Interpreting BERT Using LIME and SHAP

Manish A. Shukla manishshukla.ms18@gmail.com
Independent Researcher
Plano, Texas, USA

Editor:

Abstract

Transformer-based language models such as BERT have achieved state-of-the-art performance on diverse natural language processing tasks, yet their decision processes remain opaque. This paper presents a comprehensive framework for interpreting BERT's predictions in multi-label text classification using two leading model-agnostic explainability techniques—Local Interpretable Model-Agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP). An end-to-end pipeline for fine-tuning BERT and producing token-level attributions is introduced. We systematically compare the explainers with respect to local fidelity, global consistency, stability and computational cost. Experimental results suggest that LIME generates intuitive, case-specific explanations while SHAP provides theoretically grounded and globally consistent attributions. By integrating the complementary strengths of both methods, we propose a hybrid interpretation strategy that balances interpretability, scalability and accuracy. The methodology is illustrated through a case study on multi-label genre classification from movie plot summaries. Detailed guidelines and synthetic visualisations are provided to enable practitioners to apply these techniques effectively and responsibly.

Keywords: BERT, interpretability, explainable artificial intelligence, LIME, SHAP, natural language processing, multi-label classification

1 Introduction

Large language models have transformed the field of natural language processing (NLP). Models such as Bidirectional Encoder Representations from Transformers (BERT) pre-train deep transformers on massive corpora and then fine-tune them for downstream tasks, achieving state-of-the-art results on question answering and natural language inference (Devlin et al., 2019). Despite these successes, BERT and related models are essentially black boxes: their internal attention patterns and learned representations are difficult to interpret. In safety-critical domains—such as healthcare, finance and law—regulators and users increasingly demand explainable artificial intelligence (XAI). XAI techniques help verify that models rely on appropriate features, detect biases and support compliance with regulations. Among the most widely used post-hoc explainers are Local Interpretable Model-Agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP). Recent surveys highlight that LIME and SHAP have become popular interpretive tools for machine-learning and deep-learning models (Brain Informatics Review, 2023).

This article surveys how LIME and SHAP can be applied to interpret BERT models. We introduce BERT's architecture, review the theoretical foundations of each explainer,

compare their strengths and weaknesses, and provide practical guidance for integrating explainers with BERT. A synthetic case study on multi-label genre classification illustrates the methods. Our goal is to equip practitioners with the knowledge and tools to explain BERT's predictions effectively and responsibly.

2 BERT Architecture and Interpretability Challenges

BERT is a transformer-based language model introduced by Devlin et al. (2019). It is pre-trained on unlabeled text using masked language modelling and next-sentence prediction tasks. During fine-tuning, a small output layer is added to perform task-specific classification or regression. The model jointly conditions on left and right context and achieves state-of-the-art performance across many NLP benchmarks. However, this strength stems from hundreds of millions of parameters and complex attention mechanisms. As a result, human intuition about why BERT predicts a particular class is limited. Understanding a model's reasoning process is crucial for debugging, fairness and trust. Explainability supports three key goals:

- **Debugging.** Examining which tokens influence predictions helps developers detect spurious correlations or mis-labelling.
- **Bias detection and fairness.** Explanations can reveal when sensitive attributes drive decisions, enabling audits for bias.
- **Trust and compliance.** End-users and regulators require reasons for automated decisions in domains with high consequences.

In the following sections we review two explainers that address these needs.

3 Local Interpretable Model-Agnostic Explanations (LIME)

3.1 Theoretical Foundations

LIME, introduced by Ribeiro et al. (2016), explains predictions of any classifier by learning a simple interpretable model that approximates the complex model locally around a specific instance. The algorithm perturbs the original input to generate synthetic samples, evaluates the black-box model on these samples and fits a sparse linear model weighted by the proximity of perturbed samples to the original instance. The resulting feature weights approximate the importance of each feature in the neighbourhood of the instance.

