Noname manuscript No.
(will be inserted by the editor)

Explainability of Transformer Models in Sentiment Analysis and Comparison of the interpretability tools

Fahim Ishrak (Corresponding author) ·
Amir Jafari Ph.D.

Received: date / Accepted: date

Abstract With so many popular Deep learning models being developed and solving complex issues, they still, however, remain as black boxes. As a result, there is a untrustworthy notion towards them because of the lack of transparency. Many tools were created to work out the interpretability of such models however, there are few such examples in the domain of Natural Language Processing, more specifically, Transformer models. This experimental project aims to look at the interpretability of Transformer models using two recent tools (LIME and Captum) against a baseline CNN model on a Binary classification dataset. Finally the tools are compared to look at the different interpretability results.

Keywords Natural Language Processing · Transformer Models · BERT · LIME · Integrated Gradients

1 Introduction

As the need for solutions to complex problem in computer vision and natural language processing continues to increase, so does the development of complex models to solve such problem. When dealing with simple models, like logistic regression, it is possible to work out exactly how the inputs are mapped to the outputs. However, as models become more complex with millions of trainable parameters, they become blackboxes, providing almost little to no explanations for their predictions with the human-readable explanations remaining absent. As a result, we can only see the input and the output and trust that the model

Fahim Ishrak
Tel.: (312)-438-6960
E-mail: fahim_ishrak@gwu.edu

Amir Jafari Ph.D.
Tel.: (202)-994-1239
E-mail: ajafari@gwu.edu

is learning correctly. Tools were created to tackle these interpretability problems, however there is little such example in natural language processing and newly transformer models. Applying modern NLP for real-world applications requires interpretability and to make the system more robust and transperent. So, using established explainability tools such as LIME and captum in various sentiment analysis datasets, we hope to bring some transperency into the NLP models with a degree of certainty and finally compare the two tools used for interpretability.

2 Literature Review

The works on BERT [1], LIME [6] and Captum (Integrated Gradients) [3][8] gives the necessary background on the tools and models required to work with this project.

The work with sentiment analysis [4] looks at comparing two models and the tradeoff between them in terms of computation and accuracy of the explainability. It didn't dive deep into the datasets or the interpretability surrounding the dataset. The authors also built their own generative framework for interpretability and didn't use any tools. The work on Explaining Sentiment in the field of medicine [5] Looks into the need for explainable systems, compares different explainable systems, and brings up the need for a better explainable system. This paper also does not deal with any datasets or any specific tools used.

3 Background

Before diving into the task, some high level understanding and intuition is presented in this section to show how the models and the tools work.

3.1 CNN

Usually, Convolutional Neural Networks (CNN) is used for analyzing visual images. The basic architecture of CNNs is an alternating order of Convolution layers and pooling layers. Afterward, the feature matrix is flattened out into a vector and then followed by a Multi-Layer Perceptron (MLP) with the number of classes in the output layer. Similar to how images can be represented as pixel values, text data can be represented as word vectors which can be processed using a CNN. When working with text data, one-dimensional convolutions are used, however, the process remains the same

The first task is to Vectorize a given corpus into numbers. Each value then maps to a value in a dictionary that encodes the entire corpus. Each text also has a different length of words so padding with 0 is used to make sure every text has the same length.

Finally, GloVe (Global Vectors for Word Representation) is used to get the word embeddings and an embedding matrix is created for the CNN.

3.2 Bert

BERT [1] is a multi-layer bidirectional Self-attention encoder. It is based on the implementation described in Vaswani et al. (2017) [9]. It gives an elaborate background of the model as well as looks at excellent guides like "The Annotated Transformer." [7]. To give a high-level overview of BERT, it is a series of encoder layers stacked on top of each other. The output layer of the model is modified according to specific tasks. For example, in the case of sentiment analysis, a feed-forward network with a softmax layer is used in the output layer.

An encoder layer is a series of attention layers and a feed-forward layer. In the attention layer, self-attention works by looking at other positions in the input sequence for hints which can help lead to a better encoding for the word. The feed-forward layer is a stack of Recurrent Neural Networks.

