Data Mining and Knowledge Discovery (2024) 38:128–153
https://doi.org/10.1007/s10618-023-00962-4

An attention matrix for every decision: faithfulness-based arbitration among multiple attention-based interpretations of transformers in text classification

Nikolaos Mylonas[1] · **Ioannis Mollas**[1] · **Grigorios Tsoumakas**[1]

Received: 27 November 2022 / Accepted: 12 July 2023 / Published online: 28 August 2023

Abstract
Transformers are widely used in natural language processing, where they consistently achieve state-of-the-art performance. This is mainly due to their attention-based architecture, which allows them to model rich linguistic relations between (sub)words. However, transformers are difficult to interpret. Being able to provide reasoning for its decisions is an important property for a model in domains where human lives are affected. With transformers finding wide use in such fields, the need for interpretability techniques tailored to them arises. We propose a new technique that selects the most faithful attention-based interpretation among the several ones that can be obtained by combining different head, layer and matrix operations. In addition, two variations are introduced towards (i) reducing the computational complexity, thus being faster and friendlier to the environment, and (ii) enhancing the performance in multi-label data. We further propose a new faithfulness metric that is more suitable for transformer models and exhibits high correlation with the area under the precision-recall curve based on ground truth rationales. We validate the utility of our contributions with a series of quantitative and qualitative experiments on seven datasets.

Keywords Interpretable machine learning · Transformers · Attention · Text classification · Multi-label learning

Responsible editor: Charalampos Tsourakakis.

✉ Nikolaos Mylonas
myloniko@csd.auth.gr

Ioannis Mollas
iamollas@csd.auth.gr

Grigorios Tsoumakas
greg@csd.auth.gr

[1] School of Informatics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece

1 Introduction

Transformers have become the dominant approach for tackling Natural Language Processing (NLP) tasks, surpassing previous convolutional and recurrent neural network architectures (Vaswani et al. 2017; Wolf et al. 2020). Transformers are considered black boxes. The large number of their parameters and their complex architecture makes it difficult to understand how they reach their decisions (Schwenke and Atzmueller 2021). Interpretability is important in high-risk applications, where decision-making systems can have a significant impact on human lives (EU 2021). The importance of interpretability in such applications, along with the high performance that transformers can achieve in them, raises the need for techniques that can explain the decisions of these models.

The most popular transformer-specific interpretability approach is the use of self-attention scores. Since these scores are computed during inference, obtaining interpretations from them adds no computational overhead. However, the use of attention to produce explanations has been met with skepticism by some researchers (Jain and Wallace 2019). Other transformer-specific interpretability approaches combine attention with gradient information (Chefer et al. 2021) or compute new attentions based on the network's residual connections (Abnar and Zuidema 2020). Nevertheless, such techniques introduce new elements in the model's architecture, necessitating the model's pre-training from scratch.

While reviewing the literature on attention-based interpretation of transformers, we ran across different ways of integrating attention information across heads and layers, as well as different ways of extracting an interpretation vector from a final attention matrix. This motivated us to study whether some particular ways perform better than the rest. The inconclusiveness of this study led us to propose a novel family of local interpretation techniques for transformers, dubbed Optimus. Given an input instance, Optimus Prime evaluates the faithfulness of the interpretations obtained by a number of different combinations of head, layer and matrix operations and selects the best one as the final interpretation. In multi-label learning tasks, Optimus Label selects the most faithful combination separately for each predicted label, leading to improved results. To reduce the computational complexity, at the cost of performance, Optimus Batch selects the most faithful combination across an initial set of instances, and then uses this fixed setup for subsequent instances. In addition, we propose a new faithfulness evaluation metric, *Ranked Faithful Truthfulness*, that correlates highly with the area under the precision recall curve computed on top of ground truth rationales.

Our contributions are empirically assessed on seven datasets from four different domains: sentiment analysis, natural language understanding, hate speech detection, and biomedicine. Decision-making in the latter two can in some cases have significant impact on human lives. The first one by affecting freedom of speech or allowing the incitement of violence, and the second one by recommending inappropriate treatments. Results show that attention-based interpretations can compete with state-of-the-art techniques, and even exceed them in certain cases, while being less computationally expensive in others.

The remainder of this article is organized as follows. Section 2 examines related work on transformer interpretability and interpretability evaluation. Section 3 presents our attention-based interpretability technique alongside our new faithfulness metric. Section 4 introduces the setup of our experiments and presents quantitative and qualitative evaluation results. Finally, Sect. 5 discusses the conclusions of our work and points to future research directions.

2 Related work

A model's ability to provide insights for its decisions or inner working, whether intrinsically or not, is referred to as interpretability. Complex models, such as transformers, cannot provide interpretations out of the box, and therefore post-hoc techniques are typically applied. The representations of an interpretation include, among others, rules, heatmaps, and feature importance. This work focuses on feature importance, also known as attribution importance or saliency map, which quantifies the influence of a model's input features on its output. The rest of this section presents related work on transformer interpretability, mainly in the context of text classification, as well as on interpretability evaluation methods.

2.1 Transformer interpretability

We first review model agnostic and neural-specific feature importance techniques that are applicable to transformers. Then, we present interpretability techniques that have been designed specifically for transformers. Finally, we discuss two studies that have discovered interesting patterns and properties of transformers' attention module.

2.1.1 Transformer-applicable techniques

Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al. 2016) and SHapley Additive exPlanations (SHAP) (Lundberg and Lee 2017), two model-agnostic, local interpretation approaches, can be easily applied to a transformer by just probing it for predictions. Backpropagation-based neural-specific techniques such as Layer-wise Relevance Propagation (LRP) (Bach et al. 2015) and Integrated Gradients (IG) (Sundararajan et al. 2017) can be modified to provide interpretations for transformer models. Such techniques that consider model architecture and employ back-propagated gradients are expected to yield more meaningful interpretations. However, some studies have shown that model-agnostic methods achieve competitive performance in transformer explainability (Mathew et al. 2021).

Making use of techniques such as LIME, IG, and SHAP, Thermostat (Feldhus et al. 2021) provides a collection of ready-to-use interpretations in the form of feature importance scores for different transformer models and datasets. This can reduce the environmental impact and economic barriers associated with the repetitive execution of common experiments in interpretable NLP. Ecco (Alammar 2021) is an open-source library offering a variety of techniques for analyzing the inner workings of a transformer, such as how the model's hidden states change from layer to layer, providing feature importance interpretations, as well as enabling the examination of activation vectors.

2.1.2 Transformer-specific techniques

Extracting information from the attention module of transformers has been a popular method for interpreting their decisions (Mullenbach et al. 2018; Wiegreffe and Pinter 2019), especially before its criticism (Jain and Wallace 2019; Bastings and Filippova 2020). Recent work introduced an interpretation technique based on reinforcement learning that uses attention matrices in order to build a perturbation-based game environment that provides explanations for transformer models (Niu et al. 2022). An explainability method based on hierarchical transformer models was proposed in Bacco et al. (2021). Two transformer-based model architectures were introduced to classify and extract explanations for sentiment analysis. Explanations were extracted based on attention weights and compared to ones provided by human users.

A recent method that does not solely rely on raw attention to provide explanations, is combining relevance and gradient information (Chefer et al. 2021). Specifically, relevance scores are produced for each attention head in each layer, leveraging the theory underpinning LRP. These results are then integrated with gradient information. The produced explanation is a matrix of size $S \times S$, where S denotes sequence length. The final relevance map is derived from the row of the matrix corresponding to the [CLS] token, which is the special classification token that is added by certain transformers at the start of each sequence.

A process aimed at quantifying how attention information flows from layer to layer is introduced in Abnar and Zuidema (2020). Specifically, two methods based on Directed Acyclic Graphs are proposed, *Attention Rollout* and *Attention Flow*, that compute attentions for each input token. Both methods take into account the models' residual connections to obtain token attentions. These attentions were found to retain more information and can serve as a visualization tool. A tool developed specifically for attention visualization, is BertViz Vig (2019), which provides insight about how tokens of a particular sentence affect each other, while also shedding light on what each attention head and layer focuses on.