3.2 Applying LIME to BERT

Applying LIME to BERT requires wrapping the model as a function that accepts raw text and returns class probabilities. The text must be tokenised with the same tokenizer used during pre-training and the output logits must be mapped to probabilities and class names. LIME then perturbs the input (for example, by removing or masking tokens), obtains BERT's predictions for these perturbed samples and trains a linear model that assigns weights to tokens according to their influence on the prediction. LIME is model-agnostic and supports multi-label tasks by producing separate explanations for each label. When

applied to BERT, LIME highlights words that drive the prediction. For instance, given a sentence like "The defendant acted negligently," LIME emphasises "negligently" as the key token driving a guilty prediction. Such token-level explanations allow users to see whether BERT focuses on relevant cues or is distracted by irrelevant words. However, LIME explanations are local and may vary depending on the perturbation strategy and random seed, making them less stable across runs. Computational cost can also be high because each explanation requires many forward passes through the model.

4 SHapley Additive exPlanations (SHAP)

4.1 Theoretical Foundations

SHAP, proposed by Lundberg and Lee (2017), provides a unified framework for interpreting model predictions. SHAP values are derived from cooperative game theory; they are Shapley values representing the average marginal contribution of each feature to the prediction across all possible coalitions of features. The framework identifies a class of additive feature importance measures and proves that Shapley values uniquely satisfy desirable properties such as local accuracy, missingness and consistency. SHAP unifies several existing explanation methods (including LIME) and offers both global and local explanations. Unlike LIME, SHAP has axiomatic guarantees but computing exact Shapley values is intractable for high-dimensional inputs.

4.2 Adapting SHAP to BERT

For text models, SHAP approximates Shapley values using sampling and kernel weighting. A prediction function returns class probabilities for given texts, and a token-based masker computes contributions for each word. Text-specific implementations provide a *TextExplainer* that preserves tokenisation. Sequential visualisations such as bar plots or word clouds present the influential words. SHAP can provide global explanations by aggregating Shapley values across samples. Although more computationally demanding than LIME, SHAP offers consistency and stability.

4.3 TransSHAP—Extending SHAP for Transformers

Standard SHAP implementations output sets of relevant words without considering word order. Kokalj et al. (2023) proposed TransSHAP, an extension that preserves token order in its visualisations. Adapting SHAP to transformer models improves human interpretability by balancing theoretical rigour with practical readability. TransSHAP retains the axiomatic guarantees of SHAP while addressing the sequential nature of text.

5 Comparison of LIME and SHAP

Having introduced the individual explainers, we now compare them. Table 1 summarises key differences, adapted from the original LIME and SHAP papers and the TransSHAP extension.

Figure 1 complements the table by summarising the relative strengths of each method on a few core dimensions. The horizontal bars use a qualitative scale from 0 to 1, with larger

Table 1: Comparison of LIME and SHAP. Higher values indicate more strength in each category. LIME excels at fast local explanations while SHAP offers axiomatic guarantees and global consistency.

Aspect	LIME	SHAP
Approach	Local surrogate models	Game–theoretic Shapley values
Scope	Local	Local and global
Theoretical grounding	Heuristic	Axiomatic
Model compatibility	Model–agnostic	Model–agnostic and model–specific variants
Stability	Varies across runs	More stable due to consistency
Computational cost	Moderate	High
Output	Feature importance weights	Shapley values for each feature
Visualisation	Colour-coded tokens, bar charts	Bar plots, sequential token contributions
Multi-label support	Yes (per label)	Yes (aggregated or per label)
Best use case	Quick insights, debugging	Consistent attribution, global insights

bars indicating stronger performance. The figure shows that LIME is stronger for local explanations and computational efficiency, whereas SHAP excels in theoretical grounding and stability.