The model takes as inputs the input IDs, token type IDs, and position IDs. These are generated using the Bert tokenizer [10]. Input IDs are sequence tokens in the vocabulary. Token type IDs are segmented token indices to show the different portions of the inputs. Position IDs are Indices of positions of each input sequence tokens in the position embeddings.

The BERT model used in this project is BERTBASE (Number of Layers=12, Hidden Size=768, self attention, head=12, Total Parameters = 110M)

The model was pre-trained using a combination of the BooksCorpus [2]and English Wikipedia which has a combination of 3000m words. This pre-trained model is then trained on our custom dataset by only training the output layer of the model.

3.3 Lime

LIME (Local Interpretable Model-agnostic Explanations) [6] is an explanation technique that explains the prediction of any classifier by learning an interpretable model locally around the prediction. Locally, interpretable means looking at why did the model make a specific prediction and What effect did this specific feature value have on the prediction.

The output of LIME reflects the contribution of each feature to the predicted data. This provides local interpretability, thus showing us which changes in the feature will have the most impact on the prediction.

An explanation is created by estimating the underlying model locally by an interpretable one. Interpretable models are e.g., linear models, decision tree's, etc. The interpretable models are trained on small sections of the original data and provide a local approximation. The section is augmented by e.g., adding noise, removing words or hiding parts. By only approximating the black-box locally (in the neighborhood of the data sample) the task is significantly simplified.

3.4 Integrated Gradients

Given a deep network, an input, and a baseline input. For image networks, the baseline could be a black image, while for text models it could be a zero embedding vector. Integrated gradients are obtained by summing up the gradients along The straightline path from the baseline x' to the input x. It is defined as the intergral of the gradients along the straightline path from the baseline x' to the input x.

The integrated gradient along the i^th dimension for an input x and baseline x' is defined as follows. Here, $\partial F(x) \partial x_i$ is the gradient of F(x) along the ith dimension.

IntegratedGrads $_i(x) ::= (x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial F\left(x'+\alpha\times\left(x-x'\right)\right)}{\partial x_i} d\alpha$

The gradient informs which embedding vector has the strongest effect on the models' predicted class probabilities. Varying variable changes the output, and the variable will receive some attribution to help calculate the feature importances for the input image. A variable that does not affect the output gets no attribution.

4 Methodology

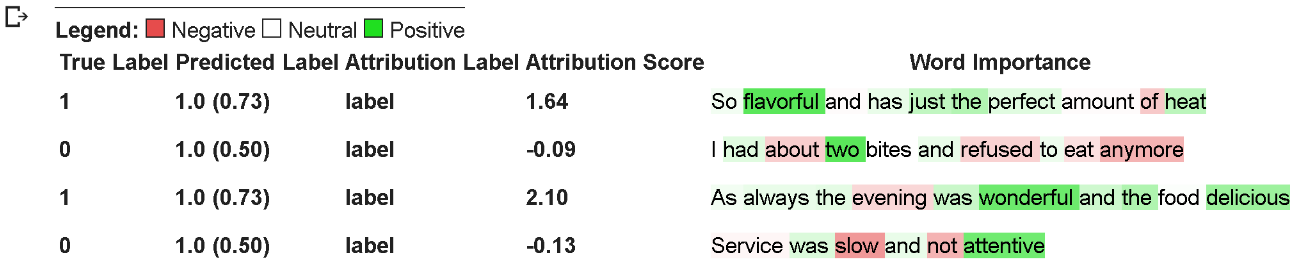
The dataset picked for this task is the restaurant reviews dataset and the Stanford Treebank dataset. Both of them are binary classification datasets. The dataset structure is simple, containing only the texts and corresponding sentiment. For preprocessing, it is made sure that the classes are not imbalanced and this is resolved by under-sampling the majority class. Afterward, the dataset is trained on a pre-trained BERT model and a CNN model. The CNN model is a baseline against which the BERT model will be compared. The BERT model is trained by using a tokenizer specific to BERT. The tokenizer encodes the text into input ids, attention masks, and position ids. These then act as an input to the BERT model which then outputs a value within one and zero. The loss - sum of the masked language modeling loss and the next sequence prediction loss- is then calculated and weights are updated. The model is then ready to be shipped to LIME and captum to see how it predicts on specific sentences from the dataset.

