining much of BERT's capacity. It offers a favorable trade-off between efficiency and performance, making it particularly suitable for tasks where computational resources or response time are limited.

The SST-2 dataset is composed predominantly of short to very short sentences and phrases derived from user reviews on Rotten Tomatoes. This contrasts with the IMDB dataset, which typically features longer, more complex reviews. Given the relatively simple structure of the SST-2 texts, and the desire to evaluate the model's pre-trained capabilities in a zero-shot setting, no additional training or fine-tuning is conducted in this work. For simplicity purposes, in the following, we will use "original BERT model" to refer to the base "distilbert-base-uncased-finetuned-sst-2-english" for which no further training has been performed.

To obtain explanations as to how the model has reached a particular prediction, we relied on three different explainability techniques. These are all popular choices with strong open-source libraries and more importantly, model-agnostic techniques, which can be applied to any type of black-box model. The first one is LIME [12] and it works by building a linear explanation model when attempting to explain a specific input, by evaluating the original model on additional inputs sampled around the original input. To apply this technique, we rely on the open-source *lime*[2] library, which besides scores associated with each word, also provides a straightforward visual explanation, highlighting the words contributing the most to the model's decision. We note that because of the sampling mentioned above, the scores of each feature can be different even when LIME is applied to the same input multiple times.

Secondly, SHAP [8] unifies seven other explainability methods by assigning scores to each feature of an input. These scores are obtained by computing the Shapley value

[1]https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

[2]https://github.com/marcotcr/lime

of each feature, following results from game theory. In our case, we make use of the open source *shap*[3] library, which assigns positive or negative scores to each word from the input. The last technique that was used in this work is Anchors [13]. In its theoretic version, it is based on if-then rules, with the anchor itself representing a logical predicate. In practice, we will be using the *alibi* library and its *AnchorText*[4] explainer. Its output is actually a list of words from the input, rather than a predicate, and we can interpret these words as the ones that are sufficient to guarantee, with a high precision, the same model prediction, even if the rest of the input is modified. Precision and coverage (recall) are both important notions when talking about anchors, and the library does provide access to them.

As stated, one of the objectives of this work is to expand on the approach of Zytek et al. [19]. In this light, our approach involves carefully designing a series of prompts that guide LLMs to take the outputs generated by the explainability techniques discussed earlier and transform them into coherent, natural language narratives. The objective is for these narratives to be crafted in a way that makes the often complex and technical results of these techniques accessible and easily comprehensible to a broad audience, including those without specialized knowledge in the field. We have used OpenAI's ChatGPT [1], Anthropics' Claude [2] and Meta's LlaMA [17].

METHODOLOGY

In order to assess the performance of the chosen model, we evaluated it on the entire dataset and computed several key evaluation metrics, including accuracy, precision, recall, and F1 score. To obtain these scores, we have compared the predicted sentiment of each review ("positive" or "negative") with their original label. The computed metrics, summarized in Table 1, indicate that our original BERT model achieves sufficient performance, enforcing our argument that additional training may not be necessary. Moreover, this evaluation enables us to investigate the model's potential biases in its decision-making process, ensuring that the results are not influenced by unintended patterns or discrepancies.

Accuracy	0.8896
Precision	0.9154
Recall	0.8586
F1 Score	0.8861

Table 1: Model performance on IMDB dataset

We now proceed to the core experiments of this study. For each input, we begin by utilizing our pre-trained model to generate a prediction. Following this, we apply each of the three explanation techniques described earlier - each independently - to gain insights into the model's decision-making process. The outputs of these explainability techniques are then fed into all three of the LLMs, each accompanied by a carefully crafted prompt. It is important to note that we focus on using zero-shot prompts, meaning that the LLMs are provided with the explainability results as input without any prior fine-tuning specific to the task at hand. While all the five prompts we use (P1-P5, see Table 2) share the common goal of generating a clear, human-friendly narrative that can be easily understood by the general public, each prompt is subtly distinct to influence the style and tone of the produced explanations. The specific variations in the prompts, designed to adjust how the explanation is framed, are detailed in Table 2.

