arXiv:2610.41425v1 [cs.IR] 30 Oct 2026

How BERT-based Cross-Encoders Model Relevance in Information Retrieval?

MATHIAS VAST, Sinequa by ChapsVision, France and Sorbonne Université, CNRS, ISIR, France
BASILE VAN COOTEN, Sinequa by ChapsVision, France
LAURE SOULIER, Sorbonne Université, CNRS, ISIR, France
BENJAMIN PIWOWARSKI, Sorbonne Université, CNRS, ISIR, France

Cross-encoders are the most effective but also the least efficient models in Information Retrieval. Despite their success, the mechanisms underlying their relevance predictions remain poorly understood, particularly how they process to match queries and documents. This lack of interpretability limits their optimization, can lead to suboptimal retrieval, especially out-of-domain, and raises efficiency concerns for over-parameterized models. To address these challenges, this paper investigates two under-explored aspects of cross-encoder behavior. First, to understand the relevance prediction process, we trace the flow of information at the different stages of the process and between different input segments. We study the nature of the signals that compose this flow and that allow cross-encoders to predict the relevance of a passage to a query. Our work particularly focuses on the matching performed by the cross-encoders between queries and documents. Second, we explore how cross-encoders behave with out-of-domain documents and/or queries. Through a series of interpretability experiments, we evaluate three BERT-based cross-encoder models, varying in size and pre-training strategies, across both in-domain and out-of-domain datasets. Overall, this study both consolidates what is known about how cross-encoders model relevance and extends it by exploring the role of matching signals.

Additional Key Words and Phrases: Interpretability, Cross-Encoders, Relevance, Matching

ACM Reference Format:
Mathias Vast, Basile Van Cooten, Laure Soulier, and Benjamin Piwowarski. 2026. How BERT-based Cross-Encoders Model Relevance in Information Retrieval?. 1, 1 (November 2026), 68 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

With the rise of transformer-based models [94], currently culminating in Large Language Models (LLMs), information retrieval (IR) has changed drastically since the BM25 era [1]. Current transformer-based IR models can be categorized according to how they model the interaction between queries and documents: implicitly (cross-encoders [67, 68]), explicitly (dense [46] or sparse encoders [33]) or a mix of both (late interactions [47]). With their ability to model semantics and produce contextualized representations, these models have overcome the query-document exact match limitation of traditional lexical methods. This advancement, however, has come with an important trade-off: interpretability. Unlike lexical approaches such as BM25 – where relevance can be directly traced to term-statistics – neural models are mostly black-boxes. For end users, this opacity may be acceptable, provided that the retrieved results or the LLM-generated outputs are relevant [50]. For researchers and practitioners, however, this opacity hinders both systematic improvement

Authors' Contact Information: Mathias Vast, mathias.vast@isir.upmc.fr, Sinequa by ChapsVision, Paris, France and Sorbonne Université, CNRS, ISIR, F-75005 Paris, France; Basile Van Cooten, Sinequa by ChapsVision, Paris, France, bvancooten@chapsvision.com; Laure Soulier, Sorbonne Université, CNRS, ISIR, F-75005 Paris, France, laure.soulier@isir.upmc.fr; Benjamin Piwowarski, Sorbonne Université, CNRS, ISIR, F-75005 Paris, France, benjamin.piwowarski@cnrs.fr.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
Manuscript submitted to ACM

and the diagnosis of failure cases. In particular, cross-encoders are widely used in the re-ranking stage of retrieval pipelines – whether in search engines or RAG frameworks [51] – despite limited understanding of their internal mechanisms and failure modes. While we know that these models are over-parameterized [108] and could be made more efficient by removing unnecessary layers or attention heads [40, 97], our limited comprehension prevents us from designing more efficient and/or effective alternatives.

This inability to diagnose failure cases (as well as the scientific challenge that understanding these models represents) has motivated a surge in *interpretability* and *explainability* studies, not limited to IR. From attribution methods [74, 79, 81, 84] to probing [5, 90, 101], many techniques have been developed to interpret and explain the behavior of neural networks. In IR, efforts have largely concentrated on elucidating the nature of the signals involved when computing document relevance, though with mixed outcomes [2]. Some works study the agreement of neural IR models with established IR axioms [15, 56] or use probing [5] to search for specific encoded features [60, 101], while other studies have highlighted the roles of specific attention heads in detecting matching signals [56].

