Is Attention Truly All We Need?

Alexander Karpekov [akarpekov3@gatech.edu], Sidney Miller [smiller324@gatech.edu]

December 10, 2023

Abstract

In this paper we are investigating whether Transformer attention weights can be used directly to derive feature importance. Our findings suggest that while the absolute attention weights are indeed correlated with the most important tokens in NLP tasks such as sentiment analysis, augmenting them with gradient information results in even better feature importance gain. This holds true for BERT-based encoder-only Transformer architectures. Our findings apply to both encoder-only and encoder-decoder Transformer models like BART, showcasing the power of attention weights to the explainability of the model decision making. As a part of this project, we are also sharing our GitHub repository that we built from scratch. It lets anyone to test their own Transformer model and use gradient-based attention mechanism to obtain input token feature importance.

1 Introduction

1.1 Motivation

Large Language Models have been gaining a lot of interest since the creation of OpenAI's ChatGPT. Because of this, there have been more conversations surrounding business use cases for LLMs than have ever been seen before. At the same time, given the black-box nature of these models, it is oftentimes hard to ensure the interpretability and, more importantly, verifiability of these large models. This can sometimes prevent these models from having broader business, academic, and personal application. This concern is not unfounded, as there are many documented cases of LLMs providing generally wrong information (Zhang et al., 2023) and other forms of harmful information (Bhattacharyya et al., 2023). This problem is generally caused by a lack of model transparency, which then leads to the creation of harmful information and hallucinations (Zhao et al., 2023).

In order to both better address stakeholder concerns and get a better understanding of what causes LLM hallucinations, it is important to make models more interpretable. If this can be done, we can better correct and tune LLMs and ease the concerns from stakeholders. Put differently, "...explainability builds appropriate trust by elucidating the reasoning mechanism behind model predictions in an understandable manner, without requiring technical expertise. With that, end users are able to understand the capabilities, limitations, and potential flaws of LLMs" (Zhao et al., 2023)

1.2 Goal

Our goal in this project was to explore if we can extract feature importance information from large Transformer models. Understanding what inputs the model is relying upon the most to make its decisions improves

our understanding of the inner workings of the model and increases the trust in model application. In doing so, we hope to be able to better explore the explainability aspect of LLMs and be able to get a better idea of what an LLM is actually doing. In order to explore this concept, we decided to replicate the results from the paper "On Exploring Attention-based Explanation for Transformer Models in Text Classification". This paper explores the use of different metrics to better explain what a transformer model finds important. Given that the paper came out in 2021, we wanted to find out if, first, its findings could be replicated, and second, if they are still relevant given recent advances in LLMs.

We specifically decided to replicate this paper as the authors only really explore explainability for a custom BERT model that they created from scratch and fine tuned. We felt that it would be better to not only explore the metrics they generated on more standard language models, but to also explore different architecture types (e.g., also explore encoder-decoder models such as BART).

1.3 Our Contribution

The two core contributions of this project beyond the replication and the verification of the previous paper results are as follows:

1. Providing an end-to-end codebase which allows the user to get both the attention and gradient attention ("AGrad") for any classification model using HuggingFace transformer models (the original paper did not share their code with the general public): `https://github.com/karpekov/tranformer_attention_importance`
2. Extending analysis attention layer to multiple Transformer architectures, both encoder-only (e.g., DistilBERT) and encoder-decoder models (e.g., BART).

1.4 Related Work

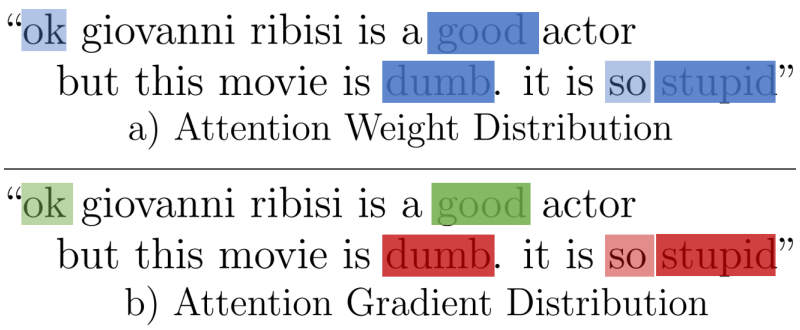
As transformers grew in popularity following the seminal paper "Attention is All You Need" (Vaswani et al., 2017), many researchers turned to self-attention mechanisms as a potentially powerful tool for attributing feature importance to the input tokens. This method could be used as a means of explaining what a NLP model was doing. For example, some argued that high attention weights associated with particular tokens are largely responsible for the final model output. (Jain & Wallace, 2019) However, the paper "Attention is not Explanation" argued that attention is not necessarily a good measure for LLM interpretability. They stated the following:

> "In this work we perform extensive experiments across a variety of NLP tasks that aim to assess the degree to which attention weights provide meaningful "explanations" for predictions. We find that they largely do not. For example, learned attention weights are frequently uncorrelated with gradient-based measures of feature importance, and one can identify very different attention distributions that nonetheless yield equivalent predictions. Our findings show that standard attention modules do not provide meaningful explanations and should not be treated as though they do." (Jain & Wallace, 2019)

It is important to note that the paper written by Jain & Wallace (2019) only used attention mechanisms within the bi-directional LSTM architecture, and not a Transformer block (it is not surprising given that both (Jain & Wallace, 2019) and (Vaswani et al., 2017) came out around the same time). As a response to this, in "On Exploring Attention-based Explanation for Transformer Models in Text Classification" (the paper we are replicating) the authors sought to explore how different variants of attention layer weights could be used as a measure of explainability. Essentially, the authors do this by taking the label associated with the text entry into account by incorporating the gradients of the attention weights to determine the individual token importance. In other words:

> "...the key reason why attention weights cannot be directly used as effective relevance indications is because they do not contain the directional information for relevance (i.e., whether the input tokens contribute towards or against the prediction)"

From this paper, we are specifically replicating the "Attention Gradient Distribution" or AGrad, which takes into account the directional information. For example:



In the first example, it is difficult to determine how each word contributes to the final classification just based on the token weight. Some of the tokens which have the highest weights do not necessarily correlate to the same class. Incorporating the gradient with the initial attention gradient distribution can be used to fix it. When we incorporate the gradient in the following example, it is more clear how each word is contributing to the final classification.

2 Methodology

To extract feature importance for a given sentence , we will be using two main methods: Attention-based and Gradient-Attention based (AGrad). The average attention weight for the [CLS] token, denoted as $\overline{w}_{\text{CLS}}$, is calculated as:

$$\overline{\alpha}_{j,\text{CLS}} = \frac{1}{|A|} \sum_{i=1}^{|A|} \alpha_{i,j}, \quad \forall j \text{ in the input sequence tokens} \tag{1}$$

where:

- A represents the set of attention heads in the last layer of a transformer model, such that $A = \{a_1, a_2, \ldots, a_m\}$.

- $\alpha_{i,j}$ denotes the attention weight given to the j-th token by the i-th attention head.

In case of BERT, $|A| = 12$ for the BERT Base model. To obtain gradient-attention weights we follow the computation in (Liu et al., 2021): We first compute the gradient of the loss with respect to the attention weights, i.e., $\partial L/\partial \alpha_i$ with L denoting the loss, and α_i $(1 \leq i \leq |x|)$ the attention weights. The gradient of the loss w.r.t. an attention weight can indicate whether that attention weight is decreasing or increasing the loss. Then, we compute the product $\partial L/\partial \alpha_i \times \alpha_i$, which helps to mitigate the gradient saturation problem. In the end, we get:

$$\overline{\text{AGrad}}_{j,\text{CLS}} = \frac{1}{|A|} \sum_{i=1}^{|A|} \left(\frac{\partial L}{\partial \alpha_{\text{CLS},i,j}} \times \alpha_{\text{CLS},i,j} \right), \quad \forall j \text{ in the input sequence tokens} \tag{2}$$

where:

- L represents the loss function.
- α_i denotes the attention weight for the i-th position in the input sequence x, where $1 \leq i \leq |x|$.
- $\frac{\partial L}{\partial \alpha_i}$ represents the gradient of the loss function L with respect to the attention weight α_i.

