

Axiomatic Causal Interventions for Reverse Engineering Relevance Computation in Neural Retrieval Models

Catherine Chen
catherine_s_chen@brown.edu
Brown University
Providence, Rhode Island, USA

Jack Merullo
john_merullo@brown.edu
Brown University
Providence, Rhode Island, USA

Carsten Eickhoff
c.eickhoff@acm.org
University of Tübingen
Tübingen, Germany

ABSTRACT

Neural models have demonstrated remarkable performance across diverse ranking tasks. However, the processes and internal mechanisms along which they determine relevance are still largely unknown. Existing approaches for analyzing neural ranker behavior with respect to IR properties rely either on assessing overall model behavior or employing probing methods that may offer an incomplete understanding of causal mechanisms. To provide a more granular understanding of internal model decision-making processes, we propose the use of causal interventions to reverse engineer neural rankers, and demonstrate how mechanistic interpretability methods can be used to isolate components satisfying term-frequency axioms within a ranking model. We identify a group of attention heads that detect duplicate tokens in earlier layers of the model, then communicate with downstream heads to compute overall document relevance. More generally, we propose that this style of mechanistic analysis opens up avenues for reverse engineering the processes neural retrieval models use to compute relevance. This work aims to initiate granular interpretability efforts that will not only benefit retrieval model development and training, but ultimately ensure safer deployment of these models.

CCS CONCEPTS

• **Information systems** → **Retrieval models and ranking**; Document representation.

KEYWORDS

interpretability, neural ranking models, information retrieval axioms, search

ACM Reference Format:

Catherine Chen, Jack Merullo, and Carsten Eickhoff. 2024. Axiomatic Causal Interventions for Reverse Engineering Relevance Computation in Neural Retrieval Models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*, July 14–18, 2024, Washington, DC, USA. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3626772.3657841>

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SIGIR '24, July 14–18, 2024, Washington, DC, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0431-4/24/07
<https://doi.org/10.1145/3626772.3657841>

1 INTRODUCTION

State-of-the-art neural ranking models achieve high performance on a variety of tasks. Despite their success, how these models arrive at their decisions remains largely unknown. Uncovering these decision-making behaviors is crucial, not only for diagnosing model errors and improving ranking performance but also for addressing potential biases in the model. As neural retrieval models become larger and more inscrutable, there is a need for methods unveiling the various relevance criteria considered throughout the parameters of a ranking model.

Axiomatic IR constructs formal constraints, or *axioms*, outlining specific properties that an effective ranking model should satisfy. For instance, the *TFC1 axiom* [11] asserts that a ranking model should prioritize documents with a higher frequency of query term occurrences. IR axioms provide a significant advantage in diagnosing model behavior by testing a model's adherence to desired properties. Retrieval axioms have been instrumental in identifying and rectifying shortcomings in traditional retrieval models to enhance their ranking capabilities [3]. However, modern neural retrieval models are sophisticated black boxes, and it is unclear whether they learn structured features that directly correspond to interpretable mechanisms for, e.g., tracking query term frequencies.

To gain a better understanding of how neural retrieval models make predictions, causal intervention-based methods emerge as a solution. Interpretation of language models often uses methods based on causal mediation analysis [25] to localize model behaviors [30]. More recently, *Mechanistic Interpretability* focuses primarily on understanding learned behaviors of the Transformer architecture [29] underlying modern NLP systems [9, 14, 21, 33]. These methods are extremely effective at isolating important model components and more significantly, understanding how these components interact to complete a task. From an explainability perspective, this form of analysis provides a level of granularity that surpasses existing explainable IR (XIR) work, such as probing which yields correlational but not causal insights.

In this paper, we combine the inherently human-interpretable nature of IR axioms with diagnostic datasets to propose a causal-intervention based hypothesis testing framework to explain and localize the ranking behavior of neural models. First, we design a novel activation patching setup for retrieval, highlighting differences in evaluation compared to existing activation patching efforts on generative language tasks. Next, we discuss the shortcomings of current diagnostic datasets and provide guidelines for systematically curating diagnostic datasets for activation patching. Finally, we demonstrate the effectiveness of activation patching for targeted hypothesis testing in neural retrieval models. Specifically, we test if such models adhere to the TFC1 axiom, and further

analyze if this axiom is implemented in an interpretable way. On a pre-trained DistilBERT-based encoder, TAS-B [19], we find evidence for an attention head-based mechanism that acts as a term frequency identifier.

Overall, this perspectives paper aims to initiate interpretability efforts to localize model ranking behavior, potentially reshaping our approach to isolating axiomatic behavior in neural models. Such efforts can pave the way for constructing a compositional definition of relevance, thereby enhancing both ranking capabilities and safety. Specifically, we make the following contributions:

- Extend activation patching to retrieval models, uncovering the concrete mechanisms capturing retrieval axioms.
- Establish best practices for constructing diagnostic datasets for activation patching.
- Demonstrate that TAS-B learns a latent mechanism for tracking term frequencies, congruent with the TFC1 axiom.
- Propose new directions for explainable IR (XIR) research based on causal interventions

The remainder of this paper is structured as follows: In Section 2, we present existing work on axiomatic IR and mechanistic interpretability and describe previous attempts to understand ranking concepts learned by neural models. Section 3 introduces our activation patching methodology for retrieval settings. In Section 4, we outline our experimental setup, and in Section 5 we present the results of our causal interventions. In Section 6, we discuss the implications of axiomatic mechanistic interpretability work and propose future XIR research directions, and then conclude our paper in Section 7.

Author Perspectives. The perspectives presented in this paper reflect the views of academic authors based in North America and Europe. Our study introduces methods aimed at enhancing our understanding of the underlying mechanisms of neural retrieval models and relevance computation. The implications of this work extend beyond the academic context, offering potential benefits to industrial research and practice by facilitating a hypothesis testing framework for assessing desirable or undesirable model properties prior to deployment.

2 RELATED WORK

2.1 Axiomatic IR

Retrieval axioms were first introduced by Bruza and Huibers [3] and since then, have been applied in a number of ways to enhance ranking effectiveness through axiomatic re-ranking [17] or regularizing neural retrieval models [5, 6, 27]. Recent research in explainable IR (XIR) has leveraged axioms to uncover and explain the ranking concepts learned by neural retrieval models [4].