However, we still lack a *mechanistic* understanding of how these signals are detected or processed, let alone incorporated to form the *relevance prediction*. In addition, few works have attempted to provide a holistic view of the entire relevance prediction process, as most of them generally focus on a specific aspect. This also raises the question of whether any single interpretability technique (probing, ablations, attention) can bridge this gap. Notable exceptions in IR are studies of the processing stages within MonoBERT [45, 73, 111], but even these lack supporting evidence for the hypotheses they formulate. More recently, Lu et al. [56] identified a mechanism similar to a semantic variant of BM25 within a MiniLM-based cross-encoder, while Chen et al. [12] used mechanistic interpretability to interpret the roles of some attention heads for duplicate tokens inside TAS-B [43]. These works provide an end-to-end picture of a specific mechanism inside neural IR models, yet they also highlight another overlooked aspect of interpretability studies: their generalization to other models and datasets.

Cross-encoders stand as the most effective models for relevance estimation, while their internal mechanisms remain poorly understood. This paradox makes them a high-potential target for interpretability research. Better understanding how they work could help overcome their two main limitations in search pipelines: computational inefficiency and dependence on a first-stage retriever.To address these limitations, our experimental plan combines 1) multiple interpretability techniques, since signals in transformer-based models are entangled [27, 28]; 2) multiple BERT-based cross-encoders of different sizes and pre-training strategies; and 3) multiple datasets, both from the training domain and from domains unknown to the models. This last point is motivated by the release of the BEIR benchmark [87] which highlighted disparities between the in-domain and out-of-domain performance of neural IR models. Although smaller for cross-encoders than for bi-encoders, this gap, together with their reliance on semantic knowledge inherited from pre-training, raises the question of whether cross-encoders behave consistently In-Domain (ID) and Out-of-Domain (OOD).

Thanks to the points 1) and 3) of this experimental plan, i.e., multiple techniques applied across multiple datasets, we further gain the ability to draw stronger conclusions, as we can cross observations.

Our objective is to consolidate and extend previous findings on the role of attention heads and the nature of signals processed by cross-encoders. Beyond unifying these findings, we seek to shed light on the full process through which these models arrive at a relevance prediction. Specifically, we address the following research questions:

- **RQ1**: What is the contribution of the different models' components in predicting relevance?
- **RQ2**: How does information flow within the model, particularly through its most important components?

- **RQ3**: Which signals flow through the model and contribute to predicting relevance or non-relevance?
- **RQ4**: How is the relevance signal constructed within the [CLS] representation of cross-encoders?

These main research questions structure our work. We study each one for different models and across different domains to answer a series of orthogonal research questions that strengthen our findings:

- **RQ5**: To what extent are these mechanisms consistent across different sizes of cross-encoders?
- **RQ6**: How robust are these findings to domain shifts?

In Section 2, we present an overview of the current state of understanding of how transformer-based models [94] work in general (Subsection 2.2) and more precisely in IR (Subsection 2.3). Then, we detail our experimental protocol in Section 3, including the different models and datasets we used to answer **RQ5** and **RQ6**. Once the setup is explained, we detail in sections 4 to 7 the results of our experiments and how they help us shape answers to our research questions. We summarize below the main highlights of these different sections.

To identify the most crucial components involved in the relevance prediction (**RQ1**), we follow our previous work in [93], leveraging the Neuron Integrated Gradients (NIG) method [82] (see section 4).

> *Key highlights.* Some components are important *across* domains, while others are important *specifically* in OOD, particularly for large models. In particular, our method highlights the key role of a few attention heads and FFN neurons in the relevance prediction process.

