How Does BERT Rerank Passages? An Attribution Analysis with Information Bottlenecks

Zhiying Jiang, Raphael Tang, Ji Xin and **Jimmy Lin**

David R. Cheriton School of Computer Science
University of Waterloo

{zhiying.jiang, r33tang, ji.xin, jimmylin}@uwaterloo.ca

Abstract

Fine-tuned pre-trained transformers achieve the state of the art in passage reranking. Unfortunately, how they make their predictions remains vastly unexplained, especially at the end-to-end, input-to-output level. Little known is how tokens, layers, and passages precisely contribute to the final prediction. In this paper, we address this gap by leveraging the recently developed information bottlenecks for attribution (IBA) framework. On BERT-based models for passage reranking, we quantitatively demonstrate the framework's veracity in extracting attribution maps, from which we perform detailed, token-wise analysis about how predictions are made. Overall, we find that BERT still cares about exact token matching for reranking; the `[CLS]` token mainly gathers information for predictions at the last layer; top-ranked passages are robust to token removal; and BERT fine-tuned on MSMARCO has positional bias towards the start of the passage.

1 Introduction

Pre-trained language models like BERT (Devlin et al., 2019) have achieved prominent improvements in both information retrieval (IR) and natural language processing (NLP). Concurrently, researchers have raised wide awareness about the difficulty of explaining such deep learning models (Guidotti et al., 2018; Robnik-Šikonja and Bohanec, 2018; Fong and Vedaldi, 2017). Recently, many papers scrutinize BERT's behaviors in various tasks (van Aken et al., 2019; Clark et al., 2019; Tenney et al., 2019; Qiao et al., 2019; MacAvaney et al., 2020). When it comes to token-wise analysis, most of the work study intra-layer self-attention and how it relates to various linguistic characteristics. Although these analyses yield unique insights on layer-local behavior across pairs of tokens, they do not take a global perspective of how token-wise representations exactly relate to the prediction. This is crucial for answering a fundamental question in interpretability: what hidden features and tokens contribute the most to the prediction?

To faithfully compute such feature–prediction attribution maps, Schulz et al. (2020) and Jiang et al. (2020) propose to apply information bottlenecks. In this paper, we leverage this model-agnostic method to analyze passage reranking for pre-trained transformers. We first introduce the information bottleneck for attribution (IBA) method (Schulz et al., 2020) in general and elaborate its use in interpreting passage reranking. Afterwards, we compare it with two other widely adopted attribution methods to demonstrate its credibility and justify our choice. We then carry out detailed analyses on the inner mechanisms of passage reranking.

BERT reranking (Nogueira and Cho, 2019) starts a new chapter in information retrieval, as it combines the dual advantages of the speed of sparse representation (BM25) and the deep contextualization of dense representation. To be specific, given a query q, BM25 returns top-1,000 passages D. The label r is 1 if a passage $d \in D$ is relevant to q, and 0 otherwise. For BERT, the input is `[CLS] q [SEP] d [SEP]`, and the output label is r. After fine-tuning, we rerank D based on the output probabilities of relevance. This setting is different from most NLP tasks, where positive and negative labels are provided by the dataset, and only one pair of (input, probability) is required for the final output.

We use IBA to generate attribution maps for BERT-large (Devlin et al., 2019) fine-tuned on the MSMARCO dataset (Bajaj et al., 2016) in this paper. With the attribution maps, we investigate the following questions:

Q1. What are the similarities and differences between BERT and BM25?
For the two-stage pipeline, we wonder how BERT's ranking mechanism is similar to BM25 and what it provides that BM25 doesn't. Through cross-passage examination, we find that BERT still regards lexical matching as important to some extent,

similar to BM25. BERT, furthermore, manages to capture deeper-contextualized relationships between the query and the relevant passage.

Q2. How do special tokens contribute to reranking across layers?
In BERT, only the [CLS] token is designed to factor into prediction. Then how do those special tokens collect information across layers to capture a contextualized relationship? We find that, different from what attention analyses show, [CLS] starts to gather the evidence for prediction primarily after layer 16, especially in layer 24.

Q3. How robust is the top-ranked passage?
One of the special settings of ranking is that we do not care about the absolute score, as long as the relevant passage ranks higher than irrelevant ones. We conduct experiments of token removal for the top-1 positive passage to test the robustness. We find that we can truncate up to 22.5% tokens on average, given reasonable attribution scores, of the top-ranked passage without affecting its order.

Q4. Does BERT have positional bias?
We then look deeper into what makes those passages rank higher. We find that BERT, after fine-tuned on MSMARCO, prefers those passages with inverted pyramid structure—that is, passages that put important information at the start. We further confirm that it has positional bias towards the start of the passage through various experiments.

2 Related Work

Generally speaking, interpretability methods are either model specific, applying to only a single architectural family, or model agnostic, covering a broad spectrum of supervised models. Since pretrained transformers represent the state of the art in NLP, for model-specific techniques we discuss those for BERT, the prototypical, most-interpreted transformer model. As this work specifically explores passage reranking, we also provide the necessary literature about recent progress.

2.1 BERT specific

A number of works investigate the inner mechanisms of BERT. Kovaleva et al. (2019); Clark et al. (2019) carefully analyze BERT's attention heads, noting positive correlation between attention heads and linguistic features, as well as special tokens.

Looking at attention, Voita et al. (2018) find that BERT captures anaphora and dependence on position and length in machine translation. Pointing out some shortfalls of these papers, Jain and Wallace (2019); Brunner et al. (2019); Serrano and Smith (2019) argue that attentions often do not reflect how models make predictions.

Another line of work analyzing BERT use probing classifier to draw the connection between vector representation and specific linguistic knowledge (Tenney et al., 2019; Hewitt and Manning, 2019; Liu et al., 2019). Rogers et al. (2020) provide a thorough literature survey about what we already know about how BERT works and they've found different probing methods sometimes lead to contradictory interpretations. A direct remedy is to look into what BERT looks at during inference time (i.e. identify important features for prediction, also known as "attribution methods" in general). That's where our work focuses on.

2.2 Attribution maps

Although more commonly applied to convolutional neural networks in image classification, most attribution methods are model agnostic. They aim to assign weights to input features according to how the model makes predictions, with higher weights corresponding to greater contributions.

The most prevalent methods are gradient-based. Intuitively, gradients reflect how small changes in the input affect the final prediction to some extent. But previous work shows that raw gradients are noisy and limited to capturing only the local "importance" (Smilkov et al., 2017). To remedy this, some of them (Sundararajan et al., 2017; Smilkov et al., 2017) incorporate global importance to mitigate this problem, while others (Binder et al., 2016; Shrikumar et al., 2017; Kindermans et al., 2018) modify or extend the back-propagation algorithms directly to emphasize positive contributions with regard to prediction. However, Sixt et al. (2020) show that most of the modified back-propagation methods fail a basic sanity check: invariance to parameter randomization and label randomization.

LIME (Ribeiro et al., 2016) is not even limited to differentiable models. They use interpretable models like decision trees to approximate deep neural networks, and thus can theoretically interpret any classifier. However, empirically, LIME's high demand on memory may worsen its quality compared to other methods, as we will see in the later section.

