Pathway to Relevance: How Cross-Encoders Implement a Semantic Variant of BM25

Meng Lu[*]
Brown University
meng_lu@brown.edu

Catherine Chen[*]
Brown University
catherine_s_chen@brown.edu

Carsten Eickhoff
University of Tübingen
carsten.eickhoff@uni-tuebingen.de

Abstract

Mechanistic interpretation has greatly contributed to a more detailed understanding of generative language models, enabling significant progress in identifying structures that implement key behaviors through interactions between internal components. In contrast, interpretability in information retrieval (IR) remains relatively coarse-grained, and much is still unknown as to how IR models determine whether a document is relevant to a query. In this work, we address this gap by mechanistically analyzing how one commonly used model, a cross-encoder, estimates relevance. We find that the model extracts traditional relevance signals, such as term frequency and inverse document frequency, in early-to-middle layers. These concepts are then combined in later layers, similar to the well-known probabilistic ranking function, BM25. Overall, our analysis offers a more nuanced understanding of how IR models compute relevance. Isolating these components lays the groundwork for future interventions that could enhance transparency, mitigate safety risks, and improve scalability.

1 Introduction

Information retrieval (IR) is the subfield of NLP that aims to rank a collection of documents by their relevance to a specific query. Traditional ranking strategies have long relied on probabilistic models grounded in intuitive heuristics, like BM25 (Robertson and Zaragoza, 2009), to estimate relevance. BM25 leverages term frequency (TF) and inverse document frequency (IDF) to rank documents effectively, achieving strong performance across various tasks. Its simplicity and inherent interpretability have made it a cornerstone of traditional IR systems. Inspired by BM25's success, earlier neural IR models (Pang et al., 2016; Guo et al., 2016) were purposefully designed to emulate BM25's principles. These models incorporate explicit components for semantic TF and IDF computations, blending neural architectures with established IR heuristics to improve relevance estimation.

However, the advent of transformer-based models revolutionized the field of IR. These models, trained end-to-end on large numbers of query-document pairs (Nguyen et al., 2016; Thakur et al., 2021), excel at extracting context-dependent semantic signals for ranking tasks. By leveraging multi-headed attention and vast parameter spaces, transformers (Vaswani et al., 2017) capture nuanced relationships between query and document terms that go beyond traditional heuristic-based approaches. Despite their superior performance, these models come with significant trade-offs: their complexity and lack of interpretability make it challenging to understand their internal mechanisms. This raises fundamental questions: how do these models assess relevance? Do they rely on established IR principles such as TF and IDF, or do they draw on entirely different expressions of relevance?

In this work, we build upon previous correlational studies (Choi et al., 2022; Zhan et al., 2020; Formal et al., 2021, 2022; MacAvaney et al., 2022) by employing mechanistic interpretability methods (Nanda, 2022; Elhage et al., 2021; Olsson et al., 2022; Meng et al., 2024; Wang et al., 2022; Pearl, 2022) to address these questions.

Concretely, we analyze a BERT-based IR model (i.e., a cross-encoder) to isolate multiple relevance signals beyond exact match TF.[1] Our main findings are: (1) We identify attention heads that extract and process BM25-like components, including a semantic version of term frequency (soft-TF), term saturation, and document length (§4.1-§4.3). (2) We find evidence that IDF information largely already exists in the largest singular value of the

[*]Equal contribution.

[1]All code and resources are available at: https://github.com/mlu108/CrossEncoderBM25

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 25537–25559
November 4-9, 2025 ©2025 Association for Computational Linguistics

embedding matrix and can be manipulated to later control term importance (§4.4). (3) We confirm that heads in later layers aggregate all these relevance signals in a BM25-style manner by defining a linear approximation of the hypothesized relevance computation and evaluating its ability to reconstruct cross-encoder's predicted scores, confirming that our circuit captures the core mechanism of relevance computation (§5).

Overall, our mechanistic analysis uncovers insights on how IR models determine relevance, paving the way for targeted model editing to boost performance, enable personalization, and mitigate bias in transformer-based IR.

2 Background and Related Work

2.1 Axioms and BM25

Axiomatic IR constructs formal desiderata, or axioms, outlining specific properties that an effective ranking model should satisfy (Bruza and Huibers, 1994). For example, the TFC1 axiom (Fang et al., 2004) states that documents that contain more query terms should be ranked higher, and the TDC axiom (Fang et al., 2004) states that documents containing query terms with higher inverse document frequency (IDF) should receive higher scores. These axioms provide a theoretical foundation for understanding and developing ranking functions by formalizing human-interpretable and intuitive notions of relevance into mathematical constraints.

BM25 is a widely used probabilistic ranking function that exemplifies these axiomatic principles by ranking documents based on term frequency (TF), inverse document frequency (IDF), term saturation, and document length. It is defined as:

$$\sum_{t \in q} \text{IDF}(t) \cdot \frac{\text{TF}(t,d) \cdot (k_1 + 1)}{\text{TF}(t,d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{\text{avgdl}})} \quad (1)$$

where q is the query, d is the document, t is a query term, k_1 and b are hyperparameters controlling term saturation and length normalization, respectively. $|d|$ is the length of document d, and avgdl is the average document length in the corpus.

Intuitively, BM25 computes a relevance score by multiplying each query term's term frequency (TF) in the document with its inverse document frequency (IDF), capturing both how relevant and how informative the term is. It then sums these scores across all query terms with two adjustments: diminishing returns for repeated terms (term saturation via k_1), and downweighting for longer documents (length normalization via $|d|$), which prevents overly long documents with additional query term occurrences from being unfairly favored. As shown in Table 1, each component of BM25 aligns with a specific axiom. For instance, TFC1 reflects the TF factor, which favors documents containing more query terms. Framing BM25 in terms of axioms allows us to mechanistically analyze whether and how neural IR models internalize these principles when estimating relevance.

2.2 Mechanistic Interpretability for IR

Despite not being explicitly trained to encode traditional relevance concepts, previous work provides correlational evidence suggesting that BERT-based IR models encode BM25-like information (Wang et al., 2021; Rau and Kamps, 2022; Yates et al., 2021; MacAvaney et al., 2022; Choi et al., 2022; Zhan et al., 2020; Formal et al., 2021, 2022).

To move beyond correlation and gain a causal understanding of how relevance is computed, we turn to mechanistic interpretability (Nanda, 2022; Elhage et al., 2021; Pearl, 2022), which has been instrumental in uncovering how transformer-based NLP models perform certain tasks, such as indirect object identification (Meng et al., 2024) and greater-than computation (Hanna et al., 2024).

In the context of IR, Chen et al. (2024) apply activation patching (Vig et al., 2020; Meng et al., 2024) to identify attention heads responsible for exact term matching. We build on this work by decomposing BM25 into intuitive IR axioms and constructing diagnostic datasets to localize each component. Rather than focusing solely on exact matches, we extend the analysis to semantic matches and trace how these components interact to compute the final relevance score using path patching (Wang et al., 2022; Goldowsky-Dill et al., 2023). This work is the first to causally uncover an internal circuit for the relevance estimation mechanism of neural IR architectures.

3 Methodology

3.1 Model

While many early neural models were purposefully designed to emulate BM25's principles, transformer-based models have revolutionized the field of IR. Among them, cross-encoders represent a specific architectural approach to neural ranking

BM25 Component	Axiom	Perturbation	Axiom-Based Diagnostic Dataset Examples (b: baseline doc, p: perturbed doc)
TF (*soft-TF*)	**TFC1 (Fang et al., 2004)**: Prefer documents with more query term occurrences.	Given a baseline document, we perturb it by appending one more selected query term.	***b*:** Quebec is a small city in Canada. ***p*:** Quebec is a small city in Canada. Quebec.
	STMC1 (Fang and Zhai, 2006): Prefer documents with semantically similar terms.	Given a baseline document, we append a semantically similar term - the synonym with highest embedding cosine similarity with query term from 20 candidates generated by GPT-4o.	***b*:** Quebec is a small city in Canada. ***p*:** Quebec is a small city in Canada. Toronto.
IDF	**TDC (Fang et al., 2004)**: Prefer documents with more discriminative query terms.	N/A	N/A
k	**TFC2 (Fang et al., 2004)**: Additional occurrences yield smaller improvements.	We create a baseline document using GPT-4o to create five relevant sentences starting with selected query term's pronoun and perturb by incrementally restoring the term.	***b*:** It is in Canada. It is fun. ***p*$_1$:** Quebec is in Canada. It is fun. ***p*$_2$:** Quebec is in Canada. Quebec is fun.
b	**LNC1 (Fang et al., 2004)**: Penalize longer documents for non-relevant terms.	Given a baseline document, we create five perturbations by sequentially appending an increasing number of random sentences from a non-relevant query in the base dataset.	***b*:** Quebec is in Canada. ***p*$_1$:** Quebec is in Canada. Road not taken is written by Frost. ***p*$_2$:** Quebec is in Canada. Road not taken is written by Frost. Happiness is a Butterfly.

Table 1: Mapping of BM25 components to IR axioms. Note: Since IDF is a distinct component in BM25 (unlike term saturation and document length, which are tied to TF), we use alternative methods to isolate it (§4.4).

models that process query-document pairs jointly.

Given input `x` in the format: `<CLS> query <SEP> document <SEP>`, the model is trained to optimize a binary classification task where the `<CLS>` token is passed to a classifier to determine whether the provided query is relevant to the document. In this work, we choose to examine ms-marco-MiniLM-L-12-v2 (Face, 2025), a BERT-based cross-encoder, for its high performance on common IR benchmarks (Nguyen et al., 2016; Craswell et al., 2020).

3.2 Diagnostic Datasets Construction

To investigate if and how the model implements BM25, we map BM25 components to IR axioms (§2.1) and construct diagnostic datasets that isolate individual axiomatic components. We build off of Chen et al. (2024), who create a diagnostic dataset for analyzing TFC1 using MS-MARCO (Nguyen et al., 2016), consisting of 10k web search query-document pairs. To create a diagnostic dataset for the remaining axioms, we perturb each query-document pair in this base dataset following their formal axiomatic definition (Table 1). Each perturbed sample differs minimally from the original, with the perturbation introducing an additional signal corresponding to the axiom. Our perturbation strategies and examples of input pairs are shown in Table 1. Additionally, we expand the analysis on TF to *soft-TF* to include semantic matches, motivated by findings that neural models often score documents highly even in the absence of exact lexical matches (Rau and Kamps, 2022).

3.3 Path Patching on Diagnostic Datasets

We first begin by tracking term matching, the core feature of classical IR models like BM25, by applying path patching on minimal pairs in TFC1 and STMC1 diagnostic datasets to uncover the soft-TF mechanism. For other components in BM25, we use these datasets to further inspect how they interact with the main soft-TF circuit.