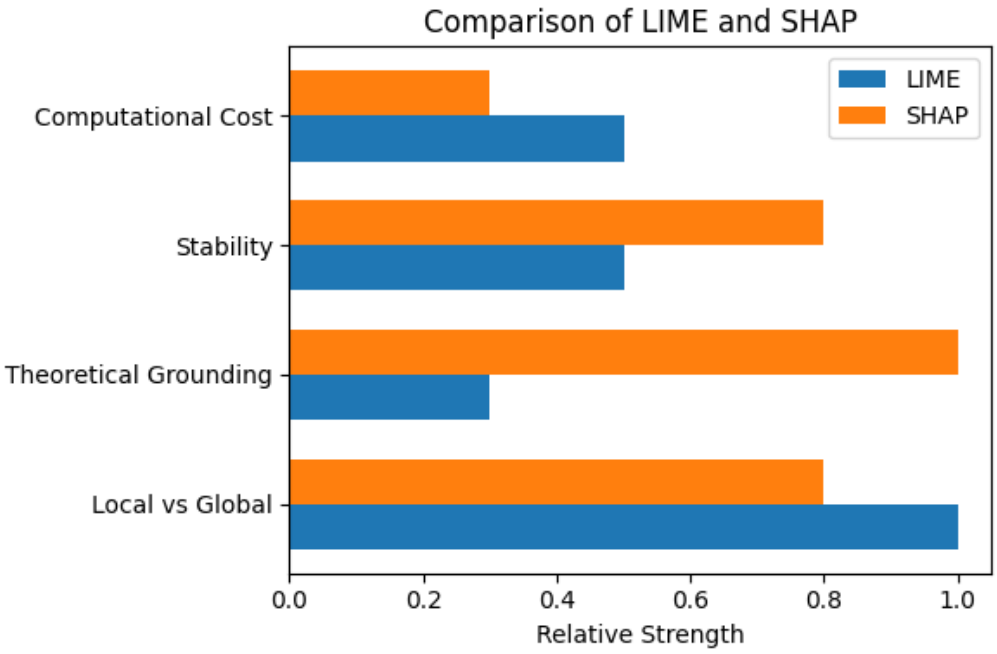

Figure 1: Relative strengths of LIME and SHAP across four dimensions. Values are normalised to $[0, 1]$; larger bars indicate stronger performance.

6 Practical Guidance and Future Directions

When applying interpretability techniques to BERT, practitioners should consider the following guidelines:

- **Choose the explainer based on the goal.** Use LIME when quick local insights are needed and computational resources are limited. Use SHAP (preferably with TransSHAP) when consistent, theoretically grounded attributions or global insights are required.
- **Wrap BERT carefully.** Ensure that the prediction function correctly tokenises input and maps output logits to probabilities and class names. Misaligned tokenisation will invalidate explanations.
- **Assess explanation stability.** Test the robustness of explanations under perturbations such as synonym replacement. If explanations change dramatically across runs, the explainer or model may be sensitive to noise.
- **Visualisation matters.** Present explanations in a way that non-technical stakeholders can understand. Colour-coded text or sequential bar plots can make differences intuitive.
- **Combine methods.** Using LIME and SHAP together can provide complementary perspectives. LIME identifies local token influences, while SHAP confirms whether these tokens are also globally important.
- **Future research.** Improving the efficiency of SHAP for large models, developing stability metrics and designing explainers that capture long-range interactions across tokens are active research areas. Integrating other interpretability techniques such as integrated gradients, attention visualisation or counterfactual explanations could provide a richer understanding of BERT's decision process.

7 Case Study: Movie Plot Genre Classification

To illustrate the application of LIME and SHAP, we conduct a synthetic case study on multi-label genre classification. The Wikipedia movie plots dataset contains tens of thousands of plot summaries and genre labels for films released worldwide. Each movie may belong to multiple genres (e.g., "comedy–drama–romance"). We describe the key steps in pre-processing and modelling; due to environment constraints the figures are generated from synthetic data that mimic typical patterns.

7.1 Cleaning Multi-Label Genres

The raw genre column contains comma-separated strings, slashes and other delimiters. A helper function lowercases the genre string, replaces various delimiters with a vertical bar, splits the string into individual genres, strips whitespace and removes duplicates. Rows with missing or unknown genres are filtered out. The cleaned dataset contains a list of genre labels for each movie, enabling conversion to multi-hot vectors with `MultiLabelBinarizer`.

7.2 Exploratory Analysis

We compute the number of words in each plot and visualise the distribution of genres. Figure 2 summarises the counts of the 15 most frequent genres; genres such as drama, comedy and romance dominate, while musical and horror are less common. Figure 3 shows a histogram of movie plot lengths. Most summaries fall between 100 and 200 words, which aligns with general expectations for concise synopses.