Information-theoretic methods are often unconstrained by tasks and models as well, while additionally providing a unified view of how infor-

mation flows across models. Guan et al. (2019) use mutual information to estimate tokens importance across layers but don't provide quantitative evaluation. Bang et al. (2019) also take advantage of information bottlenecks to interpret predictions, but they restrict the information by sampling tokens, which doesn't generate a complete attribution map for every token and limits the interpretation to be token-wise only. More recently, Schulz et al. (2020) propose the information bottleneck method for attribution, which empirically achieves the best result on multiple evaluation metrics in interpreting images. Jiang et al. (2020) further leverage this method in NLP and also surpass other model-agnostic methods on multiple datasets.

2.3 Neural IR

BERT is a game changer for information retrieval. Lin et al. (2020) even separate neural reranking techniques into "pre-BERT" and "post-BERT" eras. Nogueira and Cho (2019) start the post-BERT era by proposing a two-stage pipeline, using sparse representations like BM25 to generate candidates and then neural models like BERT to rerank them. More recent work explores merging the two-stage pipeline into an end-to-end dense retrieval, like DPR (Karpukhin et al., 2020), which still use BERT as the basic building block for neural information retrieval. Therefore, understanding BERT's behavior for reranking in the original setting still helps.

Toward this, a few previous works specifically analyze BERT for reranking: Qiao et al. (2019) analyze attention to see how BERT attends to stop words and regular words across layers. MacAvaney et al. (2020) does a more thorough study of various reranking models, using carefully designed textual manipulation methods. Different from them, we use a model-agnostic method to generate a token-wise attribution map, as it provides us with the flexibility to carry out a layer-wise analysis. Besides, to the best of our knowledge, no previous work has done a cross-passage analysis to see patterns across the ranks of different passages.

3 IBA Method

3.1 General Introduction to IBA

The starting point of IBA is to keep only the feature-level information that's most helpful toward the final prediction. After a given layer in the target neural network, we insert an information bottleneck, which restricts the total amount of information in the representation. Simultaneously, we maximize the amount of information important toward the final prediction.

To be concrete, given an input $\mathbf{X} \in \mathbb{R}^N$ and output $\mathbf{Y} \in \mathbb{R}^M$, an information bottleneck is an intermediate representation $\mathbf{T}$ that maximizes the following function:

$$\mathrm{I}(\mathbf{Y};\mathbf{T}) - \beta \cdot \mathrm{I}(\mathbf{X};\mathbf{T}), \tag{1}$$

where I denotes mutual information, and β is a hyperparameter that balances the trade-off between reconstruction $\mathrm{I}(\mathbf{Y};\mathbf{T})$ and information restriction $\mathrm{I}(\mathbf{X};\mathbf{T})$. A larger β means a narrower bottleneck and hence less information through the network.

Intuitively, maximizing $\mathrm{I}(\mathbf{Y};\mathbf{T})$ keeps information for accurate prediction, while minimizing $\mathrm{I}(\mathbf{X};\mathbf{T})$ filters out unnecessary information. To obtain the condensed representation $\mathbf{T}$, we construct a loss function based on the intuition above. For $\mathrm{I}(\mathbf{Y};\mathbf{T})$, we can directly use the cross-entropy loss for classification $\mathcal{L}_{\mathrm{CE}}$. For $\mathrm{I}(\mathbf{X};\mathbf{T})$, we will derive it step by step below.

Formally, for a given layer l of a model, let $\mathbf{X} = f_l(\mathbf{H})$, meaning the output of each layer, where $\mathbf{H}$ is the input of layer l. We then restrict the information by injecting noise ϵ into input $\mathbf{X}$, which results in

$$\mathbf{T} = \boldsymbol{\mu} \odot \mathbf{X} + (\mathbf{1} - \boldsymbol{\mu}) \odot \epsilon, \tag{2}$$

where $\odot$ refers to element-wise multiplication, $\mathbf{1}$ is an all-one vector, $\boldsymbol{\mu} \in \mathbb{R}^N$ is the weighting parameter controlling the balance between signal and noise whose dimension is the same as input $\mathbf{X}$. For each dimension, we constrain $\boldsymbol{\mu}_i \in [0, 1]$, setting $\boldsymbol{\mu}_i = \sigma(\boldsymbol{\alpha}_i)$, where σ is the sigmoid function, to simplify the training process. And $\boldsymbol{\alpha}_i$ is the parameter that we are learning for each dimension. From Eq. 2 we can see that when $\boldsymbol{\mu} = 0$, that is, all the information is discarded; only noise is passed through ($\mathbf{T} = \epsilon$). Taking that into account, in order to preserve the magnitude of the input for the next layer, it's desirable to keep ϵ the same mean and variance as $\mathbf{X}$. Therefore, we have $\epsilon \sim \mathcal{N}(\mu_{\mathbf{X}}, \sigma^2_{\mathbf{X}})$. This condition doesn't always ensure $\mathbf{T}$ to have exactly the same mean and covariance with $\mathbf{X}$ though. And the model is recovered after training the bottleneck to ensure the covariance shift doesn't affect interpreting subsequent instances.

After obtaining $\mathbf{T}$, we can now evaluate $\mathrm{I}(\mathbf{X};\mathbf{T})$. By definition,

$$\mathrm{I}(\mathbf{X};\mathbf{T}) = \mathbb{E}_{\mathbf{X}}[D_{KL}[P(\mathbf{T}|\mathbf{X})\|P(\mathbf{T})]], \tag{3}$$

where D_{KL} means Kullback–Leibler (KL) divergence and $P(\mathbf{T}|\mathbf{X})$, $P(\mathbf{T})$ are probability distributions. As $P(\mathbf{T}) = \int P(\mathbf{T}|\mathbf{X})P(\mathbf{X})\mathrm{d}\mathbf{X}$, there is no analytical expression for $P(\mathbf{T})$. We use the standard variational approximation $Q(\mathbf{T}) = \mathcal{N}(\mu_{\mathbf{X}}, \sigma^2_{\mathbf{X}})$ to substitute $P(\mathbf{T})$. Note that we estimate each $\mu_{\mathbf{X}}$ and $\sigma_{\mathbf{X}}$ empirically. The variational approximation assumes that each dimension is distributed independently and normally. The normal distribution comes from the observation that activations after linear and convolutional layers tend to be Gaussian-like (Klambauer et al., 2017; Borovykh, 2018). The independence assumption, on the other hand, does not hold in general, but it just overestimates the mutual information, so it gives an upper bound of mutual information between $\mathbf{X}$ and $\mathbf{T}$:

$$\mathrm{I}(\mathbf{X};\mathbf{T}) \leq \mathbb{E}_{\mathbf{X}}[D_{KL}[P(\mathbf{T}|\mathbf{X})\|Q(\mathbf{T})]]. \quad (4)$$

Proof can be found in Appendix A.

An upper bound means when the approximation between $\mathbf{X}$ and $\mathbf{T}$ is 0, their mutual information is guaranteed to be 0, which is a desired property, as we expect $\mathrm{I}(\mathbf{X}, \mathbf{T})$ to be small.