Specifically, we isolate soft-TF by selecting a query, a baseline document b, and a perturbed document p, where p introduces an additional semantic or lexical term frequency signal. We then substitute activations from p into b at specific model components, such as individual attention heads or MLP layers. If this substitution causes the model's relevance score for b to shift toward that of p, we

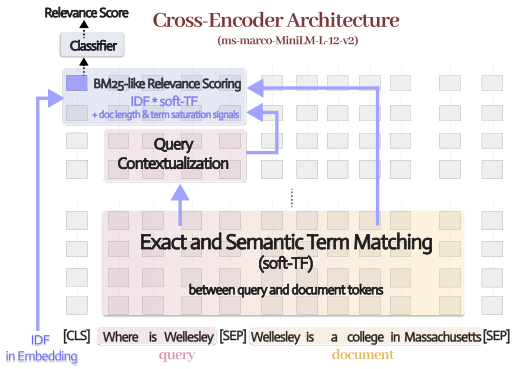


Figure 1: Overview of relevance mechanisms in the model. The model first jointly analyzes query and document tokens to identify matching terms (exact and semantic), then contextualizes each query term within the full query, and finally calculates a relevance score by weighting query terms by their importance (IDF) and aggregating them, similar to BM25.

interpret it as causal evidence that the patched component encodes a term matching signal.

The path patching algorithm proceeds iteratively in a backward fashion: it first identifies components directly responsible for changes in the output logits, then recursively traces upstream to uncover the full set of causally relevant components. This backward search reveals the components through which specific information, introduced by the perturbation, flows (more details in Appendix C).

4 Semantic Scoring Circuit

In the course of this section, we will identify and localize the following components in the model's *Semantic Scoring Circuit* (overview in Figure 1, detailed walk-through example in Figure 10):

- *Matching Heads* locate exact and semantic term matches in the document for each query term, while also encoding term saturation and document length signals.
- *Query Contextualization Heads* distribute the matching signal from higher-IDF query tokens to all query tokens.
- *IDF* of each term is stored in a dominant low-rank vector of the model's embedding matrix.
- *Relevance Scoring Heads* aggregate query-term importance by combining the matching signal (soft-TF) and IDF for each query term in a manner similar to BM25.

As described in §3, we begin by uncovering the core soft-TF mechanism via a "backward pass" of the model: path patching to the output logits, then progressively patching to earlier components to sequentially uncover important heads that compute and propagate soft-TF signals.

After localizing the soft-TF mechanism, we apply Singular Value Decomposition (SVD) on the embeddings to investigate IDF. After isolating all BM25-like components in this section, we verify that Relevance Scoring Heads perform a BM25-like computation in §5.

4.1 Relevance Scoring Heads

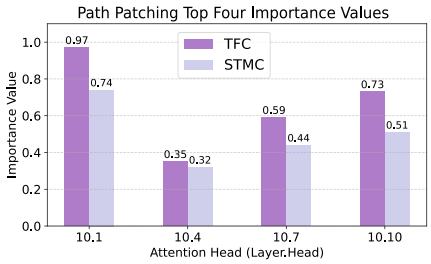


Figure 2: Path patching identifies heads 10.1, 10.4, 10.7, and 10.10 as the most important carriers of soft-TF signals to [CLS] on both TFC1 and STMC1, with similar patching effects.

Information Flow. To identify the components that directly transmit soft-TF signals to the relevance scores, we path patch to the logits using the TFC1 and STMC1 diagnostic datasets. We observe a high Pearson correlation between the patching effects of TFC1 and STMC1 ($r = 0.99$, $p < 0.001$), indicating that path patching on both datasets identifies essentially the same set of influential heads. This suggests that the model treats exact and semantic matches similarly in the final relevance computation (see Figure 15 in Appendix D for detailed patching results).

Figure 2 displays the heads with the highest patching effects that exceed the top 30% of causal importance for both TFC1 and STMC1: 10.1 (Layer 10, Head 1), 10.4, 10.7, and 10.10. Notably, these heads contribute a larger change in relevance scores for TFC1 compared to STMC1, suggesting that the model may prioritize exact matches over semantic ones.

Component Behavior. Qualitative analysis reveals that the [CLS] token selectively attends to query tokens, indicating the movement of soft-TF signals into its representation to prepare for the final relevance score in the model's classification head. We find that the heads distribute attention to

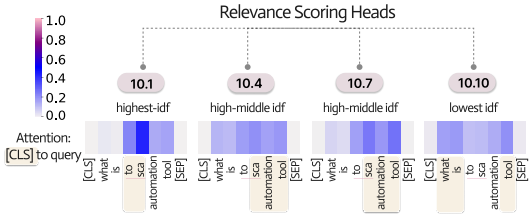


Figure 3: Example of the attention pattern from [CLS] to query tokens in the Relevance Scoring Heads, illustrating how these heads process *soft-TF* for specific query tokens based on their IDF values.

different parts of the query. Specifically, 10.1 focuses on high-IDF query terms, 10.10 on low-IDF, and 10.7 and 10.4 on mid-to-high-IDF (Figure 3). We verify this distributive behavior by calculating the Pearson correlation between each head's attention distribution and the IDF values of query tokens and find a moderately high average (r=0.67, $p <$ 0.01).

Comparison to BM25. Because these heads prepare the [CLS] token for the final scoring in the classification head, we call them *Relevance Scoring Heads*. They combine IDF and soft-TF signals per query term, resembling BM25's term weighting mechanism. Thus, we hypothesize that the model combines the outputs of the Relevance Scoring Heads in a similar manner and verify this hypothesis in §5.

Important Upstream Components. We path patch to the Relevance Scoring Heads' value vectors to identify which upstream components transmit soft-TF signals. On both TFC1 and STMC1, patching effects show high average correlation of 0.83 (p <0.001), revealing a shared set of attention heads, divided into two groups: (1) Query Contextualization Heads (§4.2) which redistribute soft-TF among query tokens and (2) Matching Heads (§4.3) which detect exact and semantic query-document matches (soft-TF) (see Appendix D).

4.2 Query Contextualization Heads

Query Contextualization Heads (8.10 and 9.11) aggregate soft-TF signals of higher-IDF query tokens and distribute them across all query tokens, strengthening their representations for the Relevance Scoring Heads' final computation.

Component Behavior. At the Relevance Scoring Heads, the [CLS] token retrieves soft-TF from all query tokens. Thus, we analyze the attention patterns among query tokens in these two intermediary heads to understand how they modify the query token representation in the residual stream. We find that all query tokens in these heads consistently focus on one or two higher-IDF query tokens, with strong correlations to IDF values (9.11: $r = 0.829$, 8.10: $r = 0.781$) with both $p < 0.001$. This suggests that 8.10 and 9.11 redistribute the soft-TF signals of higher-IDF tokens to contextualize the entire query for the Relevance Scoring Heads.

Comparison to BM25. BM25 weights each query term independently, with no exchange of information between terms. In contrast, Query Contextualization Heads seem to learn to contextualize the entire query, redistributing soft-TF signals to amplify high-IDF tokens and dynamically reweight terms, before the final scoring. Further study of this learned contextualization is left for future work.

Important Upstream Components. Path patching to 8.10 and 9.11 confirms that these heads receive soft-TF signals from the Matching Heads (more details in Appendix D).

4.3 Matching Heads

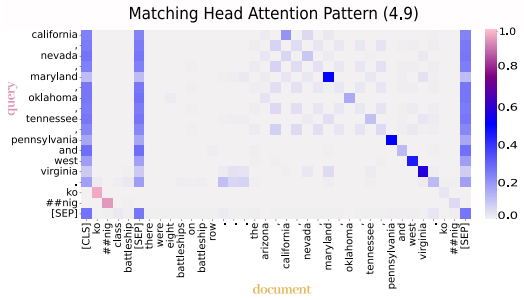


Figure 4: Example attention pattern of Matching Head 4.9: tokens attend most strongly to duplicates but also mildly to similar tokens.

Alongside Query Contextualization Heads, a separate set of heads, named *Matching Heads* (0.8, 1.7, 2.1, 3.1, 4.9, 5.7, 5.9, 6.3, 6.5, 7.9, 8.0, 8.1, 8.8), in the model's early and middle layers detect exact and semantic token matches (soft-TF) between the query and document and pass these signals to the Relevance Scoring Heads.

Component Behavior. Qualitative inspection reveals that Matching Heads actively attend to both exact and semantically similar matches across the entire input sequence (i.e., query and document). As shown in Figure 4, head 1.7 exhibits this behavior clearly: both query and document tokens strongly attend to duplicated terms and, to a lesser extent, to semantically related terms.

To quantitatively verify this behavior, we compute the Pearson correlation between atten-

tion weights and semantic similarity for each query–document token pair. If a head indeed matches semantically similar words, its attention from a query token q_i to a document token d_j should increase with $\cos(\mathbf{q}_i, \mathbf{d}_j)$. On average, Matching Heads exhibit a substantially stronger correlation ($r = 0.500$, $p < 0.001$) compared to other heads ($r = 0.132$). This evidence suggests that Matching Heads employ an attention mechanism that scales proportionally with semantic proximity: higher semantic similarity corresponds to higher attention scores. Since each attention weight quantifies the strength of token matching, the sum of attention values from a query token to all document tokens effectively approximates the soft-TF of a query term.

We validate the Matching Heads' importance by mean-ablating them, and find that the average logits across both TFC1 and STMC1 perturbed samples decrease (TFC1: 5.146 to -4.394, STMC1: -1.998 to -4.611), indicating that these heads are critical for computing a semantic version of term matches.

Additional Relevance Signals. Since BM25's TF term takes two additional signals, term saturation and document length, into account, we also investigate whether they affect Matching Heads. To isolate these effects, we use two diagnostic datasets: (1) TFC2, which increases repeated query term occurrences to test for term saturation, and (2) LNC1, which adds irrelevant sentences to simulate the isolated effect of increased document length (Table 1).

Because each dataset varies only in one factor, we track each Matching Head's average attention across these controlled groups. Consistent with how BM25 controls for term saturation and document length effects, we observe two trends (additional details in Appendix E): (1) in TFC2, attention to a term increases sharply after the initial occurrence, but plateaus with additional term repetitions, consistent with term saturation, and (2) in LNC1, attention decreases as irrelevant content increases, despite constant relevance, indicating sensitivity to document length.

These findings suggest that Matching Heads integrate soft-TF with saturation and document-length signals rather than operating in isolation. We define this composite signal as the *Matching Score*, reflecting its incorporation of soft term frequency, term saturation, and document length effects, three core signals of BM25.