As a "naive" usage of NIG cannot provide enough details on the nature of the contributions from the critical components, we investigate the flow of information inside the model (**RQ2**). In Section 5, we combine insights drawn from improving the granularity of the original NIG technique, extending the ablation analysis from Zhan et al. [111], and using a variant of the Information Bottleneck Attribution [78]. While NIG and ablations allow for the extraction of the most salient informational directions throughout the model, the Information Bottleneck yields a more holistic view of the information flow. Altogether, they map the pathways through which information flows between input segments.

> *Key highlights.* We empirically validate the existence of the multi-stage process described in Zhan et al. [111] across both ID and OOD, but provide more details: Our results confirm that the relevance prediction process begins with the contextualization of the query and document tokens and, in parallel, query-to-document information transfers (i.e., the document tokens receive information from the query tokens through the attention mechanism). Contrary to [111], we observe that document-to-query transfers are equally important at this stage. These transfers last for most of the models' layers, until the later layers. At this point, relevance information shared across document and query tokens is not directly transmitted to the [CLS] but is first accumulated in the query tokens before being sent to the [CLS] in the very last few layers. In addition, we confirm the role of the "no-op" operation [18] for [SEP] and further show that [CLS] might play a similar role through the early to the middle layers.
> Finally, our study reveals two new insights: 1) not all cross-encoders share the same process of accumulating relevance information inside the [CLS] and 2) MiniLM, contrary to MonoBERT cross-encoders, does not seem to rely on "relevance" signals to predict relevance, but instead on "non-relevance" signals.

By answering **RQ1** and **RQ2**, we notice the key role of some high-attribution attention heads in query-to-document and document-to-query transfers, a behavior we link to the detection of matching signals across them. To derive the nature of these matching signals (**RQ3**), we analyze in Section 6 the attention patterns of every attention head by contrasting their attributions when exposed to relevant pairs versus non-relevant ones. Our objective here is twofold: firstly, to

uncover the nature of the query-document interactions detected by high-attribution heads, and secondly, to discover heads specialized in detecting *relevant* matching signals.

Furthermore, we extend this experiment with a mechanistic interpretability study (see subsection F.3) to better understand the mechanism at stake that allows these heads to match together (contextualized) query tokens and (contextualized) document tokens.

> *Key highlights.* Our results confirm that heads specialized in detecting matching signals between queries and documents exist in the model. By analyzing their attention patterns, we further characterize the nature of the matching signals they capture and report two distinct stages. From early to middle layers, these interactions match occurrences of the same token in the query and the document (*syntactic matching*), while in the last layers, matching occurs between semantically-contextualized tokens (*semantic matching*) that are close to each other. These heads are also among the most important components of cross-encoders, which confirms the significance of matching signals, and our mechanistic study explains the behavior of most of them: their Query-Key circuit favors cross-attention (query to document or document to query) over self-attention, and associates highly similar source and target tokens.

After discovering the nature of the signals that are leveraged by cross-encoders to predict relevance, we look into the process behind the construction of the "relevance" signal itself (**RQ4**). We target key stages of the process, i.e., when the information stored inside the query tokens is transferred into the `[CLS]` representation, and investigate the geometry of the hidden state's subspaces to understand if a "relevance" subspace exists.

> *Key highlights.* Our analysis reveals differences between MiniLM, MonoBERT-base and MonoBERT-large, in both the construction and the nature of their respective "relevance" subspace. In addition, we report that relevance signals emerge in the `[CLS]` token only in the final layers, which supports the view that it performs a "no-op" function in earlier layers, akin to the `[SEP]` tokens [18].

Taken together, our findings both reinforce and refine prior insights into cross-encoder behavior. In particular, we are able, not only to confirm, but also to specify, the flow of information between input parts inside BERT-based cross-encoders. Crucially, we go beyond the existing work by testing the consistency of this finding across multiple architectures and domains (**RQ5**, **RQ6**). In doing so, we help move the field toward a more unified and generalizable understanding of how cross-encoders compute relevance.

2 Related Work

2.1 Comparisons of interpretability approaches

To interpret neural models, researchers rely on a large variety of interpretability techniques that can be categorized into distinct families. In IR, some architectures such as SPLADE [33] or ColBERT [47] are "more explainable" than cross-encoders which must rely on the interpretability approaches we describe next.