5 Results

For the integrated gradients results, the labels are given. For the LIME results, blue means towards positive and orange mean towards negative

5.1 Restaurant reviews

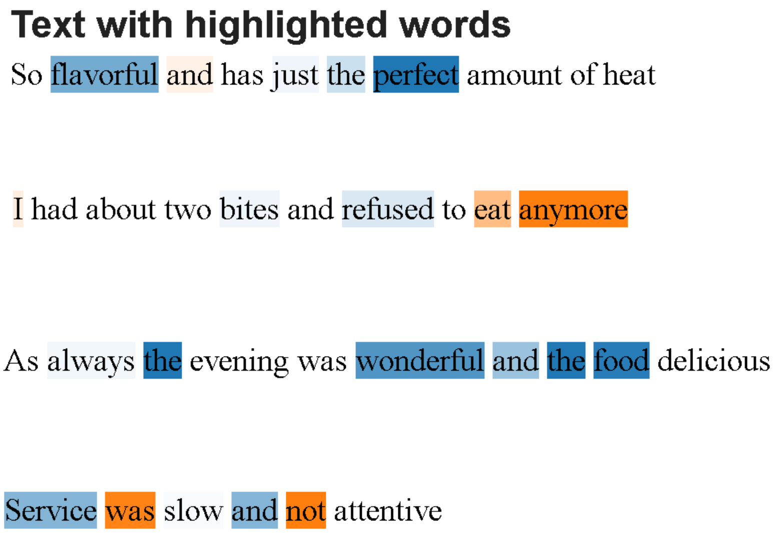
5.1.1 CNN



Legend: 🟥 Negative □ Neutral 🟩 Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
1	1.0 (0.73)	label	1.64	So flavorful and has just the perfect amount of heat
0	1.0 (0.50)	label	-0.09	I had about two bites and refused to eat anymore
1	1.0 (0.73)	label	2.10	As always the evening was wonderful and the food delicious
0	1.0 (0.50)	label	-0.13	Service was slow and not attentive

Fig. 1 Result of integrated gradients on CNN



Text with highlighted words

So flavorful and has just the perfect amount of heat

I had about two bites and refused to eat anymore

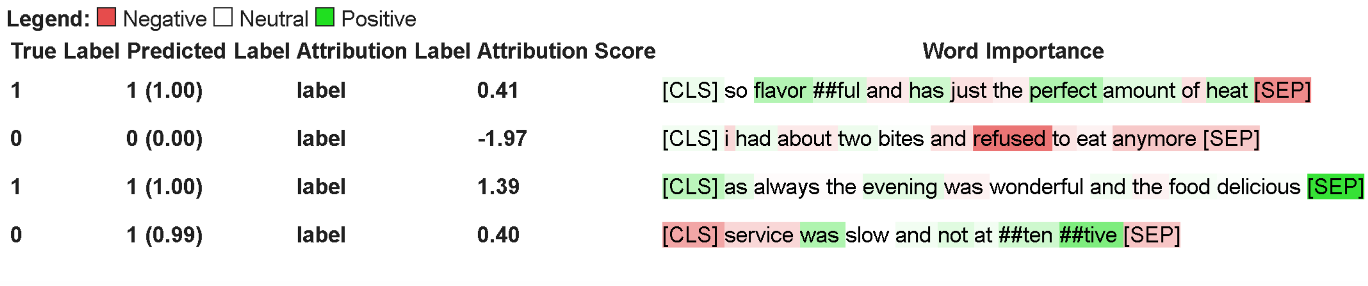
As always the evening was wonderful and the food delicious

Service was slow and not attentive

Fig. 2 Result of Lime on CNN

As it can be seen for CNN, the highlights from the integrated gradients make sense and is consistent throughout, some of the results from LIME matches with that of the integrated gradients but LIME overall is bit more inconsistent.