Comparison to BM25. Matching Heads implement a semantic variant of the TF component in BM25 by identifying and weighting query-document token matches. Our TFC2 and LNC1 experiments show that they also go beyond simple match counting: their attention outputs are modulated in ways that mirror BM25's additional relevance adjustments for term saturation and document length normalization. Thus, Matching Heads jointly capture all three core components of BM25, TF, term saturation, and length normalization, while also incorporating semantic similarity.

Important Upstream Components. Path patching reveals that the embeddings are the primary input to Matching Heads (see Appendix D.2), confirming they generate soft-TF based on semantic similarity from the embeddings. This completes the backward tracing for the soft-TF circuit.

4.4 IDF in the Embedding Matrix

In addition to TF, IDF plays a critical role in BM25 by weighting each query token's TF contribution, allowing the model to prioritize more uncommon terms. As shown in §4.1, IDF similarly emerges as a key signal for the Relevance Scoring Heads in our model. While Choi et al. (2022) provide correlational evidence for IDF in the embedding matrix, we use SVD and low-rank interventions to causally localize their functional role in the model.

SVD allows us to decompose a matrix into a sum of orthogonal rank-1 components, outer products of a column and row vector, ordered by their contribution to the overall matrix. Recall that the embedding matrix W_E is structured such that each row corresponds to a word's representation in the vocabulary, while the columns represent latent feature dimensions. By applying SVD to W_E, we can analyze the dominant directions to see what signals (e.g., IDF) drive the model's behavior. Mathematically, the SVD of W_E is expressed as:

$$W_E = USV^T = \sum_{i=1}^{r} \sigma_i \, u_i \, v_i^T$$

Here, σ_i are the singular values, which quantify the importance or strength of each rank-1 component; u_i and v_i left and right singular vectors, respectively, forming orthonormal bases that capture patterns in the row (token embeddings) and column spaces (feature dimensions); and r is the rank of W_E, representing the number of non-zero singular values and independent directions in the matrix. By focusing on the largest singular values (σ_i) and their corresponding singular vectors

(u_i, v_i), we can study whether IDF is stored in dominant components of the embedding matrix.

We find that the top singular vector U_0 is highly correlated ($r = -71.36\%$) with MS-MARCO IDF values (Nguyen et al., 2016), the model's training dataset, indicating that IDF is encoded in the embedding matrix's dominant low-rank component.

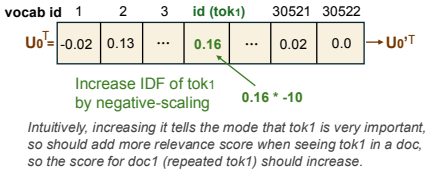


Figure 5: U_0 editing example (transposed for visualization). Since U_0 is negatively correlated with IDF, increasing `tok1`'s IDF, representing its importance in relevance computation, requires negatively scaling its U_0 component.

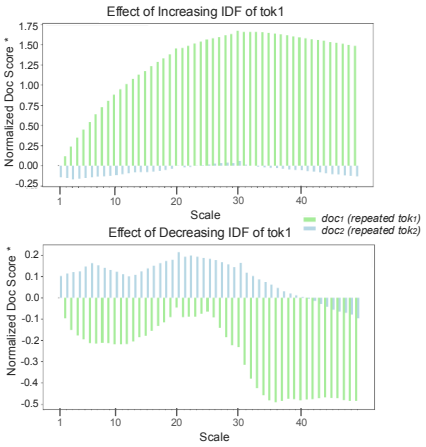
Causal Experiment. To validate this correlation, we perform an intervention to demonstrate that the IDF values from U_0 have a causal effect on downstream components (i.e., relevance scoring heads) and thus, the overall relevance computation.

Given previous understanding on U_0, we can interpret U_0 as a 1-D IDF dictionary, where $\text{idx}(q_i)$ corresponds to the vocabulary index of q_i, mapping to its respective IDF value. Modifying the value at $\text{idx}(q_i)$ in U_0 allows us to adjust the importance of the Matching Score for q_i (Figure 5).

If the model uses the IDF values encoded in U_0, then modifying these values should result in corresponding changes to the ranking score. Specifically, according to the BM25-based Scoring Hypothesis in §4.1, there is a linear relationship between the IDF of a query token q_i and the relevance score: increasing the IDF of q_i should increase the relevance score. We have shown that U_0 is rank-1 and negatively correlated with IDF. If the model indeed uses the IDF values encoded in U_0, we can increase $\text{IDF}(q_i)$ by decreasing q_i's value in U_0, which would directly increase the relevance score. The converse should hold true for decreasing IDF.

To test this, we design an experiment with controlled, minimal examples. Given each query from our base dataset (e.g., "computer science department number"), we create two documents where `doc1` repeats the first query token (e.g., "computer computer computer" and `doc2` repeats the second query token (e.g., "science science science"). Then, we can edit the IDF of `tok1` and measure the effect on the relevance scores of `doc1` and `doc2`.

Figure 6 shows that scaling `tok1`'s IDF up or down causes a corresponding monotonic increase (or decrease) in `doc1`'s score, providing causal evidence that the model uses U_0 to encode IDF and suggests that it sums soft-TF by IDF values as hypothesized in §4.1 (see additional details about IDF and Relevance Scoring Heads' attention pattern correlation in Appendix F.1).



* We normalize the scores so that the average score of the unscaled doc1 is 0.

Figure 6: *Top*: Changes in scores for `doc1` (repeated `tok1`) and `doc2` (repeated `tok2`) when increasing `tok1`'s IDF at different scales, averaged across all samples. Increasing `tok1`'s IDF raises `doc1`'s score more than `doc2`'s. *Bottom*: Decreasing `tok1`'s IDF produces the inverse effect, with slight non-monotonic deviations suggesting optimal editing windows.

Comparison to BM25. This completes the BM25 component set: the embeddings encode IDF, and Relevance Scoring Heads modulate the amount of soft-TF extracted by the `[CLS]` token based on that IDF. This is similar to BM25, where each query term's TF is weighted by its IDF to prioritize informative terms. In the next section, we formally validate that Relevance Scoring Heads combine soft-TF and IDF in a BM25 manner to compute the final relevance score.

5 Validation of BM25-like Computation

To validate whether Relevance Scoring Heads perform a BM25-style computation as hypothesized in §4.1, we first formalize the hypothesized function of the heads and the information flowing into them as a BM25-style linear function. Next, we evaluate this linear model's ability to reconstruct the cross-encoder scores by examining how well the hypothesized linear model fits the data.

5.1 Formalizing the Hypothesized Function

In §4.1, we hypothesize that Relevance Scoring Heads compute a summation of Matching Scores weighted by IDF. The Matching Score incorporates soft-TF, along with term saturation and document length signals, all of which are components of the BM25 function. If this hypothesis that these components interact in a BM25-like manner holds, then the *Semantic Scoring Circuit* can be expressed as a linear function:

$$\sum_{i=1}^{N} \text{linear\_combo}\Big(-U_0(q_i),\ \text{MS}_{\text{total}}(q_i, d_j), \\ -\ U_0(q_i)\ \cdot\ \text{MS}_{\text{total}}(q_i, d_j)\Big) \tag{2}$$

where the components of the linear combination are defined as follows:

1. $U_0(q_i)$: The value of q_i in the U_0 vector, representing the model's interpretation of q_i's IDF.

2. $\text{MS}_{\text{total}}(q_i, d_j)$: The total Matching Score, computed as the sum of Matching Scores from individual heads (MS_{H_k}) weighted by learned weights α_k. Each Matching Score represents the sum of attention values from q_i to all document tokens:

$$\text{MS}_{\text{total}}(q_i, d_j) = \sum_{k=1}^{13} \alpha_k \cdot \text{MS}_{H_k}$$

3. $-U_0(q_i) \cdot \text{MS}_{\text{total}}(q_i, d_j)$: The interaction term, modeling the product of IDF and TF in BM25.

By incorporating both the hypothesized computation function and the earlier components U_0 and MS, the linear model effectively represents the hypothesized circuit.

5.2 Assessing the Linear Model Fit

We test whether our hypothesis holds by comparing the linear model's effectiveness against the cross-encoder's actual relevance scores.

First, we train a linear regression model using our base dataset, limiting queries to five tokens to keep the number of coefficients manageable. For each forward pass, we extract two features for each query token: (1) the MS (Matching Score), calculated as the sum of the query token's attention over document tokens from the 13 Matching Heads, and (2) its value along the top singular vector in U_0 of the decomposed embedding matrix. These features form the input x to the linear regression model, while the cross-encoder's relevance scores are the target y. Finally, we evaluate how well the linear representation of the Semantic Scoring Circuit predicts the cross-encoder's relevance scores with an 80/20 train-test split.

The linear regression model achieves a high Pearson correlation ($r = 0.8157$, $p < 0.001$) with ground-truth relevance scores, showing that it captures the core of the cross-encoder's scoring mechanism in a simplified and interpretable form. This correlation surpasses that of the traditional BM25 scoring function under optimized parameters ($k = 0.5, b = 0.9$; corr = 0.4200, $p < 0.001$), which demonstrates that the discovered U_0 and MS components effectively capture the signals that the cross-encoder utilizes for ranking. The linear model's strong generalization to unseen datasets and varying query lengths (details in Appendix G) further confirms that our circuit understanding captures the cross-encoder's core mechanism for relevance computation.

6 Discussion

6.1 A Two-Stage Process for Relevance Computation

Similar to Tenney et al. (2019), who find that BERT rediscovers the classical NLP pipeline, encoding syntax in its lower layers and semantics in its upper, our circuit analysis reveals a corresponding two-stage process. In the early and middle layers, Matching Heads extract lexical signals (e.g., soft-TF matches), reflecting prior work that argues document representations are query-dependent (Qiao et al., 2019). In the upper layers, Contextual Query Representation and Relevance Scoring Heads aggregate these signals into BM25-like relevance scores, consistent with Zhan et al. (2020), who argue that document tokens are largely query-independent. This dual-stage mechanism reconciles these two conflicting viewpoints by showing how query-specific signals migrate from document to query representations across layers and provides a more nuanced understanding of how relevance is computed.

6.2 Potential Downstream Applications

In §4.4, we show how model editing can be used to *upscale* or *downscale* term importance by modifying the encoded IDF values in the embeddings, effectively introducing "tuning dials" for fine-grained control over model behavior. These insights enable

two potential downstream applications.