The main advantage of Optimus compared to the discussed approaches is that the interpretations provided are solely based on attention allowing for fast response times. In addition, Optimus can be used in different tasks with minimal adjustments, as attention remains the same regardless of the downstream task of the transformer.

2.1.3 Attention analysis in transformers

Five distinct patterns of self-attention that are used across attention heads were discovered in Kovaleva et al. (2019) by displaying attention score heatmaps for BERT. Additionally, it was shown that by disabling certain attention heads or layers, the model does not necessarily display a decrease in performance and can even exhibit improvements in specific cases.

The topic of transformer identifiability was explored in Brunner et al. (2020). Attention weights are defined as identifiable if they can be uniquely determined from the transformer's output. It was discovered that if the sequence length used is higher than the attention head dimension, then attention weights are not identifiable. This is because certain rows of the attention matrix can be linear combinations of others. Using attention as an explanation may be unwarranted, since different combinations of weights may produce the same output.

2.2 Interpretability evaluation

The most appropriate way to evaluate an interpretability technique is via a user study, where end users compare interpretations (Lertvittayakumjorn and Toni 2019). However, this kind of experimental procedure is not always feasible, due to its costly and time-consuming nature. Furthermore, human evaluation is prone to bias (Herman 2017).

Ground truth interpretations provided by human annotators are called *rationales* (DeYoung et al. 2020). In text classification, these rationales can be words, sentences or spans of text that strongly associate each instance with its label. When rationales are available, we can evaluate interpretations using standard metrics, such as F_1, or the Area Under the Precision-Recall Curve (AUPRC). Unfortunately, datasets accompanied by rationales are scarce. Moreover, since rationales are provided by humans, erroneous, noisy, and biased annotations may occur, as in human evaluation.

There are also metrics that can do without human input, by evaluating certain properties of the produced interpretations. *Robustness* (Melis and Jaakkola 2018) concerns the stability of a technique. By slightly modifying the examined instances, robustness measures the degree of change between the interpretations for the initial and modified instances. The smaller this change is, the higher the robustness of the technique. *Comprehensibility* (Robnik-Sikonja and Bohanec 2018) calculates the percentage of non-zero weights in an interpretation. The lower this number, the easier for end users to comprehend the interpretation.

A frequently used family of metrics are those emulating the behavior of a user that interacts with the model to explore the validity of a given interpretation, known as faithfulness evaluation metrics. *Faithfulness score* (Du et al. 2019), the most popular one, eliminates the token with the highest importance score from the examined instance and measures how much the prediction changes. Higher changes signify better interpretations. *Truthfulness* (Mollas et al. 2022) removes all tokens of an instance, one at a time, and awards or penalizes the technique based on the model's behavior. Other metrics of this family include Comprehensiveness, Sufficiency, Monotonicity and Faithfulness Violation Test (Chan et al. 2022; Liu et al. 2022).

3 Our approach

Given a transformer model f, and an input sequence $x = [t_1, \dots, t_S]$, consisting of S tokens $t_i, i = 1 \dots S$, our goal is to extract a local interpretation $z = [w_1, \dots, w_S]$, where $w_i \in \mathbb{R}$ signifies the influence of token t_i on the model $f(x)$, based on the model's self-attention scores. We first present the Optimus family of techniques for selecting the most faithful interpretation among several different ones. Then, we discuss a novel faithfulness metric that exhibits high correlation with AUPRC computed on top of ground truth rationales.

3.1 The Optimus family of techniques

We first review the self-attention layer of transformers, where attention scores are computed, and introduce a variation for obtaining feature importance scores that include negative values. Then, we present the different ways that are commonly used to turn attention scores into an interpretation in the form of feature importance, as well as introduce some new ones. Finally, we present three techniques for selecting the most faithful among these interpretations.

3.1.1 Attention scores

The input to each self-attention layer is a matrix of dimensions $S \times E$, where S denotes sequence length, and E refers to embedding size. At first, this matrix is passed through three linear layers, namely *Query*, *Key*, and *Value*, to produce matrices Q, K, V of the same dimension as the input. Next, the dot product of Q and K is calculated, and divided by the square root of the embedding size. Subsequently, the attention mask is added. The product of those operations is a matrix of dimensions $S \times S$ which contains both negative and positive values. This matrix is then passed through a softmax function, which outputs a matrix, A, of the same dimensions containing only positive values which correspond to the attention from each token of the sequence to the rest:

$$A = softmax\left(\frac{Q \cdot K^T}{\sqrt{E}} + mask\right) \tag{1}$$

In fact, transformers employ a *multi-head attention* architecture, where the input to each self-attention layer is logically split to R attention heads. Each head operates on a different part of the input matrix of size $S \times \frac{E}{R}$ allowing the transformer to learn different relationships between tokens.

Due to the use of the softmax function, which maps all input values to the range [0, 1], attention matrices contain only positive numbers. Consequently, any interpretations extracted from these matrices will contain only positive values. However, interpretations containing both positive and negative values are often desirable (Liu et al. 2021), as the presence of polarity within feature importance scores can facilitate a better association of input elements to a decision. To illustrate this, consider a situation where a model is uncertain about a decision and predicts a label with low probability. By highlighting the input elements that have a negative influence on the decision, users can comprehend the reasons behind the uncertainty and adjust or modify those elements to increase the certainty of the decision. Therefore, our experiments consider a modification on Eq. (1), which ignores the softmax function to allow negative values to appear in the resulting interpretations (Eq. 2). We denote the corresponding matrix by A^*:

$$A^* = \frac{Q \cdot K^T}{\sqrt{E}} + mask \tag{2}$$

3.1.2 Interpretation extraction

An interpretation is typically obtained by first aggregating the attention matrices across all heads of each self-attention layer, then aggregating the resulting matrices across all self-attention layers, and finally extracting from the resulting matrix the one-dimensional interpretation vector. Head operations commonly found in the literature are averaging (Chefer et al. 2021; Mathew et al. 2021; Wang et al. 2019) and summing (Hoover et al. 2020; Schwenke and Atzmueller 2021) the attention matrices of each head. These operations are equivalent in the context of interpretability evaluation, as they lead to the same ordering of the tokens by importance, differing only in the magnitude of the scores assigned to the tokens. Therefore, the summing operation is not included in our pipeline. Operations concerning the resulting matrices of self-attention layers include averaging (Schwenke and Atzmueller 2021) and multiplying (Chefer et al. 2021). We therefore also consider multiplying as an operation for heads.

We further propose an additional operation for both heads and layers: selection of the attention matrix corresponding to a certain head/layer. As different heads are learning different relationships among the input tokens, including the [CLS] token in classification tasks, they are essentially learning different ways that input tokens influence the class. We therefore hypothesize that this new selection operation can be crucial for obtaining local interpretations tailored to a particular input sequence.