Combining Eq. 4 with the cross entropy for classification, we have our loss function:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \beta \cdot \mathbb{E}_{\mathbf{X}}[D_{KL}[P(\mathbf{T}|\mathbf{X})\|Q(\mathbf{T})]]. \quad (5)$$

Note that we negate the sign for minimization. β can be viewed as the gate controlling the relative importance between the two loss components. After getting $\mathbf{T}$ from above, we calculate how much information $\mathbf{T}$ still contains about $\mathbf{X}$ using Eq. 3. This gives us the contribution of each dimension in every token. In order to generate the token-wise attribution map, we sum over the feature–token axis to obtain an attribution score for each token.

3.2 IBA for Passage Reranking Analysis

The procedure of using BERT to rerank passages (Nogueira and Cho, 2019) can be characterized as follows: Given query q, and a list of passages D, $d \in D$ is returned by BM25. BERT then assigns the relevance score $R(q, d)$, the logits for the probability that the passage is regarded as relevant, to each pair of q and d. $\mathcal{L}_{\mathrm{CE}}$ in this case is the same as the cross entropy in Nogueira and Cho (2019). We use BERT-large model fine-tuned on MSMARCO dataset for experiments. In order to get $\mathbf{T}$, we optimize the learning parameter $\boldsymbol{\alpha}$. At the beginning of the training, we start with $\mathbf{T} \approx \mathbf{X}$ to keep the information of $\mathbf{X}$ in $\mathbf{T}$ as much as possible. Thus, we initialize $\boldsymbol{\alpha}_j = 5$ for each dimension j as it results in $\boldsymbol{\mu}_j = 0.993$, which is close to 1 as desired. During optimization, we fix the training steps to 10 and repeat a sample 10 times to inject different noise, which altogether requires 100 total steps to generate an attribution map for a single instance. Another important hyperparameter is β. We empirically pick $\beta \approx 10 \times \frac{\mathcal{L}_{\mathrm{CE}}}{\mathcal{L}_{\mathrm{IB}}}$, as suggested in (Jiang et al., 2020).

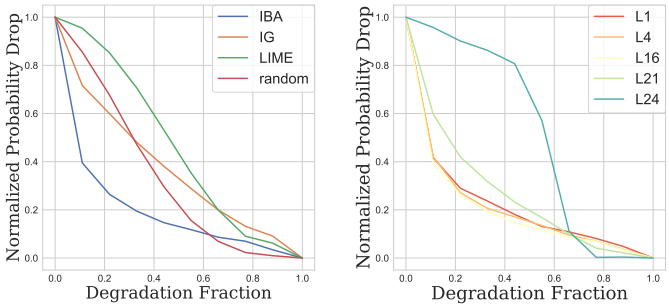
To compare the effectiveness of IBA with other attribution methods, we carry out a degradation test. The essential idea of a degradation test is to remove the most important k% tokens, excluding special tokens, identified by different attribution methods and measure the drop of the probability with respect to the given label.

The initial value of k is 11 and we increase k until all the tokens are removed, shown as the x-axis in Figure 1a. y-axis means the normalized average probability drop after removing a certain percentage of tokens: $\frac{\tilde{p}(y|x')-m}{o-m}$ where x' represents input with certain tokens removed, o is the original probability before tokens removal, and m is the minimum of the fully degraded instance's probability across all attribution methods. We conduct the experiment across the entire MSMARCO dev set (6980 queries).

We compare the result with two other popular model-agnostic attribution methods, LIME (Ribeiro et al., 2016) and Integrated Gradients (IG) (Sundararajan et al., 2017), each representing a different category of attribution methods: LIME uses interpretable models like decision trees and linear models to approximate the black box, while IG is a variation of using the gradient of the predicted output with respect to given input features. To provide a simple baseline, we also compare the result with "Random," where tokens are removed randomly. We expect a better attribution method will have a steeper slope, meaning removing important tokens identified by the method significantly deteriorates the performance. As shown in Figure 1, IBA outperforms all other three methods with a 61.3% probability drop comparing with second-placed IG, which makes for a 29.0% drop. The absolute probability drop value can be seen in Table 1.

4 Experiments and Analyses

Figure 1c shows an example of important tokens in the query and the passage identified by IBA. Aside from token matching like "pH" and "water", deeper semantic relatedness like "acid" and "neutral" are



(a) Comparison among attribution (b) Comparison among layers

what is the ph of water mean

a ph number measures from 0 to ##14 how acidic or al ##kali ##ne a liquid is - - anything above 7 is al ##kali ##ne and anything below 7 is acid . water has a ph level of 7 - - it ' s neutral , meaning it has the same amount of acids and al ##kali ##s , which balance each other out .

(c) Qualitative example

Figure 1: Degradation test on MSMARCO with BERT-large, followed by an attribution map example.

Methods	$o - \bar{p}(y\|x')$
Random	0.135
LIME	0.043
IG	0.265
IBA	**0.565**

Table 1: Probability drop after removing first k% important tokens identified by these methods.

also stressed. More examples and analysis are shown in Appendix G.

Given the attribution maps, we are now able to study which tokens BERT looks at for reranking. To be specific, we exploit IBA to extract the top-20 most important tokens $\mathbb{M}$ for each $(q, d), q \in Q, d \in D$, where Q and D represent the query list and the passage list. We carry out our experiment under two different settings:

1. Q consists of 1,000 randomly selected queries from the entire MSMARCO passage reranking dev set. D is composed of the human annotated relevant passages. We then apply IBA to all 24 layers to get top-20 tokens $\mathbb{M}$ for each (q, d).

2. Q consists of 105 queries from a subset of the MSMARCO passage reranking dev set, provided by Pyserini (Lin et al., 2021). D comprises top-50 passages that BERT-large retrieves for each query. For these experiments, we fix the layer l that we insert the information bottleneck after.

For setting 1, we aim at cross-layer analysis for relevant passages. Specifically, we identify if lower layers show different focus from higher layers. This setting is similar to GLUE-like (Wang et al., 2018) classification tasks where we want to find general patterns about BERT. The reason for using the top 20 is that, in our sampled instances, the average tokenized query length is 9.2, and we also want to see the emphasized tokens in passages. For setting 2, we perform cross-passage analysis to investigate different patterns between higher-ranked passages and lower-ranked passages. The choice of the top-50 cutoff is due to frugality: the recall@50 (0.817) is comparable to the recall@1000 (0.848), with much less computation.

4.1 Passage-Level Patterns

It's well known that two-stage ranking pipelines use both exact token matching and semantic relatedness (Lin et al., 2020). As BM25 estimates relevance purely by lexical matching, we wonder if BERT still relies on exact match and what else BERT provides.

Q1. What are the similarities and differences between BERT and BM25?

To answer this question, we first study the correlation between higher ranking scores and higher lexical matching between queries and passages.