6.3 Mitigating Adversarial Attacks

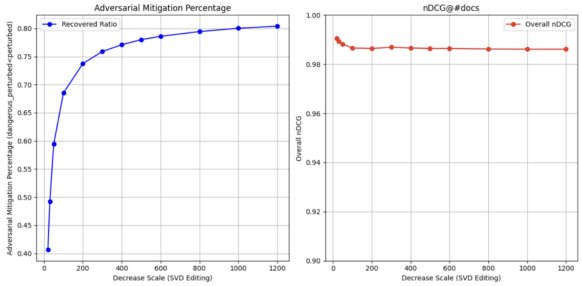


Figure 7: *Left*: Decreasing the IDF of dangerous tokens shows an increasing trend in adversarial mitigation proportion. *Right*: The intervention maintains a high NDCG at all levels, demonstrating that the general ranking capability of the cross-encoder remains unaffected.

First, targeted model editing could mitigate unsafe or adversarial content without impairing the model's ranking effectiveness.

As an initial test, we construct a dataset using obscene and offensive words (LDNOOBW, 2015), filtering out multi-word entries and injecting these unsafe tokens into the safe samples of our TFC datasets. Next, we inspect the subgroup of queries and documents where inserting an unsafe token in the document significantly increases the relevance score. Our goal is to "erase" the effect of the dangerous token by reducing its importance.

Among 17,537 adversarial samples, our approach achieves a 80.396% success rate (i.e., the unsafe document is ranked lower than the safe document). For example, when we downweight the target unsafe term by a large factor (e.g., -1200), the model maintains a high nDCG across all ranks, with a score of 0.9861, which is only a 1.39% drop in ranking performance.

These results provide preliminary evidence for the potential of localized editing in the IDF-storing low-rank matrix.

6.4 Parameter Efficient Fine-Tuning

Second, aligning internal representations with ground-truth IDF scores could serve as a powerful initialization strategy for fine-tuning, leading to more efficient retrieval pipelines. In other words, rather than updating the entire parameter space, fine-tuning the embedding matrix to better encode IDF values aligned with the retrieval domain may suffice.

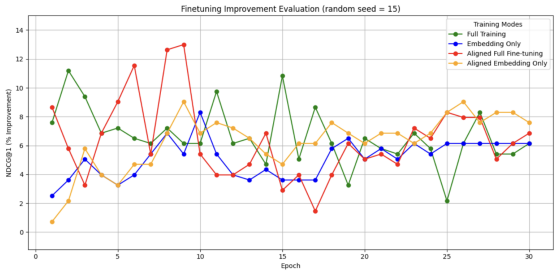


Figure 8: For random seed 15, aligned full fine-tuning achieves the best performance, and aligned embeddings only achieve performance compared to full fine-tuning.

To test this hypothesis with a preliminary experiment, we use the nfcorpus data, a Nutrition Fact retrieval dataset, from BEIR (Thakur et al., 2021). We fine-tune on 10 epochs and evaluate performance across four conditions: (A1) full fine-tuning, (A2) IDF-aligned full fine-tuning (aligning token IDFs to the dataset using the model-editing method from §4.4), (B1) embedding-only fine-tuning, and (B2) IDF-aligned embedding-only fine-tuning.

The results in Figure 8 across three random seeds show that IDF-aligned full fine-tuning (A2) achieves the highest NDCG@1, while IDF-aligned embedding-only fine-tuning (B2) outperforms unaligned embedding-only fine-tuning (B1). Additionally, full fine-tuning (A1) and embedding-only fine-tuning (B1) show no significant difference, supporting the idea that adaptation primarily refines the embedding matrix rather than the full model weights.

These findings reveal preliminary indications that fine-tuning just on the embedding matrix could potentially be sufficient to adapt cross-encoders to new domains and reveal IDF-aligned initialization as a possible pathway for efficient adaptation.

7 Conclusion

In this work, we mechanistically uncover the core components of the relevance scoring pathway in a BERT-based IR model, revealing how it leverages contextual representations to implement a semantic variant of BM25. Our fine-grained analysis lays the foundation for applying interventions to build more transparent and controllable models, enabling personalization, bias mitigation, and parameter-efficient adaptation. More broadly, identifying universal components or architecture-specific mechanisms contributes to the interpretability of IR models and informs the design of controllable, efficient ranking systems for real-world deployment.

Limitations

In this work, we focus on analyzing the behavior of a single cross-encoder in order to deeply analyze its mechanisms. Future work should investigate the generalizability of this behavior to other neural IR models with the same or different architectural base.

Additionally, while the Semantic Scoring circuit bears a strong resemblance to BM25, there may also be other factors influencing relevance computation that are out of the scope of investigation for this work. In this work, we compare it to BM25 because of its well-known status and the fact that previous research has shown that neural IR encode a BM25-like relevance signal (Wang et al., 2021; Rau and Kamps, 2022; Yates et al., 2021; MacAvaney et al., 2022). However, there are other term-matching models (e.g., TF-IDF, QL, etc.) that also rely on similar retrieval heuristics, and it is possible that the model implements these as well.

Furthermore, we note that our BM25-like approximation of the Semantic Scoring circuit does not fully approximate the cross-encoder's true relevance scores. There are a couple potential explanations for this: (1) the non-linearity of neural models, which our linear regression model does not capture, (2) potentially unexplored components such as additional attention heads that could be uncovered through a different choice of dataset or multi-layer perception (MLP) layers not covered in our analysis, and (3) the incorporation of signals beyond traditional relevance heuristics, such as real-world knowledge learned during pre-training (MacAvaney et al., 2022). We leave the investigation of these avenues for future work.

Finally, although we intend for our analysis to inform future work on efficiency, transparency, and performance improvements, like most research, it is possible that a malicious actor could apply these insights to produce IR models that are robust to adversarial use cases.

Acknowledgments

We thank the Health NLP Lab for valuable suggestions and discussions that improved this work.

References

Peter Bruza and Theo W. C. Huibers. 1994. Investigating aboutness axioms using information fields. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

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*, pages 605–618. Springer.

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)*, pages 1401–1410. ACM.

Jaekeol Choi, Euna Jung, Sungjun Lim, and Wonjong Rhee. 2022. Finding inverse document frequency information in bert. *Preprint*, arXiv:2202.12191.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. In *Proceedings of The Twenty-Eighth Text REtrieval Conference (TREC 2019)*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, and 1 others. 2021. A mathematical framework for transformer circuits. Transformer Circuits Thread Blogpost. Accessed: 2025-01-13.

Hugging Face. 2025. ms-marco-minilm-l-12-v2. `https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-12-v2`.

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*, pages 49–56.

Hui Fang and ChengXiang Zhai. 2006. Semantic term matching in axiomatic approaches to information retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 115–122.

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*, pages 257–263. Springer.

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*, pages 120–127. Springer.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. 2023. Localizing model behavior with path patching. *Preprint*, arXiv:2304.05969.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 55–64.

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2024. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Advances in Neural Information Processing Systems*, 36.

LDNOOBW. 2015. List of dirty, naughty, obscene, and otherwise bad words. Accessed: 2025-05-19.

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*, 10:224–239.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2024. Locating and editing factual associations in gpt. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Neel Nanda. 2022. A comprehensive mechanistic interpretability explainer & glossary. Accessed: 2025-01-13.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, and Rangan Majumder. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches*.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, and 7 others. 2022. In-context learning and induction heads. *Preprint*, arXiv:2209.11895.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Judea Pearl. 2022. *Direct and Indirect Effects*, 1 edition, page 373–392. Association for Computing Machinery, New York, NY, USA.

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

David Rau and Jaap Kamps. 2022. How different are pre-trained transformers for text ranking? In *European Conference on Information Retrieval*, pages 207–214. Springer.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

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.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 6019–6031.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

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. *Preprint*, arXiv:2211.00593.

Shuai Wang, Shengyao Zhuang, and Guido Zuccon. 2021. Bert-based dense retrievers require interpolation with bm25 for effective passage retrieval. In *Proceedings of the 2021 ACM SIGIR international conference on theory of information retrieval*, pages 317–324.

Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. 2021. Pretrained transformers for text ranking: Bert and beyond. In *Proceedings of the 14th ACM International Conference on web search and data mining*, pages 1154–1156.

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*, SIGIR '20, page 1941–1944, New York, NY, USA. Association for Computing Machinery.

A Activation Patching on Attention Heads

Prior to path patching, we conduct an exploratory analysis to compare the similarity between exact and semantic matches. Figure 9 shows that the patching effect has a high correlation of 0.96, suggesting that the model employs highly similar components for both exact and soft matches. These initial results on the indirect effect of these components provide a foundation to investigate their direct effects further with path patching.

B Example of Semantic Scoring Circuit

Figure 10 shows an example of the hypothesized semantic scoring circuit.

In the figure, the model receives an example input: "`[CLS] Where is Wellesley [SEP] Wellesley is a college in Massachusetts. [SEP]`".

(1) Matching Heads: These heads generate $MS(\texttt{where})$, $MS(\texttt{is})$, and $MS(\texttt{wellesley})$ that mainly capture query term soft-TF values. For example, the query token `wellesley` strongly attends to `wellesley` in the document, and more weakly to related terms like `wellesley` and `college`. This attention pattern forms a Matching Score $MS(\texttt{wellesley})$ that also reflects document length normalization (the longer the document, the smaller the soft $MS(\texttt{wellesley})$) and term saturation effects (the more `wellesley` exists in the document, the less additional gain in MS).

(2) Query Contextualization Heads: These heads redistribute the soft-TF signal from key content words to surrounding query tokens. For instance, `where` and `is` may receive part of `wellesley`'s soft-TF signal, allowing the model to amplify high-IDF query terms' soft-TF signals.

(3) IDF storage: The IDF of all query terms is stored in a dominant low-rank vector of the model's embedding matrix.

(4) Relevance Scoring Heads: Finally, the model aggregates the reweighted soft-TF signals across query tokens using a mechanism similar to BM25.

C Detailed Path Patching Methodology

The path patching algorithm is an iterative backward process that starts with identifying which upstream components send important information to the logits (Figure 11). Specifically, it involves four forward passes through the model to identify which upstream (sender) components send information to the target (receiver) component (i.e., logits): (1) Run on the baseline input x_b and cache activations. (2) Run on the perturbed input x_p and cache activations. (3) Select the sender set s, the components whose activations are patched in, and the receiver set r, the components where the effect of patching s is analyzed. Run the forward pass on x_b, patch in s, freeze all other activations, and recompute r', which is the same as r from the x_b run except for the direct influence from s to r. Cache r'. (4) Run the model on x_b, and patch in r' values. Measure the difference in logits between x_b and x_p to quantify the effect of s on r in terms of passing the additional signal.