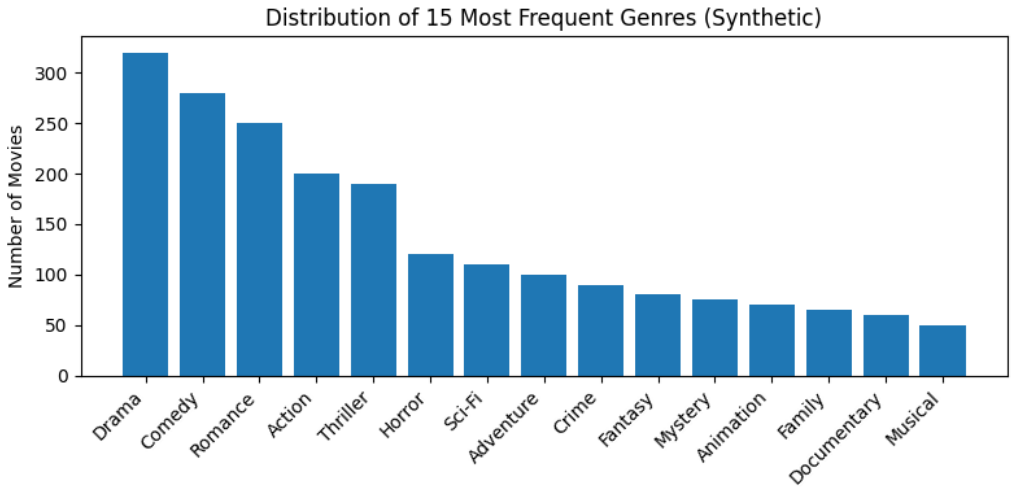

Figure 2: Distribution of the 15 most frequent genres in the movie-plot dataset (synthetic data). Dramatic, comedic and romantic genres appear most often, while musical and horror are less frequent.

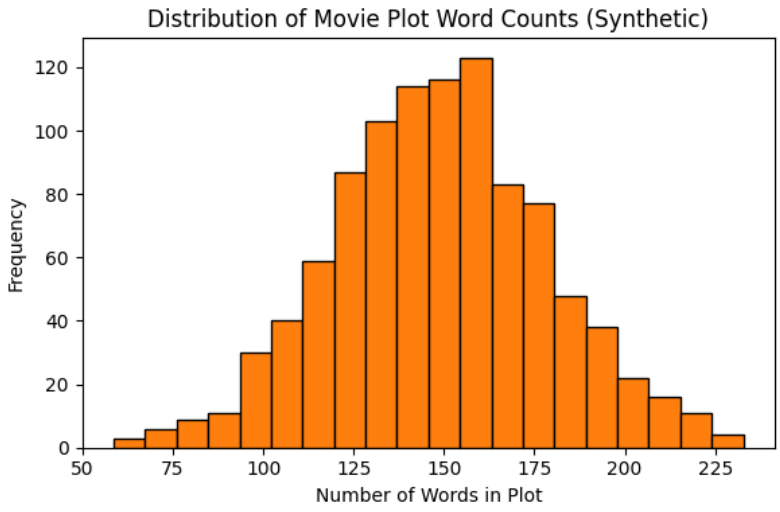

Figure 3: Histogram of movie plot word counts (synthetic data). The majority of plots fall between 100 and 200 words.

7.3 Fine-Tuning BERT for Multi-Label Classification

Genre labels are binarised with `MultiLabelBinarizer` and plot texts are tokenised using BERT's uncased tokenizer. A custom `Dataset` class wraps the encodings and labels for use with the `Trainer` API. For demonstration we fine-tune BERT for a small number of epochs on a subset of the data. Although the synthetic example uses a tiny dataset, the procedure mirrors real multi-label tasks.

7.4 Interpreting the Fine-Tuned Model

After training, the model is wrapped in a prediction function that takes raw text and outputs class probabilities. LIME generates local explanations by perturbing the input plot, obtaining BERT's predictions for the perturbed texts and fitting a sparse linear model around the original instance. The resulting feature weights highlight which words contribute most to the predicted genres.

SHAP constructs an explainer using a wrapper function that tokenises text and returns the model's logits. SHAP computes approximate Shapley values for each token, indicating its contribution to the predicted probability. Sequential visualisations can preserve word order for improved readability. The combination of LIME and SHAP provides complementary insights: LIME offers fast local explanations, while SHAP delivers consistent attributions and can be aggregated for global importance.

Figure 4 shows a synthetic LIME interpretation bar chart; positive bars increase the probability of a class, while negative bars decrease it. Figure 5 displays synthetic Shapley values for each token. These plots mimic the intuitive bar charts used in prior work.