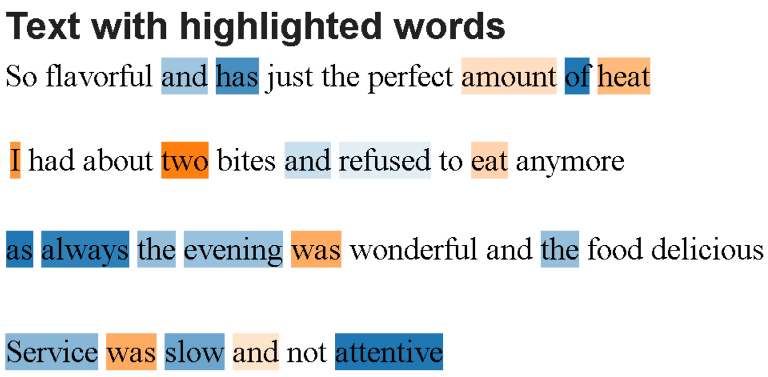
5.1.2 BERT



Legend: 🟥 Negative □ Neutral 🟩 Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
1	1 (1.00)	label	0.41	[CLS] so flavor ##ful and has just the perfect amount of heat [SEP]
0	0 (0.00)	label	-1.97	[CLS] i had about two bites and refused to eat anymore [SEP]
1	1 (1.00)	label	1.39	[CLS] as always the evening was wonderful and the food delicious [SEP]
0	1 (0.99)	label	0.40	[CLS] service was slow and not at ##ten ##tive [SEP]

Fig. 3 Result of integrated gradients on BERT



Text with highlighted words

So flavorful and has just the perfect amount of heat

I had about two bites and refused to eat anymore

as always the evening was wonderful and the food delicious

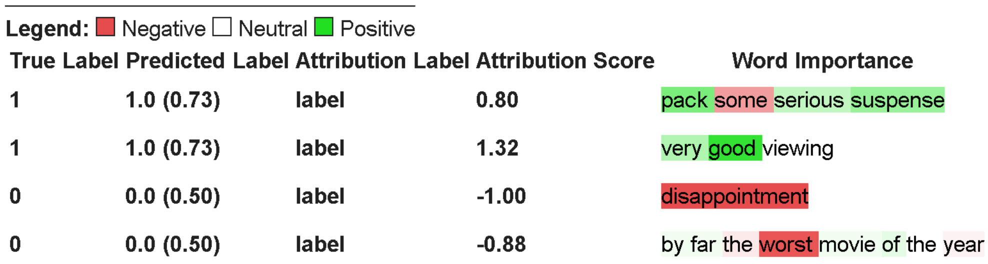
Service was slow and not attentive

Fig. 4 Result of Lime on BERT

For BERT, the results for integrated gradients also make sense but it seems less ’confident’ given by the intensity of the colors. For LIME, the results are a bit inconsistent as words like ’slow’ and ’refused’ were given a positive connotation.

5.2 Stanford Tree Bank

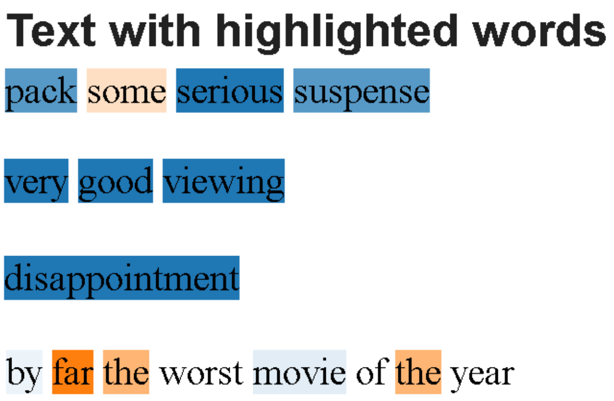
5.2.1 CNN



Legend: 🟥 Negative □ Neutral 🟩 Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
1	1.0 (0.73)	label	0.80	pack some serious suspense
1	1.0 (0.73)	label	1.32	very good viewing
0	0.0 (0.50)	label	-1.00	disappointment
0	0.0 (0.50)	label	-0.88	by far the worst movie of the year