From an explainability perspective, axioms offer a significant advantage over alternative interpretability methods such as feature attribution due to their grounding in concepts that are inherently intuitive to humans. For example, *diagnostic datasets* have been used to systematically test ranking axioms in neural models [4, 26]. Furthermore, axioms have been used to explain neural ranking decisions by investigating the extent to which the decisions can be explained by retrieval axioms [31]. While prior approaches holistically shed light on the end-to-end behavior of neural models by

identifying satisfied axioms, we extend this work by localizing ranking concepts to specific components. This approach will allow us to gain a more granular understanding of how ranking models make their decisions.

2.2 Understanding NRM Learned Concepts

Probing is a popular method that has been used previously to assess a model’s acquisition of certain concepts and localize the network components responsible for such behavior. This technique involves training a light-weight classifier on top of a model’s components (e.g., embeddings or attention maps) to evaluate the information encoded in its representations [7, 10, 12, 13, 20, 28, 32, 34].

Although these methods reveal the information learned by the network based on correlational data, there are ongoing debates regarding the reliability of probing in determining actual causality and confirming the concrete utilization of learned concepts in the final inference [1, 2]. In this paper, we opt for a different approach, developing a causal-intervention based method for the analysis of neural retrieval models.

2.3 Mechanistic Interpretability

Mechanistic interpretability aims to unravel the internal mechanisms of neural models, mapping them to human-understandable concepts, typically through causal interventions. The primary objective is to localize model behavior to specific components, such as individual attention heads, and analyze the interactions among these components to determine how they complete a task. One way this is done is with activation patching, which replaces the output of a component from one forward pass (e.g., an attention layer) with that from a similar input (‘patching’). It is also commonly referred to as causal mediation analysis [30], causal tracing [21], or interchange interventions [14]. In generative language modeling, causal interventions have proven valuable to detect gender bias [30], investigate where models store factual information [15, 21], identify a collection of components that interact with each other to perform concrete tasks [18, 33], and correct model errors through editing [21, 22].

Existing XIR methods for interpreting neural ranking models currently lack this level of granularity and causal understanding. This paper aims to address this gap by introducing causal interventions for retrieval.

3 METHODOLOGY

In this section, we provide the technical details on activation patching in the context of generative language tasks and outline our specific activation patching setup tailored for retrieval. Additionally, we detail our process for curating a dataset intended for activation patching purposes.

3.1 Activation Patching

To understand how activation patching can localize model behavior to specific components, it is important to recap how information flows through the model. Transformer models are comprised of several stacks of multi-headed attention and multi-layer perceptron (MLP) layers [29]. The *residual connection* that updates the hidden

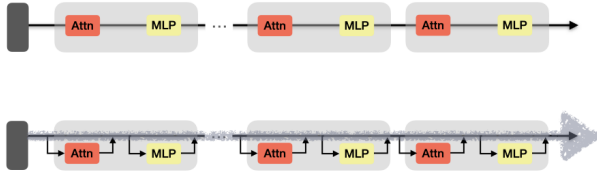


Figure 1: *Top*: Traditional transformer diagram depicting a linear information flow between blocks. *Bottom*: Non-sequential transformer diagram demonstrating read and write operations to an assumed common residual stream. Layernorms are not shown for simplicity.

representation in a model by adding the output of an MLP or attention block to its input induces a helpful intuition for analyzing these models: transformers move information through a *residual stream* that network components “read” from and “write” to [9]. This reinterpretation of information flow in transformer models carries a crucial implication for interpretability work: Each layer possesses the capability to “communicate” with downstream layers by transmitting information through the residual stream by adding their outputs to it. Figure 1 visually represents this information flow.

Previous work on activation patching for generative language tasks involves running the model on pairs of inputs: (1) a *clean* input, denoted as X_{clean} , that produces a correct answer (e.g., *input*: “Paris is the capital of”, *answer*: “France”) and (2) a corrupted input, denoted as $X_{corrupt}$, which changes the input in a minimal way such that the expected answer changes (e.g., *input*: “London is the capital of”, expected answer: “England”). The model runs on each input and stores all of the intermediate activations (MLP outputs, hidden states, etc.). Notably, in the clean run, the model produces the correct answer, whereas in the corrupted run, it does not. In a third, *patched*, run, the model runs on $X_{corrupt}$. But during this run, an activation from the clean run is patched in to replace the corresponding corrupted activation. After the model completes this run, the evaluation focuses on gauging how much the output has shifted away from the corrupt answer (England) and toward the correct answer (France), typically by examining the difference between the relevant logits. The intervention is iteratively repeated for all possible activations to localize those activations most instrumental for the task. If an intervention on a specific activation significantly improves performance, it signifies the importance of that activation for producing the right answer.

We can patch activations into a transformer in several different places: (1) residual stream, (2) attention outputs, (3) individual attention heads, and (4) MLP outputs. Additionally, we can also patch at specific input positions (i.e., we can patch in activations for individual tokens in the input). This allows for a nuanced exploration of how different components contribute to model behavior.

To make activation patching suitable in a retrieval setting, we propose several modifications to this general scheme: (1) to the input pairs and (2) to the evaluation metric. To construct a suitable dataset for activation patching in the context of retrieval, we create clean and corrupt query-document-document triples with respect

to a target axiom, whose behavior we aim to isolate in the network (more details in Section 3.2). We refer to the clean and corrupt documents in these triples as $X_{baseline}$ and $X_{perturbed}$, respectively.