We then use these weights to rank each input token and assign its importance to the classification task. Following the methodology in (Liu et al., 2021), we will evaluate the relevance scores generated by different explanation techniques using faithfulness and consistency tests, described below.

2.1 Faithfulness

Definition: An explanation provided by a feature attribution method is faithful if it reflects the actual information and its importance degree as used by the model to make the decision.

Test: We adopt the occlusion test outlined in (Arras et al., 2017) for the most important tokens in the input. We will drop tokens from the input sentence by masking them, and then compute the accuracy of the sentiment classification. We use three strategies to drop tokens from the input:

- Random strategy: Pick and mask X random tokens in the input sentence. This is used as a baseline evaluation.
- Attention-based: Pick top-X tokens with the highest attention weights $\overline{\alpha}_{j,\text{CLS}}$, mask them and compute accuracy.
- Gradient-Attention based: same as attention based, but using $\overline{\text{AGrad}}_{j,\text{CLS}}$ instead.

A steep decrease in accuracy for a small number of top tokens dropped indicates that those tokens were indeed important for making that particular classification decision.

2.2 Consistency

Definition: An explanation method is consistent if for samples from the same class, their identified important words by the explanation have contextual embeddings that form distinct clusters in the latent space.
A well trained deep learning model is expected to learn latent space representations (i.e., contextual embeddings) that are closer for data samples from the same class (intra-class cohesion) and further for samples belonging to different classes (inter-class separation) (Liu et al., 2021).

Test: To test for this, we select the top 5 words in decreasing importance for each correctly identified sample. We then get the hidden layer representation for each token in each correctly identified sample and use them as token vectors. These vectors are then clustered together and measured based on the consistency using the mean Silhouette score. The score of one indicates appropriate clustering and high consistency, and negative one indicates mismatched clusters or low consistency. Values close to zero indicate overlapping clusters. A Silhouette score of one indicates appropriate clustering and high consistency, a score of negative one indicates mismatched clusters or low consistency, and a score of zero indicates overlapping clusters.

3 Experiment

3.1 Data

For this task, we chose to train a model and evaluate the attention importance using the IMDb dataset created by Maas et al. (2011). This dataset contains 50,000 movie reviews which all contain a positive or negative sentiment. This is a balanced dataset with 50% positive and 50% negative labels, meaning that the baseline accuracy of a random guess is 0.5. We decided to split the dataset half and half between the train and test sets. We chose this specific dataset for two reasons:

1. This dataset was one of the primary datasets the original paper was trained on
2. This dataset has the most tokens in a given review. This means that we could see how the accuracy decays at each step at a more granular level.

3.2 Model Selection and Fine-Tuning

When trying to find the importance of attention layer metrics, we wanted to make sure that the changes and differences we found were due solely to each model's architecture. For this reason, we decided to train several different types of models including encoder-only models (TinyBERT, DistilBERT, RoBERTa) and an encoder-decoder model (BART).
When preparing a model for analysis, we started with a pretrained model of the specified architecture from HuggingFace. Once we downloaded the model, we fine-tuned each one on the IMDb dataset training split for several epochs. After this, the model was exported for future analysis. By training all of the models, we got the following results:

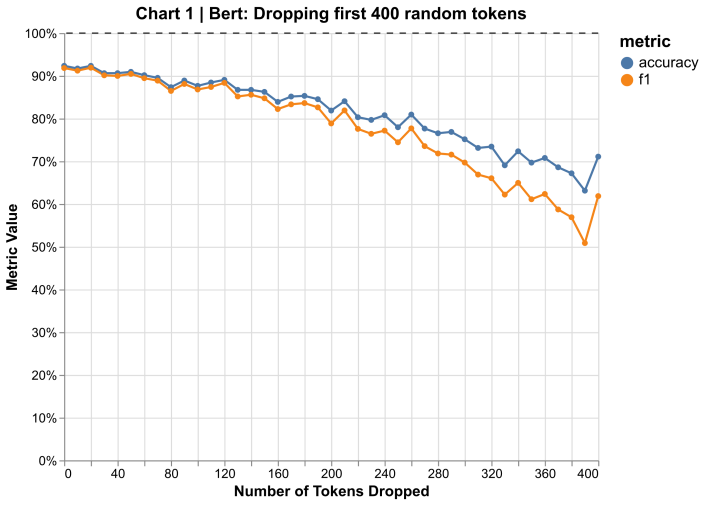
Model	# Parameters	# Layers	# Attention Heads	Accuracy	Precision	Recall	F1
TinyBERT	4.39M	2 Layers	2 Heads	0.883	0.885	0.878	0.882
DistilBERT	66.4M	6 Layers	12 Heads	0.891	0.919	0.864	0.891
BART	70.7M	6 Layers	8 Heads	0.938	0.893	0.962	0.926
RoBERTa	125M	12 Layers	12 Heads	0.953	1.000	0.914	0.955