P1 is our starting point, which clearly states the request for the LLM, as well as provides some basic general context. P2 expands the previous prompt, by explicitly mentioning the results should be easily comprehendible by people without any ML experience or knowledge. Additionally, it instructs the LLM to focus on succinct explanations.

P3 applies only to outputs from SHAP or LIME and it explores whether adding specific directives to focus on the most relevant features, based on their scores, changes the explanation produced by the LLM. P4 explores whether specifying the structure of the output of the explainability technique upfront improves the LLM's explanation.

Finally, P5 was introduced after conducting multiple experiments and noticing a recurring pattern in which the LLMs occasionally provided their own reasoning or justifications for the scores assigned to certain words, or even questioned whether these words should have influenced the decisions of the initial model in the manner they did. This observation highlighted the need for a more objective and straightforward explanation. By using this prompt, we aim to obtain an explanation that is focused solely on the factual details of the model's decision-making, without introducing any unnecessary biases.

We showcase an example of an input/output pair for ChatGPT in Figure 1 for which the explainability technique used was SHAP - its output was truncated for display purposes. We note this example uses a different prompt than the ones mentioned in Table 2, as the LLM's explanation references the SHAP values, but we deem this example more easily understandable.

RESULTS

Upon considering the use of Anchors as an explainability technique, it quickly became apparent that conducting all the planned experiments was unnecessary, as will be elaborated in the following paragraph. However, had we included those

[3] https://pypi.org/project/shap/

[4] https://pypi.org/project/alibi/

additional experiments, the total number of experiments would have increased to approximately 500. Without the inclusion of Anchors, we have carried out a total of 480 experiments - 3 (number of LLMs) x 2 (explainability techniques used) x 5 (number of prompts) x 2 (true positive examples) x 2 (true negative examples) x 2 (false positive examples) x 2 (false negative examples).

Our findings are based on observations made through manual inspection of the outputs generated by the explainability technique and the explanations produced by the LLMs. These patterns are not isolated cases but frequent occurrences, consistently appearing across multiple experiments and suggesting a degree of generalizability.

We commence our analysis with the most straight-forward result. Anchors are not a well-suited technique for this type of data. As mentioned, the output is a list of words that are sufficient to guarantee, with a high precision, the same model prediction, even if the rest of the input is modified.

Code	Possible values
[EXP_SHAP]	(SHAP_base_val, [(f_1, f_score_1), (f_2, f_score_2), ...])
[EXP_LIME]	[(feature_1, f_score_1), (feature_2, f_score_2), ...]
[EXP_ANCHORS]	(precision, coverage, [feature_1, feature_2, feature_3, ...])
[EXP_FORMAT]	[EXP_SHAP] / [EXP_LIME] / [EXP_ANCHORS]
[EXP_TECH]	SHAP / LIME / Anchors
Code	**Repetitive Unit**
[INTRO]	You will be explaining the output of the following explainability technique [EXP_TECH] applied on a model that performs sentiment analysis for textual data representing movie reviews from IMDB.
[TASK]	You will receive the input to the model and the output of the explainability technique. Turn these results into a narrative explanation, easy to follow and understand. Do not mention any of the values.
[EXP_RESULT]	This is a placeholder for the results of the explainability technique applied.
Code	**Prompt**
P1	[INTRO][TASK][EXP_RESULT]
P2	[INTRO][TASK] The explanations you will provide will be read by people without machine learning knowledge or experience. Make the explanations as human-friendly and succinct as possible. [EXP_RESULT]
P3	[INTRO][TASK] Add details about the most important features and how they contribute to the model's decision. [EXP_RESULT]
P4	[INTRO][TASK]. The output of the [EXP TECH] is in the format [EXP_FORMAT] and is the following: [EXP_RESULT]
P5	[INTRO][TASK]. Interpret the provided data objectively. Do not give personal reasons as to why the model interpreted a word in a specific way. Do not make any side comments of your opinion of the review. Focus solely on the provided output. Make the explanation as succinct as possible, without discarding the most relevant information. [EXP_RESULT]