These head and layer operations lead to a single matrix, integrating the attention scores from the different heads of the different layers in a model. To obtain the final interpretation vector, a common approach is to consider the attention that each input token receives from the special [CLS] token that is prepended at the beginning of sequences in text classification tasks (Chefer et al. 2021; Mathew et al. 2021). We call this operation “From [CLS]”. Assuming that each row of the attention matrix corresponds to the attention a token pays towards the others, while each column corresponds to the attention it receives from the others, this operation amounts to extracting the [CLS] row of the final attention matrix. We also consider a “To [CLS]” operation, by extracting the [CLS] column of the final matrix, containing the attention that the [CLS] token receives from each input token. Two additional operations that were identified in the literature, are selecting the maximum value from each column “Max

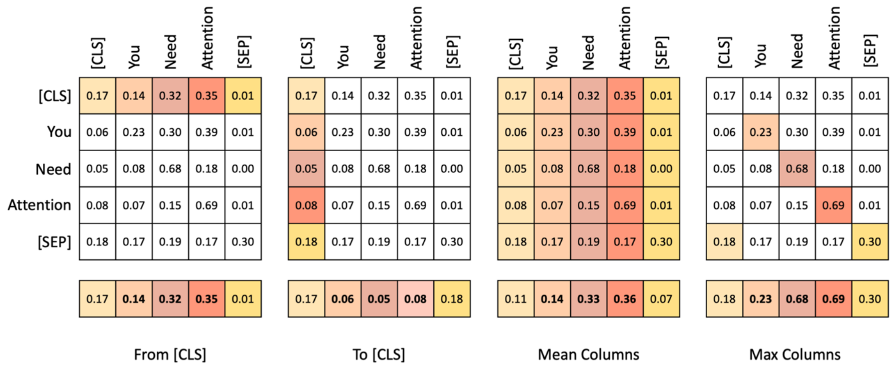


Fig. 1 Interpretation extraction operations from an attention matrix

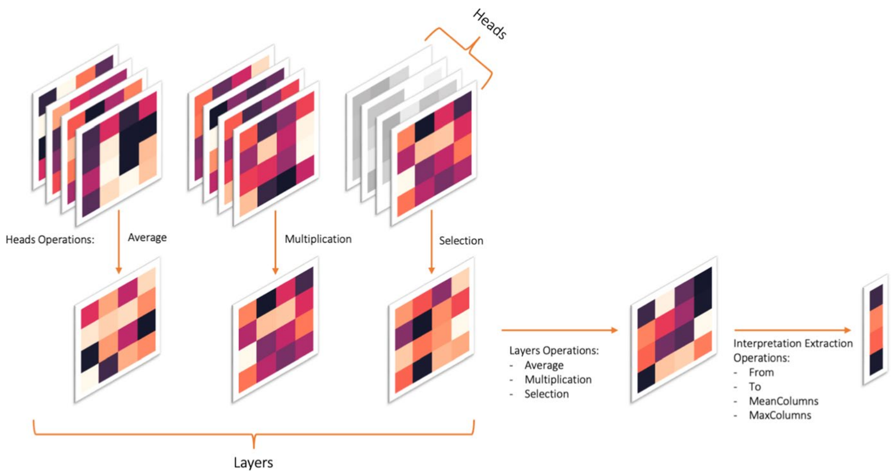


Fig. 2 Head, layer and matrix operations

Columns" (Schwenke and Atzmueller 2021) and averaging the columns of the attention matrix "Mean Columns" (Clark et al. 2019). All these four operations are presented in Fig. 1.

In summary, we considered the following operations that are also depicted in Fig. 2: (a) averaging, multiplying and selection for heads, (b) averaging, multiplying and selection for layers, and c) From [CLS], To [CLS], Max Columns and Mean Columns at the matrix level. The combinations of these operations lead to a total of $(2 + H) \times (2 + M)$

$\times$ 4 potentially different attention-based interpretations, where H denotes the number of heads and M the number of layers.

3.1.3 Selecting the most faithful interpretation

Given a transformer model f, an input sequence x, Optimus Prime extracts interpretations using all combinations of the available operations at the head, layer and matrix level (see Algorithm 1). To arbitrate among all these interpretations, it uses an unsupervised faithfulness evaluation metric e, in order to output the most faithful interpretation (see Algorithm 2).

Algorithm 1 ExtractInterpretations

Require: model f, instance x
M = number of layers in f
H = number of heads in each layer of f
A_{ij} = attention matrix of layer i and head j in f given x
for head operation $h \in [mean, multi, 1, \ldots, H]$ **do**
  for $i \in [1, \ldots, M]$ **do**
    $B_i = h(A_i)$ ▷ apply h across the head matrices of each layer
  for layer operation $l \in [mean, multi, 1, \ldots, M]$ **do**
    $C = l(B)$ ▷ apply l across the layer matrices
    for matrix operation $m \in [from, to, mean\_col, max\_col]$ **do**
      $w[h, l, m] = m(C)$ ▷ extract interpretation
return w

Algorithm 2 Optimus Prime

Require: model f, instance x, faithfulness evaluation metric e
$w = ExtractInterpretations(f, x)$
$best = [mean, mean, from]$
$best\_score = 0$
for head operation $h \in [mean, multi, 1, \ldots, H]$ **do**
  for layer operation $l \in [mean, multi, 1, \ldots, M]$ **do**
    for matrix operation $m \in [from, to, mean\_col, max\_col]$ **do**
      $score = e(w[h, l, m], f, x)$
      if $score > best\_score$ **then**
        $best\_score = score$
        $best = [h, l, m]$
return $best$

The first variation, Optimus Batch, finds the most faithful combination of operations for a set of instances (Algorithm 3). This combination is then used to provide interpretations for future instances. Optimus Batch is faster and more environmentally friendly because the search for the most faithful combination occurs only once. However, it is expected to yield lower results than Optimus Prime, which performs the process separately for each instance.

Algorithm 3 Optimus Batch

Require: model f, set of instances X, faithfulness evaluation metric e

```
best = [mean, mean, from]
best_score = 0
for i ∈ [1, ..., |X|] do
    w_i = ExtractInterpretations(f, x_i)
for head operation h ∈ [mean, multi, 1, ..., H] do
    for layer operation l ∈ [mean, multi, 1, ..., M] do
        for matrix operation m ∈ [from, to, mean_col, max_col] do
            score[h, l, m] = 0
            for i ∈ [1, ..., |X|] do
                score[h, l, m] = score[h, l, m] + e(w_i[h, l, m], f, x_i)
            if scores[h, l, m] > best_score then
                best_score = scores[h, l, m]
                best = [h, l, m]
return best
```

Selecting one combination to extract an interpretation in a multi-label task with L labels, where each instance relates to more than one label may be insufficient, as the positive prediction of different labels may have different interpretation. Therefore, Optimus Label acquires multiple interpretations, one for each different label predicted for an examined instance x (see Algorithm 4). While in a binary classification task this makes no difference, in multi-label tasks it can enhance the performance, as there is more flexibility to match the different ground truth rationales of the positive labels.

Algorithm 4 Optimus Label

Require: model f, instance x, faithfulness evaluation metric e

```
w = ExtractInterpretations(f, x)
for predicted label p in f(x) do
    best_p = [mean, mean, from]
    best_score_p = 0
    for head operation h ∈ [mean, multi, 1, ..., H] do
        for layer operation l ∈ [mean, multi, 1, ..., M] do
            for matrix operation m ∈ [from, to, mean_col, max_col] do
                score = e(w[h, l, m], f, x, p)
                if score > best_score_p then
                    best_score_p = score
                    best_p = [h, l, m]
return best
```

3.2 Ranked faithful truthfulness

The selection process presented in the previous subsection relies heavily on the use of a faithfulness metric. While metrics like faithfulness score (Du et al. 2019) and truthfulness (Mollas et al. 2022) could be used, this work introduces a novel feature importance metric named Ranked Faithful Truthfulness (RFT). Inspired by both faithfulness score and truthfulness, this metric combines their qualities to provide a more complete evaluation. RFT examines the whole interpretation, assigning each token a different penalty based on its importance.

Considering model f, input sequence x of size S, and the interpretation z of $f(x)$, as discussed in the beginning of Sect. 3, RFT performs S independent modifications to x, each time removing a different token, t_i, leading to instance $x^{(-i)}$. For each modification, it computes a faithfulness score v based on w_i and the difference of $f_p(x)$ and $f_p(x^{(-i)})$, where f_p returns the probability of x belonging in a certain label, as follows:

$$v(x,z,i) = \begin{cases} f_p(x) - f_p(x^{(-i)}), & \text{If } w_i > 0, \\ f_p(x^{(-i)}) - f_p(x), & \text{If } w_i < 0, \\ -|f_p(x) - f_p(x^{(-i)})|, & \text{If } w_i = 0 \end{cases} \tag{3}$$

For non-zero weights, this score is positive (negative) when the change in prediction aligns (contrasts) with our expectations given the weight of the model. For zero weights, it is negative or zero. In all cases, its magnitude corresponds to the absolute value of the difference in predictions.