4 Results

In this section we are presenting the experiments that we ran and the results that we obtained.

4.1 Faithfulness Test Results

First, we will evaluate the faithfulness of different feature importance methods. To set a baseline, we first want to see what impact a random token occlusion strategy has on accuracy. Chart 1 shows how dropping N number of random tokens from the input affects the accuracy of the BERT base model. Note that dropping as many as 100 tokens (given the max review limit of 512) barely affects the model performance: BERT is still able to determine the correct sentiment of the review in 90% of cases. Even dropping 300 tokens only leads to the deterioration of accuracy to 75%. Other models, including DistilBERT and BART, exhibited the same behavior: they were barely sensitive to the loss of up to 100 tokens from the input. This helps us set a baseline on the relative token importance.



Next, we conduct a Faithfulness test for all the fine-tuned models. We use 4 different strategies to drop up to 40 input tokens:

- Random: Same strategy as above which allows us to create a baseline for the faithfulness test.
- Attention based: Drop top-K tokens as determined by the absolute value of the attention weights from each input to the [CLS] token: $\overline{\alpha}_{j,\text{CLS}}$
- Gradient-Attention based, Top: Drop top-K tokens from attention weights times the gradient: $\overline{\text{AGrad}}_{j,\text{CLS}}$
- Gradient-Attention based, Bottom: Same as the previous bullet but reversed. The interpretation for these tokens is as follows: drop tokens that are contributing to the opposite direction of the gradient, or in other words, tokens that are most associated with the opposite class.

Chart 2 shows the results from the DistilBERT model. Note that unlike chart 1, the X-axis only extends up until 40 tokens instead of 400 like in the previous example. The y-axis shows the relative drop from the original accuracy, where at tokens_dropped = 0, the relative accuracy is at 100%. While the random baseline maintains the same level of accuracy like in chart 1, all other strategies appear to have a significant impact on the model accuracy. Attention based strategy leads to a 30% drop in accuracy at 40 tokens dropped, which means that those top-K tokens were correlated with the model classification decision-making. However, the gradient-attention based strategy sees an even higher drop in accuracy of 70% at 40 tokens. Dropping as few as 18 tokens leads to 50% reduction in accuracy, meaning that the model is now more likely to start predicting the opposite class. Note the orange line on the chart: dropping the tokens whose gradient goes in the opposite direction of the loss leads to a 12% improvement in baseline accuracy, increasing it from 91% to almost 100%! These results show that the gradient-weighted attention is very successful at identifying the most important token for and against the classification decision, and therefore represents the most faithful explainability results.

