IR Lens: A Tool for Interpreting Cross-Encoder Models

Mihai Branga-Peicu
branga@isir.upmc.fr
CNRS, ISIR, Sorbonne Université
F-75005 Paris, France

Mathias Vast
mathias.vast@isir.upmc.fr
Sinequa by ChapsVision
CNRS, ISIR, Sorbonne Université
F-75005 Paris, France

Basile Van Cooten
bvancooten@chapsvision.com
Sinequa by ChapsVision
Paris, France

Laure Soulier
laure.soulier@isir.upmc.fr
CNRS, ISIR, Sorbonne Université
F-75005 Paris, France

Jules Françoise
jules.francoise@lisn.fr
CNRS, LISN, Université Paris-Saclay
Orsay, France

Benjamin Piwowarski
benjamin.piwowarski@cnrs.fr
CNRS, ISIR, Sorbonne Université
F-75005 Paris, France

Baptiste Caramiaux
caramiaux@isir.upmc.fr
CNRS, ISIR, Sorbonne Université
F-75005 Paris, France

Abstract

Transformer-based ranking models, such as MonoBERT, are central to Information Retrieval; yet their inner workings remain largely opaque. This hinders not only our understanding of the systems implementing them, but also our ability to improve them. To alleviate this limitation, we introduce IR Lens, a new interpretability tool tailored to cross-encoders based on two key components: 1) Neuron Integrated Gradients to expose the contributions of model parts at multiple levels, and 2) targeted ablations to support hypothesis tracking. With its interactive graphical interface, IR Lens enables IR practitioners to explore, analyze, and manipulate neuron-level mechanisms in cross-encoders, facilitating a deeper understanding of neural ranking models. By extending the reach of existing interpretability methods, we believe IR Lens has the potential to support the improvement of cross-encoders.

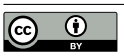
CCS Concepts

• **Information systems → Retrieval models and ranking**; • **Human-centered computing → Visualization systems and tools**.

Keywords

Interpretability, Web Interface, Visualization, User interactions, Cross-Encoders, Neuron Integrated Gradients, Ablations

ACM Reference Format:
Mihai Branga-Peicu, Mathias Vast, Basile Van Cooten, Laure Soulier, Jules Françoise, Benjamin Piwowarski, and Baptiste Caramiaux. 2026. IR Lens: A Tool for Interpreting Cross-Encoder Models. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26), July 20–24, 2026, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3805712.3808382



SIGIR '26, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-2599-9/2026/07
https://doi.org/10.1145/3805712.3808382

1 Introduction

While Transformer-based models [23] have been widely used in Information Retrieval (IR), their inner workings remain largely unknown. The "black-box" nature of these systems has led to the development of various interpretability (XAI) approaches [5, 6, 16, 17, 19]. In IR, such approaches have revealed the role of specific attention heads in capturing matching signals [12], as well as directional patterns in the flow of information between query and document representations [27]. Recently, Vast et al. [22] applied Neuron Integrated Gradients (NIGs) [18] to cross-encoders, uncovering the relative importance of different model components in the relevance prediction process. While these interpretability studies provide valuable insights for the IR research community, their practical impact remains limited, as the underlying methods lack interactivity, which greatly hinders the discoverability of the system's behavior.

To address this, we introduce IR Lens[1], a web-based interpretability tool for cross-encoders that makes NIG-based analysis interactive and accessible to IR practitioners with different levels of expertise. Rather than exposing raw attribution scores, IR Lens emphasizes progressive exploration and interaction with cross-encoders' internal components, allowing users to inspect their hidden mechanisms. The interface helps identify model components that contribute most to relevance predictions: for a given query-document pair, users can inspect components at multiple levels of granularity—from entire layers down to individual FFN neurons or information transfers in the attention—and formulate hypotheses about their role by performing side-by-side comparisons of ablation strategies. Beyond standard NIG computation, users can also compute conditional NIGs to trace which components most influenced a given intermediate output. By integrating these capabilities into a unified tool, IR Lens targets a broad audience—from PhD students to IR practitioners looking to debug deployed models—and represents a first step toward bridging the gap between interpretability methods and usable interactive tools in IR.

[1] IR Lens can be accessed here: https://ir-lens.isir.upmc.fr and the code is publicly available: https://gitlab.isir.upmc.fr/branga/IR-Lens.