However, unlike prior methods which consistently patch activations from $X_{baseline}$ into $X_{perturbed}$ during the third patched run through the model, we determine the patch based on the perturbation’s expected effect. Specifically, we always patch activations from the document with higher expected performance into the run on the document with the lower expected performance. This modification accounts for axiomatic perturbations that may either add or remove crucial relevance concepts from a document. For instance, in evaluating term frequency, query terms could be introduced to a document to observe how the ranking score increases, or conversely, query terms might be removed or replaced to assess how the ranking score decreases. Our full activation patching algorithm¹ is shown below and visualized in Figure 2:

- (1) *Baseline run*: Run the model on $X_{baseline}$. If the ranking score of $X_{baseline}$ is expected to be greater than $X_{perturbed}$, cache activations and record the ranking score.
- (2) *Perturbed run*: Run the model on $X_{perturbed}$. If the ranking score of $X_{perturbed}$ is expected to be greater than $X_{baseline}$, cache activations and record the ranking score.
- (3) *Patched run*: Run the model on one of $X_{baseline}$ or $X_{perturbed}$ (whichever has the lower expected ranking score), replacing a specific activation (e.g., attention layer output) with the cached values from the other run, and record the final ranking score.

To assess the effect of the patch on model performance, we evaluate using the normalized difference in ranking scores. A value of 1 indicates that the intervention increases the ranking score such that it fully recovers the performance of the higher ranked document, while a value of 0 indicates the patch had no effect on performance. In other words, a value of 1 suggests that the patched activations encode important information for the ranking score calculation.

3.2 Diagnostic Dataset Curation

Activation patching requires pairs of inputs to isolate the effects of a model’s ability to complete a task. For retrieval, we define input document pairs with respect to a query to form query-document-document triples. To construct our diagnostic dataset triples, we modify documents according to the TFC1 axiom as defined by Fang et al. [11]:

TFC1 Let $q = w$ be a query with only one term w . Assume the length of document d_1 equals the length of document d_2 . If the number of occurrences of w in d_1 is greater than the number of occurrences of w in d_2 , then for query q , the relevance score of d_1 should be higher than d_2 .

We define two perturbations to observe the effects of TFC1 along two lenses: injection and replacement. Thus, TFC1-Inject (TFC1-I) and TFC1-Replace (TFC1-R) are outlined below. For a given query and document,

¹We modify the *TransformerLens* library [23] and our activation patching code can be found at <https://anonymous.4open.science/r/sigir-submission-4647>.

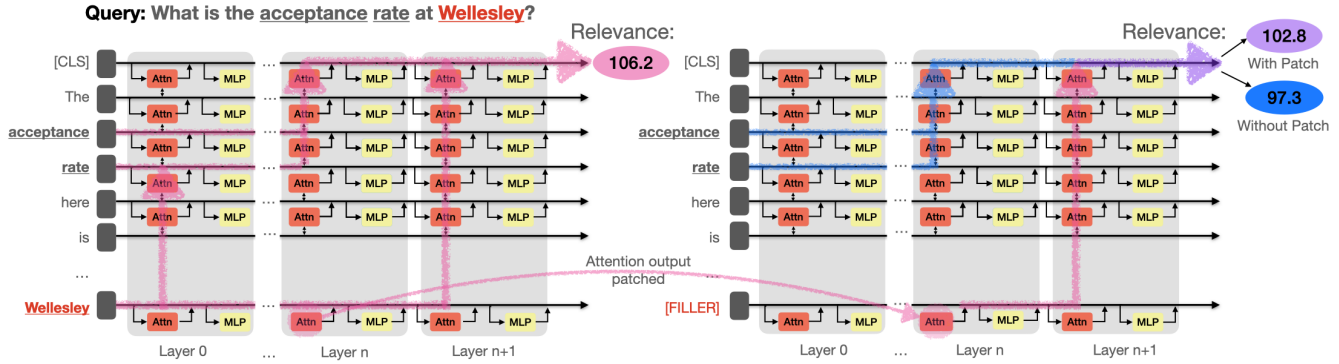


Figure 2: Activation patching setup for retrieval. In this example, a document pair is constructed to observe term frequency effects in the model. A perturbed document $X_{perturbed}$ (left) is created by injecting a selected query term (“Wellesley”) at the end, and a baseline comparison document $X_{baseline}$ (right) is created by adding filler tokens to equalize document lengths. Each input is run through the model, with network components reading and writing information to their respective residual streams. In a third *patched* run, the model runs on $X_{baseline}$, and an activation (e.g., attention head output) from the cached $X_{perturbed}$ run is patched in. The model continues its run, with the residual stream now affected by the patch to produce a new ranking score.

- TFC1-I We randomly sample a query term and insert it at the end of the document d to create our perturbed document d_p . To create a baseline document d_b equal in length to our perturbed document, we insert a filler token(s) (e.g., ‘a’) at the end of document d .
- TFC1-R We randomly sample one query term and replace all its occurrences in document d with a filler token(s) to create a perturbed document d_p . The original document d acts as the baseline document d_b .

Figure 3 includes an example of a query-document-document triplet for TFC1-I. For a given query “average snowfall nyc”, the term “snowfall” is randomly selected for injection. The perturbed document is constructed by injecting “snowfall” to the end of the original document but before the SEP token. The baseline document is constructed to match the length of the perturbed document by inserting a filler token (i.e., “a”) that has minimal impact on the ranking score.

We curate our diagnostic dataset using MS-MARCO [24]. For each query in the development set (approximately 6.8k queries) we retrieve the top 100 relevant documents and perturb them. Subsequently, we recalculate retrieval scores for all queries on the perturbed corpus and identify the 100 queries exhibiting the highest average change in score per document. This procedure is conducted independently for TFC1-I and TFC1-R. Overall, by strategically perturbing the corpus and selecting queries based on their impact on retrieval scores, we can effectively leverage activation patching to glean insights into specific areas of the network that contribute to variations in ranking performance.

To enable a comprehensive analysis of the most crucial tokens across various documents, we establish several token classes for standardized comparison. Tokens are categorized primarily according to their relation to the original query, considering factors such as whether the token appears in the full query and/or is the chosen

Table 1: Token type classifications for documents. TFC1-I perturbed documents include all six token types, while TFC-R perturbed documents have five token types since no terms are injected during perturbation.

Label	Definition
tok_{CLS}	The CLS token.
tok_{inj}	The selected query term injected into the document.
tok_{qterm+}	Occurrences of the selected query term that already exist in the original document.
tok_{qterm-}	Occurrences of the non-selected query terms in the original document.
tok_{other}	Terms in the original document that are not query terms.
tok_{SEP}	The SEP token.

term for injection. The detailed breakdown of token types and their definitions is presented in Table 1, while an illustrated example of a document with labeled token types is depicted in Figure 3.