In addition, RFT normalizes this score proportionally to the importance of each token. An intuitive way to achieve this would be to multiply it by the absolute value of the token importance $|w_i|$. However, this would result in information loss, as the prediction changes of zero weights would not be considered. We instead divide the score by the rank $r(t_i)$ of token t_i based on the absolute value of its weight. For example, the ranks of 3 tokens with importance values −0.1, 0.3, 0.2, would be 3, 1 and 2, respectively. Equation (4) provides the definition of RFT. Higher RFT values indicate better performance.

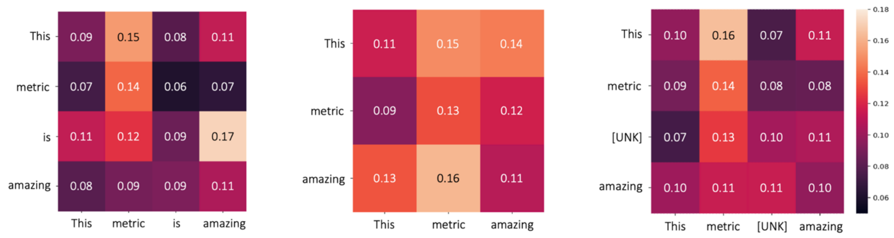
$$\mathrm{RFT}(x, z) = \frac{1}{S} \sum_{i=1}^{S} \frac{v(x, z, i)}{r(t_i)} \tag{4}$$

3.2.1 Token replacement by [UNK]

Faithfulness-oriented metrics, including RFT, evaluate the performance of an interpretability technique based on how the model's decision changes when one or more tokens of the input is removed. This however affects the context for the rest of the tokens, which is important in sequence processing models, like recurrent neural networks and transformers, and even more so, if they use positional encoding, which is the standard for transformer models. This is even more apparent when measuring the distributional shift between the original text and the one altered after removing the token (Rychener et al. 2020). This finding suggests that simply removing words or tokens produces texts that are out-of-distribution for the transformer, greatly hindering the model's performance.

To address this issue, we propose to replace tokens with [UNK], the special token for representing tokens that are not in the vocabulary, instead of deleting them. This way, we nullify the influence of the replaced token, while minimally affecting the context. Another similar option would be to use [MASK] which the transformer is already familiar with, rather than [UNK] (Liu et al. 2022). However, this would also lead to erroneously deflated scores, since the model is trained to replace [MASK] with words fitting to the context.

Figure 3 shows an example of the change in attentions when using [UNK], where image (a) presents the attentions of the initial sequence, (b) after the most important token is removed, and (c) when replaced with [UNK]. We can see that removing the token affects attentions between the remaining tokens more than replacing it with [UNK]. For example, the attention "metric" has towards "amazing" is 0.07 in the original sequence. By removing "is" attention increases to 0.12, which is to be expected since the context of the sequence changed, as these tokens are next to each other. On the other hand, replacing "is" with [UNK], increases attention slightly to 0.08, since this change does not affect the positions of the tokens.



(a) Original attention scores

(b) After removal of token *is*

(c) After replacement with [UNK]

Fig. 3 Example of attention with token removal and replacement with [UNK]

4 Experiments

This section presents our experimental setup, a comparative evaluation of RFT against faithfulness, and a comparative evaluation of the proposed attention-based explanation methods against LIME, IG and the standard attention-based technique in terms of the quality of their explanations (both quantitatively and qualitatively) and their computational requirements. The code used in our experiments, as well as a ready to use tool for interpreting transformers using Optimus and RFT are available in GitHub.[1]

4.1 Setup

We use BERT and DistilBERT transformer models in our experiments. BERT was chosen due to its high prominence in the literature, while DistilBERT as a lighter alternative. The base implementation for both models was selected since it is the most common choice for text classification tasks. We compare the performance of our technique to two state-of-the-art competitors, LIME (Ribeiro et al. 2016) with 200 neighbours for larger datasets and 2000 for smaller ones, and IG (Sundararajan et al. 2017) with 50 interpolations steps. These methods were covered in Sect. 2.

We experiment on 7 datasets coming from 4 different domains: hate speech, biomedicine, sentiment analysis, and Natural Language Understanding (NLU). We selected datasets of sequence length that could fit into the 512 token capacity of the base implementations of BERT and DistilBERT, ideally being accompanied by (token/sentence-level) rationales and covering both single and multi-label classification tasks. Table 1 presents key statistics about each dataset, along with the performance of the two transformer models.

HateXplain (HX) (Mathew et al. 2021) is a single-label and Ethos (Mollas et al. 2022) a multi-label dataset from the hate speech domain. They contain hate speech posts collected from Twitter/Gab and YouTube/Reddit, respectively. The former classifies these posts as hateful, normal, offensive or undecided, out of which we

[1] https://tinyurl.com/bdh3v2nw.

Table 1 Key statistics for each dataset. Information about mean size is presented in token/sentence-level

Dataset	Rationales	Samples	Mean size	Labels	Performance	Domain
HX	Token	13.749	23.9	1	87.97/87.61	Hate speech
Ethos	–	433	20.4	8	85.71/79.71	Hate speech
AIS	–	3.024	44.8/7.1	1	98.28/98.96	Biomedicine
HoC	Sentence	1.852	244/10.1	10	82.44/78.46	Biomedicine
MV	Token	421	388/18.7	1	92.61/93.86	Sent. analysis
HB	Token	500	18.5	6	61.10/57.39	Sent. analysis
ESNLI	Token	10.000	24.4	1	90.13/90.05	NLU

Performance is measured in terms of F_1 macro (%) for BERT/DistilBERT

disregard the last two classes. In Ethos, each post is associated with eight labels regarding violence, target and type of hate speech.

From the biomedicine domain, we use Acute Ischemic Stroke (AIS) (Kim et al. 2019), a single-label dataset, and Hallmarks of Cancer (HoC) (Baker et al. 2015), a multi-label one. The first concerns medical notes of brain MRI scans for acute ischemic stroke. HoC, contains biomedical abstracts indexed with 10 hallmarks of cancer.

From the sentiment analysis domain, two datasets, Movies (MV) (DeYoung et al. 2020) and Hummingbird (HB) (Hayati et al. 2021), were employed for single and multi-label tasks, respectively. Movies contains reviews with positive or negative sentiment. Due to the larger sequences of the samples in this dataset, we applied a filtering step, only keeping the ones with size less than or equal to 512. Hummingbird concerns a textual style classification task, with labels including politeness, sentiment, offensiveness, and five emotions.

ESNLI (Camburu et al. 2018) is a single-label classification dataset for natural language understanding. Given two sentences, the premise and hypothesis, the objective is to determine their relationship: entailment, contradiction, or neutral. Due to the ambiguous nature of the *neutral* class in the examples, we only kept the ones related to the *entailment* and *contradiction* classes. Furthermore, we limit the number of examples to the first 10,000.

Table 2 Pearson and Spearman correlations between faithfulness score (F) and RFT with AUPRC

	BERT				DistilBERT			
	Pearson		Spearman		Pearson		Spearman	
	F	RFT	F	RFT	F	RFT	F	RFT
HX	0.93	**0.96**	0.93	**0.96**	0.91	**0.96**	0.93	**0.95**
HoC	**0.67**	**0.67**	**0.58**	**0.58**	0.82	**0.85**	0.82	**0.86**
MV	0.33	**0.51**	0.31	**0.40**	0.67	**0.77**	0.37	**0.52**
HB	0.69	**0.76**	0.67	**0.76**	**0.86**	**0.86**	**0.83**	**0.83**
ESNLI	0.67	**0.77**	0.63	**0.75**	0.67	**0.77**	0.69	**0.74**
Average	0.66	**0.73**	0.62	**0.69**	0.79	**0.84**	0.72	**0.78**

Higher correlation denoted with bold

We used 20% of each dataset as test set, while the remaining 70 and 10% were used as training and validation sets, respectively, for fine-tuning the models. We ran our experiments once for each dataset, as all examined techniques besides LIME are deterministic, and the computational complexity of LIME makes multiple runs prohibitive.