2 Related Works

Most existing interpretability systems for explaining transformer language models rely on attention visualizations to expose token-to-token interactions across layers, typically using bipartite representations as done by BertViz [24], which supports head-level inspection alongside global overviews of attention patterns. Other tools extend this analysis by linking attention and hidden representations to corpus-level evidence, allowing users to retrieve similar contexts and assess whether observed behaviors generalize beyond a single input [1, 9]. Another example, AttentionViz [26], uses scatterplots to compare attention heads and highlight recurring attention patterns across many inputs and layers. While these approaches are effective for inspecting general language behavior, they are not designed to reflect task specific structures, such as the asymmetric roles of query and document tokens or relevance oriented decision processes, which limits their applicability to IR.

In addition to attention-based inspection tools, interactive systems embed attribution methods and intervention mechanisms within visual analytic environments. LIT [20], for instance, supports dataset inspection together with salience-style explanations and attention views inside a modular interface. Other systems emphasize tracing how predictions are formed by extracting and visualizing token level pathways through transformer computations. VISIT [10] presents these pathways using a Sankey-style diagram and logit-lens projections, and the LM Transparency Tool [21] highlights influential components and routes through the model using related flow based views. TransformerLens [13] is a library commonly used that supports logit lens inspection through direct access to activations. Finally, other interactive systems enable direct intervention on model internals, such as ablating neuron contributions [8] or pruning attention heads [11] and inspecting the resulting behavioral changes. However, these tools are primarily designed for interpreting general language model behavior and do not explicitly target task-specific models such as cross-encoders for IR.

A broader view on task-specific models within explainable IR is summarized by Anand et al. [2]. In this context, Zhan et al. [28] analyze BERT based ranking models using attribution based token importance including breakdowns by token types (query, document, and special tokens) to study query document interaction behavior. Vast et al. [22] apply Neuron Integrated Gradients (NIG) to cross encoders to attribute relevance predictions to internal components such as neurons and attention substructures, while Lu et al. [12] develop a Mechanistic Interpretability framework tailored to IR further formalize how internal components influence relevance predictions. Including, Parry et al. [15] who propose a library to conduct Mechanistic Interpretability on IR models, XAI works in IR remain largely analytical and code-driven. This offers limited support for interactive exploration or hypothesis driven analysis, motivating the need for tools that bridge mechanistic interpretability methods and IR specific workflows.

3 Neuron Integrated Gradients

Cross-encoders classify a query-document (Q, D) couple as relevant or not relevant. Their input is typically composed by the following sequence: $\mathcal{P}$ = {[CLS], Q, [SEP1], D, [SEP2]}.The transformer architecture updates the representation of the input tokens after each layer $\ell \in [1 \dots L]$ by performing a sequence of two main processing steps, namely residual updates through self-attention and feed-forward networks (FFN). Finally, the classification head of a cross-encoder is applied to the token representation [CLS] of the last layer L, which outputs a relevance score for the document *w.r.t.* the query.

To study the importance of these processing steps, we use the Neuron Integrated Gradients [18] approach (NIG). NIG is an interpretability technique that extends Integrated Gradients [19] to attribute the importance of individual neurons within a model. In a transformer, NIG scores are meaningfully defined for the outputs of attention heads and for the intermediate outputs of FFN layers, as these are the rotation-invariant components of the architecture—their individual dimensions carry interpretable meaning independently of the chosen basis.

Following [22], we compute the NIG score for each such neuron. IR Lens then aggregates these per-neuron scores at two levels. First, scores are summed within each component (e.g., across an attention head or all neurons of an FFN layer). Second, scores are grouped by *input parts* from $\mathcal{P}$: for an attention head h at layer ℓ, this means separating contributions by source–target input part pairs (e.g.,aggregating the NIG scores of all query-to-document attention connections within head h), while for an FFN layer, this means grouping by which input part's tokens are being processed (e.g., all document tokens D). This two-level aggregation provides a finer-grained analysis that reveals the nature of the operation performed by each component.

4 Design goals

IR Lens was designed to support the interpretability of transformer-based cross-encoders by enabling users to explore neuron-level hidden mechanisms, provided by NIG values, and examine their robustness through targeted ablations. The design follows the observation that understanding neural ranking models requires more than static explanations: practitioners must be able to progressively inspect internal components, form hypotheses about their usefulness, and evaluate how these hypotheses hold under controlled modifications of the model. Our design goals are threefold:

(1) *Modeling the relative importance of internal components.* The system is intended to allow users to investigate how relevance predictions emerge from interactions between query and document representations inside a cross-encoder. Users should be able to select query document pairs and obtain an overview of which parts of the model contributed the most to the relevance prediction.
(2) *Supporting the exploratory analysis of internal components.* From this global perspective, the system should support more fine grained exploration. Users should be able to inspect attribution values at different levels of granularity, ranging from entire feed forward layers and attention blocks down to individual neurons and attention heads. Complementary views are intended to reveal both structural patterns and value distributions, allowing users to compare components and reason about their relative importance.
(3) *Offering a complementary hypothesis evaluation.* In addition to exploration, the system must support hypothesis evaluation.