Chart 2 | Distilbert: Faithfulness Test using Various Dropout Strategies

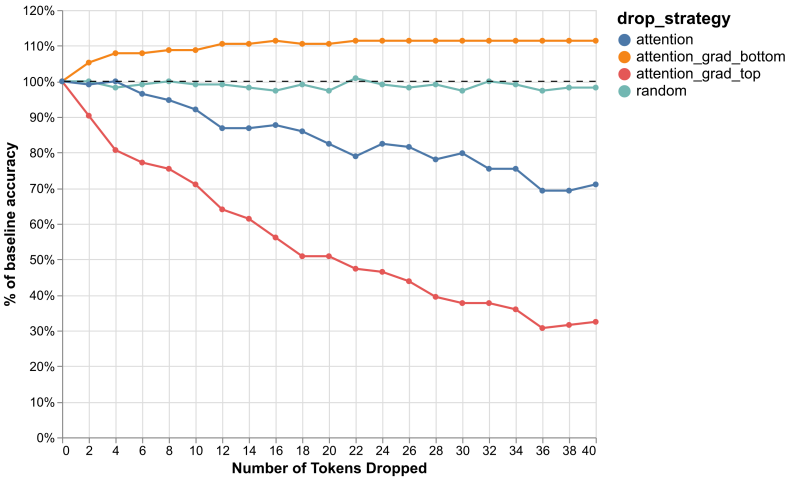
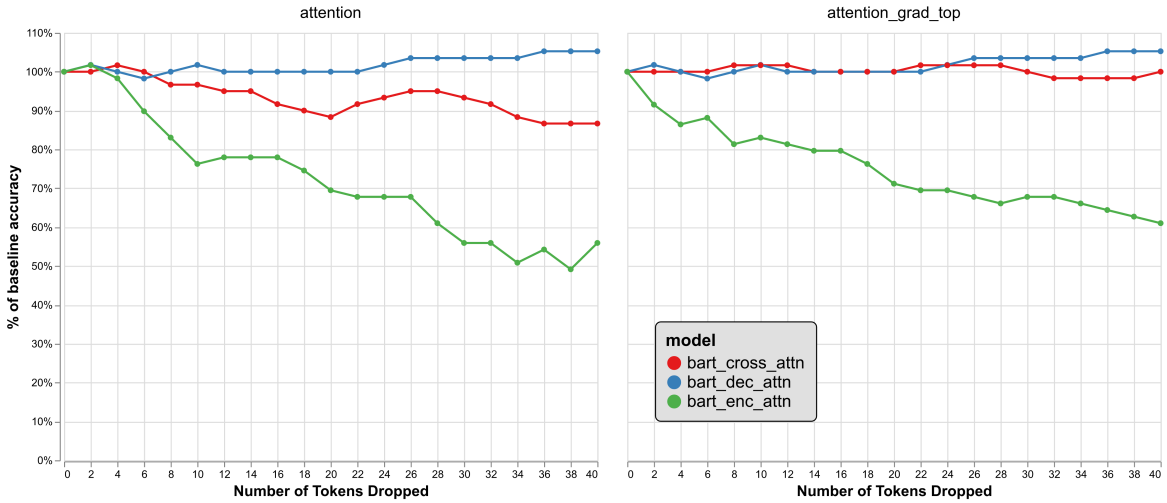


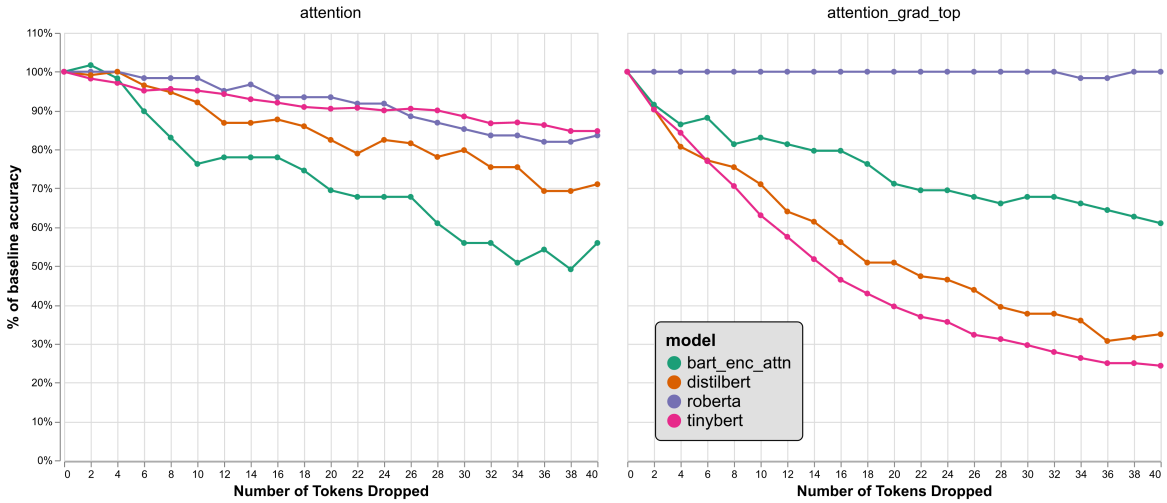
Chart 3 | BART: Comparing Attention Mechanisms: Encoder Attn, Decoder Attn, Cross Attn