Table 2: The give prompts provided to LLM

Input

You will be explaining the output of the following explainability technique, SHAP, applied on a model that performs sentiment analysis for textual data representing movie reviews from IMDB. You will receive the input to the model and the output of the explainability technique. Turn these results into a narrative explanation, easy to follow and understand. Identify the most important features based on their score and add details how they contribute to the model's decision

Model input: "This a fantastic movie of three prisoners who become famous. One of the actors is george clooney and I'm not a fan but this roll is not bad. Another good thing about the movie is the soundtrack (The man of constant sorrow). I recommand this movie to everybody. Greetings Bart"
SHAP output: base value -2.48816133; [('', 0.07088954205391929), ('This ', 0.1145877672242932), ('a ', 0.10569845483405516), ('fantastic ', 0.6042013957048766), …]

Output

What is this about?
The model is predicting whether a movie review is **positive or negative**. Before reading the review, it starts with a **strong bias toward negative** (called the *base value*, which is **-2.49**). Each word in the review then either pushes the prediction **toward positive** or **further negative**.
Top Words That Made the Review Seem Positive
These words increased the model's confidence that the review was positive:
- **"fantastic"** (+0.60): This was the **most powerful positive word**. It told the model the reviewer felt strongly positive.
- **"not bad"** (split across "not" +0.42 and "bad" +0.18): Despite the presence of the word "bad", the model correctly understood "not bad" as praise.

Words That Made the Review Seem More Negative
These pulled the model back toward a negative prediction:
- **"but"** (−0.74): This was the **strongest negative signal**. The word "but" often introduces criticism, and the model picked up on that pattern.
- **"not a fan"** (made of "a" and "fan", each −0.08): This phrase expressed mild disapproval, which the model correctly marked as negative.

Overall Model Decision
While the model began assuming the review would be negative, the **strong positive words—especially "fantastic", "not bad", and "famous"—completely flipped the prediction**. These outweighed the more subtle negatives like "but" and "not a fan". So, the model correctly decided the **review was positive**.

Figure 1: Input/output example for ChatGPT

Because there are no scores, positive weighting and negative weighting words may both be part of the output. What we observed was that generally LLMs added their own interpretation on whether those words had a positive or negative impact. Moreover, in numerous instances the word "not" was included in the anchor's output. While the word "not" is a key negator in language, it is unclear and impossible to know whether its impact was positive or negative, for example in the case of a review containing both "not bad" - positive sentiment and "would not watch again" - negative sentiment. Because of this, one cannot trust that the explanations produced by the LLMs are faithful, as they are free to interpret such usages in whichever way they prefer. Ideally, we want our LLMs to summarize the output, not make their own explanation. Moreover, the anchor output only contains a short list of words, and because in our case we are dealing with movie reviews which are often verbose, the anchor can only capture a small part of the entire review and that could lead to a misleading sentiment prediction. As a result, in the upcoming analysis, we no longer include Anchors.

We continue by focusing on explanations obtained when the model's prediction matches the sentiment of the review. Generally, all three LLMs produced good explanations. All LLMs, unless instructed, included references to the provided values. Moreover, both LIME and SHAP generally provide a score for each word, based on the tokenizer, and while the LLMs correctly identify positive or negative sequences of words due to all words in the sequence having positive or negative values in the explanation output, LLMs struggled to add those values together, instead mentioning only the value of the first word. However, their explanation, where applicable, did mention that certain sequences heavily influenced the model correctly, even if displaying the wrong scores for the sequence. We find this misleading, as users would see some values whose meaning they might not understand, affecting the quality of the explanation. Out of the three LLMs, LLaMA seemed to address this issue the best. Nonetheless, this is why we have explicitly mentioned in our task description that the explanation should not mention any of the provided values.