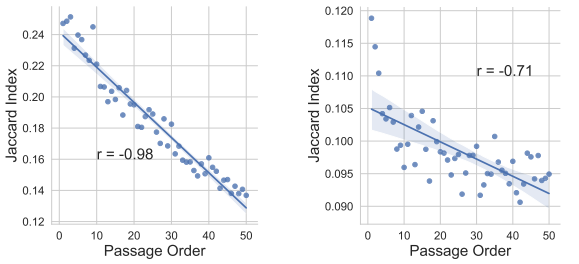
To measure the degree of lexical matching, we use the Jaccard index under experimental setting 2: $\mathbf{J} = \frac{|u_i \cap v_j|}{|u_i \cup v_j|}, \{u_i, v_j | u_i \in q \cap \mathbb{M}, v_j \in d \cap \mathbb{M}\}$, where $i \in [1, |q|], j \in [1, |d|]$, remember that $\mathbb{M}$ is the top-20 tokens extracted by IBA.

For each query q, we calculate the Jaccard index for every passage d_i in the top-50 passages. We then average them across all queries. We choose to insert the information bottleneck after layer 16, as it is the most informative one according to our degradation test.

As we see in Figure 2a, the Jaccard index decreases as the rank of the passages becomes lower. In general, the higher[1] the rank is, the more overlapped the important tokens between the query and

[1]"Higher" rank actually means lower order: passages with order 1 have higher rank than passages with order 2, etc.

passages are. We also calculate Spearman's correlation (Spearman, 1961) r_s to gauge the degree of monotonic association. We find that $r_s = -0.98$, indicating a strong monotonic relation between the Jaccard index and passages order. Does this correlation hold among all tokens between the query and passage? Figure 2b shows the Jaccard index $\mathbf{J}' = \frac{|u'_i \cap v'_j|}{|u'_i \cup v'_j|}$, $\{u'_i, v'_j | u'_i \in q, v'_j \in d\}$ across passages. We see it shows a similar trend to $\mathbf{J}$, confirming that even if BM25 returns passages that have higher lexical matching with query, token matching between queries and passages still plays an important role when BERT is reranking. But using all of the tokens between the query and the passage obtains a correlation coefficient of $r_s = -0.71$, which is lower than using important tokens only. We argue that it's because IBA interprets in a way that's more aligned with the specific tokens that BERT looks at when reranking.



(a) Important tokens. (b) Whole query and passage.

Figure 2: Across top-50 passage analysis.

We further investigate what BERT provides that BM25 doesn't. Specifically, we look into what d_{BERT} gets right but d_{BM25} gets wrong. We notice that BERT captures more contextualized relevance between the query and passage, while the BM25-returned answer has more "superficial" relevance - d_{BM25} seems to talk about the topic but doesn't really answer the question. The example shown in Figure 3 demonstrates that passage returned by BM25 seems highly related to the topic—"cognitive impairment" but instead of explaining what the goal is, it is explaining what "cognitive impairment"'s definition is. On the contrary, BERT not only returns the passage related to "cognitive impairment" but also the goal. More discussions about semantic similarity are in Appendix F.

4.2 Layer-Level Patterns

Downstream tasks often rely on BERT's [CLS] vector at the last layer as input, and that's also true for

Query: what is the goal for the child with a cognitive impairment

BM25 ranked 1st: A cognitive impairment is a condition where your child has some problems with ability to think and learn. Children with a cognitive impairment often have trouble with such school subjects as math and reading. cognitive impairment is a condition where your child has some problems with ability to think and learn. Children with a cognitive impairment often have trouble with such school subjects as math and reading.

BERT ranked 1st: Promoting optimum development. The goal for children with cognitive impairment is the promotion of optimum social, physical, cognitive, and adaptive development as individuals within a family and community. Vocational skills are only one part of that goal. The focus must also be on the family and other aspects of development.

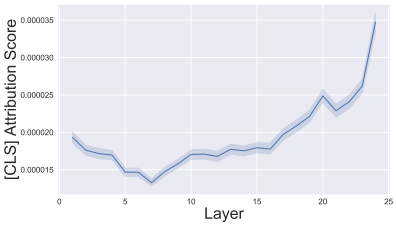
Figure 3: Top-1 passage by BM25 and BERT.

reranking. It's intriguing to know the layer at which [CLS] starts to learn the relevance. Clark et al. (2019) thoroughly analyze BERT's self-attention mechanism for each layer. While they provide insight into how tokens attend to one another, the attention weights themselves often do *not* correlate with measures of feature importance (Jain and Wallace, 2019).

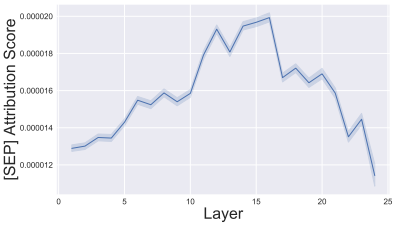
Q2. How do special tokens contribute to reranking across layers?

We insert an information bottleneck after each layer for 24-layer attribution maps. First we inspect how the [CLS] token gets emphasized across the layers. Figure 4a shows the attribution score across 24 layers in experimental setting 1, with 95% confidence intervals. Note that the score is normalized between 0 to 1 for each token but it doesn't add up to 1 for each instance. We further normalize the attribution score by dividing the sum of the attribution scores at each layer to account for different layers' scale. As we can see in the plot, the attribution score for [CLS] across layers first decreases from layers 1–7, then goes up and fluctuates between layers 7–16, until finally increasing from layer 16 to 24. This differs from what attention analysis reveals in Kovaleva et al. (2019) and Clark et al. (2019), where they demonstrate that attention heads attend to [CLS] in earlier layers but attend to [SEP] in later layers. It's not contradictory, though, because we inspect feature importance with respect to the predicted output. Since [CLS] at the final layer is treated as a summary representation for the whole sentence to perform classification, it's intuitive that [CLS] is regarded as an important feature in the final layers.

What about the [SEP] tokens? Figure 4b shows the attribution score averaged between the two present [SEP] tokens—recall that BERT inserts two for every input. They become increasingly important with a certain amount of fluctuation from



(a) CLS attribution score across layers.



(b) SEP attribution score across layers.

Stats	Query	passage
Avg. Length	9.2	74.8
Med. Length	9.0	69.0
Min. Length	4.0	16.0
Max. Length	44.0	214.0

(c) Stats of 1,000 relevant (q, d).

Figure 4: Normalized attribution scores of special tokens across layers; dataset statistics under setting 1.

layer 1 to layer 16, after which [SEP]'s attribution scores drop, around the point where BERT starts to emphasize [CLS]. We combine the two [SEP] tokens because we find that both of them behave similarly, with the first [SEP] having a slightly higher attribution score. The possible explanation is that the first [SEP] is also responsible for identifying the boundary between the query and the passage, thus more important for reranking than the final [SEP]. Plots for separate [SEP]'s scores and weights can be seen in Appendix D.

Combining the above plots and the degradation tests across layers in Figure 1b, we conjecture that the [CLS] token initially serves as a classification prior to condition the tokens in the early layers (1–7) with [SEP] increasing participation. Then, BERT gathers more general syntactic information (Hewitt and Manning, 2019), until layer 16, after which the [CLS] token slowly aggregates class-specific information and at layer 24 becomes the most important token for classification. Figure 1b (the full 24-layer degradation test is shown in Appendix B) echos the findings from previous work (Liu et al., 2019), demonstrating that the middle layers are the most informative ones for prediction. To be exact, layer 16 ($\frac{2}{3}$ of the total number of layers) is the most informative one in our experiment with BERT-large, the same fraction as what Jiang et al. (2020) find with BERT-base.