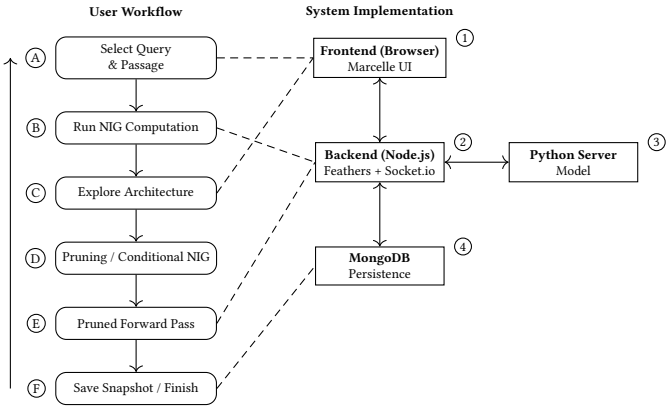


Figure 1: On the left, the User workflow, illustrating the refinement process present in deciding on inputs as well as prunes of the model. The right shows the supporting system.

After a first phase of exploration where users have identified key components, the system should further offer them the possibility to easily test the actual importance of a set of neurons or attention heads. Following Vast et al. [22], the most straightforward way to do so, is through ablations. As such, users should be able to remove selected components of the model, such as neurons, attention heads, or specific layer regions, and observe the impact on the original prediction. Feedback from ablations should be integrated into the same views used for exploration, enabling side-by-side inspection and hypothesis tracking without disrupting the analysis workflow.

5 Workflow and interface description

In this section, we describe the functionalities of IR Lens as well as the interface that implement each one of them. The main interface of IR Lens, shown in Figure 2, consists of three custom, resizable visual panels: the architecture view (A), the NIG table (B), and the distribution plots (C). These visualizations are designed to support the progressive exploration of neuron-level mechanisms and to remain tightly coordinated during interaction. The architecture view (Figure 2, center) provides a layer-wise representation of attention and FFN components based on the aggregation of NIG values per input part. It renders nodes and edges corresponding to internal model components and supports two-step selection (source followed by target) to enable focused inspection of interactions. For additional granularity, each attention module or FFN layer is decomposed based on the different input parts. This design allows for contextualizing the importance scores in terms of *transfers of information* (for attention) or *the integration of the information stored in specific tokens* (for the FFN).

We now describe the user workflow, as outlined in Figure 1 (left).

Ⓐ First, IR Lens requires the user to type their own query-passage pair or to select a query from MS MARCO and then choose one of the passages from the pool of annotations of that query. This is controlled by the "Inputs" interface (Figure 2, right), accessible in the left panel of the main interface. Note that panel interfaces can be hidden to leave more room to the architecture view.

Ⓑ After NIG computation, the visibility of component connections is controlled via a global log-scale threshold t, which by default highlights the $t = 1.0\%$ most important nodes and edges on the main interface (Figure 2, in the architecture view (A)). This view supports global pattern discovery and serves as a navigation mechanism for more detailed analysis.

Ⓒ Each node and edge becomes clickable, allowing the user to access more fine-grained information about the importance of a transfer of information between two input parts (inside the attention) or of a linear transformation inside the FFN. In addition, the NIG table presents per neuron or per head attribution values corresponding to the current selection of token types in the architecture view. It supports sorting by token type and dynamically reflects ablation states and conditional analysis modes. This tabular view enables comparison of internal components and complements the structural overview provided by the architecture view. Distribution and Empirical Cumulative Distribution Function plots summarize NIG values both globally and at the level of individual layers (Figure 2-C). These plots use symmetric log scaling and share the same threshold state as the architecture view, allowing users to reason about how threshold choices affect visible structure and ablation strategies. Together, these visualizations provide both structural and statistical perspectives on attribution patterns. As users explore the architectures, they start to formulate hypotheses about the importance of some components and their roles.