One of the most widespread ways to access the internal knowledge of neural networks is through probing [5]. Probing studies rely on training external models from hidden states on auxiliary but related tasks (e.g., for IR, it can be Named-Entity Recognition, Semantic Similarity or co-reference Resolution [90, 101]), revealing the specific abilities that the model learned implicitly during its training to solve the task. Probing is popular because it is easy to use: given a dataset for the auxiliary task, one can test whether a model's activations encode the corresponding information. In IR, probing studies often rely on "diagnostic datasets" related to specific IR axioms [10, 12, 15]. However, because probing relies on external classifiers [5], its results are only correlational and can be hard to interpret, especially with overly

expressive nonlinear probes. In addition, probing methods cannot be used to target specific neurons as they use each hidden representation as a whole and do not provide any explanation of the interplay between these abilities, as well as their relative importance in the model output.

Other simple introspection techniques include the study of the attention process (e.g., the distribution of the attention weights across the tokens) in transformer-based architectures. Similarly to probing, these techniques have been criticized for providing explanations uncorrelated to the predictions of the model [44]. This claim has been challenged since [107], and researchers now address this issue by combining attention patterns analysis with other interpretability methods to strengthen their conclusions [111].

A way to test the importance of specific components is through ablation studies. By comparing the performance of the model with and without a specific part, one can directly assess its importance. Ablations belong to the extended family of *perturbation-based* attribution approaches [32, 70, 74], as they compare the outputs of the original model with those of the ablated model. Similarly, mechanistic interpretability [11] approaches[1], such as activation patching [64, 96] and path patching [38], introduce perturbations inside the computational graph of the model to deduce the importance of specific components. While they led to significant breakthroughs [30, 69], including in IR [12, 56], mechanistic interpretability approaches and perturbation-based ones are hard to scale in an exhaustive study like ours. Even for more straightforward approaches, such as ablations, perturbations need to be conducted in isolation and put into perspective with the perturbations of the surrounding components. On larger models than BERT, this can make them intractable.

More tractable attribution methods are the *backpropagation-based* ones. They usually have more desirable theoretical properties than perturbation-based methods, such as sensitivity or additivity [79, 81, 84]. Their main drawback is the difficulty of giving a human-understandable meaning to their attribution. Among such techniques, the Integrated Gradients [84] and Neuron Integrated Gradients (NIG) [26, 58] respectively allow tracing back the contributions of input parts and individual neurons to the predictions of the model. However, these approaches fail to provide insights into the nature of these contributions. Notably, attribution patching [85] alleviates this limitation for mechanistic interpretability approaches by making use of gradient-based approximation (borrowing from NIG-related techniques), but it can lead to the appearance of false negatives [48], potentially harming its conclusions.

While attribution methods help to understand the relative importance of input parts or components of the model (at various granularity levels), they fall short of providing a holistic view of information flow without being applied repeatedly across the architecture. In contrast, the Information Bottleneck [78] allows for automatically determining the most important parts in the input. This approach estimates the minimal information flow between an input X and an output Y, necessary to preserve Y, limiting the components to their most important features. As it acts on all the components at once, it further accounts for their relations, alleviating one of the main limitations of attribution methods – but at the cost of some potential instability during training. However, its results need to be further refined to understand what makes a part with a high attribution score important.

These different approaches, along with their associated findings, strengths, and weaknesses, are summarized in Table 1. Based on this comparison, several techniques emerge as especially promising for our study. The analysis of attention patterns will be useful for examining the information exchanges between the queries and the documents (or matches) and for determining the nature of these exchanges (**RQ3**). Both perturbation-based (ablations) and backpropagation-based (NIG) attribution methods are efficient and easy to interpret. They give the contribution of each component

[1]Mechanistic interpretability is a line of work that tries to "*reverse engineer transformers into human-understandable computer programs*" (quote taken from the Transformer Circuits blog: https://transformer-circuits.pub/)

Table 1. Summary of interpretability approaches in IR and whether we used them to answer our Research Questions (RQ).