	# required	% required
(q, d_1)	8.9	10.39
(d_1)	18.52	22.49

Table 2: Truncation test on top-1 pair/passage.

4.3 Truncation Test

Different from other downstream tasks, passage reranking usually involves scores for 1,000 passages to generate the final result. Instead of absolute scores for passages, we only care if relevant passages have higher scores than irrelevant passages. Recent work (Bai et al., 2020; Formal et al., 2021) starts to incorporate sparse mechanisms (adding and removing tokens) in order to elevate efficiency for the first-stage ranking. We ask how token removal affects those true positive passages.

Q3. How robust is the top-ranked passage?

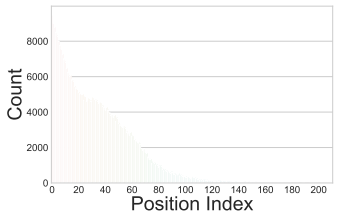
Specifically, we want to know how many unimportant tokens we can remove before the top-1 passage falls to second place. Once again, we use the IBA-generated attribution map and then remove those tokens with lowest attribution scores, until the ranking score for the top-1 passage drops below the second one. As in the reranking setting, the input is always a query–passage pair (q, d), and we have two experimental settings: (1) removing tokens that appear in both q and d; and (2) removing tokens that appear in only d. We include the result under both settings and report the truncated number, as well as the percentage needed in Table 2.

Surprisingly, even if BERT assigns an extreme score to the passage, making the score close to one another (Qiao et al., 2019), it still takes up to 22.5% tokens on average for top-1 passage to downgrade to the second place.

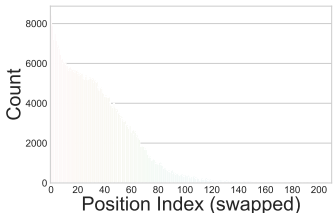
Obviously, removing tokens from the query quickly deteriorates the ranking score. Passages-only seem to have more redundant tokens that can be safely removed, even though sentences in the passage will become incomplete and broken after token removal. Note that this experiment removes only tokens of the top-1 passage. We attach the comprehensive results and discussion of truncating tokens in all passages in Appendix G.

4.4 Positional Bias

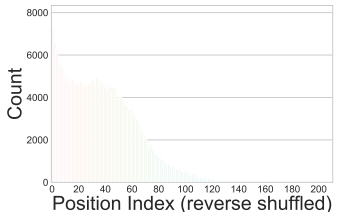
MacAvaney et al. (2020) observe that changing the sentence order has negative effects when reranking with BERT. They suggest that either the model is affected by the discourse-level signal (e.g., topics discussed earlier in passages) or the model encodes positional bias.



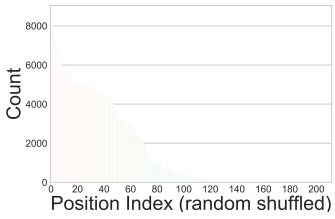
(a) Original sentence order.



(b) First two sentences swapped.



(c) Reversed sentence order.



(d) Randomized sentence order.

Figure 5: Position index of important tokens in passages; stats of tested passages.

	Precision@1	MRR	MRR@10	Recall@3	Recall@50	Recall 1000
Original	0.276	0.411	0.403	0.427	0.817	0.848
Swapped	0.219	0.362	0.352	0.408	0.813	0.848
Randomized	0.200	0.343	0.332	0.384	0.803	0.848
Reversed	0.181	0.332	0.321	0.390	0.803	0.848

Table 3: Reranking metrics after changing sentence order.

Q4. Does BERT have positional bias?

To investigate if BERT has positional bias, we first plot the position index of x_i where $\{x_i | x_i \in \mathbb{M} \cap d\}$. Specifically, we insert the bottleneck after every layer under experimental setting 1—1000 relevant pairs of (q, d)—and then accumulate the count of each position index for all 24 layers (plots for each layer is also shown in Appendix H). Statistics about randomly selected (q, d) are shown in Figure 4c. As we show in Figure 5a, tokens at the start of passages (e.g., position index from 0 to 20) have significantly higher occurrences than tokens appearing later.

We then conduct three controlled experiments: (1) swapping the first two sentences; (2) reversing the order of all sentences; and (3) randomizing the order of sentences. We maintain the order of within-sentence tokens in order to keep the discourse complete and coherent. The plots are shown in Figure 5. We see that, although changing the order results in more later-appearing tokens emphasized, the start of the passages still have incomparable dominance. To quantify the effect of swapping sentences, randomizing sentences, and reversing sentences, we calculate $p(\text{relevant}|\{\text{original, swap, random, reverse}\})$. We find that $p(\text{relevant}|\text{original}) = 0.939$, $p(\text{relevant}|\text{swap}) = 0.920$, $p(\text{relevant}|\text{random}) = 0.918$, $p(\text{relevant}|\text{reverse}) = 0.897$. The probability drops after every change of sentence order. The more the order changes, the more the probability drops (i.e., the negative effect is reversed order > randomized > swapped). Given that BERT assigns extreme reranking scores to most (q, d) pairs (e.g., scores are mostly either close to 0 or close to 1), it's unclear whether changing the order of sentences affects the final result.

Therefore, we also conduct experiments of changing the order sentence with the subset of the MS-MARCO passage reranking dev set. We present these results in Table 3. Swapping sentences substantially deteriorates the result; randomizing and reversing the sentences further worsens the result.

The above experiments suggest that the sentence order in the passage carries high importance in reranking with BERT. Specifically, passages with the inverted pyramid structure would be preferred, as they present important information at the beginning of the passages. More discussion on positional bias can be found in Appendix E.

5 Conclusions

In this work, we leverage IBA to examine BERT for reranking. We compare ranking mechanisms between BM25 and BERT, finding that BERT still

values token matching, and it also learns deeper relevance between queries and passages. We further analyze special tokens across layers and demonstrate patterns that [CLS] aggregate evidence. We then investigate the robustness of top-ranked passages. Finally, we find that BERT fine-tuned on MSMARCO has positional bias towards the start of the passage. In summary, attribution maps can explain models' predictions and serve well as an observation tool that helps us visualize patterns, resulting in improved hypothesis formulation and experimental design.

Acknowledgements

This research was supported in part by the Canada First Research Excellence Fund and the Natural Sciences and Engineering Research Council (NSERC) of Canada; computational resources were provided by Compute Canada.

References

Yang Bai, Xiaoguang Li, Gang Wang, Chaoliang Zhang, Lifeng Shang, Jun Xu, Zhaowei Wang, Fangshan Wang, and Qun Liu. 2020. SparTerm: Learning term-based sparse representation for fast text retrieval. *arXiv preprint arXiv:2010.00768*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. MS MARCO: A human generated machine reading comprehension dataset. *arXiv:1611.09268*.

Seojin Bang, Pengtao Xie, Heewook Lee, Wei Wu, and Eric Xing. 2019. Explaining a black-box using deep variational information bottleneck approach. *arXiv:1902.06918*.

Alexander Binder, Sebastian Bach, Gregoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. Layer-wise relevance propagation for deep neural network architectures. In *Information science and applications (ICISA) 2016*, pages 913–922. Springer.