Moving on to the case where our model's prediction did not match the review's sentiment, we could draw some interesting observations. In some cases, the LLMs were able to detect that the prediction of the model was incorrect and it became challenging to get them to produce an explanation that focused only on the provided output of the explainability technique. Several times, the explanations would contain side comments, seemingly added by the LLM. That was the main issue prompt P5 was trying to address and it did so up to a limit. It was observed using Claude that its explanations are not always faithful. Claude was provided with the output of the explanation technique and because it identified that the model's prediction was wrong, its explanation focused around the values that contributed positively to the correct sentiment. This is extremely misleading and takes away much of the trust in its explanations. After being instructed that the prediction was in fact the opposite, Claude only changed a small part

of its explanation, where it was mentioning the predicted value, with the rest of the explanation, about most relevant features staying the same. To further explore, we took the restriction of not providing reference values from the explanation away. We could see that Claude, although attempting to explain the correct model output, gave stronger corroborating values for the opposite sentiment. We did not observe such issues with either ChatGPT or LLaMA. In Figure 2, we include an example of the final parts of the output produced by Claude when using the same input and prompt, with the only difference being that for the second explanation, we explicitly mentioned what the prediction of the original model was.

The model analyzed a detailed review of the TV show "Oz" and produced a base value of -1.15, which indicates a slightly negative sentiment starting point. However, the overall sentiment prediction was likely positive based on the contribution of key words and phrases throughout the review.

The model analyzed a detailed review of the TV show "Oz" and produced a base value of -1.15, which indicates a slightly negative sentiment starting point. The overall sentiment prediction was negative, despite some positive contributions from certain phrases. This suggests that the negative elements ultimately outweighed the positive ones.

Figure 2: Unfaithful explanations provided by Claude

Another interesting result was observed when looking at apparently incorrect model predictions. Because the reviews are quite verbose, it is very difficult to manually inspect each feature and its score in the output of the explanation technique. Streamlining the whole process of obtaining an initial prediction, using an explainability technique and then LLMs to objectively extract the most relevant information from the output of the explainability technique allowed us to quickly notice whether there are any discrepancies between the input, prediction or explanation. Because of this, we were able to identify what we believe are incorrect labels in the provided IMDB dataset. For illustration purposes, we include a few examples in Table 3.

Review	Original label
Masterpiece. Carrot Top blows the screen away. Never has one movie captured the essence of the human spirit quite like "Chairman of the Board." 10/10... don't miss this instant classic.	negative
Some TV programs continue into embarrassment (my beloved 'X-Files' comes to mind.) I've been a fan of Dennis Farina since 'Crime Story,' another late, lamented show. 'Buddy Faro' never had a chance. The series had a good premise and great actors. It's really, really a shame.	positive

Table 3: Mislabeled data in IMDB dataset

Moreover, doing some experiments on synthetic data provided an insight into our base model. We have observed that negative very short reviews have a much higher tendency to be misclassified by our model as positive. We hypothesised that the presence of even a slightly positive word in a short review is able to skew the model's prediction to positive, thus uncovering an existing bias of our base model. Considering "Would never watch again. Absolutely terrible overall. The ending was decent." as an example, we see that the review is certainly negative, with the only positive part being that the ending was decent - note the usage of decent, which is rather a neutral- positive word in this context. Inspecting the values of the features, we were able to see that the model placed huge emphasis on "decent", assigning it a score higher than the sum of all the scores for negative segments like "never watch again", "absolutely terrible". When allowed to make comments, the LLMs also correctly identified this issue and the abnormal weight placed on the positive word.