Figure 3: Example of a perturbed and baseline document pair for TFC1-I, labeled by token types.

4 EXPERIMENTAL SETUP

We run all our experiments on TAS-B [19], a DistilBERT-based model with 6 layers and 12 attention heads per attention layer. TAS-B independently encodes queries and documents and uses a pooled representation of the CLS token for ranking score calculation. Beyond its status as a high-performing neural ranking model, prompting our interest in understanding its inner workings, TAS-B is an interesting target due to its simplified architecture. Since activation patching involves iterative interventions on model components with multiple runs for each input, models with fewer parameters, such as TAS-B, demand fewer computational resources. Additionally, the smaller architecture aids in precisely localizing the impact of interventions by narrowing down the search space.

Recall that activation patching involves patching in activations from a high-performing run into a low-performing run. Considering the opposing perturbation effects of TFC1-I and TFC1-R (where TFC1-I raises ranking scores through injection of query terms while TFC1-R lowers ranking scores of perturbed documents by removing query terms), the activation patching setups for these scenarios are inverse. Specifically, in the experiments on TFC1-I, we run the model on $X_{baseline}$ and the activations from $X_{perturbed}$ are patched in. Conversely, for the TFC1-R experiments, the model runs on $X_{perturbed}$ and activations from $X_{baseline}$ are patched in. In both cases, the model runs on the input with fewer occurrences of the selected query term, allowing observation of the effects of patching in an activation from a run on a document that contains more instances of the selected query term.

5 RESULTS

In this section, we present the results from our activation patching experiments and describe the components in TAS-B that encode a term frequency signal.

5.1 Importance of Added/Deleted Query Terms

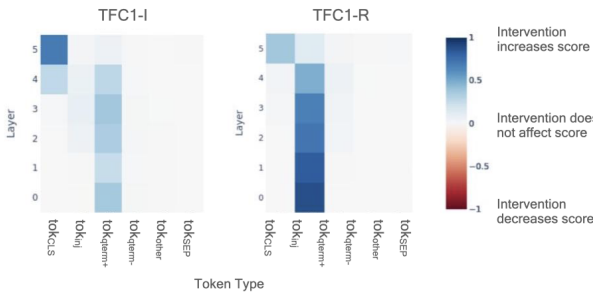


Figure 4: Results from patching into the residual stream at the start of each layer over all tokens positions in the document.

Figure 4 displays the results of activation patching for TFC1-I and TFC1-R. In these experiments, we patch in the residual stream at the beginning of each layer for each token in the document. To identify which tokens exhibit the most significant impact across all documents, we categorize the results for all document tokens based on their token type, as defined in Table 1. In the patching result

figures, blue squares highlight the tokens that increase performance when patched.

First, we find that the model becomes confident and reaches a decision in the later layers, specifically in Layers 4 and 5 (Figure 4). At this point, the term frequency information is transferred from the existing selected query term tokens tok_{qterm+} to the CLS token. This shift towards the CLS token aligns with expectations, considering that the ranking score is derived from a pooled representation of the CLS token.

Second, for TFC1-I, the injected tokens (tok_{inj}) surprisingly are not the most important for recovering performance. Rather, the instances of the selected query term already present in the original document (tok_{qterm+}) are the most impactful. We postulate that the model may store important information in query terms situated toward the beginning of the document. To further investigate this hypothesis, we run an additional experiment changing the location of the perturbation, injecting the selected query term at the beginning of the document rather than the end. By doing so, we find that this leads to a shift in importance towards the injected tokens at the beginning of the document (Figure 5), suggesting that the model stores the majority of the term frequency information in the initial occurrence of duplicate terms.

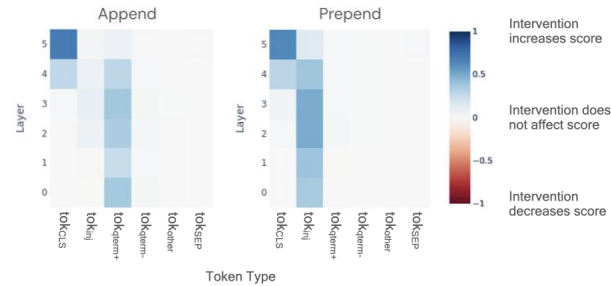


Figure 5: TFC1-I residual stream patching results by location of term injection. The left shows results from the original patching setup, with the selected query term injected at the end of each document. On the right, selected query terms are injected at the beginning of the document.

Third, for TFC1-R, patching in the activations from the baseline run into the perturbed run, precisely at the positions where the query term instance was removed, leads to significant performance improvements. These outcomes, when considered alongside the observations from patching for TFC1-I, indicate that, as anticipated, term frequency information is remarkably localized to the selected query term.

Additionally, we conduct experiments on the outputs of attention layers and MLPs, yet we do not observe any indications of either component type significantly influencing performance. Consequently, we hypothesize that the term frequency signal is likely localized to individual attention heads. To explore this hypothesis further, we proceed to perform activation patching specifically on attention head outputs in the next subsection.

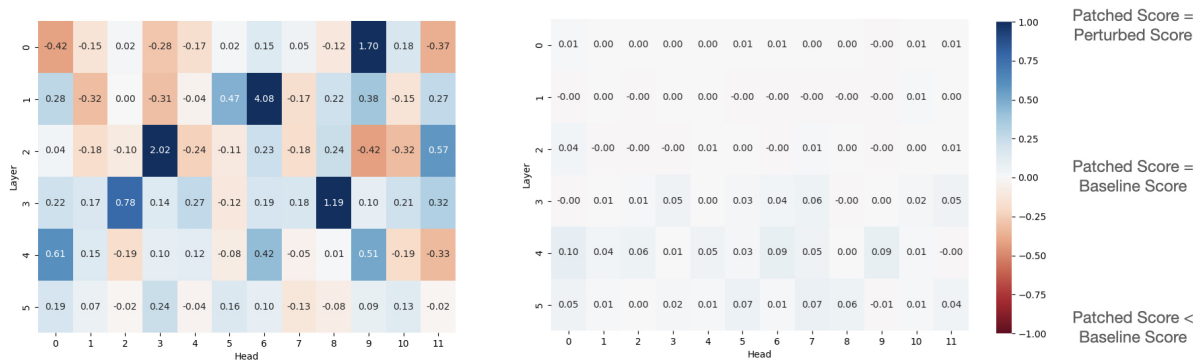


Figure 6: Activation patching on individual attention heads reveals four heads (0.9, 1.6, 2.3, 3.8) that encode the TFC1 axiom. These heads fully recover and exceed the perturbed performance when the model runs on the baseline inputs. Here, we present results for the top (left) and bottom (right) 10% of relevant documents per query. Even though all documents have at least one occurrence of a query term, the attention heads are only effective when there is an existing relevance signal.