Next, we want to compare how attention and gradient-attention based strategies work for BART, (an encoder-decoder model). We want to see which attention mechanism is correlated with token importance. On chart 3, we compare Encoder attention, Decoder attention, and Cross attention mechanisms for attention-based and gradient-attention based occlusion tests. As the graph shows, it seems like only the encoder-attention weights based strategy is associated with a drop in model accuracy: dropping 40 tokens leads to 40-50% reduction in classification accuracy. Dropping tokens according to Decoder and Cross attention weights (both

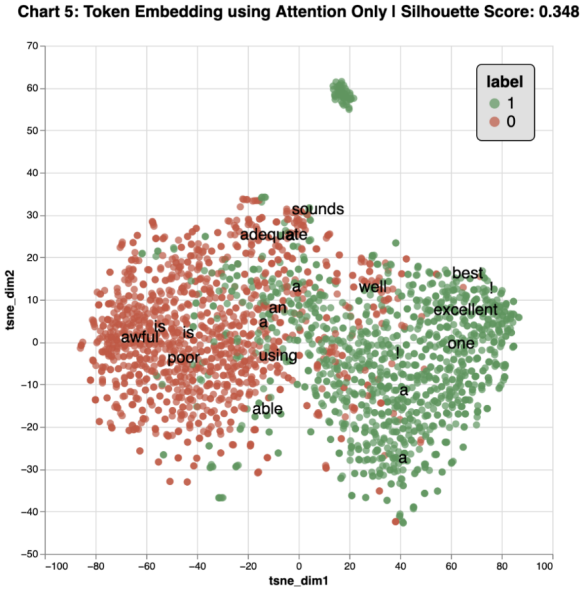
absolute and gradient-based) doesn't seem to decrease the model performance. Also note that non-gradient based encoder attention seems to pick up more important tokens than the gradient-based one, as seen in the sharper accuracy drop of the green line on the left. This finding tells us that we can only rely on the encoder attention in encoder-decoder models such as BART to determine token importance. This finding is intuitive: decoder attention is used to generate text and/or predictions and attends to the model output only, while encoder attention block actually attends to the input, picking on the most relevant features that are related to the task at hand. It is interesting to see that cross-attention doesn't correlate with token importance, even though it is supposed to be connecting the input representation with the output representation.

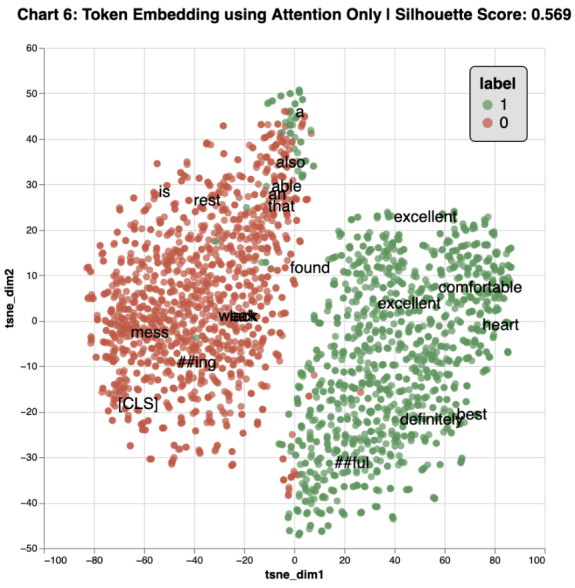
Chart 4 | BERT & BART: Attention vs Grad Attention



Finally, chart 4 compares four different models: 3 encoder-based (DistilBERT, RoBERTa, and TinyBERT) and 1 encoder-decoder one (BART, using encoder-attention block only). Gradient attention occlusion strategy leads to the sharpest drops in accuracy for DistilBERT and TinyBERT models with 50% drops after just 12 tokens. Surprisingly, BART has the best performance in absolute attention-based occlusion (left graph) – it is the only model that gets to 50% drop in accuracy within the top 40 token occlusion. Note that RoBERTa doesn't seem to be sensitive to gradient-based attention – we believe that this is most likely due to the error in how the gradient backprop was set up for that particular model on Huggingface, and the follow up analysis is needed to investigate this further. In conclusion, it appears that gradient-based attention works based on encoder-only models to determine the most important features, and absolute attention strategy works best for encoder-decoder architecture.