The effect of a patch is measured by the difference in logits (which in the case of cross-encoders, is equivalent to the difference in relevance scores). This algorithm is then iteratively repeated for each important upstream component. For more information, we refer the reader to Wang et al. (2022) and Goldowsky-Dill et al. (2023).

We begin by applying path patching to track the path of term-matching, the most fundamental component of BM25. To carry this out, we use the TFC1 and STMC1 diagnostic datasets to identify the components responsible for encoding the relevance signal of an additional query term in a document, providing the foundation for our analysis of model behavior.

D Detailed Path Patching Results

D.1 Path Patching to Final Logits and Intermediate Heads

We perform path patching on 2000 random query and document pairs from the STMC1 and TFC1 diagnostic datasets and collect the results.

In this section, we present the results of path patching on TFC1 and STMC1. First, we patch to the final logits and identify the shared Relevance Scoring Heads (Figure 15). Next, we patch to these heads and discover the shared Query Contextualization Heads (Figures 16, 17). Finally, by patching to the Query Contextualization Heads, we identify the shared Matching Heads (Figure 18).

All resulting heatmaps are located at the end of the appendix for clearer flow for the other appendix sections.

D.2 Path Patching to Matching Heads

We path patch the residual stream (resid_pre) on each of the Matching Heads using 100 random

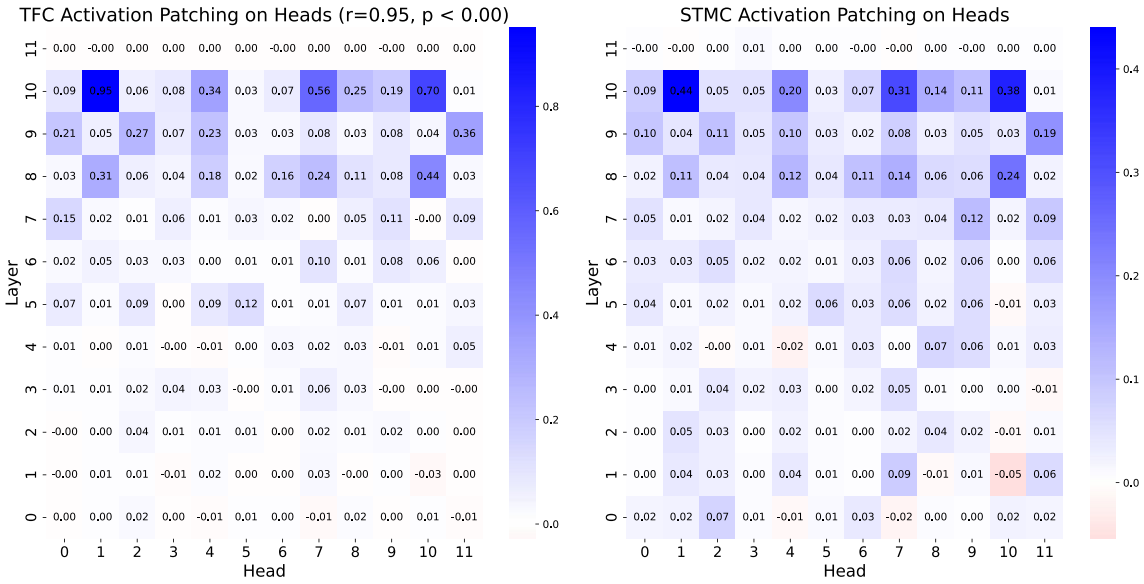


Figure 9: Activation Patching Results for TFC and STMC diagnostic datasets show a correlation of 0.96, which suggests that the model employs highly similar components for exact and soft matches.

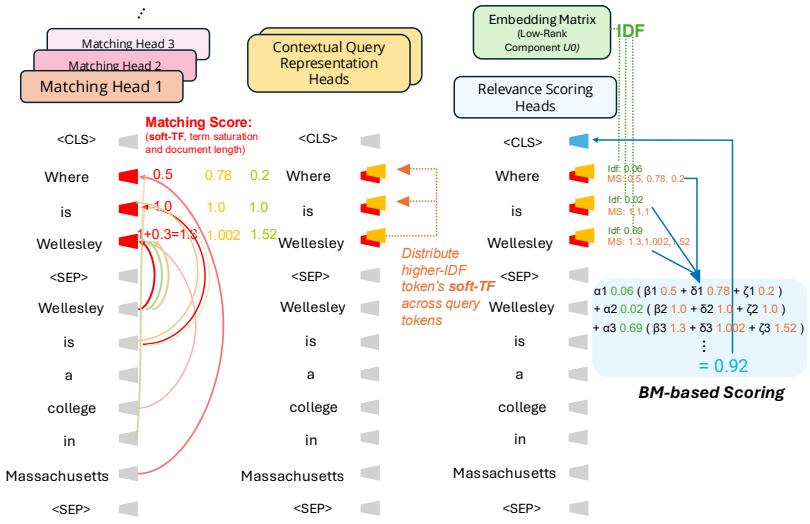


Figure 10: Example walkthrough of the hypothesized circuit. We find that the cross-encoder rediscovers a semantic variant of BM25. Specifically, Matching Heads, Contextual Query Representation Heads, and the embedding matrix compute, process and send BM25-like components' information to the Relevance Scoring Heads which finally compute the relevance score in a BM25-like manner. The trapezoid represents the residual stream representation for each token position.

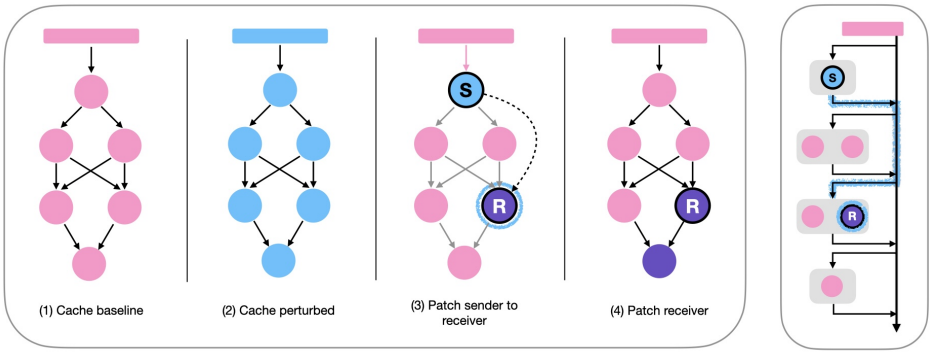


Figure 11: Path patching methodology. *Left:* There are four forward passes: (1,2) Run model on baseline and perturbed inputs and cache activations. (3) To measure the effect of an upstream sender component (S) on a downstream receiver component (R), run the model on the baseline input, patch in S, freeze all other components, and cache the activation of R. (4) Run the model on the baseline input and patch in R. *Right:* Alternative visualization of step (3) using the residual stream. By allowing only the downstream receiver R to be recomputed when the sender S is patched, we effectively isolate the direct path from S to R, while preserving all other paths to R as they were in the baseline run.

pairs from the TFC1 and STMC1 datasets (Figures 19, 20, 21, 22). Path patching to the Matching Heads, aimed at tracking the flow of soft-TF information, reveals that at the beginning of the residual stream at layer 0, which corresponds to the embeddings, has the most direct influence on the Matching Heads. In contrast, there is minimal impact from both the attention and MLP layers. This supports the conclusion that the Matching Heads are the primary "generator" of soft-TF, attending to soft matches based on semantic similarity from the embeddings. Thus, we conclude our tracking of soft-TF information, with the Matching Heads identified as the most upstream heads.

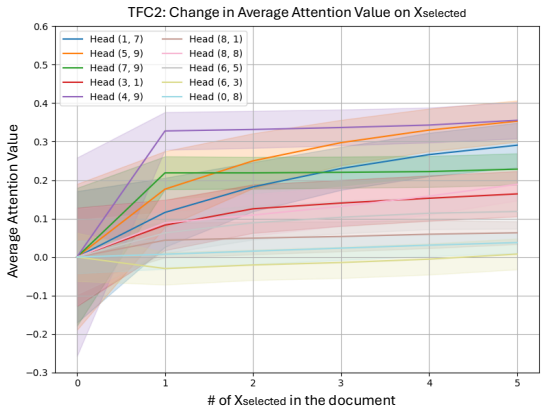


Figure 12: For the majority of Matching Heads, after the initial occurrence of X_{selected}, which causes a sharp increase in attention, subsequent occurrences result in only minimal incremental increases. This aligns with TFC2, which states that additional term occurrences yield smaller improvements.

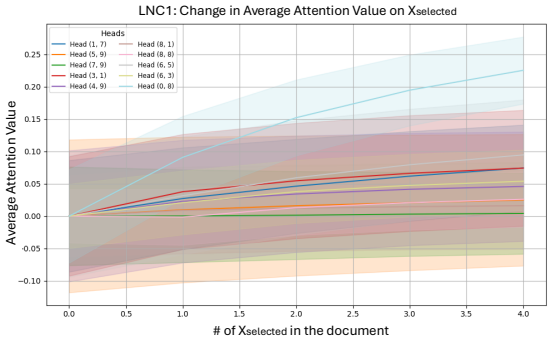


Figure 13: Attention trends for selected terms are shown, with unselected terms following a similar pattern: longer documents generally increase the average attention assigned to all document tokens, regardless of whether they match a query token.

E Additional Details on Matching Heads and Additional Signals

In this section, we provide additional details on the analysis presented in §4.3.

Matching Score Contains Term Saturation and Document Length Signals. The average soft-TF correlation score for Matching Heads (0.500) indicates that their attention values from query tokens to document tokens capture not only semantic similarity but also additional signals. To examine whether these attention values capture term saturation or document length effects, we analyze their behavior under using two diagnostic datasets: (1) TFC2 (increasing occurrences of a query term), and (2) LNC1 dataset (increasing number of irrelevant sentences to simulate the isolated effect of longer document length).

Saturation Test (TFC2). We define x_{selected} as the selected query term, which is the duplicate term in TFC2 samples shown in Table 1. Similarly, x_{others} refers to document tokens that are not duplicates of any query token. For each case, we calculate the normalized sum of attention values from all query tokens to x_{selected} and x_{others}, representing the "total semantic match" of the matched token or the average across unmatched tokens, respectively.

We hypothesize that if the Matching Score contains the term saturation signal, then given more duplicate query terms (TFC2), the summed attention for that term will rise sharply at first and then plateau. As the occurrences of x_{selected} increase, its average attention value grows (Figure 12). On the other hand, the attention for x_{others} remains relatively constant, which aligns with our soft-TF understanding: only document tokens that are semantically similar to a query token get nonzero attention values, proportional to the extent and frequency of semantic similarity.