4.2 Evaluating RFT

In this experiment, we focus on the five datasets where rationales are available and measure the Pearson and Spearman correlation of the faithfulness score (F) and our RFT metric with the supervised AUPRC metric, which uses ground-truth rationales. For each dataset, we measure the F, RFT and AUPRC performance of each interpretation technique (for Optimus we separately consider each combination of head, layer and matrix operation), by averaging the scores of the techniques across the test instances. Then, we compute the correlation values between the performance of F and AUPRC as well as RFT and AUPRC across the techniques.

Table 2 shows that RFT is more correlated to AUPRC than F. Specifically, in 4 out of the 5 examined datasets RFT has higher correlation to AUPRC than F when evaluating the interpretations for BERT's decisions, while in HoC the 2 metrics have the same correlation. For DistilBERT, we observe a similar pattern, with RFT being more aligned to AUPRC in 4 out of 5 cases and the same in HB, with respect to F. Therefore, we use RFT as the faithfulness evaluation metric of Optimus, as well as for the comparison of the interpretability techniques in the following section.

4.3 Quantitative results

In the tables below, B denotes a baseline attention setup, namely *mean* for heads, *mean* for layers, and *From [CLS]* at the matrix level. These operations are the ones most commonly found in the literature. Regarding our technique, OP represents Optimus Prime, OB the Optimus Batch variant and OL the Optimus Label one. It is worth noting that, OP and OL, yield the same results in binary classification.

For datasets with larger sequences, namely AIS, HoC, and MV, the evaluation using RFT was performed both at the token and at the sentence level, with the latter designated by (S) next to the dataset name. For the sentence level experiments, the weight of a sentence s_k is obtained by computing the average weight of its tokens: $\frac{1}{|s_k|}\sum_{t_i \in s_k} w_i$.

The top part of Table 3 summarizes the RFT performance of the examined techniques across the different datasets in relation to the explanations provided for BERT. We can see that even the baseline attention setup achieves competitive performance in all datasets when using A as raw attention with a mean rank of 7.7. Furthermore, our proposed unsupervised process further boosts those results, with OP obtaining a mean rank of 2.4. OL increases the performance in multi-label datasets achieving a mean rank of 1.7. These two techniques, however, are computationally demanding since they require a search step for each instance and label. In contrast,

Table 3 Performance of interpretability techniques in terms of RFT (top) and AUPRC (bottom) when explaining BERT on different datasets

Dataset	LIME	IG	A				A*			
			B	OB	OP	OL	B	OB	OP	OL
HX	.180	.487	.455	.465	<u>.528</u>	<u>.528</u>	.133	.453	**.548**	**.548**
Ethos	.483	.515	.422	.444	.543	<u>.620</u>	.181	.450	.498	**.635**
AIS	.068	.079	.063	.081	**.110**	**.110**	.001	.077	<u>.105</u>	<u>.105</u>
AIS (S)	.132	.164	.139	.185	**.202**	**.202**	.008	.173	<u>.201</u>	<u>.201</u>
HoC	.114	.307	.239	.240	.349	**.417**	−.013	.208	.326	<u>.399</u>
HoC (S)	.141	.270	.244	.273	.360	<u>.404</u>	−.132	.203	.372	**.436**
MV	.011	.062	.132	.137	**.242**	**.242**	.001	.085	<u>.201</u>	<u>.201</u>
MV (S)	.063	.053	.110	.144	<u>.293</u>	<u>.293</u>	−.124	.144	**.341**	**.341**
HB	.356	.151	.226	.269	.408	<u>.429</u>	−.079	.259	.410	**.437**
ESNLI	.292	.277	.246	.453	**.636**	**.636**	−.074	.394	<u>.627</u>	<u>.627</u>
Avg. Rank	7.9	6.7	7.7	5.6	2.4	1.7	10.0	7.0	2.9	1.9
HX	.296	**.391**	.366	.374	.369	.371	.358	<u>.387</u>	.367	.369
HoC	.367	**.646**	.554	<u>.557</u>	.511	.547	.469	.421	.493	.546
MV	.139	.169	.176	<u>.183</u>	**.188**	**.188**	<u>.183</u>	.147	.177	.177
HB	**.506**	.366	.399	.483	.479	.454	.391	<u>.498</u>	.456	.443
ESNLI	.477	.446	.411	**.510**	<u>.500</u>	<u>.500</u>	.433	.452	.491	.491
Avg. Rank	7.4	5.6	7.2	2.4	3.6	3.6	7.8	5.8	5.6	5.4

Best performance denoted with bold, second best denoted with underline. The average rank of each technique across the datasets is also available

OB which also has performance higher than state-of-the-art (mean rank 5.6), is less costly and friendlier to the environment. The results, when using A^*, show a slight decline for OP (1.9) and OL (2.9). In addition, the baseline setup has the worst performance for A^*, consistently being last (10). Investigating the interpretations provided by the A^* baseline setup, we found that many of them consist solely of negative weights, which is the cause for the low RFT performance. The OB results with A^* are much worse than those with A.

Similarly, the bottom part of Table 3 presents the results for the AUPRC metric in datasets with rationales. The baseline attention yet again outperforms LIME in mean rank, while our unsupervised process increases the obtained AUPRC values in most datasets. Here, however, we can see that the best technique is OB using A, with a rank of 2.4, instead of OP and OL, which achieve a mean rank of 3.6. The cause of this phenomenon is that our unsupervised procedure finds the best setup per instance according to RFT, which may not necessarily increase the AUPRC scores. Interpretations provided using A^* seem to be less effective than those of A for the AUPRC metric as well, specifically OB's ranking goes from 2.4 to 5.8, OP's from 3.6 to 5.6 and OL's from 3.6 to 5.4. Nevertheless, the results of A^*'s RFT evaluation are much lower than AUPRC's. This is due to the two metrics' distinct natures, with RFT considering and evaluating polarity as well as ranking, while AUPRC only

Table 4 Performance of interpretability techniques in terms of RFT (top) and AUPRC (bottom) when explaining DistilBERT on different datasets

Dataset	LIME	IG	A				A*			
			B	OB	OP	OL	B	OB	OP	OL
HX	.166	.309	.293	.349	**.477**	**.477**	.066	.355	<u>.475</u>	<u>.475</u>
Ethos	.499	.540	.471	.497	.591	<u>.651</u>	.200	.482	.596	**.658**
AIS	.064	.082	.093	.095	**.116**	**.116**	.001	.091	<u>.113</u>	<u>.113</u>
AIS (S)	.111	.193	.207	.216	<u>.223</u>	<u>.223</u>	−.079	.202	**.225**	**.225**
HoC	.079	.291	.216	.216	.306	**.354**	.003	.204	.293	<u>.334</u>
HoC (S)	.117	.266	.245	.245	.326	<u>.356</u>	−.098	.165	.337	**.376**
MV	.023	.098	.073	.080	**.185**	**.185**	−.003	.079	<u>.173</u>	<u>.173</u>
MV (S)	.069	.140	.136	.159	<u>.239</u>	<u>.239</u>	−.103	.158	**.253**	**.253**
HB	.329	.167	.281	.319	.399	**.406**	.194	.319	.394	<u>.402</u>
ESNLI	.413	.353	.326	.377	**.619**	**.619**	−.035	.356	<u>.612</u>	<u>.612</u>
Avg. Rank	7.9	6.8	7.4	6.0	2.4	1.6	9.9	6.9	2.8	2.0
HX	.322	.378	.342	**.402**	.381	<u>.392</u>	.375	**.402**	.382	.387
HoC	.387	**.671**	<u>.607</u>	<u>.607</u>	.525	.596	.535	.442	.494	.547
MV	.147	.240	.205	**.305**	.238	.238	.176	<u>.275</u>	.229	.229
HB	**.514**	.372	.386	.470	.488	.453	.388	<u>.496</u>	.473	.447
ESNLI	.463	.447	.409	.489	<u>.497</u>	<u>.497</u>	.487	**.501**	.496	.496
Avg. Rank	7.8	6.0	7.6	3.0	4.4	3.8	7.6	3.2	5.4	5.2