5.2 Term Frequency Signal Components

Patching individual attention heads for TFC1-I reveals that attention heads 0.9 (Layer 0, Head 9), 1.6, 2.3, and 3.8 heavily influence ranking performance (Figure 6) when fielding term frequency interventions. When these four heads are patched in, the model fully recovers (and even surpasses) the perturbed performance. This suggests that these heads contain a high concentration of information important to the ranking score calculation. Interestingly, we observe that these heads are most effective when there is an existing relevance signal. In other words, these components may amplify existing indications of relevance but do not, by themselves signal relevance in the absence of other evidence. Figure 6 presents the results categorized by the top and bottom 10% of relevant documents per query. We find that heads 0.9, 1.6, 2.3, and 3.8 have a substantial positive impact on the top relevant documents and nearly no impact on the least relevant documents. This observation might explain why, when replicating the same experiments for TFC1-R, no important heads are identified. To further verify the importance of these four heads, we perform ablation experiments and observe significant performance decreases (Figure 7).

5.3 Attention Head Behavior

To determine which tokens contribute to the relevance signal within these heads, we analyze the average attention scores from the injected tokens to other document token types. Figure 8 illustrates that in heads 0.9 and 1.6, the injected tokens primarily attend to other occurrences of the selected term in the document. However, in heads 1.6 and 2.3, attention begins to shift or is entirely concentrated on the SEP token. In other words, it appears that the term frequency signal is stored in duplicate token occurrences in earlier layers, but in later layers, it may be dispersed broadly across the document representation. Due to this difference in behavior, we posit that these two groups of heads are interacting with each other via the residual stream to compose a relevance signal for the document, and further discussion on this hypothesis is provided in the following section.

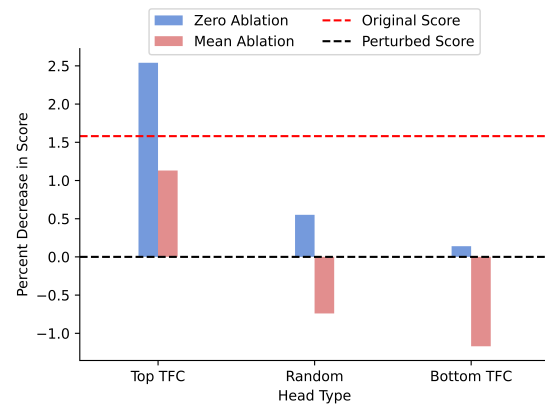


Figure 7: Results from ablation experiments on duplicate token heads. In one experiment, we zero ablate the attention heads. In a second experiment, we replace the values with the mean activation value across all documents for each query.

6 DISCUSSION

Our results demonstrate the feasibility of employing axiomatic causal interventions to localize relevance computation within specific model components. This introduces several novel directions for XIR research. In this section, we discuss the implications of our findings on future work for reverse engineering neural retrieval models.

6.1 Implications for Axiomatic Datasets

In this work, we design diagnostic datasets to successfully isolate components in a neural retrieval model that encode the TFC1 term frequency axiom. This is promising for future axiomatic model diagnosis, given the framework’s flexibility to seamlessly test various existing or novel axioms. However, the curation of diagnostic datasets for activation patching requires careful consideration, specifically:

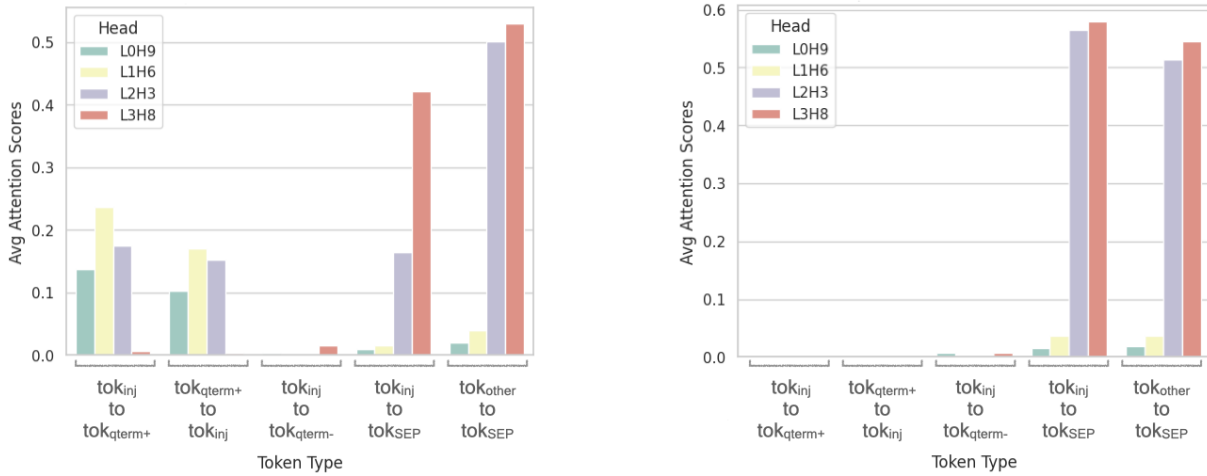


Figure 8: Average attention scores for duplicate token heads. Left: documents with at least one occurrence of the injected term in the original document before perturbation. Right: documents with no occurrences of the injected term in the original document before perturbation.