4.2 Consistency Test Results





We also conducted a consistency test for the DistilBERT model only to see how well the top tokens are separated in the latent space. Chart 5 shows the 2-dimensional projection of the last hidden layer for top 5 tokens from 100 samples: red ones are the ones associated with the negative reviews and the green ones with the positive ones. The silhouette score for these two clusters is 0.348 indicating the positive and negative clusters are relatively well separated, but there is still some overlap between them. Note that because we are using the last hidden layer of the transformer blocks, this embedding is context aware: the same word can appear in both the positive cluster and the negative cluster but have very different embeddings (e.g., see the "a" token or the "##ful" which can be both part of "awful" and "wonderful"). Chart 6 shows the same contextual embeddings but for the top words that were identified by the gradient-based attention. The silhouette score is much higher at 0.569 and visually the positive and negative clusters are much better separated. These results confirm what the faithfulness test showed for the DistilBERT model: gradient-based attention for feature importance gives better, more robust results than pure attention-based one. A follow up analysis would be needed to see which method is preferable for BART model: if consistency test results follow the faithfulness test obese, then absolute attention-based tokens should have a higher silhouette score.

4.3 Discussion

Our experiment results followed the findings in the original paper (Liu et al., 2021): gradient-based attention methods do seem to be better at picking out the most important features when it comes to using decoder BERT-based models for classification tasks. Both faithfulness and consistency tests confirmed this. Additionally, we discovered that this does not necessarily hold true for encoder-decoder models such as BART: attention-based occlusion proved to be stronger than the gradient-attention one. We also found that the encoder attention block is the one to be used for feature importance, not the decoder or the cross attention

ones. Our findings confirm that Attention is, indeed, *all* we need: it is a powerful tool that can be used for model interpretability and feature importance extraction. We can envision how this methodology can go beyond binary classification and even beyond the NLP tasks. Further experiments would be needed, but we believe that our open code repository allows for easy adjustments to new tasks and can be used as a feature extractor library in the future.

4.4 Limitations and Future Work

In this project, we have replicated the results from Arras, Montavon, Müller, & Samek (2017) and provided a codebase to apply AGrad on both encoder-only and encoder-decoder models. That being said, there are limitations in both the analysis we have done and the approaches presented in the paper. A major limitation in our study was we only used the IMDb dataset whereas the original paper used multiple different datasets in their experiments. A major limitation of the paper itself is that AGrad only works when we have a binary prediction problem. A future direction for research on the topic could be incorporating ways to apply this method to multi-classification problems and other text generation problems.

5 Citations

Arras, L., Montavon, G., Müller, K. R., & Samek, W. (2017). Explaining recurrent neural network predictions in sentiment analysis. arXiv preprint arXiv:1706.07206.

Bhattacharyya, M., Miller, V. M., Bhattacharyya, D., Miller, L. E., & Miller, V. (2023). High Rates of Fabricated and Inaccurate References in ChatGPT-Generated Medical Content. Cureus, 15(5).

Jain, S., & Wallace, B. C. (2019). Attention is not explanation. arXiv preprint arXiv:1902.10186.

Liu, S., Le, F., Chakraborty, S., & Abdelzaher, T. (2021, December). On exploring attention-based explanation for transformer models in text classification. In 2021 IEEE International Conference on Big Data (Big Data) (pp. 1193-1203). IEEE.

Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011, June). Learning word vectors for sentiment analysis. In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies (pp. 142-150).

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

Zhang, Y., Li, Y., Cui, L., Cai, D., Liu, L., Fu, T., ... & Shi, S. (2023). Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. arXiv preprint arXiv:2309.01219.

Zhao, H., Chen, H., Yang, F., Liu, N., Deng, H., Cai, H., ... & Du, M. (2023). Explainability for large language models: A survey. arXiv preprint arXiv:2309.01029.