Best performance denoted with bold, second best denoted with underline. The average rank of each technique across the datasets is also available

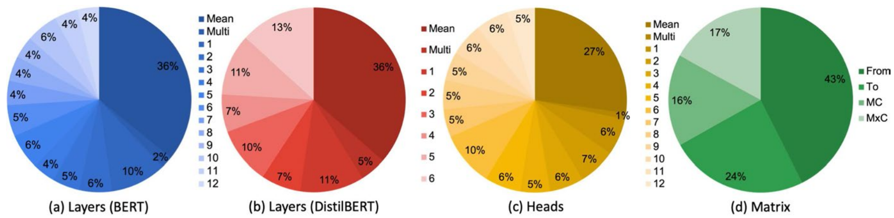


Fig. 4 Frequency of operations in layer, head and matrix level

ranking. The A^* baseline setup, however, has a better ranking when evaluated by AUPRC (7.8) when compared to RFT's evaluation (10). In this scenario, where we detected numerous interpretations with exclusively negative values, RFT evaluated them harshly due to their polarity, although AUPRC considered the ranking to be relatively correct.

The findings for DistilBERT showcased in Table 4 are similar to the aforementioned ones, suggesting that our technique is stable across different encoder-based Transformer architectures.

Fig. 5 Interpretation example for each examined technique on HX. Red (blue) denotes positive (negative) influence (Color figure online)

Another noteworthy discovery is an analysis of the most commonly used options for layer, head, and matrix operations as indicated by OP's per instance best interpretations based on AUPRC, with both A and A^* (Fig. 4). This analysis indicates that, while the most commonly used operations in the literature appear more frequently (mean, multi, From [CLS]), they are insufficient for providing the best attention-based interpretations. As a result, using Optimus, our suggestions (layer and head selection, To [CLS]) can improve those interpretations.

Specifically, Fig. 4a, b show the percentage of occurrence for each operation at the layer level for BERT (12 layers) and DistilBERT (6 layers). We can see that the mean operation is the most common, occurring in 36% of the interpretations, with multi being selected in only 2 and 5% of the interpretations. In case of BERT, the first layer is more commonly employed (10%), whereas DistilBERT favors the last layer (13%). In both scenarios, the rest of the layers occur with similar frequency. In Fig. 4c, we can see the frequency of operations for heads concerning both BERT and DistilBERT. Similarly to before, mean is the most prevalent (27%) and multi is the least dominant (1%). This time, though, all the heads appear at a similar frequency (5–7%), with only the sixth one standing out (10%). Finally, among matrix operations (Fig. 4d), the most common is From [CLS], with 43%, followed by To [CLS], with 24%. Mean Columns (MC) and Max Columns (MxC) occur with comparable frequency in BERT and DistilBERT.

4.4 Qualitative results

In addition to the quantitative results, we present two examples, one from the domain of hate speech and one from biomedicine, to qualitatively evaluate the interpretations provided by LIME, IG, Baseline and OL. Red highlighting indicates a positive influence on the prediction, while blue highlighting indicates a negative influence. The intensity of the colors reflects the magnitude of the influence, with more intense colors representing a stronger impact.

Starting with the hate speech domain, a random instance from the HX dataset predicted by DistilBERT as "*hate speech*" is selected. In Fig. 5, we showcase the selected instance, having removed any tokens that may correspond to offensive or derogatory words or imply them. The first line corresponds to ground truth rationales for the examined instance. Baseline attention seems to highlight the important tokens in the sequence, however, giving higher importance to only one of them,

Fig. 6 Interpretation example for each examined technique on HoC. Red (blue) denotes positive (negative) influence (Color figure online)

according to the ground truth. On the other hand, OPTIMUS correctly identifies the most important tokens, giving minimal weight to others. IG behaves similarly, assigning correct values to important tokens and also including a few irrelevant ones. LIME seems to miss some important tokens, even assigning a negative score to one.

One intriguing aspect of this example is that the two hidden tokens preceding "*strong*" are not included in the ground truth rationale, even though they should have been. This is also supported by the interpretations, which assigned an influence score to those tokens. Indeed, removing these two tokens, has a negative impact on the prediction's probability. This suggests that even ground truth information might be prone to inaccuracies or annotator bias.

Having presented a token-level example, the second one concerns sentence-level interpretations. As such, we select HoC, where sentence-level rationales are available. Choosing a random instance, we obtain its prediction, and select one label among the predicted, namely *enabling replicative immortality*. Figure 6 illustrates the interpretations provided by the techniques for that instance and label. Baseline correctly highlights all three important sentences, but gives less importance to one of them, while also highlighting an unrelated one, according to the ground truth. OL accurately identifies all three sentences. However, it assigns importance weight to an unrelated one as well. IG and LIME both erroneously give one of the three ground truth sentences a negative importance score, but correctly underline the other two.

In both examples, Optimus appears to provide a closer approximation of the ground truth rationales compared to the other techniques. However, it also assigns importance to additional tokens or sentences that are not part of the ground truth rationales. This outcome is expected because human-annotated ground truth rationales often have sparse coverage, and in some cases, they may even miss critical information. In contrast, interpretability techniques provide dense interpretations, with nearly all tokens or sentences receiving a feature importance score. In cases where sparse interpretations are preferred, we can introduce a threshold to these weights, resulting in sparse interpretations for Optimus as well.

4.5 Computational overhead analysis

One key advantage of using attention interpretations, is the low computational overhead since they are already computed during inference. This leads to faster response times and lower environmental impact. This is especially true in cases where one combination of operations is identified beforehand (B and OB and then applied on incoming instances. In contrast, identifying the most faithful combination for each instance individually (OP and OL as seen in Sect. 4.3) and techniques such as LIME and IG require additional procedures, resulting in increased computational cost.

Let's begin by discussing the theoretical complexity of each technique for interpreting a single label. Starting with the competitors, LIME has a complexity of $\mathcal{O}(g \times s + g + g \times s^2 + s^3)$, where g is the number of neighbors and s is the number of tokens in the input sequence. The complexity breakdown is as follows: $\mathcal{O}(g \times s)$ for neighborhood generation, $\mathcal{O}(g)$ for similarity computation, and $\mathcal{O}(g \times s^2 + s^3)$ for ridge regression training. The complexity of IG is $\mathcal{O}(g \times s)$, with g representing the number of interpolation steps and s representing the number of tokens. The baseline attention setup B has a complexity of $\mathcal{O}(1)$ since it only requires three operations to create the interpretation.

OP has a complexity of $\mathcal{O}(g \times s)$, where s represents either the number of tokens in the input sequence or the number of sentences, depending on the specific problem. Meanwhile, g corresponds to the number of different combinations, as discussed in Sect. 3.1.2. On the other hand, OB has a preparation complexity of $\mathcal{O}(g \times s \times v)$, where v denotes the number of instances. During inference, OB has a complexity of $\mathcal{O}(1)$, as the setup has already been selected through the preparation procedure. Lastly, for OL, the runtime is the same as OP since this analysis focuses on a single label.

In smaller input sequences (smaller s), the primary factor influencing complexity among the examined techniques is variable g. However, the value of g varies for each technique. For instance, in LIME, g represents the number of neighbors, with a default value of 5000. In the case of IG, where g represents interpolation steps,

Table 5 Average time response and tCO_2e emissions

	HX		HoC (S)		MV	
	Seconds	tCO_2e	Seconds	tCO_2e	Seconds	tCO_2e
LIME	38.252	4.892	134.903	17.252	27.422	3.507
IG	0.393	0.050	14.699	1.880	3.451	0.441
B/OB	4.21E−05	5.38E−06	0.010	0.001	0.009	0.001
OP/OL	3.050	0.390	5.304	0.678	132.204	16.907

the default value is 50. Regarding OP, g refers to the number of possible attention combinations and varies depending on the specific transformer being examined. For example, in BERT, the value of g is 784.