Ⓓ To allow them to test their hypothesis, IR Lens provides a "Pruning" interface (Figure 2, top left). Pruning can either be specified by separate log scale thresholds for attention and FFN components or interactively selected on the model view. The pruning interface further exposes rule and target lists that specify which layers, neurons, attention heads, or token connections should be ablated. Users can store and manage multiple ablation configurations for comparison and hypothesis tracking. A dedicated pruning cursor mode provides immediate feedback during interactive selection, and prune previews allow users to hover over saved configurations to visualize their effect before applying them.

To further refine their hypotheses, IR Lens lets users compute NIGs *conditionally* to a given output, rather than to the final relevance estimation. (Figure 2, bottom left). This allows for traversing the computational graph, first identifying the most important components for the prediction, and then the ones that influenced these components the most.

Ⓔ Once the target sections have been pruned, a pruned forward pass (Figure 2-D) can be computed to compare the original prediction with the new prediction without the ablated components. This is particularly useful for users interested in understanding the importance of a set of components, not just in terms of NIG score, but also regarding its impact on the model's predictions.

Ⓕ The experimental setup is saved automatically on NIG computation. users satisfied with their findings can save their prunes to restart working on them later or to reuse them for other query-passage pairs.

6 Implementation

IR Lens is implemented as a web-based system that combines a reactive frontend with a backend for attribution computation and

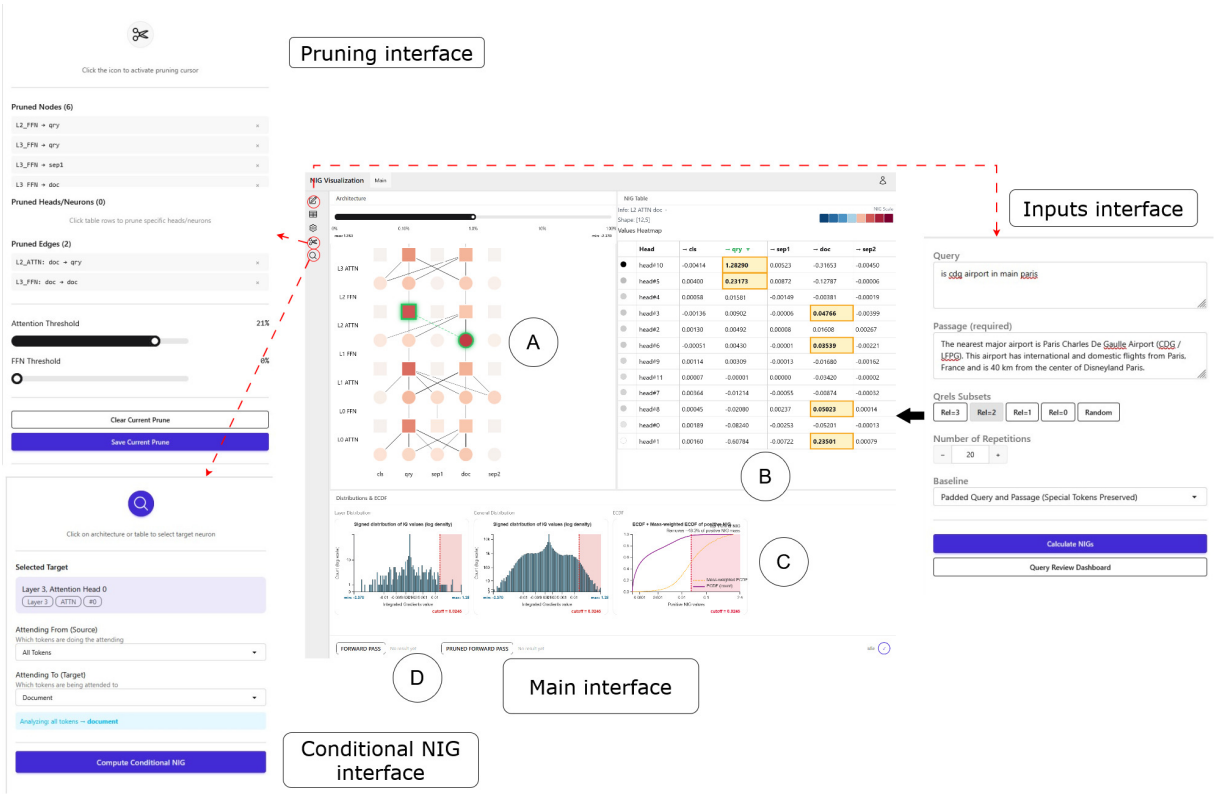


Figure 2: Overview of IR Lens main interface and some of its features. (A) is the architecture view, where the clickable cross-encoder components are represented. (B) is the NIG's scores table of the currently selected component. (C) are the different plots representing the distribution of the NIG scores. (D) are the buttons that trigger the original forward pass and the pruned forward pass across the model.

model inference (see Figure 1, right). Below we describe the implementation of the elements that constitute the system architecture.