Another recurring challenge we encountered was the LLMs' inability to accurately perform floating-point addition. This limitation led to frequent errors in score calculations, with the most significant discrepancies arising in cases involving multiple segments. However, the calculations were typically correct when dealing with individual words, where the associated values were often extracted accurately. In Figure 3, we present an example for a "negative" prediction generated by ChatGPT, which highlights this issue. For negative predictions the smaller the score of a segment, the higher its contribution towards a negative sentiment. In the output, ChatGPT claims that the most influential segment had an impact score of -0.174, but then proceeds to mention another segment with a score that is approximately twice the value of the one it previously identified as the most influential - segment which should have instead been identified as the most influential, as it carried the most negative weight. This discrepancy exposes a notable problem: we specifically instructed our LLMs not to include any numerical scores in their explanations. Consequently, a user reviewing this output would not have been able to identify the underlying mathematical inconsistency. Instead, they would have been forced to rely entirely on the LLM's narrative without any way to verify the accuracy of the figures presented, which could potentially undermine trust in the explanation provided.

Top Negative Contributors:
These words pushed the prediction more strongly toward negative sentiment:

- "privacy is not", "high on the agenda": Combined SHAP impact of approximately -0.174, making this the most influential segment. It indicates that the reviewer pointed out a lack of privacy as a significant downside, which the model interprets as a negative sentiment indicator.
- "not a show for the faint-hearted or timid": These words contribute a total SHAP value of approximately -0.36. Although this phrase may be descriptive, the use of "not," "faint-hearted," and "timid" likely registers with the model as a strong negative emotional tone.

Figure 3: LLMs do not make sound mathematical statements

CONCLUSIONS

In this work, we focused on generating explanations understandable to the general public using LLMs. We explored the capabilities of a pretrained base BERT model specifically trained for sentiment analysis. To guide explanation generation, we designed five distinct prompts that shared a common objective but introduced subtle variations in emphasis. We experimented with different LLMs and explainability techniques to assess how effectively they could produce these explanations.

We found that, on their own, the outputs of explainability techniques are somewhat limited. While undeniably useful - especially for users with technical or ML background - they can be difficult for the general public to interpret. This limitation became even more evident in our study, which focused on movie reviews that often contain substantial text. For longer reviews, even expert users may struggle to extract meaningful insights from raw explainability outputs due to their complexity and length. LLMs, however, are well-suited to handle large volumes of data, and in this work, we leveraged their ability to summarize and distill information into more accessible explanations.

Although the task we focused on - sentiment analysis of IMDB movie reviews - is relatively straightforward and likely manageable by most adults, our work highlights a more nuanced challenge. In the context of NLP, the direct use of explainability technique outputs becomes significantly more difficult when dealing with long textual inputs. Our results show that ChatGPT is particularly well-equipped to handle this complexity and can generate useful explanations, provided it receives appropriately designed instructions. Furthermore, by streamlining this process, we uncovered what appear to be mislabeled examples in the original dataset.

Another key insight from this work is the critical importance of faithful explanations. As we have seen, explanations produced by Claude were not always faithful, in that the LLM would initially detect the original model produced the incorrect prediction and alter its explanation towards the correct prediction. Ideally, an LLM should avoid subjective additions when generating explanations, as this is essential for building user trust in both the model's prediction and its rationale. While introducing an LLM into the explanation pipeline adds another layer of complexity - on top of an already opaque predictive model and its associated explainability technique - it also plays a vital role. We observed that the raw outputs of explainability techniques do not scale well as the number of features (in our case, words) increases, making interpretation increasingly difficult. However, for the LLM to be genuinely helpful, it must remain grounded in the provided explanation output. When an LLM introduces its own reasoning, it effectively bypasses both the original model and the explainability method, reducing the explanation to the LLM's internal logic - another black-box with no applied interpretability.

Finally, regarding the pretrained sentiment analysis model, we identified a potential bias toward short inputs, where the presence of even mildly positive words appears to sway the prediction toward a "positive" label. Despite this, the base model generally performed well on the dataset. Notably, it correctly predicted the sentiment for several instances we believe were mislabeled in the original data.

For future work, several directions can be pursued. First, a user study could be conducted to evaluate the perceived usefulness of the generated explanations, supporting the notion that these explanations make XAI more accessible to the general public and help build trust in black-box models. Second, future efforts could focus on improving the faithfulness of explanations, which remains a critical challenge in our pipeline. Finally, our observation that long reviews are misclassified as "positive" roughly five times more often than as "negative" suggests the presence of additional biases in the base model, warranting further investigation.