In larger sequences (larger s, up to 512), the most influential variable shifts to s in LIME and IG. However, for OP, the influence of both s and g remains relatively similar. Notably, OP offers an advantage at the sentence level because s represents sentences rather than tokens. Consequently, the influence of s becomes minimal, resulting in a lower time response as g once again becomes the most influential variable. Hence, we believe that as the sequence length increases, OP performs better than LIME but is surpassed by IG in terms of time response. Nevertheless, this scenario changes in sentence-level situations where OP outperforms both techniques.

To empirically validate this, we performed a time response analysis for each technique in HX (small sequences), HoC and MV (large sequences) using BERT. Additionally, we computed their carbon footprint in terms of metric tons of CO_2 equivalent emissions (tCO_2e) based on (Patterson et al. 2021). We performed our experiments in Google Colab using a single GPU with an average power consumption of 271W. We used the formula tCO_2e=KWh×kg CO_2e per KWh/10^3, assuming kg CO_2e/KWh to be 0.429 (Patterson et al. 2021) and the formula KWh= (interpretation time×# of GPUs × avg. power/GPU × PUE)/10^3, assuming PUE[2] (Power Usage Effectiveness) to be 1.10.

The average time response (seconds) and tCO_2e emissions for each examined technique are showcased in Table 5. When interpretations are provided at sentence-level, OP and OL seem to have lower time responses and emissions, compared to LIME and IG, while at token-level the opposite holds true. This is due to RFT, which is an important part of the OP and OL procedures, needing to examine fewer interpretation elements at the sentence-level. This is also backed up by our theoretical complexity analysis performed on this section. It is worth mentioning that in the case of LIME, we used 2000 neighbors in HX, and 200 in HoC (S) and MV, as LIME cannot be efficiently applied to datasets with larger sequences. Finally, B and OB require the least amount of time.

Figure 7 presents the cumulative tCO_2e emissions for up to 100 instances. On the left, we simulate an interpretable Hate Speech detection system on a social media

[2] https://tinyurl.com/2u8zeks8.

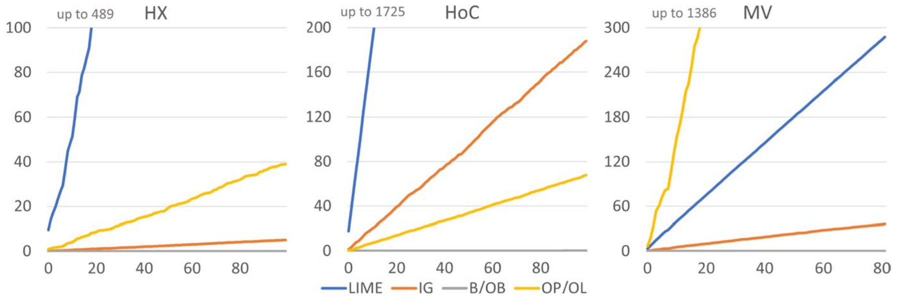


Fig. 7 Interpolated emissions for each method (Left: HX, Middle: HoC (S), Right: MV). *X*-axis: number of instances, *Y*-axis: cumulative tCO_2e emissions

platform, where a large amount of data is being produced every second. Likewise, the middle plot concerns interpretability-assisted semantic indexing of biomedical publications, which tend to have bigger sequences, in conjunction to the large number of publications processed each day in databases like PubMed (986K articles for 2020[3]). Finally, on the right, we model a database in which users constantly search for reviews, and an interpretable sentiment analysis system provides insight to these reviews.

5 Conclusions

This section serves as a conclusion to our work, where we discuss our findings, acknowledge the limitations of our technique, and outline potential directions for future research.

5.1 Findings

Some studies argue that attention should not be used as an interpretation tool, while others incorporate attention-based methods in their experiments without specifying how interpretations are produced. This work investigates various ways attention is used in the literature as interpretation and proposes an arbitration scheme to determine the best way to extract interpretations from attention information. The most faithful combination is determined via an optimization procedure based on an unsupervised faithfulness metric.

Our findings support that, when properly configured, especially using operations that select particular attention matrices from heads and layers, attention can effectively be used as an interpretation tool for text classification. In addition, we demonstrate that attention can compete with other cutting-edge techniques in a series of experiments that include a new faithfulness metric based on feature importance.

[3] https://tinyurl.com/4nprsskn.

Furthermore, compared to other techniques, attention is easier to implement, while also being faster and less harmful to the environment, in certain cases, such as B and Optimus Batch.

5.2 Limitations

Optimus does not come without certain limitations. For instance, while identifying the most faithful combination of operations per instance and per label (Optimus Prime and Optimus Label) achieves top performance, it is time-consuming. Additionally, it faces scalability issues in token-level scenarios with large texts since it requires examining every individual token to evaluate the interpretations using the proposed RFT metric. Another limitation is that Optimus has only been implemented and tested on textual data and NLP tasks, and thus, its applicability in different data types remains unexplored.

5.3 Future work

In the future, we aim to address the issue of runtime in the Optimus Prime and Optimus Label setups to make them more environmentally friendly. One approach we will explore is experimenting with alternative faithfulness metrics that are more time efficient or different unsupervised metrics, such as robustness or complexity. Additionally, we will investigate the use of twin models, where one model is responsible for making predictions and another model is dedicated to extracting attention information. This approach could offer significant benefits, as it would enable us to perform RFT queries without the need for attention information, reducing computational complexity while maintaining accuracy.

Another approach would be to explore the differences in performance between using both A and A^* as input to Optimus techniques, instead of using only one of them to extract interpretations. Since both examined transformer models, BERT and DistilBERT, are encoder based, experiments investigating encoder-decoder based models could also be conducted. Given that Optimus is independent of the downstream task, there are other tasks beyond single and multi-label classification worth exploring. For instance, we can investigate the applicability of Optimus to multi-class classification and token classification. Finally, we have plans to conduct a user study to evaluate the quality of the interpretations provided by Optimus.

Acknowledgements The research work was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "First Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment grant" (Project Number: 514)

Availability of Data and Materials All datasets used in these research are public and freely available.

Code Availability Experiments' code is available in GitHub: https://tinyurl.com/bdh3v2nw.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

Abnar S, Zuidema WH (2020) Quantifying attention flow in transformers. CoRR arXiv:2005.00928

Alammar J (2021) Ecco: An open source library for the explainability of transformer language models. In: Proceedings of the 59th Annual Meeting of the ACL and the 11th International Joint Conference on Natural Language Processing: System Demonstrations, pp 249–257. ACL, Online. https://doi.org/10.18653/v1/2021.acl-demo.30

Bacco L, Cimino A, Dell'Orletta F, Merone M (2021) Explainable sentiment analysis: a hierarchical transformer-based extractive summarization approach. Electronics 10(18):2195. https://doi.org/10.3390/electronics10182195

Bach S, Binder A, Montavon G, Klauschen F, Müller K-R, Samek W (2015) On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS One 10(7):1–46. https://doi.org/10.1371/journal.pone.0130140

Baker S, Silins I, Guo Y, Ali I, Högberg J, Stenius U, Korhonen A (2015) Automatic semantic classification of scientific literature according to the hallmarks of cancer. Bioinformatics 32(3):432–440. https://doi.org/10.1093/bioinformatics/btv585

Bastings J, Filippova K (2020) The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In: BlackboxNLP@EMNLP, pp 149–155. ACL, Online

Brunner G, Liu Y, Pascual D, Richter O, Ciaramita M, Wattenhofer R (2020) On identifiability in transformers. In: 8th International Conference on Learning Representations, ICLR. OpenReview.net, Online. https://openreview.net/forum?id=BJg1f6EFDB