Length Test (LNC1). We hypothesize that if the Matching Score contains the term saturation signal, given irrelevant sentences appended (LNC1), the overall attention to the query's matches will fall. As expected, injecting more irrelevant sentences leads to an increase in the attention value of all query tokens, regardless of whether they match specific query terms (Figure 13). Notably, this *increase* in attention occurs even as the overall relevance score *decreases* (as expected from LNC1), suggesting that attention values encode mixed and composite signals of soft-TF, term saturation, and document length, which the model has learned to disentangle to produce appropriate relevance score changes. The varying degrees of influence on the heads' attention values shown in Figure 12 and 13 suggest that some Matching Heads play a more significant role in regulating these effects, and they may collectively approximate the ideal term effects. Thus, we define this complex attention value as the ***Matching Score*** to reflect that it encapsulates soft-TF, term saturation, and document length signals, three important signals of BM25.

Further Patching Experiment to Confirm Matching Heads Are Not Just Writing a Binary Signal of Existence. The signals written by most Matching Heads increase as the number of duplicate tokens rises, rather than remaining constant, suggesting that these signals are discrete rather than binary. To demonstrate this, we design an activation patching experiment (§4.3) that patches only the Matching Heads using the TFC2 Diagnostic Dataset. Figure 12 shows that for most Matching Heads, increasing the number of duplicate tokens results in a monotonically increasing pattern in their output signals. This behavior further supports our conclusion that Matching Heads compute and encode soft-TF signals rather than merely relaying binary information.

F Additional Details on BM25-like Computation

In this section, we provide additional details on the Relevance Scoring Heads and the Semantic Scoring Hypothesis in §4.1 and §5.

F.1 Correlation Between IDF Values and Attention Distribution

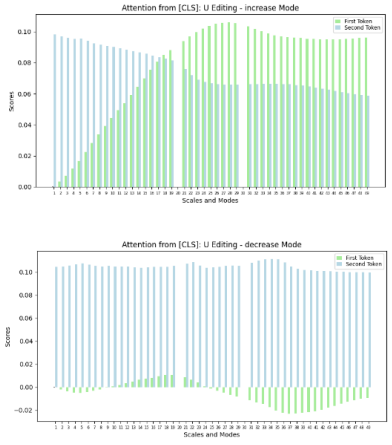


Figure 14: *Top*: Changes in attention at 10.1 from `[CLS]` to `tok1` when increasing `tok1`'s IDF at different scales, averaged across all samples. Increasing `tok1`'s IDF raises attention from `[CLS]` to `tok1`, thereby increasing the final score. *Bottom*: Decreasing `tok1`'s IDF produces the inverse effect, with slight nonmonotonic deviations suggesting optimal editing windows.

In Section 4.1, we hypothesize that Relevance Scoring Heads combine IDF and soft-TF information to compute final relevance scores in a BM25-like manner. Specifically: (A) increasing or decreasing a token's IDF leads to (B) the Relevance Scoring Head 10.1[2] allocating more or less attention from `[CLS]` to the token, which (C) increases or decreases to the weighting of the token's soft-TF and thereby the final relevance score.

[2]We focus on the attention pattern of 10.1 because it is most positively correlated with IDF (§4.1) and has highest path-patching value.

Model Pair	Median	Mean ± SD
Pearson Correlation		
Cross-encoder & SemanticBM	**0.8401**	0.8301 ± 0.0314
Cross-encoder & Random Features	0.0005	0.0003 ± 0.0147
Cross-encoder & BM25	0.4570	0.4609 ± 0.0366
Spearman Rank Correlation		
Cross-encoder & SemanticBM	**0.7619**	0.6629 ± 0.3137
Cross-encoder & Random Features	0.0000	0.0011 ± 0.3844
Cross-encoder & BM25	0.4643	0.3912 ± 0.3992
NDCG@10		
Cross-encoder	0.5000	0.5097 ± 0.4110
SemanticBM	**0.4307**	0.4511 ± 0.3769
BM25	0.5000	0.5269 ± 0.4196
Random Features	0.3562	0.3498 ± 0.2986

Table 2: Comparison of Pearson, Spearman Rank, and NDCG@10 for SemanticBM and Baselines.

A → B. Using the same IDF-scaling setup as in §4.4, we measure the average attention weight from Head 10.1's `[CLS]` token to the manipulated token. The resulting curve closely mirrors Figure 6: tokens with higher IDF receive proportionally more attention, while those with lower IDF receive less. This directly demonstrates the A→B link.

A → B → C. In Section 4.4, we modulate IDF by editing the first SVD component of the embedding matrix. Figure 6 shows that scaling a token's IDF upward or downward produces a monotonic change in the relevance score.

Taken together with Section 5 (which shows that Relevance Scoring Heads perform BM25-style computations on natural IR datasets), these results provide stronger causal evidence for the Semantic Scoring hypothesis.

F.2 Independence between IDF and Soft-TF

In this subsection, we check whether term-frequency (TF) and inverse document frequency (IDF) factors are fully separable.

Low correlation between IDF Soft-TF. If soft-TF scores are already IDF-weighted, we would expect their one-dimensional values to correlate at least mildly with $U0$. However, across all 13 Matching Heads, the average Pearson correlation between $U0$ and each head's Matching-Score is low ($r = 0.143$, $p < 0.01$), suggesting minimal dependence.

Relevance Scoring Heads Do Not Inherit IDF Information From Matching Heads. Path patching from the Matching Heads to the query vectors of the Relevance Scoring Heads produces almost no change in output score. This indicates that the soft-TF signals produced by the Matching Heads do not strongly convey IDF information, leaving the Relevance Scoring Heads without sufficient signal to prioritize which tokens should receive higher weight.

G Generalization Across IR Datasets

We previously test our linear model on an MSMARCO-based dataset with a fixed number of query tokens as an initial proof of concept. Now, we ask the question: how well does this linear model represent the cross-encoder's relevance computation on datasets it was not trained on? In this section, we extend our analysis to 12 different IR datasets[3] and investigate various query lengths.

Dataset. Since our linear model relies on fixed features based on query length, we stratify the dataset based on query length. To ensure sufficient samples, we aggregate samples from 12 datasets and only retain groups with more than 500 samples. We incorporate groups with query lengths from 1 to 22, which corresponds to the 99.95th percentile of MSMARCO's query length distribution to ex-

[3]Due to computational resources, we use a subset of BEIR datasets (Thakur et al., 2021) (ArguAna, Climate-FEVER, FEVER, FiQA-2018, HotpotQA, NFCorpus, NQ, Quora, SCIDOCS, SciFact, TREC-COVID, Touche-2020), but the chosen ones sufficiently cover a wide variety of domains.

clude outliers with excessively long queries that the model has rarely seen during training. This process results in a total of 19 groups, comprising 278194 samples, with an average group size of 14641.790 samples. While this approach may be unconventional for retrieval, our goal is to simply demonstrate consistency between the linear model's relevance score and the cross-encoder to confirm the hypothesized function of the Relevance Scoring Heads.

First, as cross-encoders are typically used for re-ranking tasks, we follow the classic re-ranking setup by first retrieving a candidate set of documents (top 10) with BM25 over all queries across the test collection of the 12 datasets. These candidate sets are then split into train-test groups using an 80/20 ratio.

Experiments and Results. We train 22 linear regression models, one for each query length (1-22), to predict the cross-encoder's relevance scores. We evaluate each model using: (1) Pearson Correlation: quantifies the correlation between predicted and actual relevance scores; (2) Spearman Rank Correlation: assesses the consistency of the ranked lists between predictions and cross-encoder outputs; (3) NDCG@10: measures ranking effectiveness and ensures no significant effectiveness discrepancies. We include BM25 and a randomized set of linear regression features as baselines.

Table 2 shows the results, and we report median values to account for observable skewness caused by outliers in the data. We observe a Pearson correlation of ranking scores with a median of 0.8401 (median $p < 0.001$), a Spearman rank correlation of 0.7619 (median $p = 0.072$), and an 88.4% alignment with cross-encoder effectiveness in terms of NDCG@10.

The experimental results confirm the hypothesized function of the Relevance Scoring Heads (§4.1). Since this linear model summarizes the whole circuit as it is structured to incorporate both the computation and the necessary components, the high correlation with the cross-encoder's effectiveness shows that our circuit understanding has captured the core part of the cross-encoder's relevance ranking mechanism.

H Additional Discussion

Towards a holistic approach in axiomatic analysis. Previous research investigating neural IR models' adherence to IR axioms shows that BERT does not adhere to most of them, leading to concerns about the limitations of current axiom definitions and the need for new ones (Câmara and Hauff, 2020). Our findings offer an explanation for these shortcomings, suggesting that the challenge in axiomatic analysis lies with the way it has been applied, rather than the axioms themselves. BERT's "rediscovery" of BM25 indicates that it leverages a combination of retrieval heuristics to compute relevance, whereas previous analysis treats axioms independently. Going forward, we may consider analyzing axioms in combinations to better reflect the multidimensional nature of relevance.

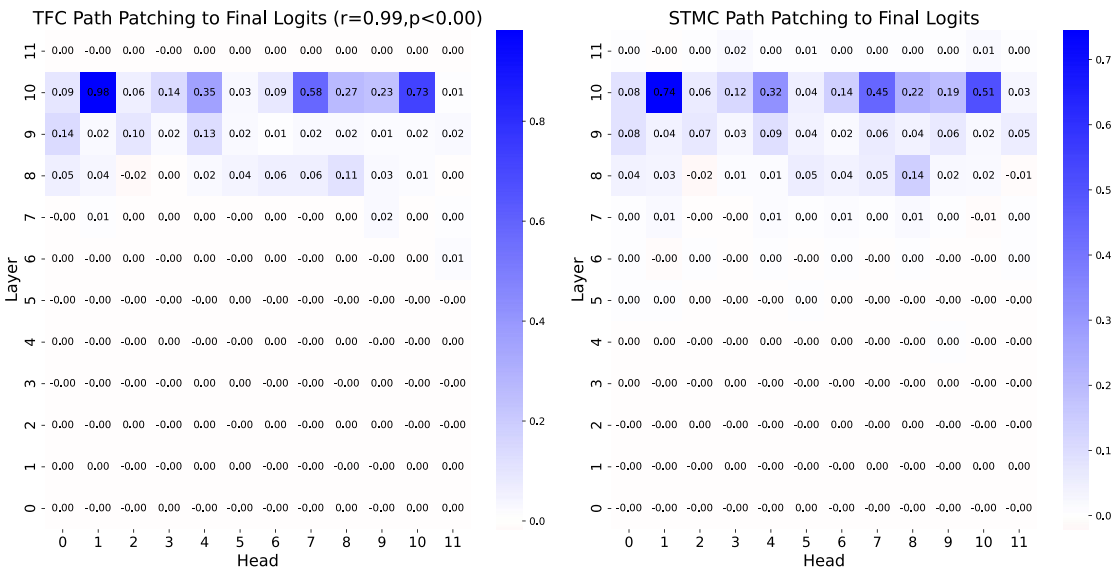


Figure 15: Path Patching results from all attention heads to the final logits. From this, we identify that 10.1, 10.4, 10.7, 10.19, those causing > 30 % increase in ranking score, as most significant heads and focus our analysis on these four heads.