① The frontend is built using Marcelle [7], a software toolkit for building interactive machine learning interfaces, which is based on SvelteKit (Svelte 5), and styled using TailwindCSS and DaisyUI. We used D3.js [3] to implement custom interpretability visualizations.

② The backend is built on the Marcelle server infrastructure, which uses FeathersJS over Socket.IO for real-time communication and MongoDB for persistence.

③ A dedicated Python service, connected to the backend, performs model inference, attribution computation, and ablation-based forward passes. The service exposes endpoints returning structured outputs for IGs and NIGs computation, conditional NIGs, and forward passes with and without ablations. To keep inference-time reasonable, we limit the choice of the cross-encoder to a single one based on the MiniLM-v2 backbone [25] [2]. In addition to the possibility for the user to experiment with their own query-passage pairs, IR Lens also provides direct access to TREC-DL19 [4] queries and annotated passages, sorted by relevance levels (TREC-DL19 is based on MS MARCO [14]).

④ Results are persisted in MongoDB and streamed back to the frontend, allowing visualizations to update in real time. Additionally, users can save snapshots of NIG values and prunes, enabling them to revisit previous analyzes.

Separating computation from interaction ensures that computationally intensive operations do not compromise interface responsiveness while maintaining a tight integration between attribution results and user-driven exploration.

7 Conclusion

By focusing on interactivity, IR Lens is designed to target an audience broader than experimented IR researchers. From students starting their PhD in IR and looking for resources to develop their understanding of cross-encoders, to IR practitioners from the industry in need of a tool to *debug* the behaviors of their deployed models, IR Lens fills a gap in the IR toolbox by making interpretability more easily accessible. To broaden the systems use, we plan to include a larger support of cross-encoder models and datasets in future works. However, we believe this first version of IR Lens will benefit the community and serve as a platform to inspire similar development across XAI approaches in IR.

[2]Checkpoint available on HuggingFace: cross-encoder/ms-marco-MiniLM-L12-v2

Acknowledgments

The authors acknowledge the ANR – FRANCE (French National Research Agency) for its financial support of the GUIDANCE project n°ANR-23-IAS1-0003 as well as the CLUSTER IA project n°ANR-23-IACL-0007.

Furthermore, the authors acknowledge the peoples of the Woi Wurrung and Boon Wurrung language groups of the eastern Kulin Nation on whose unceded lands ACM SIGIR 2026 was hosted. We pay our respects to their Elders past and present, and extend that respect to all Aboriginal and Torres Strait Islander peoples today and their continuing connection to land, sea, sky, and community.

References

[1] J Alammar. 2021. Ecco: An Open Source Library for the Explainability of Transformer Language Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, Heng Ji, Jong C. Park, and Rui Xia (Eds.). Association for Computational Linguistics, Online, 249–257. doi:10.18653/v1/2021.acl-demo.30

[2] Avishek Anand, Lijun Lyu, Maximilian Idahl, Yumeng Wang, Jonas Wallat, and Zijian Zhang. 2022. Explainable information retrieval: A survey. *arXiv preprint arXiv:2211.02405* (2022).

[3] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D^3 data-driven documents. *IEEE transactions on visualization and computer graphics* 17, 12 (2011), 2301–2309.

[4] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. In *Text REtrieval Conference (TREC)*. TREC. https://www.microsoft.com/en-us/research/publication/overview-of-the-trec-2019-deep-learning-track/

[5] Kedar Dhamdhere, Mukund Sundararajan, and Qiqi Yan. 2019. How Important is a Neuron. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SylKoo0cKm

[6] Ruth C. Fong and Andrea Vedaldi. 2017. Interpretable Explanations of Black Boxes by Meaningful Perturbation. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE. doi:10.1109/iccv.2017.371

[7] Jules Françoise, Baptiste Caramiaux, and Téo Sanchez. 2021. Marcelle: composing interactive machine learning workflows and interfaces. In *The 34th Annual ACM symposium on user interface software and technology*. 39–53.