- (1) *Thoughtful Perturbation Locations:* The choice of locations for perturbations should be deliberate and well-considered.
- (2) *Caution with Randomization:* Randomization in dataset creation may lead to sub-optimal diagnostic datasets and should be approached with caution.

First, the selection of perturbation locations demands careful consideration when constructing diagnostic datasets, as the location of perturbation may yield varying effects on the ranking score for the same document. In our initial data analysis, we discovered that, while constructing the diagnostic dataset for TFC1-I, certain queries exhibited different levels of robustness to changes in perturbation location. Notably, some queries demonstrated higher ranking scores when the selected query term was injected at the beginning of the document compared to when it was injected at the end of the document (Figure 9). An initial hypothesis for this phenomenon may be that TAS-B, trained on MS-MARCO, where document titles are concatenated to the beginning of the text, might be learning to assign higher importance to terms occurring at the document’s start. While the robustness of documents to perturbations extends beyond the paper’s scope, we advise future researchers to carefully consider the perturbation location when constructing diagnostic datasets.

Secondly, randomization has the potential to yield sub-optimal diagnostic datasets, as it may not effectively isolate axiomatic behavior. Consider the TFC1-A perturbation, defined by Rosset et al. [27] in their work to regularize neural retrieval models with axiomatic datasets. This perturbation involves randomly selecting a query term for injection:

TFC1-A We randomly sample a query term and insert it at a random position in document d . We expect the perturbed document $d^{(i)}$ to be more relevant to the query - *i.e.*, $d^{(i)} > d$.

In addition to scoring variations that may arise from the random placement of terms, the random selection of query terms may also result in low-IDF terms being selected. In this scenario, given a query such as “What is the acceptance rate at Wellesley?”, random selection without constraints could equally likely result in choosing “Wellesley” or “the” as the term to be injected. Activation patching with “Wellesley” injected at the end of a document is much more likely to isolate term frequency behavior as opposed to “the” since the former is likely to cause the score for $X_{perturbed}$ to be significantly higher than $X_{baseline}$, whereas the latter may not. To address this challenge in our experiments, we mitigate the issue by selecting queries with the highest average changes in score after perturbation, ensuring that the chosen terms are deemed “important” query terms (e.g., high-IDF terms). An alternative selection method could involve choosing query terms based on their part of speech, such as selecting only nouns.

Overall, a thorough analysis of perturbation effects is crucial during the final diagnostic dataset collection to guarantee that perturbations exert a substantial impact on ranking scores. Without significant perturbation effects, there may not be a sufficient signal for the model to effectively localize axiomatic behavior in activation patching experiments.

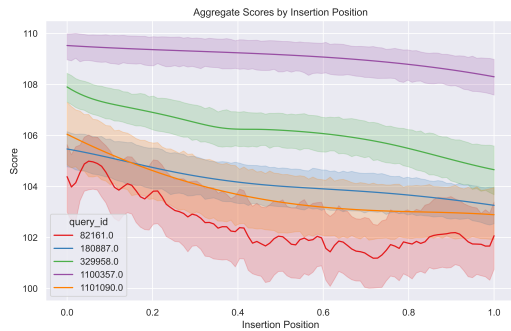


Figure 9: Document ranking scores may vary depending on the location of perturbations. In this example, we show how ranking scores change on average for documents across five queries. The horizontal axis represents a normalized position in each document where a selected query term is injected, while the vertical axis represents the ranking score. For this perturbation, document scores decrease as the position of the injection moves toward the end of the document.

6.2 Relevance Computation Heads

In this section, we provide further discussion of our activation patching results and address how this work can provide a foundation for discovering the compositional definition of relevance based on IR axioms.

What roles do these heads play? Activation patching on individual attention heads reveals four heads that significantly express a term-frequency signal aligned with the TFC1 axiom. However, as previously noted in Section 5, inspecting the attention scores and patching along token positions reveals distinct behavior among the heads. Specifically, heads in earlier layers (i.e., 0.9 and 1.6) function as *duplicate token heads*, primarily attending to duplicate instances of the selected query term (thereby potentially counting term frequencies) and storing important relevance information in these tokens. Remarkably, these heads can recover a significant amount of ranking performance on duplicate tokens alone, sometimes even exceeding the original performance, indicating that the model encapsulates a robust relevance signal in these tokens (Figure 10).

On the other hand, heads in middle layers (i.e., 2.3 and 3.8) exhibit distinct behavior compared to heads in earlier layers. These middle-layer heads do not attend to duplicate tokens, yet they still have a strong positive impact on performance. Furthermore, while patching the entire head significantly influences model performance in a positive direction, no one single token type is responsible for the majority of this behavior, suggesting that the term frequency signal may not be concentrated in any individual component (Figure 10).

Although TAS-B encodes queries and documents separately as opposed to a traditional BERT ranking model, we find that previous hypothesized behavior on the internal mechanisms of BERT aligns with the observed interactions between heads in TAS-B. Zhan et al. [34] hypothesize that BERT initially extracts representations for documents and queries in earlier layers and subsequently forms more context-specific representations to determine relevance. We

posit that duplicated token heads write the term frequency signal to the residual stream that *relevance composition heads* in the middle layers use to build a comprehensive relevance signal for the document that is dispersed among the document representations.

While we do not explore this hypothesis in detail within this paper, the implications suggest a potential avenue for future interpretability work. Specifically, future research could explore components in earlier layers responsible for extracting document representations, while components in middle layers might contribute to building a compositional definition of relevance.