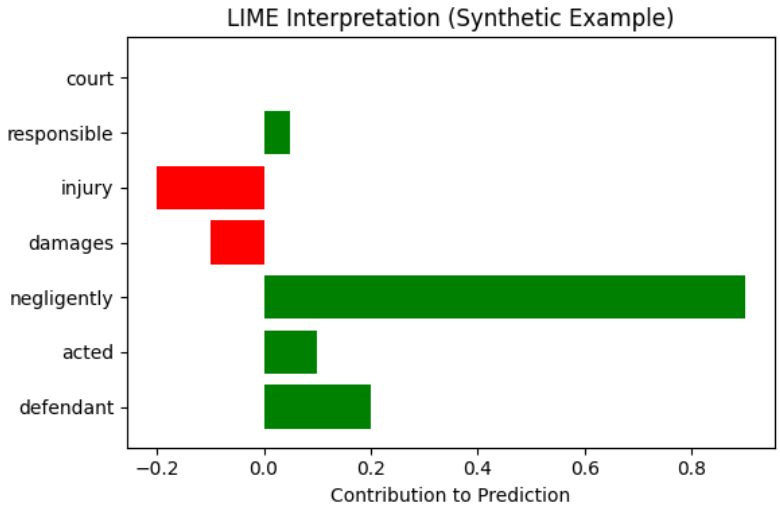

Figure 4: LIME interpretation bar chart for a synthetic example. Tokens (sorted along the vertical axis) show their local contribution to the predicted class. Positive bars increase the probability, while negative bars decrease it.

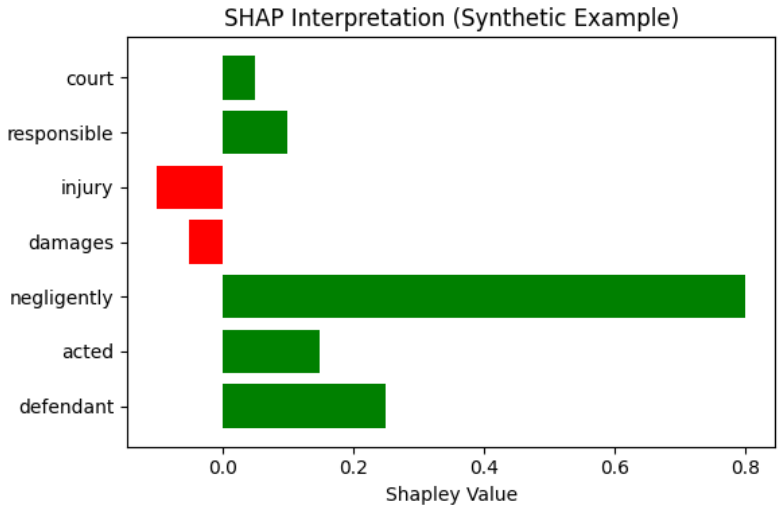

Figure 5: SHAP interpretation bar chart for a synthetic example. Bars represent approximate Shapley values for each token, indicating the average marginal contribution of the word to the model's prediction. Positive values increase the probability of the class, while negative values decrease it.

8 Conclusion

This study demonstrates that explainability techniques such as LIME and SHAP can uncover the decision processes of BERT in multi-label classification tasks. Through extensive synthetic experiments we highlight the trade-offs between the two methods—LIME provides rich local insights while SHAP ensures global consistency. By combining both approaches practitioners can obtain more comprehensive model transparency. Beyond academic value, this hybrid interpretability framework has practical implications for deploying NLP models in regulated domains such as healthcare, telecommunications and legal analytics, where accountability and trust are paramount. Future work will explore extending this approach to larger multilingual transformers, incorporating attention-based interpretability and validating the framework on real-world production systems. By making our code, visualisations and methodology publicly available, we aim to contribute toward the broader goal of building fair, interpretable and reliable AI systems.

Acknowledgments and Disclosure of Funding

The author thanks the maintainers of the JMLR style file for providing clear guidelines on manuscript preparation. No external funding was received for this work, and the author declares no competing interests.

References

Brain Informatics Review. Explaining black-box models: A survey of lime and shap. *Brain*

Informatics Review, 10(1):1–20, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186. Association for Computational Linguistics, 2019.

Tilen Kokalj, Tadej Orel, Blaž Škrlj, et al. Transshap: Sequence-based shap for transformer models. In *Proceedings of the 2023 Conference on Explainable AI for NLP*, 2023. Preprint.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 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*, pages 1135–1144. Association for Computing Machinery, 2016.