Anastasia Borovykh. 2018. A Gaussian process perspective on convolutional neural networks. *arXiv:1810.10798*.

Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2019. On identifiability in transformers. In *International Conference on Learning Representations*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Ruth C. Fong and Andrea Vedaldi. 2017. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2288–2292.

Chaoyu Guan, Xiting Wang, Quanshi Zhang, Runjin Chen, Di He, and Xing Xie. 2019. Towards a deep and unified understanding of deep neural models in NLP. In *International conference on machine learning*, pages 2454–2463. PMLR.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556.

Zhiying Jiang, Raphael Tang, Ji Xin, and Jimmy Lin. 2020. Inserting information bottleneck for attribution in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3850–3857.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Pieter-Jan Kindermans, Kristof T. Schütt, Maximilian Alber, Klaus-Robert Müller, Dumitru Erhan, Been

Kim, and Sven Dähne. 2018. Learning how to explain neural networks: PatternNet and PatternAttribution. In *ICLR (Poster)*.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. 2017. Self-normalizing neural networks. In *Advances in Neural Information Processing systems*.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: An easy-to-use Python toolkit to support replicable ir research with sparse and dense representations. *arXiv:2102.10073*.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2020. Pretrained transformers for text ranking: BERT and beyond. *arXiv:2010.06467*.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094.

Sean MacAvaney, Sergey Feldman, Nazli Goharian, Doug Downey, and Arman Cohan. 2020. ABNIRML: Analyzing the behavior of neural IR models. *arXiv preprint arXiv:2011.00696*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*.

Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the behaviors of BERT in ranking. *arXiv:1904.07531*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 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*.

Marko Robnik-Šikonja and Marko Bohanec. 2018. Perturbation-based explanations of prediction models. In *Human and machine learning*. Springer.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Karl Schulz, Leon Sixt, Federico Tombari, and Tim Landgraf. 2020. Restricting the flow: Information bottlenecks for attribution. In *International Conference on Learning Representations*.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR.

Leon Sixt, Maximilian Granz, and Tim Landgraf. 2020. When explanations lie: Why many modified BP attributions fail. In *International Conference on Machine Learning*, pages 9046–9057. PMLR.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. SmoothGrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

Charles Spearman. 1961. The proof and measurement of association between two things.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A Gers. 2019. How does BERT answer questions? a layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1823–1832.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

A Proof of Variational Upper Bound

$$
\begin{aligned}
\mathrm{I}(\mathbf{X};\mathbf{T}) &= \mathbb{E}_{\mathbf{X}}[D_{KL}[P(\mathbf{T}|\mathbf{X})\|P(\mathbf{T})]] \\
&= \int_{\mathbf{X}} p(x)\Big(\int_{\mathbf{T}} p(t|x)\log\frac{p(t|x)}{p(t)}dt\Big)dx \\
&= \int_{\mathbf{X}}\int_{\mathbf{T}} p(x,t)\log\frac{p(t|x)}{p(t)}\frac{q(t)}{q(t)}dtdx \\
&= \int_{\mathbf{X}}\int_{\mathbf{T}} p(x,t)\log\frac{p(t|x)}{q(t)}dtdx \\
&+ \int_{\mathbf{X}}\int_{\mathbf{T}} p(x,t)\log\frac{q(t)}{p(t)}dtdx \\
&= \int_{\mathbf{X}}\int_{\mathbf{T}} p(x,t)\log\frac{p(t|x)}{q(t)}dtdx \\
&+ \int_{\mathbf{T}} p(t)\Big(\int_{\mathbf{X}} p(x|t)dx\Big)\log\frac{q(t)}{p(t)}dt \\
&= \mathbb{E}_{\mathbf{X}}[D_{KL}[P(\mathbf{T}|\mathbf{X})\|Q(\mathbf{T})]] \\
&- D_{KL}[Q(\mathbf{T})\|P(\mathbf{T})] \\
&\leq \mathbb{E}_{\mathbf{X}}[D_{KL}[P(\mathbf{T}|\mathbf{X})\|Q(\mathbf{T})]]
\end{aligned}
$$

B All 24 Layer Degradation Result

Figure 6 shows the degradation test result for all 24 layers. As we can see, middle layers show the steepest slope at first, indicating they are the most capable ones of capturing important tokens. The reason why layer 24 gets a slow probability drop is because special tokens like [CLS] and [SEP] are not removed in degradation test while [CLS] is regarded as the most important token in layer 24.

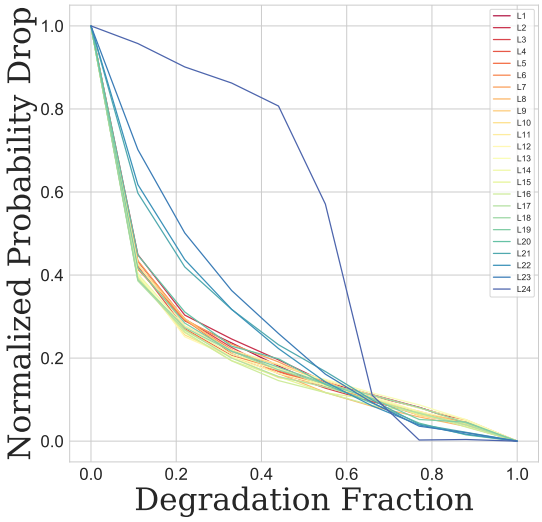


Figure 6: 24-layer degradation test result

C Qualitative Anlysis

Table 4 shows a few examples with highlighted important tokens. We can see top-10 most important tokens across query and passage not only show the token matching but also capture semantic relatedness. For example, "much" in the query of the first example is highlighted. "number", "million" and "$" sign, which are highly related to the concept of "much", are also highlighted. Similarly, in the second example, BERT identifies that the core of the question – "same document". In the corresponding passage, it emphasizes "or" as well as "mortgage" before that and "trust" after that. In the third example, the query is about "stronger", which is again, captured by BERT, and related tokens like "vs" and "roughtly equivalent" are highlighted.