Method	Strengths	Weaknesses	Used in	Sections / RQ
Probing	Versatility	Results are correlational, poor granularity	[14, 15]	None
Ablations	Ease-of-use, granularity	Needs to be scaled for better analysis	[111]	5 / RQ2
Attention patterns	Ease-of-use, can be applied to a few samples or a dataset	Results need further interpretation	[14, 56, 111]	6 / RQ3
NIG	Gives the importance of individual neurons	No info. on their role(s)	[93]	4,5 / RQ1,2
Information Bottlenecks	Get important features all at once	Needs (limited) training	[45]	5 / RQ2
Causal / Path Patching	Give the role of components	Hard to scale	[12, 56]	None

(**RQ1**) and help put other results into perspective, e.g., are the important attention heads those that detect matches? (**RQ2, RQ3**). Finally, the Information Bottleneck approach, despite involving some training, provides a suitable way to validate the findings of other methods and analyze the information flow (**RQ2**) by considering all the interactions inside the models at once. Table 1 also summarizes the Sections where each technique is deployed, as well as the Research Questions it is used to answer. In this work, the choice of methods is mainly motivated by the ease of scaling them and interpreting their results. Indeed, as we want to study three models across various datasets (**RQ5, RQ6**), we cannot afford to use methods that require too many computing resources (like mechanistic interpretability). For the same reasons, we refrain from using probing, as it is hard to link its results to the actual mechanisms of the model.

In addition to the methods listed above, we further developed our own framework to interpret neural IR models. Following our previous work involving NIG to study the most important components inside MonoBERT [93] in both in-domain and out-of-domain settings, we repeat each experiment in both setups and cross the results to differentiate the in- and out-of-domain behavior. Extending this study, we further develop a method based on NIG, which differentiates the different parts inside the input (the `[CLS]`, the query, the passage, and the two `[SEP]` tokens) to study the most important directions of the attention inside the model as well as the most important components in the FFN. To study the roles of the attention heads, we extend the traditional attention patterns analysis by contrasting the distribution of attention scores between query-relevant passage pairs and query-non-relevant passage pairs to obtain the attention heads whose roles are directly related to detecting signals important for the prediction of relevance. Lastly, to study at which points the relevance signal starts to be encoded inside the `[CLS]` token representation (**RQ4**), we analyze the representational subspace of the different tokens with Linear Discriminant Analysis.

So far, we reviewed methods designed to *interpret* models' behaviors, i.e., decipher their inner mechanisms. By design, interpretability is tied to the studied model. On the other hand, *explainability* refers to a different line of work, whose methods produce post-hoc explanations, are generally model-agnostic, and specifically target the role of the input features [57, 74]. In IR, notable works include applications of SHAP [17, 41] or LIME [16, 95] to explain pointwise predictions and even listwise rankings (for a more comprehensive view, we refer the interested readers to this tutorial [3]). We consider this line of work complementary to ours, as its focus is mostly on providing post-hoc explanations of a model's behavior, rather than on its *internal mechanisms*.

2.2 Interpreting BERT and transformer-based models

Numerous works investigate the behaviors and tendencies of BERT [25], a field of research called BERTology [75]. In particular, studies have shown that earlier layers tend to encode syntactic knowledge, such as parts of speech, syntactic structures (chunks, trees, or dependencies) [18, 42, 53, 55]. BERT models have also been found to encode IDF information inside their parameters, in particular inside their embeddings [14], and some of their behaviors appear to depend on token frequency, with attention heads in early layers attending primarily to rare tokens [97]. In addition to syntax and lexical properties, BERT's embeddings are also known to be highly contextualized [31], i.e., each token representation encodes information about its surroundings. In later layers, contextualization allows the model to capture the words' meaning [106].

Combining these insights, it is possible to draw a high-level structure underlying the different steps of a forward pass in a transformer model. In the first layers, neurons detect surface signals and shallow patterns in the input. At the same time, the input is contextualized, and moving on to the middle layers, neurons specialize in higher-level concepts. Finally, in the last layers, neurons use the signals extracted earlier to take actions and promote features in the residual stream. In the context of BERT-based models specifically, this is encapsulated by the fact that pre-trained BERT models can encode knowledge related to different NLP tasks and organize them dynamically into a pipeline [86].