[8] Mor Geva, Avi Caciularu, Guy Dar, Paul Roit, Shoval Sadde, Micah Shlain, Bar Tamir, and Yoav Goldberg. 2022. LM-Debugger: An Interactive Tool for Inspection and Intervention in Transformer-Based Language Models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Wanxiang Che and Ekaterina Shutova (Eds.). Association for Computational Linguistics, Abu Dhabi, UAE, 12–21. doi:10.18653/v1/2022.emnlp-demos.2

[9] Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. 2020. exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformer Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Asli Celikyilmaz and Tsung-Hsien Wen (Eds.). Association for Computational Linguistics, Online, 187–196. doi:10.18653/v1/2020.acl-demos.22

[10] Shahar Katz and Yonatan Belinkov. 2023. VISIT: Visualizing and Interpreting the Semantic Information Flow of Transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 14094–14113. doi:10.18653/v1/2023.findings-emnlp.939

[11] Zhen Liu, Haibo Sun, Huawei Sun, Xinyu Hong, Gang Xu, and Xiangyang Wu. 2024. BHPVAS: visual analysis system for pruning attention heads in BERT model. *Journal of Visualization* 27, 4 (Aug. 2024), 731–748. doi:10.1007/s12650-024-00985-z

[12] Meng Lu, Catherine Chen, and Carsten Eickhoff. 2025. Pathway to Relevance: How Cross-Encoders Implement a Semantic Variant of BM25. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (Eds.). Association for Computational Linguistics, Suzhou, China, 25525–25547. doi:10.18653/v1/2025.emnlp-main.1297

[13] Neel Nanda and Joseph Bloom. 2022. TransformerLens. https://github.com/TransformerLensOrg/TransformerLens.

[14] 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. *CoRR* abs/1611.09268 (2016). http://dblp.uni-trier.de/db/journals/corr/corr1611.html#NguyenRSGTMD16

[15] Andrew Parry, Catherine Chen, Carsten Eickhoff, and Sean MacAvaney. 2025. MechIR: A Mechanistic Interpretability Framework for Information Retrieval. In *European Conference on Information Retrieval*. Springer, 89–95.

[16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. ACM. doi:10.1145/2939672.2939778

[17] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2019. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision* 128, 2 (Oct. 2019), 336–359. doi:10.1007/s11263-019-01228-7

[18] Avanti Shrikumar, Jocelin Su, and Anshul Kundaje. 2018. Computationally Efficient Measures of Internal Neuron Importance. arXiv:1807.09946 [cs.LG]

[19] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70* (Sydney, NSW, Australia) *(ICML'17)*. JMLR.org, 3319–3328.

[20] Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, and Ann Yuan. 2020. The Language Interpretability Tool: Extensible, Interactive Visualizations and Analysis for NLP Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Qun Liu and David Schlangen (Eds.). Association for Computational Linguistics, Online, 107–118. doi:10.18653/v1/2020.emnlp-demos.15

[21] Igor Tufanov, Karen Hambardzumyan, Javier Ferrando, and Elena Voita. 2024. LM Transparency Tool: Interactive Tool for Analyzing Transformer Language Models. doi:10.48550/arXiv.2404.07004 arXiv:2404.07004 [cs].

[22] Mathias Vast, Basile Van Cooten, Laure Soulier, and Benjamin Piwowarski. 2024. Which Neurons Matter in IR? Applying Integrated Gradients-based Methods to Understand Cross-Encoders. In *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR '24)*. ACM, 133–143. doi:10.1145/3664190.3672528

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoeit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Poloshukin. 2017. Attention Is All You Need. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*. doi:10.48550/arXiv.1706.03762

[24] Jesse Vig. 2019. A Multiscale Visualization of Attention in the Transformer Model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Marta R. Costa-jussà and Enrique Alfonseca (Eds.). Association for Computational Linguistics, Florence, Italy, 37–42. doi:10.18653/v1/P19-3007

[25] Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021. MiniLMv2: Multi-Head Self-Attention Relation Distillation for Compressing Pretrained Transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, Online, 2140–2151. doi:10.18653/v1/2021.findings-acl.188

[26] Catherine Yeh, Yida Chen, Aoyu Wu, Cynthia Chen, Fernanda Viégas, and Martin Wattenberg. 2023. Attentionviz: A global view of transformer attention. *IEEE Transactions on Visualization and Computer Graphics* 30, 1 (2023), 262–272.

[27] 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* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 1941–1944. doi:10.1145/3397271.3401325

[28] 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.