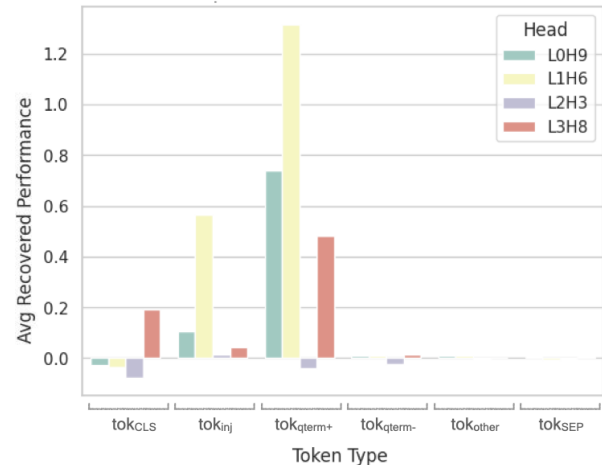


Figure 10: Activation patching results by token position for attention heads 0.9, 1.6, 2.3, and 3.8 for documents with duplicate occurrences of the selected query term. Heads in earlier layers increase in score by more than 75% just by patching on duplicate instances of the selected query term. On the other hand, heads in middle layers recover less than half (or even none at all) of the ranking score on a single token type.

Why are there no important attention heads in the last two layers? As seen in Figure 4, the model becomes confident in its decision in the final two layers, transferring information over to the CLS token. Consequently, at this stage in the model, the relevance computation has stabilized and individual attention heads no longer play a pivotal role. Notably, the information shift from the attention layers to the CLS token begins in Layer 4, where we observe a small number of heads that have a medium-positive impact when patched in. While these heads individually are not important enough to fully recover performance, there is a possibility that they collaboratively contribute to pooling contextual document representation for the determination of final relevance. This presents an intriguing avenue for future investigation.

Is relevance information stored in specific tokens? Moving the injection location from the end to the beginning of the document suggests that the model primarily stores term-frequency information in the initial instance of the duplicate token (Figure 5). However, inspecting the attention scores shows that this assumption may not always hold true. When the injection occurs

at the document’s end, although the injected term predominantly attends to earlier instances of duplicate terms, the earlier instances of query terms also attend to the injected term (Figure 8). This indicates that term-frequency information is distributed across all duplicate token instances, rather than being centralized solely in the first instance as Figure 5 might imply. One potential explanation for this behavioral difference may originate from TAS-B’s training paradigm, as discussed in Section 6.1. Given that document titles are often concise and contain essential keywords, the model may have learned to attend more heavily to tokens at the beginning of a document. Future work could investigate this more deeply by exploring how training paradigms influence the internal document representations of neural ranking models.

What is the significance of the SEP token? In prior interpretability studies focusing on attention score distributions, it has been suggested that the SEP token in BERT functions as a “no-op,” receiving redundant attention [8, 34]. In the context of ranking models, Zhan et al. [34] find that document tokens attend heavily to the SEP token but probing experiments reveal it does not carry a strong relevance signal. In contrast, our findings in the earlier layers of the model reveal that non-important tokens heavily attend to the SEP token, while repeat occurrences of relevant tokens do not. In these instances, the SEP token indeed serves as a “no-op” for non-important tokens, but this allows the model to focus on extracting more important relevance information encoded in these repeated relevant tokens. Overall, much remains unknown regarding the functionality of the SEP token, and future research could explore the feasibility of identifying important or trivial terms through an analysis of attention patterns associated with the SEP token.

6.3 Implications for Future Research

Overall, our results show that term frequency can be localized to just a few attention heads in TAS-B, suggesting promising avenues for further investigation. From a broader perspective, our exploration of causal interventions opens up several new directions for XIR research. In this section, we outline additional avenues for future research that extend beyond the scope of our preliminary study.

Generalizability. While we find evidence of specific network components that encode a term frequency signal aligning with the TFC1 axiom, our study concentrates on a single model. This necessary focus allows us to establish a deeper understanding of this model and lay the foundation for the causal intervention framework for future extension to other models. Thus, to what extent other specific neural models and architectures incorporate the TFC1 axiom is an important direction for future work. Furthermore, in a broader context, the generalizability of axiomatic mechanisms provides an interesting line of investigation for future work. Overall, the straightforward nature of this framework not only facilitates the testing of established axioms but also opens avenues for establishing potential new axioms that may be introduced in the evolving landscape of axiomatic IR.

Interaction-Based Analysis. Although we only explore the four heads capable of fully recovering performance on the patched run, other heads exhibit varied impacts—some partially recover performance, while others may even harm it (Figure 6). Analysis

of the diverse behavior could be an interesting avenue for future exploration, in addition to examining interactions between heads by patching in multiple activations simultaneously, rather than focusing on individual interventions. An interaction-based approach could potentially provide a more holistic understanding of how different components interact to influence ranking decisions.

Going beyond interactions between model components, future work should investigate the interactions between query and document representations. Our work concentrates on analyzing how axiomatic concepts are encoded in a ranking model, specifically through its document representations. This emphasis arises from the characteristic of TAS-B, which independently encodes queries and documents. Consequently, our analysis does not have access to direct interactions between queries and documents. The choice to focus on document representations aligns with the architecture of TAS-B, providing a first foundational understanding of how axiomatic signals manifest in this specific model. While our analysis is tailored to TAS-B, we view these initial results as promising for future investigations that delve into interaction-based models. Exploring such models could reveal further insights into the encoding of axiomatic concepts, paving the way for a more comprehensive understanding of neural retrieval models.

Direct vs. Indirect Causal Effects. In this paper, we use activation patching to demonstrate the potential of isolating relevance computation within specific model components. Activation patching serves as a crucial starting point in understanding the underlying mechanisms of neural retrieval models and their alignment with human intuition. Once an understanding of the general feasibility of localizing model behavior is established, we can leverage the insights learned from activation patching to explore more sophisticated interventions to gain a deeper understanding of the inner workings of neural models. For example, while patching an activation can demonstrate its influence on the ranking score, it primarily indicates an indirect causal effect. In other words, we can achieve a more nuanced understanding of which downstream model components are affected by the patch that end up changing the final ranking score. To disentangle the direct causal effect of patching, future research may employ *path patching*. The process of path patching resembles activation patching but includes an additional forward pass that patches in the original downstream activations. For more details, we refer the reader to Wang et al. [33] and Goldowsky-Dill et al. [16].