Fig. 5 Result of integrated gradients on CNN



Text with highlighted words

pack some serious suspense

very good viewing

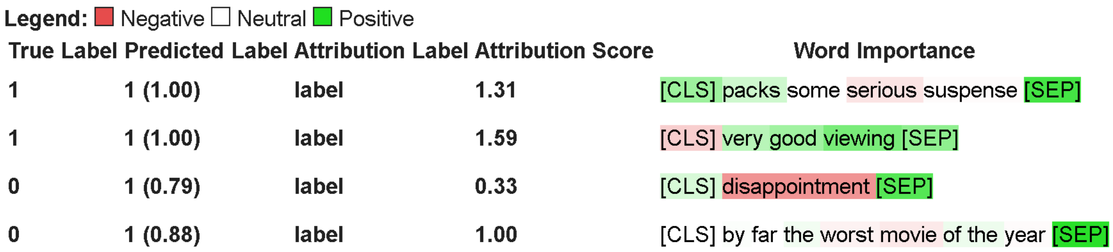
disappointment

by far the worst movie of the year

Fig. 6 Result of Lime on CNN

For the most part, the results of integrated makes sense and matches with LIME except for the word 'disappointment' which LIME gives a very positive connotation and giving the word 'worst' a neutral connotation

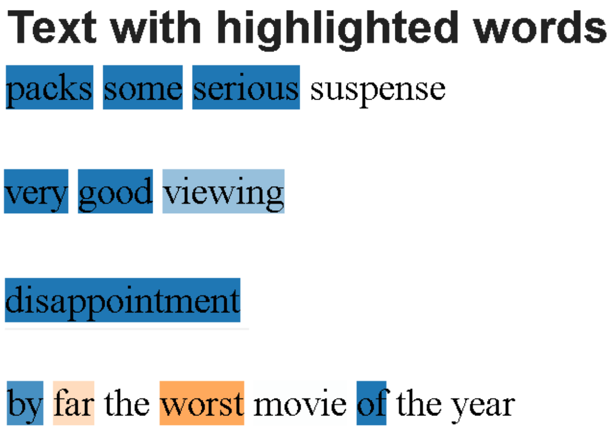
5.2.2 BERT



Legend: 🟥 Negative □ Neutral 🟩 Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
1	1 (1.00)	label	1.31	[CLS] packs some serious suspense [SEP]
1	1 (1.00)	label	1.59	[CLS] very good viewing [SEP]
0	1 (0.79)	label	0.33	[CLS] disappointment [SEP]
0	1 (0.88)	label	1.00	[CLS] by far the worst movie of the year [SEP]

Fig. 7 Result of integrated gradients on BERT



Text with highlighted words

packs some serious suspense

very good viewing

disappointment

by far the worst movie of the year

Fig. 8 Result of Lime on BERT

Once again, the results of integrated gradients with BERT makes sense and matches with LIME except for, once again, the word 'disappointment' which LIME gives a very positive connotation

6 Conclusion and Future Work

From the results, it can be concluded that integrated gradients give more reliable and consistent interpretations when compared to LIME. When comparing the transformer model, BERT, to the baseline CNN model, BERT gives similar results but it is less 'confident' and a bit inconsistent when interpreting a sentence.

More conclusive results can be provided if the models are trained on more, larger datasets. At the moment, integrated gradients doesn't handle multiclass sentiments but it would be interesting to look at how the models would interpret more than two sentiments. Finally, more transformer model can be added to the comparison as well as using more tools for comparison like SHAP and see how they are interpreting the results.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
2. Sosuke Kobayashi. Homemade bookcorpus. https://github.com/BIGBALLON/cifar-10-cnn, 2018.
3. Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*, 2020.
4. Hui Liu, Qingyu Yin, and William Yang Wang. Towards explainable nlp: A generative explanation framework for text classification. *arXiv preprint arXiv:1811.00196*, 2018.
5. Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, 2019.
6. 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, 2016.
7. Alexander M Rush. The annotated transformer. In *Proceedings of workshop for NLP open source software (NLP-OSS)*, pages 52–60, 2018.
8. Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*, 2017.
9. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
10. Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.