These insights obviously overlap with IR interpretability studies, and ours in particular, as most observed behaviors and conclusions are transferable across all NLP tasks. Yet, some specificities of IR, like the modeling of relevance, are not properly captured by generic NLP studies. IR thus developed its interpretability field, summarized in the next section.

2.3 Interpreting neural IR models

Soon after BERT was successfully applied to IR [67], studies aimed at understanding how neural IR models work and how they compare to traditional IR approaches emerged. In particular, Zhan et al. [111] and Jiang et al. [45] both study the attention process [54] of MonoBERT. Zhan et al. state that MonoBERT predicts relevance by first contextualizing queries and documents, then capturing interaction signals, and finally composing them. With the Information Bottleneck attribution method [78], Jiang et al. find that MonoBERT not only relies on lexical matching but also captures semantic relationships between the contextualized representations of the query and the document tokens. They also provide further details on the role of `[CLS]`, which starts to gather relevance information in the very last layers of the model. On this matter, Zhan et al. [111] go even further and show that across most of the model's layers, `[CLS]`, `[SEP]`, and punctuations attract a large proportion of attention but carry little relevance information, echoing a similar behavior to the "no-op" operation [18]. They further emphasize the importance of document-to-query information transfers while questioning the role of query-to-document transfers. Overall, their study provides various insights into the inner mechanisms of MonoBERT. However, they do not specify the types of interactions detected (our focus in Section 6) or how these contribute to the relevance signal (studied in Section 7), and their conclusions on attention directionality, between query and document, lack stronger empirical evidence (as illustrated in our Section 5).

More recently, Lu et al. [56] apply path patching [38], another mechanistic interpretability approach, to a BERT-based cross-encoder and observe that it "rediscovers" a semantic variant of BM25, as mentioned in Jiang et al. [45]. They identify specific attention heads responsible for computing "soft" term frequency while accounting for term saturation and document length effects (Matching Heads). Combined with the finding that the IDF is encoded in a low-rank component of the model's embedding matrix [14], Lu et al. further show that the model output can be reconstructed (correlation of 0.80) using these signals.

Aside from BERT-based ranking models, some works have also started to explore the mechanisms of LLMs for ranking. Notably, Chowdhury et al. [15] conduct a probing study to examine activations in fine-tuned LLMs for passage re-ranking, finding that certain human-engineered features, such as the number or ratio of covered query terms, are encoded in activations, while others, like BM25 scores, do not seem to be present (no linear correlation). If we further extend the scope to different architectures, other works have also explored the behavior of learned sparse retrieval models [62], late-interaction models [34], and bi-encoders [12, 65], in particular the matching behaviors of these models. For example, Chen et al. [12] discover the key roles of some attention heads in detecting duplicate instances of the same tokens while studying TAS-B [43], a dual embedding model, using activation patching [64, 96]. However, these studies are architecture-specific and cannot be generalized to the study of cross-encoders.

As we have seen in this section, our understanding of how cross-encoder models work is rapidly growing, fueled by a wide variety of interpretability approaches. Besides the aforementioned limitations, most of these studies, including [56], are conducted in a relatively limited setup, studying a single model and a single dataset. This questions the generalization of their findings across models (architectures, as well as sizes) and across domains. For example, MonoBERT has been shown to exhibit different behaviors with in-domain versus out-of-domain [93], and larger models (such as MonoBERT) have been hypothesized to behave differently than shallower neural rankers [73]. Thus, this motivates the need to conduct a more holistic interpretability study. In this work, we conduct our experiments using three cross-encoders of different sizes to compare their mechanisms and evaluate the impact of the number of parameters on their behavior (**RQ5**). In addition, we use several datasets covering various domains to verify the extent to which the mechanisms they learn can be transferred to out-of-domain data and to strengthen our findings (**RQ6**).