D Detailed [SEP] Attribution Score across Layers

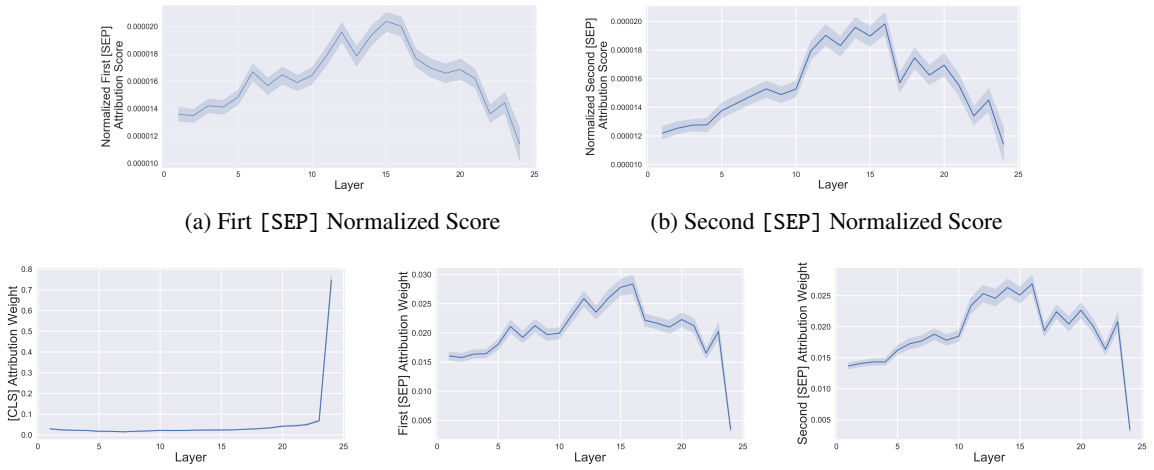
Figure 7 contains plots showing [CLS] weight and two [SEP] scores as well as weights across layers. As we've discussed in Section 4.2, Figure 7c shows how important [CLS] is compared with other tokens across layers—that is, we divide the attribution score by the sum of all of the tokens' attribution scores. It's even more clear that the [CLS] token aggregates all tokens' information in the final layer and becomes the most important token for prediction. The first [SEP] has slightly higher weight than the second one. It's probably because the first [SEP] indicates the boundary between query and document, which is an important information to learn for reranking. But in general they show similar patterns.

E Further Discussion on Positional Bias

We find that BERT prefers passages with important information emphasized at the beginning. But is this preference a real "bias"? Will it cause misjudgement because of emphasizing too much on the start of the passages? To answer this question, we design an experiment to see if passages with higher reranked scores (than the ground truth passages) also happen to get key tokens emphasized earlier. Concretely, for those instances that have incorrectly ranked negative passages higher than the positive one, we regard each token u_i in q as a query, and we find the position of corresponding token v_j that appears in d where $u_i = v_j$. Then, we calculate the mean reciprocal rank for

Query	Document
how much did nr ##a give to congress	m ##em ##bers of congress pay attention to these numbers , and they know that in the last election cycle the nr ##a spent $ 18 . 6 million on various campaigns , a says lee dr ##ut ##man , who has studied the role of gun money in politics for the sunlight foundation .
is mortgage and deed of trust the same document	the mortgage or deed of trust is recorded in the county land records , usually shortly after the borrow ##ers sign it . if the loan is fully paid off , the lend ##er will record a release (or satisfaction) of mortgage or a rec ##on ##vey ##ance of deed (which is used in conjunction with deeds of trust) in the county land records .
which is stronger hydro ##co ##don ##e or ox ##y ##co ##don ##e	dos ##age conversion : hydro ##co ##don ##e vs . ox ##y ##co ##don ##e . in terms of strength , 5 ##mg of ox ##y ##co ##don ##e is roughly equivalent of 7 . 5 of hydro ##co ##don ##e . that is the conversion required to bring about the same effects . hydro ##co ##don ##e would work better if you happen to be a lightweight person with a weak stomach .

Table 4: Top-10 most important tokens identified by IBA in three examples. '[CLS]' and '[SEP]' are ignored.



(a) Firt [SEP] Normalized Score

(b) Second [SEP] Normalized Score

(c) CLS attribution weight.

(d) Firt [SEP] Weight

(e) Second [SEP] Weight

Figure 7: Special tokens attribution scores and weights.

	USE	sent-bert(p)	sent-bert(n)
BM25 top-1	0.540	0.593	0.578
BERT top-1	0.483	0.731	0.563

Table 5: Cosine similarity between query and top-1 passage returned by different methods. "p" refers to pretrained model paraphrase-MiniLM-L6-v2", "n" refers 'bert-base-nli-mean-tokens'

all v_j — MRR = $\frac{1}{|q \cap d|} \sum_{j=1}^{|q \cap d|} \frac{1}{\text{position}(v_j)}$. We then aggregate the MRR for all higher-ranked negative passages (HRNPs) and compare it with the MRR for the lower-ranked positive passages (LRPPs). When we aggregate by the "max" function, we find that, in 86.2% of cases, HRNPs have higher MRR than LRPPs. Averaging all MRRs for HRNPs gives us 0.191, while it's 0.103 for averaging LRPPs. These numbers are 63.8%, 0.129, and 0.103, respectively if we aggregate by the arithmetic mean. We cannot say that the reason for those negative passages ranking higher is due to matched tokens appearing earlier, but we do note a correlation between HRNPs and early-appearing matched tokens.

Driven by this positional bias, we are also curious about how positional index correlates with the passages' ranks. We compute the average positional index p_i for each document's top-20 most important tokens, and then average p_i for each query. As we show in Fig. 8, higher-ranked passages do have earlier tokens emphasized, meaning that passages with important tokens stressed earlier are preferred. When comparing the top-1 document returned by BERT d_{BERT} with the top-1 document returned by BM25 d_{BM25}, this preference also exists. We compute the MRR across all tokens in the query and passages like we do

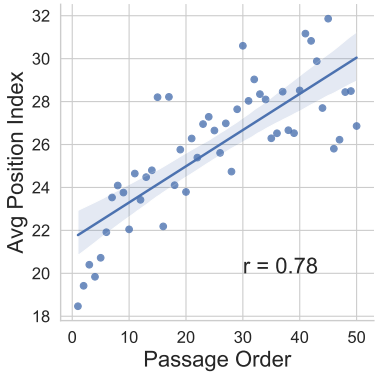


Figure 8: Positional index.

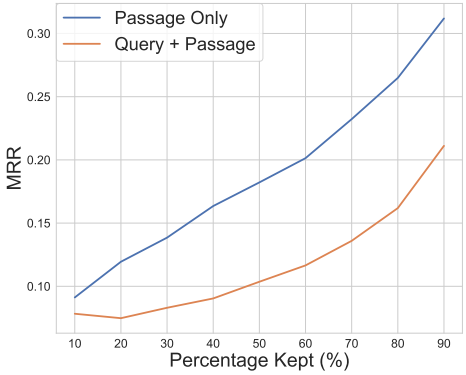


Figure 9: Average truncation MRR.

in Section 4.4 for those passages $d_{\text{BERT}} \neq d_{\text{BM25}}$ and d_{BERT} makes the correct prediction. We find that even BM25 is almost all about term matching, with $\mathbf{J}(q, d_{\text{BERT}}) = 0.062$, $\mathbf{J}(q, d_{\text{BM25}}) = 0.074$, considering the position, $\text{MRR}(q, d_{\text{BERT}}) = 0.127$ is still higher than $\text{MRR}(q, d_{\text{BM25}}) = 0.099$.