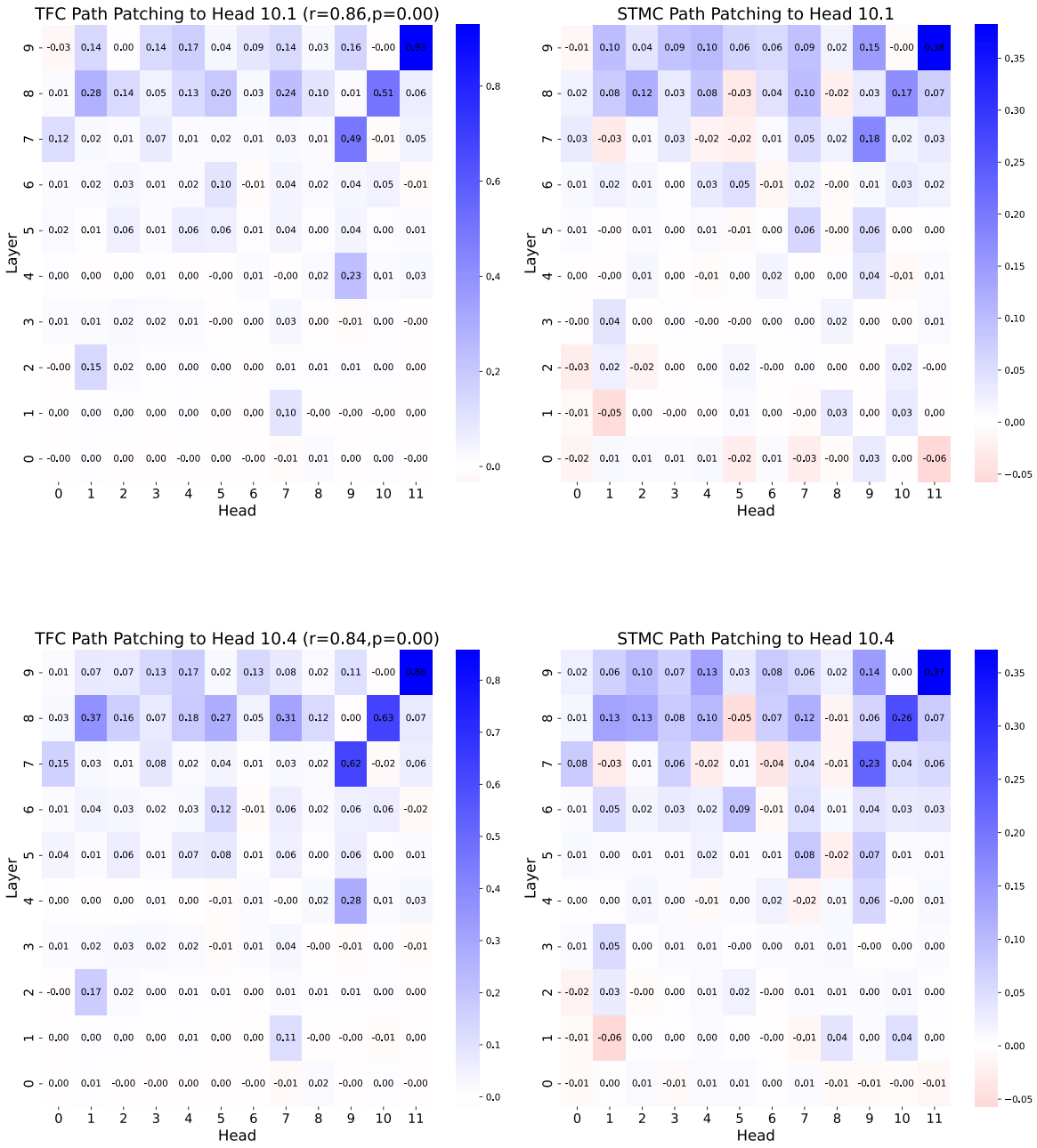


Figure 16: Path Patching results from upstream attention heads to Relevance Scoring Heads.

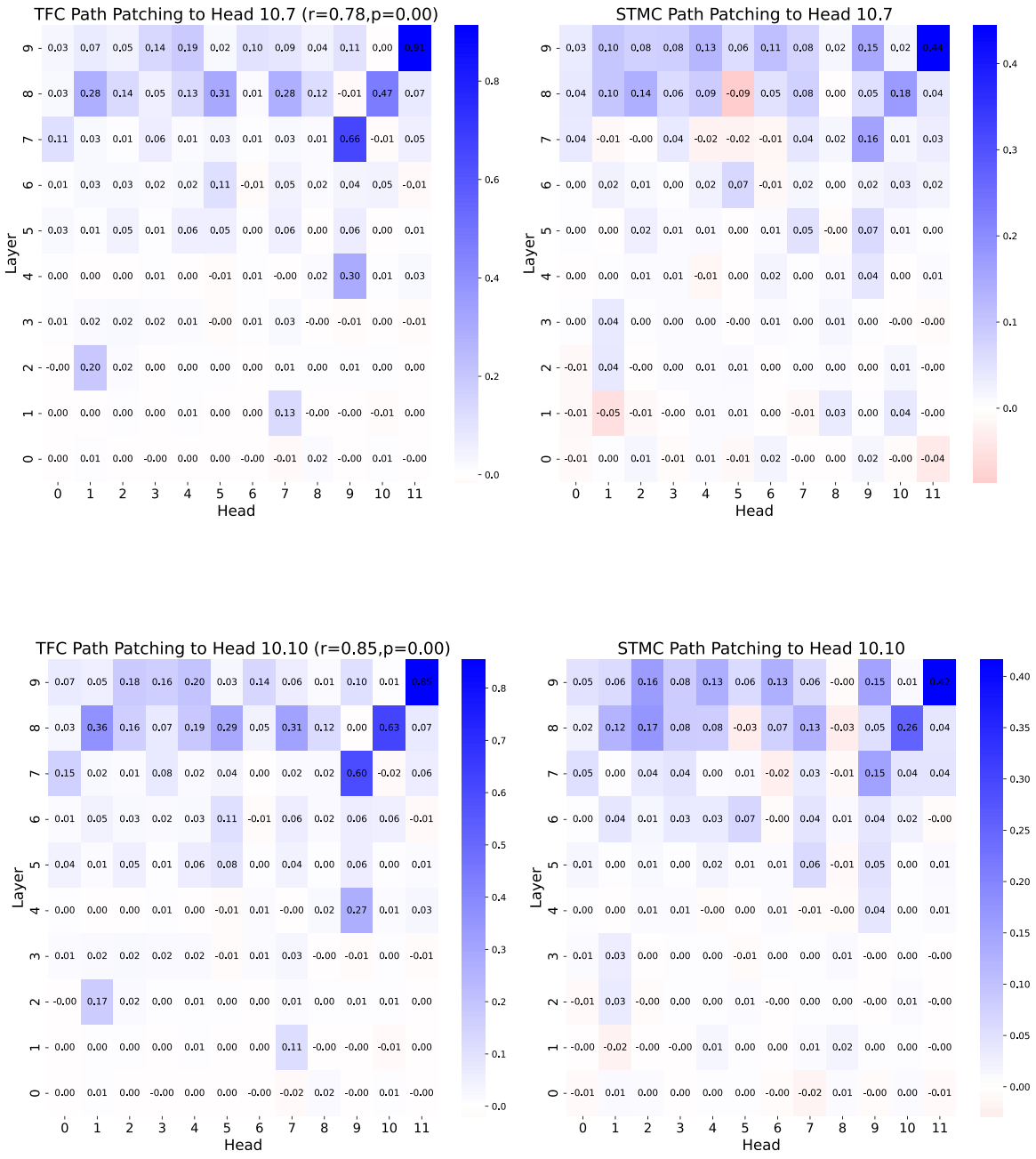


Figure 17: Path Patching results from upstream attention heads to Relevance Scoring Heads.