3 Experimental setup

3.1 Cross-Encoders Architecture

Cross-encoders classify a query-document pair (q, d). Note that in this work, we focus exclusively on BERT-based cross-encoders and consider that both the queries and the documents are textual. This excludes other re-rankers such as Qwen3-VL [52] which are based on a decoder-only backbone and/or deal with other modalities than text. We denote $\mathbf{x}$ the cross-encoder input, which is composed of the sequence of tokens `[CLS]` $q_1 \dots q_n$ `[SEP1]` $d_1 \dots d_m$ `[SEP2]` (n query tokens and m document tokens). The transformer architecture updates the representation of the input tokens after each layer $\ell \in [0 \dots L-1]$ (layers are indexed from 0 throughout the paper, consistently with the figures and tables). We denote $\mathbf{x}_j^{(\ell)} \in \mathbb{R}^{d_\mathcal{T}}$ the representation of the token j at the output of layer[2] ℓ, where j can be the token `[CLS]`, a query q_k or document token d_k, `[SEP1]` or `[SEP2]`. Each layer ℓ is composed of two main processing steps:

(1) **The attention step**, summarized as:

$$\begin{aligned} \mathbf{x}_j^{(\ell,\tilde{a})} &= \sum_{h \in \mathcal{H}_\ell} W_O^h \, \text{SelfAttention}^h(\mathbf{x}_j^{(\ell-1)}, \mathbf{x}^{(\ell-1)}) \\ \mathbf{x}_j^{(\ell,a)} &= LN\left(\mathbf{x}_j^{(\ell,\tilde{a})} + \mathbf{x}_j^{(\ell-1)}\right) \end{aligned}$$

where $h \in \mathcal{H}_\ell$ is an attention head among the H heads of layer ℓ, and $\mathbf{x}^{(\ell-1)}$ denotes the representations of all the input tokens at layer $\ell - 1$. An attention head h works in a subspace of dimension $d_\mathcal{H} = d_\mathcal{T}/H$, and is computed

[2]For consistency, we assume $\mathbf{x}_j^{(-1)}$ is the model input embedding.

using the following formula:

$$\text{SelfAttention}^h(\mathbf{x}_j^{(\ell-1)}, \mathbf{x}^{(\ell-1)}) = \sum_i \text{softmax}_i\left(\frac{\left(W_Q^h \mathbf{x}_j^{(\ell-1)}\right)^\top \left(W_K^h \mathbf{x}_i^{(\ell-1)}\right)}{\sqrt{d_\mathcal{H}}}\right) W_V^h \mathbf{x}_i^{(\ell-1)}$$

where $W_Q^h, W_K^h, W_V^h \in \mathbb{R}^{d_\mathcal{H} \times d_\mathcal{T}}$ are the query, key and value matrices, and $W_O^h \in \mathbb{R}^{d_\mathcal{T} \times d_\mathcal{H}}$ is the output matrix. Last, LN is the layer norm [4] that normalizes each vector to zero mean and unit variance across its components, before applying a learned affine transformation.

(2) **The feed-forward network (FFN) step**:

$$\mathbf{x}_j^{(\ell,\tilde{f})} = \sum_{i \in \text{FFN}_\ell} \phi(W_i^{\ell,\text{in}} \cdot \mathbf{x}_j^{(\ell,a)} + b_i^{\ell,\text{in}}) W_i^{\ell,\text{out}} + b^{\ell,\text{out}}$$

where ϕ is the activation function (GELU), $W_i^{\ell,\text{in}} \in \mathbb{R}^{1 \times d_\mathcal{T}}$ and $W_i^{\ell,\text{out}} \in \mathbb{R}^{d_\mathcal{T}}$ are respectively the i-th row of the input projection and the i-th column of the output projection, and the number of FFN neurons $|\text{FFN}_\ell|$ is a multiple (e.g., 4) of $d_\mathcal{T}$. From this formulation, "FFN neuron i", used throughout the work, corresponds to the scalar $\phi(\cdot)$.