Camburu O-M, Rocktäschel T, Lukasiewicz T, Blunsom P (2018) e-snli: Natural language inference with natural language explanations. Advances in Neural Information Processing Systems 31

Chan CS, Kong H, Guanqing L (2022) A comparative study of faithfulness metrics for model interpretability methods. In: Proceedings of the 60th Annual Meeting of the ACL (Volume 1: Long Papers), pp 5029–5038. ACL, Dublin, Ireland. https://doi.org/10.18653/v1/2022.acl-long.345

Chefer H, Gur S, Wolf L (2021) Transformer interpretability beyond attention visualization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 782–791

Clark K, Khandelwal U, Levy O, Manning CD (2019) What does BERT look at? An analysis of BERT's attention. In: BlackboxNLP@EMNLP, pp. 276–286. ACL, Florence, Italy. https://doi.org/10.18653/v1/W19-4828

DeYoung J, Jain S, Rajani NF, Lehman E, Xiong C, Socher R, Wallace BC (2020) ERASER: A benchmark to evaluate rationalized NLP models. In: Proceedings of the 58th Annual Meeting of the ACL, pp 4443–4458. ACL, Online. https://doi.org/10.18653/v1/2020.acl-main.408

Du M, Liu N, Yang F, Ji S, Hu X (2019) On attribution of recurrent neural network predictions via additive decomposition. In: The World Wide Web Conference, pp 383–393

EU (2021) Proposal for a regulation of the european parliament and the council laying down harmonised rules on artificial intelligence (AI Act) and amending certain union legislative acts. EUR-Lex-52021PC0206

Feldhus N, Schwarzenberg R, Moller S (2021) Thermostat: A large collection of nlp model explanations and analysis tools. In: EMNLP

Hayati SA, Kang D, Ungar L (2021) Does BERT learn as humans perceive? understanding linguistic styles through lexica. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP, 7–11 November, pp 6323–6331. ACL, Online. https://doi.org/10.18653/v1/2021.emnlp-main.510

Herman B (2017) The promise and peril of human evaluation for model interpretability. ArXiv arXiv:1711.07414

Hoover B, Strobelt H, Gehrmann S (2020) exBERT: A visual analysis tool to explore learned representations in transformer models. In: Proceedings of the 58th Annual Meeting of the ACL:

System Demonstrations, pp 187–196. ACL, Online. https://doi.org/10.18653/v1/2020.acl-demos.22

Jain S, Wallace BC (2019) Attention is not explanation. In: NAACL-HLT, pp 3543–3556. ACL, Minneapolis, Minnesota

Kim C, Zhu V, Obeid J, Lenert L (2019) Natural language processing and machine learning algorithm to identify brain MRI reports with acute ischemic stroke. PLoS One 14(2):1–13. https://doi.org/10.1371/journal.pone.0212778

Kovaleva O, Romanov A, Rogers A, Rumshisky A (2019) Revealing the dark secrets of BERT. CoRR arXiv:1908.08593

Lertvittayakumjorn P, Toni F (2019) Human-grounded evaluations of explanation methods for text classification. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, November 3–7, pp 5194–5204. ACL, Hong Kong, China. https://doi.org/10.18653/v1/D19-1523

Liu S, Le F, Chakraborty S, Abdelzaher T (2021) On exploring attention-based explanation for transformer models in text classification. In: IEEE International Conference on Big Data (Big Data), pp 1193–1203. https://doi.org/10.1109/BigData52589.2021.9671639

Liu Y, Li H, Guo Y, Kong C, Li J, Wang S (2022) Rethinking attention-model explainability through faithfulness violation test. In: International Conference on Machine Learning, ICML, 17–23 July, vol. 162, pp 13807–13824. PMLR, Baltimore, Maryland. https://proceedings.mlr.press/v162/liu22i.html

Lundberg SM, Lee S-I (2017) A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems 30, pp 4765–4774. Curran Associates, Inc., Long Beach, California

Mathew B, Saha P, Yimam SM, Biemann C, Goyal P, Mukherjee A (2021) Hatexplain: A benchmark dataset for explainable hate speech detection. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, February 2-9, pp 14867–14875. AAAI Press, Online. https://ojs.aaai.org/index.php/AAAI/article/view/17745

Melis DA, Jaakkola T (2018) Towards robust interpretability with self-explaining neural networks. In: Advances in Neural Information Processing Systems, Montreal, Canada, pp 7775–7784

Mollas I, Bassiliades N, Tsoumakas G (2022) LioNets: a neural-specific local interpretation technique exploiting penultimate layer information. Appl Intell. https://doi.org/10.1007/s10489-022-03351-4

Mollas I, Chrysopoulou Z, Karlos S, Tsoumakas G (2022) ETHOS: a multi-label hate speech detection dataset. Complex Intell Syst. https://doi.org/10.1007/s40747-021-00608-2

Mullenbach J, Wiegreffe S, Duke J, Sun J, Eisenstein J (2018) Explainable prediction of medical codes from clinical text. In: NAACL-HLT, pp 1101–1111. ACL, New Orleans, Louisiana

Niu R, Wei Z, Wang Y, Wang Q (2022) Attexplainer: Explain transformer via attention by reinforcement learning. In: Proceedings of the 31st International Joint Conference on Artificial Intelligence, IJCAI-22, Vienna, Austria, pp 724–731. https://doi.org/10.24963/ijcai.2022/102

Patterson D, Gonzalez J, Le Q, Liang C, Munguia L-M, Rothchild D, So D, Texier M, Dean J (2021) Carbon emissions and large neural network training. arXiv. https://doi.org/10.48550/ARXIV.2104.10350

Ribeiro MT, Singh S, 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. ACM

Robnik-Sikonja M, Bohanec M (2018) Perturbation-based explanations of prediction models. In: Zhou J, Chen F (eds) Human and machine learning - visible, explainable, trustworthy and transparent. Springer International, Cham, pp 159–175

Rychener Y, Renard X, Seddah D, Frossard P, Detyniecki M (2020) On the granularity of explanations in model agnostic NLP interpretability. arXiv. To appear in ECMLPKDD2022 proceedings of XKDD workshop. https://doi.org/10.48550/ARXIV.2012.13189

Schwenke L, Atzmueller M (2021) Show me what you're looking for: visualizing abstracted transformer attention for enhancing their local interpretability on time series data. In: The International FLAIRS Conference Proceedings, vol. 34

Schwenke L, Atzmueller M (2021) Show me what you're looking for: Visualizing abstracted transformer attention for enhancing their local interpretability on time series data. The International FLAIRS Conference Proceedings 34. https://doi.org/10.32473/flairs.v34i1.128399

Sundararajan M, Taly A, Yan Q (2017) Axiomatic attribution for deep networks. In: Proceedings of the 34th International Conference on Machine Learning, ICML 6-11 August, vol. 70, pp 3319–3328. PMLR, Sydney, NSW, Australia. http://proceedings.mlr.press/v70/sundararajan17a.html

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Advances in neural information processing systems 30

Vig J (2019) A multiscale visualization of attention in the transformer model. CoRR arXiv:1906.05714

Wang Y, Lee H-Y, Chen Y-N (2019) Tree transformer: Integrating tree structures into self-attention. In: Proceedings of EMNLP 2019 and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp 1061–1070. ACL, Hong Kong, China. https://doi.org/10.18653/v1/D19-1098

Wiegreffe S, Pinter Y (2019) Attention is not not explanation. In: EMNLP/IJCNLP, pp 11–20. ACL, Hong Kong, China

Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, Cistac P, Rault T, Louf R, Funtowicz M, Davison J, Shleifer S, von Platen P, Ma C, Jernite Y, Plu J, Xu C, Le Scao T, Gugger S, Drame M, Lhoest Q, Rush A (2020) Transformers: State-of-the-art natural language processing. In: Proceedings of EMNLP 2020: System Demonstrations, pp 38–45. ACL, Online. https://doi.org/10.18653/v1/2020.emnlp-demos.6