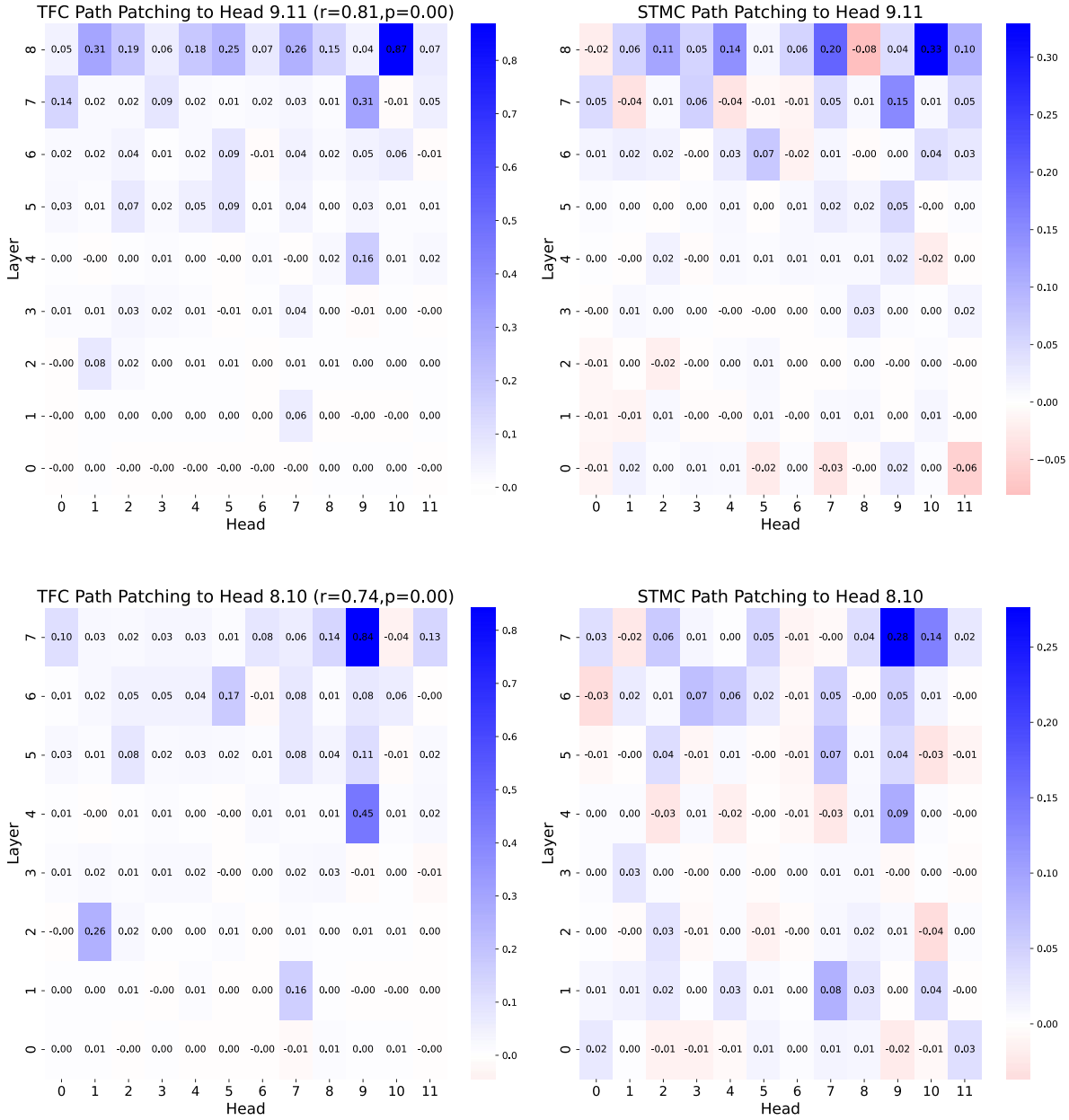


Figure 18: Path Patching results from upstream attention heads to Query Contextualization heads.

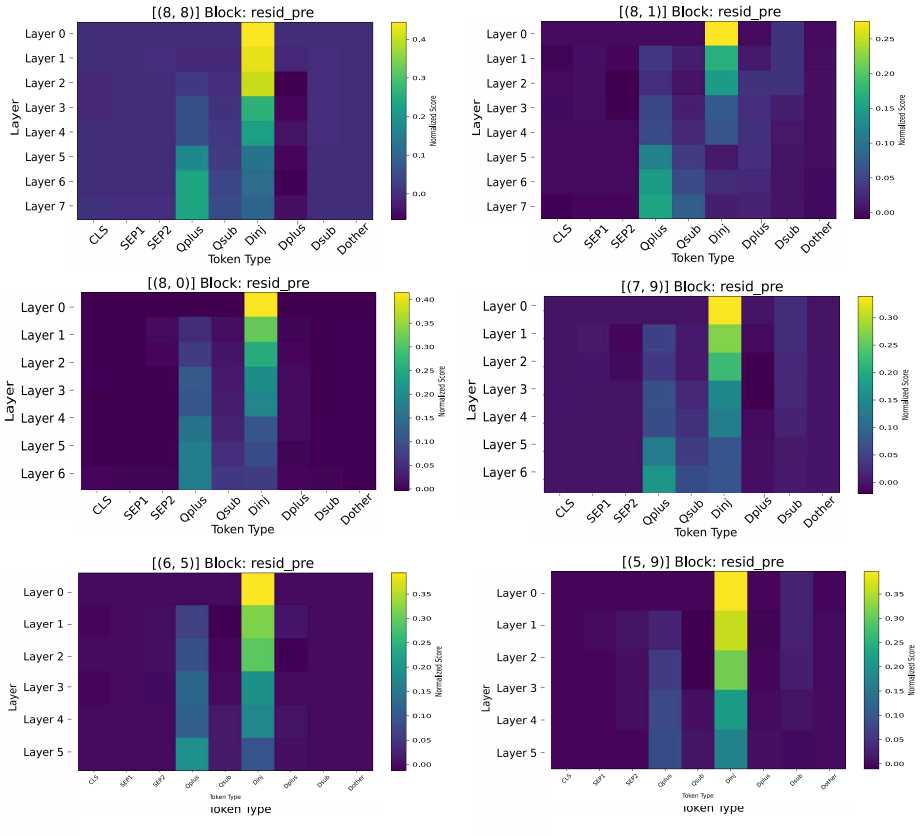


Figure 19: TFC1 path patching into Matching Heads (layers 8.8, 8.1, 8.0, 7.9, 6.5, 5.9). The residual stream values are patched, with lighter regions indicating stronger path-patching contributions (i.e., where the TFC1 signal flows), and darker regions indicating weaker contributions.

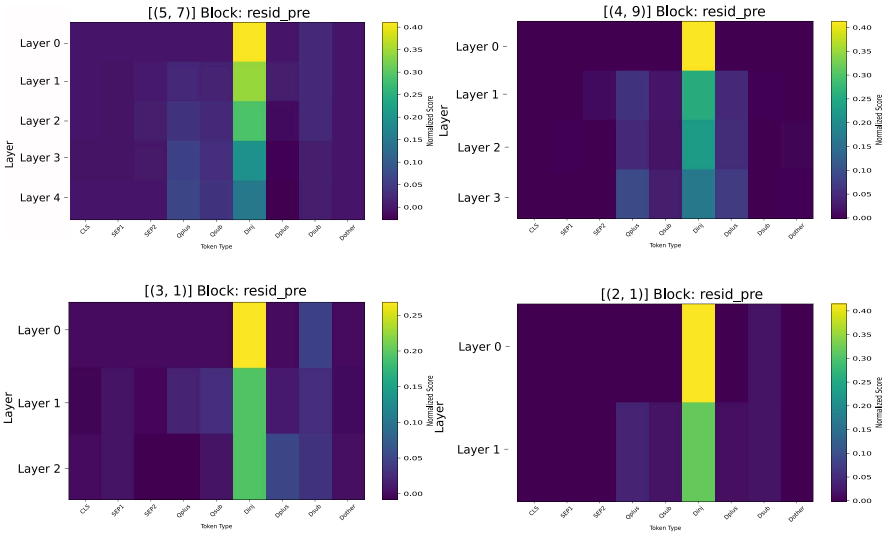


Figure 20: TFC1 Path Patching to Matching Heads (5.7, 4.9, 3.1, 2.1)

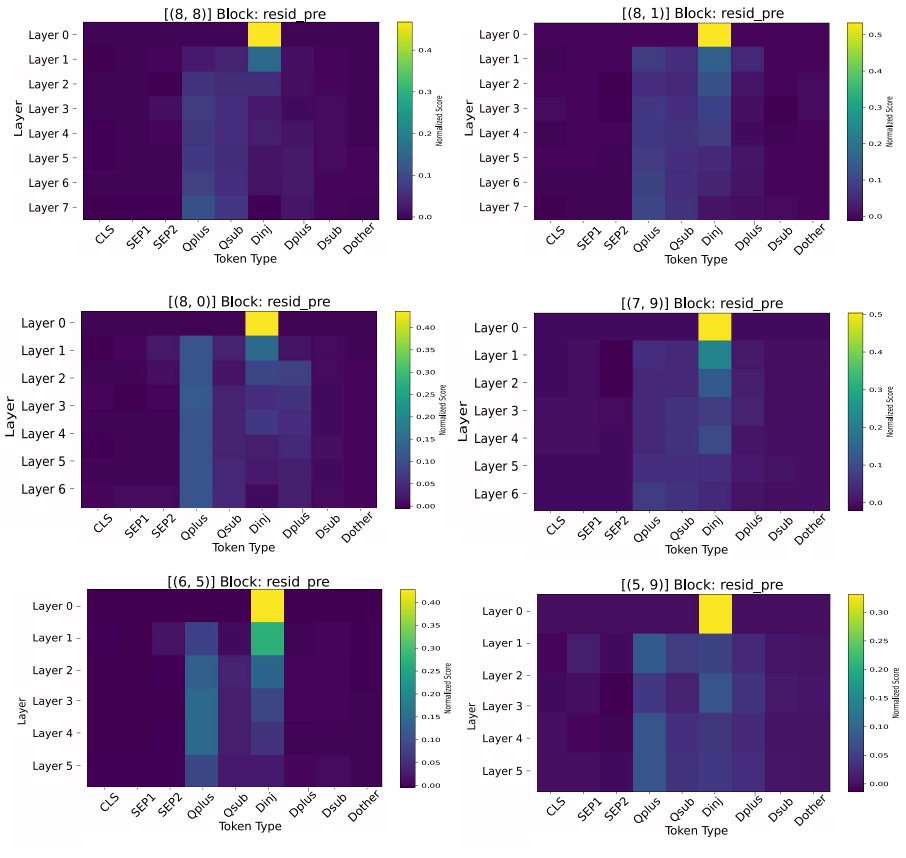


Figure 21: STMC1 Path Patching to Matching Heads (8.8, 8.1, 8.0, 7.9, 6.5, 5.9,)

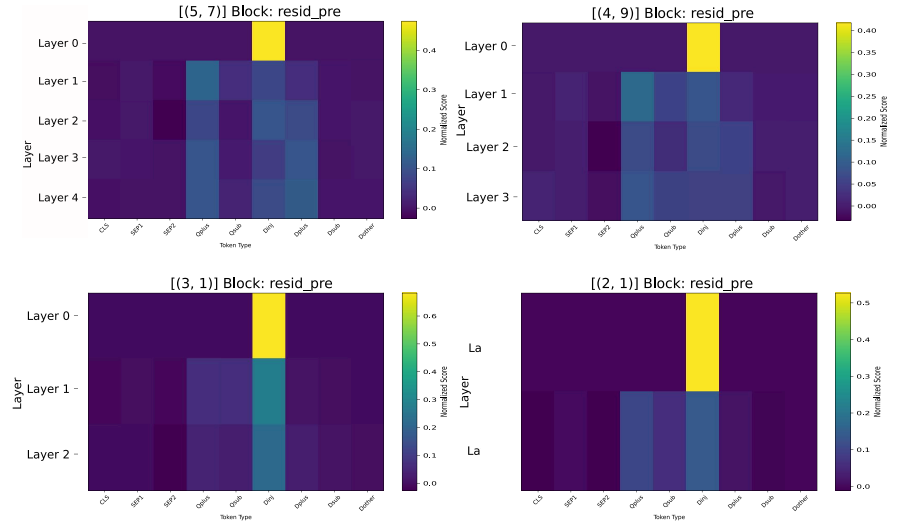


Figure 22: STMC1 Path Patching to Matching Heads (5.7, 4.9, 3.1, 2.1)