ACKNOWLEDGMENTS

This research is partially supported by the project "Romanian Hub for Artificial Intelligence - HRIA", Smart Growth, Digitization and Financial Instruments Program, 2021-2027, MySMIS no. 334906.

REFERENCES

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S. et al. (2023), 'GPT-4 Technical Report', arXiv preprint arXiv:2303.08774 .
2. Anthropic. The Claude 3 Model Family: Opus, Sonnet, Haiku. 2024
3. Bhattacharjee, A., Moraffah, R., Garland, J. and Liu, H. (2023), 'Towards LLM-guided Causal Explainability for Black-box Text Classifiers', arXiv preprint arXiv:2309.13340 .
4. Cortez, P. and Silva, A. M. G. (2008), 'Using Data Mining to Predict Secondary School Student Performance', Proceedings of 5th Annual Future Business Technology Conference, Porto, 5-12.
5. De Cock, D. (2011), 'Ames, Iowa: Alternative to the Boston housing data as an end of semester regression project', Journal of Statistics Education 19(3).
6. Jiang, H. and Senge, E. (2021), 'On Two XAI Cultures: A Case Study of Non-technical Explanations in Deployed AI System', arXiv preprint arXiv:2112.01016.
7. Kroeger, N., Ley, D., Krishna, S., Agarwal, C. and Lakkaraju, H. (2023), 'Are Large Language Models Post Hoc Explainers?', arXiv preprint arXiv:2310.05797, 2023.
8. Lundberg, S. (2017), 'A Unified Approach to Interpreting Model Predictions', arXiv preprint arXiv:1705.07874 .

9. Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y. and Potts, C. (2011), Learning Word Vectors for Sentiment Analysis, in 'Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies', pp. 142–150.
10. Nguyen, V. B., Schl¨otterer, J. and Seifert, C. (2023), From Black Boxes to Conversations: Incorporating XAI in a Conversational Agent, in 'World Conference on Explainable Artificial Intelligence', Springer, pp. 71–96.
11. Nyre-Yu, M., Morris, E. S., Moss, B. C., Smutz, C. and Smith, M. (2021), Considerations for Deploying xAI Tools in the Wild: Lessons Learned from xAI Deployment in a Cybersecurity Operations Setting., Technical Report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).
12. Ribeiro, M. T., Singh, S. and Guestrin, C. (2016), " Why Should I Trust You?" Explaining the Predictions of Any Classifier, in 'Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', pp. 1135–1144.
13. Ribeiro, M. T., Singh, S. and Guestrin, C. (2018), Anchors: High-precision Model-agnostic Explanations, in 'Proceedings of the AAAI Conference on Artificial Intelligence', Vol. 32.
14. Sanh, V., Debut, L., Chaumond, J. and Wolf, T. (2019), 'DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter', arXiv preprint arXiv:1910.01108 .
15. Slack, D., Krishna, S., Lakkaraju, H. and Singh, S. (2023), 'Explaining Machine Learning Models with Interactive Natural Language Conversations using TalkToModel', Nature Machine Intelligence 5(8), 873–883.
16. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y. and Potts, C. (2013), Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, in 'Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing', pp. 1631–1642.
17. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozi`ere, B., Goyal, N., Hambro, E., Azhar, F. et al. (2023), 'LLaMA: Open and Efficient Foundation Language Models', arXiv preprint arXiv:2302.13971 .
18. Wu, X., Zhao, H., Zhu, Y., Shi, Y., Yang, F., Liu, T., Zhai, X., Yao, W., Li, J., Du, M. et al. (2024), 'Usable XAI: 10 Strategies Towards Exploiting Explainability in the LLM Era', arXiv preprint arXiv:2403.08946 .
19. Zytek, A., Pid`o, S. and Veeramachaneni, K. (2024), 'LLMs for XAI: Future Directions for Explaining Explanations', arXiv preprint arXiv:2405.06064.