F Note on Semantic Similarity Measurement

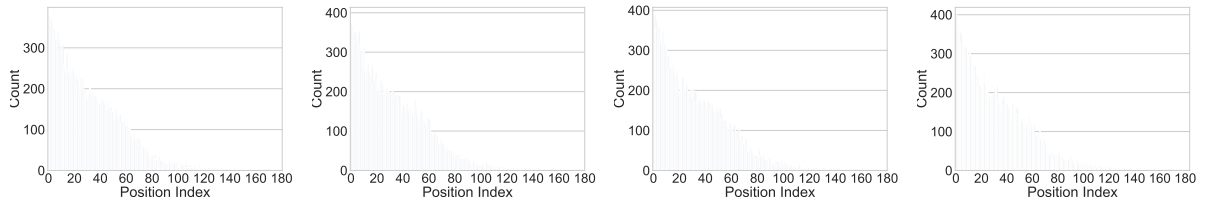
We also find that it is hard to measure the contextualized relevance between query and passages by simply calculating cosine similarity ϕ between query vectors and document vectors. We encode q, d_{BERT}, d_{BM25} and don't find that $\phi(\eta(q), \eta(d_{\text{BERT}}))$ is higher than $\phi(\eta(q), \eta(d_{\text{BM25}}))$ when using the Universal Sentence Encoder or Sentence-BERT (Reimers and Gurevych, 2019), denoted as η, pre-trained on an NLI dataset. However, if using a Sentence-BERT pre-trained on a paraphrase corpus (specifically the model "paraphrase-MiniLM-L6-v2") to measure semantic similarity, $\phi(\eta(q), \eta(d_{\text{BERT}}))$ is significantly higher than $\phi(\eta(q), \eta(d_{\text{BM25}}))$ as "paraphrase-MiniLM-L6-v2's" pretrained corpus includes MSMARCO triplet. Tempting as it is to conclude that BERT has indeed captured semantic similarity that BM25 hasn't, it's unfair to use a pre-trained model with prior knowledge on MSMARCO to measure the semantic similarity. Therefore, we think BERT has learned a deeper relevance between the query and document, but it cannot be simply measured by vaguely defined semantic similarity.

G Truncation Test across All Passages

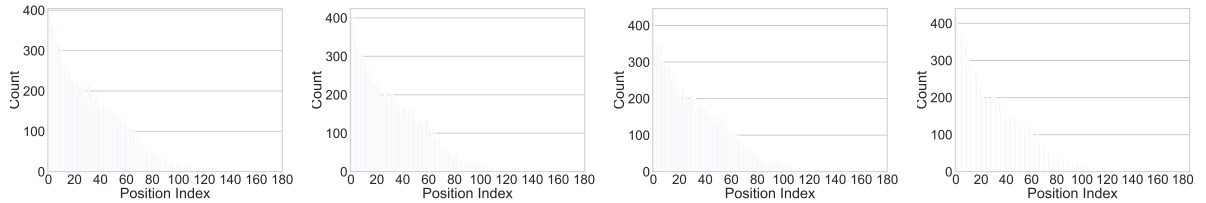
To further measure the trade-off between compression and quality, we do truncation test for all passages. Specifically, given a g% of tokens kept for every single document, we measure final ranking performance—MRR score. The result is shown in Figure 9. From the result, we can see that truncating doc only is more robust than truncating both query and doc. On average, with 90% tokens of passage kept, we have MRR = 0.311. But for the maximum, we can get MRR = 0.392 with 90% tokens, which is very close to the original score, and that depends on what tokens we remove.

H Position Index for Important Tokens across 24 Layers

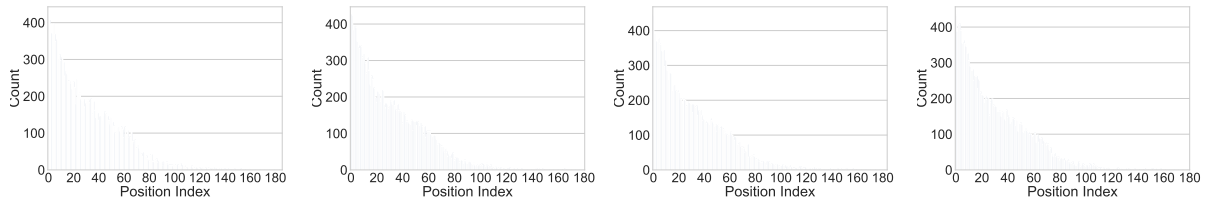
Shown in Figure 10, layer 24 is the outlier, where most tokens emphasized are in the middle of the document. For other layers, it's still the start of passages that is emphasized.



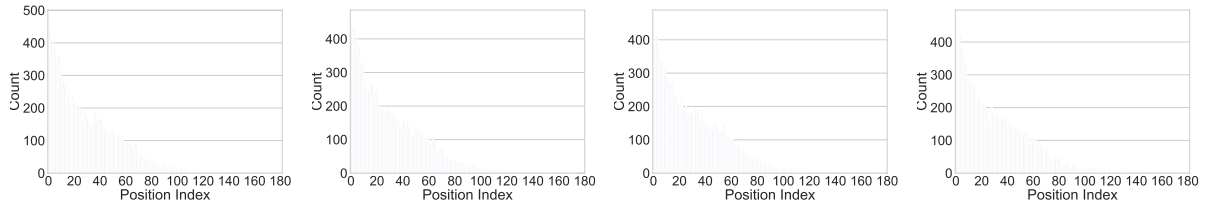
(a) Layer 1 (b) Layer 2 (c) Layer 3 (d) Layer 4



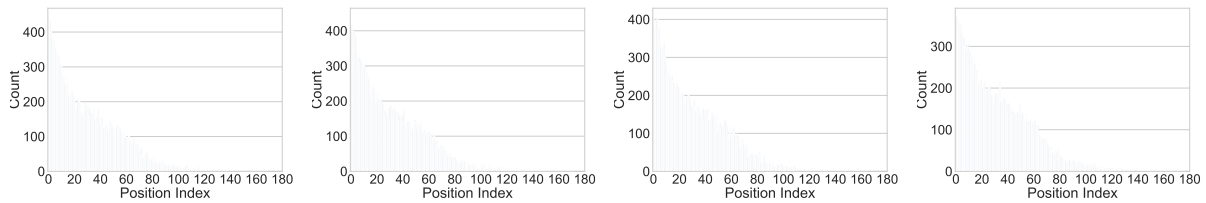
(e) Layer 5 (f) Layer 6 (g) Layer 7 (h) Layer 8



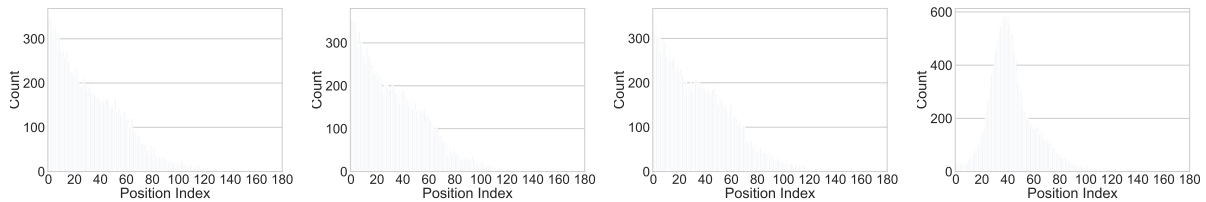
(i) Layer 9 (j) Layer 10 (k) Layer 11 (l) Layer 12



(m) Layer 13 (n) Layer 14 (o) Layer 15 (p) Layer 16



(q) Layer 17 (r) Layer 18 (s) Layer 19 (t) Layer 20



(u) Layer 21 (v) Layer 22 (w) Layer 23 (x) Layer 24

Figure 10: Positional index across 24 layers