Model Editing. Reverse engineering relevance computations to understand model behavior also serves as an initial step for various other avenues of XIR innovation. Localizing the internal mechanisms for relevance can lead to advancements in model ranking performance through component editing. For instance, in cases where the model may demonstrate erroneous behavior, model weights [21] or attention patterns [22] can be directly modified to promote more accurate performance. Similarly, employing causal interventions can help identify the presence and location of biases encoded within retrieval models, thus enabling corrective measures. As an example, Vig et al. [30] use causal interventions to detect gender bias in pre-trained Transformer language models. Future work in retrieval could test for similar sensitive concepts by constructing diagnostic datasets to reverse engineer where certain biases may

reside within neural retrieval models. Furthermore, this hypothesis testing framework can be used to detect the impact of adversarial attacks on model ranking performance and subsequently aid in designing interventions to mitigate such attacks.

7 CONCLUSION

In this perspectives paper, we design causal interventions to identify the concrete attention heads that encode a robust term frequency signal aligned with the TFC1 axiom. Our findings hold promise for future research, indicating the potential of employing mechanistic interpretability methods alongside diagnostic datasets to precisely identify where axiomatic concepts reside in neural ranking models and how relevance is computed. Going beyond the scope of diagnosing ranking models, the applications of causal intervention methods are widespread. They include model editing to enhance ranking performance, correction of potential biases, designing systems resilient to adversarial attacks, and investigating the generalizability of relevance components in ranking models. Overall, we hope this work can be a starting point for the information retrieval community for mechanistic interpretations of neural models that will lead to further insights into the inner workings of neural retrieval models.

ACKNOWLEDGMENTS

We thank Aaron Traylor for research discussions. This work was funded by NIH 3U54GM115677-08S1.

REFERENCES

- [1] Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics* 48, 1 (2022), 207–219.
- [2] Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics* 7 (2019), 49–72.
- [3] Peter D Bruza and Theo WC Huibers. 1994. Investigating aboutness axioms using information fields. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*. Springer, 112–121.
- [4] Arthur Câmara and Claudia Hauff. 2020. Diagnosing BERT with retrieval heuristics. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part I 42*. Springer, 605–618.
- [5] Jia Chen, Yiqun Liu, Yan Fang, Jiaxin Mao, Hui Fang, Shenghao Yang, Xiaohui Xie, Min Zhang, and Shaoping Ma. 2022. Axiomatically Regularized Pre-training for Ad hoc Search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1524–1534.
- [6] Zitong Cheng and Hui Fang. 2020. Utilizing Axiomatic Perturbations to Guide Neural Ranking Models. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*. 153–156.
- [7] Jaekel Choi, Euna Jung, Sungjun Lim, and Wonjong Rhee. 2022. Finding Inverse Document Frequency Information in BERT. *arXiv preprint arXiv:2202.12191* (2022).
- [8] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341* (2019).
- [9] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova Das-Sarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A Mathematical Framework for Transformer Circuits. *Transformer Circuits Thread* (2021). <https://transformer-circuits.pub/2021/framework/index.html>.
- [10] Yixing Fan, Jiafeng Guo, Xinyu Ma, Ruqing Zhang, Yanyan Lan, and Xueqi Cheng. 2021. A linguistic study on relevance modeling in information retrieval. In *Proceedings of the Web Conference 2021*. 1053–1064.
- [11] Hui Fang, Tao Tao, and ChengXiang Zhai. 2004. A formal study of information retrieval heuristics. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. 49–56.
- [12] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. A white box analysis of ColBERT. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II 43*. Springer, 257–263.
- [13] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2022. Match your words! a study of lexical matching in neural information retrieval. In *European Conference on Information Retrieval*. Springer, 120–127.
- [14] Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. *Advances in Neural Information Processing Systems* 34 (2021), 9574–9586.
- [15] Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. *arXiv preprint arXiv:2304.14767* (2023).
- [16] Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. 2023. Localizing model behavior with path patching. *arXiv preprint arXiv:2304.05969* (2023).
- [17] Matthias Hagen, Michael Völske, Steve Göring, and Benno Stein. 2016. Axiomatic result re-ranking. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 721–730.
- [18] Michael Hanna, Ollie Liu, and Alexandre Variengien. 2023. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *arXiv preprint arXiv:2305.00586* (2023).
- [19] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. In *Proc. of SIGIR*.
- [20] Sean MacAvaney, Sergey Feldman, Nazli Goharian, Doug Downey, and Arman Cohan. 2022. ABNIRML: analyzing the behavior of neural IR models. *Transactions of the Association for Computational Linguistics* 10 (2022), 224–239.
- [21] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems* 35 (2022), 17359–17372.
- [22] Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2023. Circuit Component Reuse Across Tasks in Transformer Language Models. *arXiv preprint arXiv:2310.08744* (2023).
- [23] Neel Nanda and Joseph Bloom. 2022. TransformerLens. <https://github.com/neelnanda-io/TransformerLens>.
- [24] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *choice* 2640 (2016), 660.
- [25] Judea Pearl. 2022. Direct and indirect effects. In *Probabilistic and causal inference: the works of Judea Pearl*. 373–392.
- [26] Daniël Rennings, Felipe Moraes, and Claudia Hauff. 2019. An axiomatic approach to diagnosing neural IR models. In *Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part I 41*. Springer, 489–503.
- [27] Corby Rosset, Bhaskar Mitra, Chenyan Xiong, Nick Craswell, Xia Song, and Saurabh Tiwary. 2019. An axiomatic approach to regularizing neural ranking models. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 981–984.
- [28] Procheta Sen, Debasis Ganguly, Manisha Verma, and Gareth JF Jones. 2020. The curious case of IR explainability: Explaining document scores within and across ranking models. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2069–2072.
- [29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
- [30] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems* 33 (2020), 12388–12401.
- [31] Michael Völske, Alexander Bondarenko, Maik Fröbe, Benno Stein, Jaspreet Singh, Matthias Hagen, and Avishek Anand. 2021. Towards axiomatic explanations for neural ranking models. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*. 13–22.
- [32] Jonas Wallat, Fabian Beringer, Abhijit Anand, and Avishek Anand. 2023. Probing BERT for ranking abilities. In *European Conference on Information Retrieval*. Springer, 255–273.
- [33] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593* (2022).
- [34] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. An analysis of BERT in document ranking. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 1941–1944.