For any token j, the final representation $\mathbf{x}_j^{(\ell)}$ on layer ℓ is computed with

$$\mathbf{x}_j^{(\ell)} = LN(\mathbf{x}_j^{(\ell,\tilde{f})} + \mathbf{x}_j^{(\ell,a)})$$

Finally, the classification head of a cross-encoder is a two-layer neural network which is applied to the `[CLS]` token representation $\mathbf{x}_{\texttt{[CLS]}}^{(L-1)}$ at the last layer $L-1$, formally

$$P(\mathbf{x}_{\texttt{[CLS]}}^{(L-1)}) = \sigma(w_2 \cdot \tanh(W_1 \mathbf{x}_{\texttt{[CLS]}}^{(L-1)} + b_1) + b_2)$$

where W_1, w_2, b_1 and b_2 are parameters, σ the sigmoid function that ensures that the value is between 0 and 1[3]. As a shorthand, for an input $\mathbf{x}$, we denote $P(\mathbf{x})$ the output of a cross-encoder when applied to the last layer representation of the token `[CLS]`, i.e., $\mathbf{x}_{\texttt{[CLS]}}^{(L-1)}$. We denote $z(\mathbf{x})$ the corresponding logit, i.e., the input of the sigmoid σ in P.

In this paper, we consider three BERT-based cross-encoders with different numbers of parameters but also pre-training, and study their respective behaviors. More precisely, we study cross-encoders based on MiniLM-L12-v2 [102], BERT-base and BERT-large [67], i.e., MonoBERT-base and MonoBERT-large. In the remainder of the paper, we simply refer to the former as MiniLM. While there exist fine-tuned checkpoints for MonoBERT-large and MiniLM, we had to fine-tune a version of MonoBERT-base ourselves, using the training split of MS MARCO[4]. We report the training recipe and the performance of our MonoBERT-base in Appendix A.1, along with the architectural details of the models in Table 4.

These three models let us vary three factors: size (from 33M to 340M parameters), depth (12 or 24 layers) and pre-training strategy (the classical Masked Language Modeling and Next Sentence Prediction of BERT [25], or the deep self-attention distillation of MiniLM [102]). MonoBERT-base serves as a "control" model: it has the depth of MiniLM (12) and the pre-training of MonoBERT-large, so that the influence of each factor can be studied separately (see **RQ5**).

[3]In the case of MonoBERT, the network has two outputs, one for not relevant and another for relevant, and the probability distribution is obtained through a softmax, which is equivalent to a sigmoid. We assume in the remainder of this paper that there is only one output corresponding to the prediction of relevance.

[4]All models are available on the Hugging Face Hub: castorini/monobert-large-msmarco / xpmir/cross-encoder-bert-base-infoNCE / cross-encoder/ms-marco-MiniLM-L12-v2

3.2 Datasets

For our experiments, we rely on the Deep Learning tracks from TREC (TREC-DL) between 2019 and 2022 [20–23] to observe the behavior of the models on in-domain data (ID), as well as on a subset of the datasets composing the BEIR benchmark [87] for out-of-domain data (OOD).

TREC-DL datasets complement the MS MARCO dataset [66] with manually annotated relevance judgments, with multiple annotated passages per query. It is thus more reliable (fewer false negatives). We restrict ourselves to the queries that have been manually annotated by the NIST assessors (*judged* version of the data on *ir-datasets* [61]). This amounts to 226 topics and more than 400k query relevance assessments along 4 degrees of relevance $r \in [0..3]$.

For OOD (see **RQ6**), we carefully select the datasets corresponding to tasks that resemble the most a traditional IR setup in the BEIR benchmark [87] according to their classification[5]: FiQA [63], TREC-Covid [98], TREC-News [83], NFCorpus [6], BioASQ [89], Natural Questions[6] [49] and TREC-Robust04 [99]. From this reduced pool of datasets, we further eliminated TREC-News because of the format of the task, as "queries" are in fact documents from the corpus. Since some datasets are not directly available in BEIR and *ir-datasets* [61] offers several versions with different pre-processing, we report in Appendix A.3 the performance of the three models on the versions we use.

As documents labeled as "not relevant" could be semantically related (they were retrieved by an IR model, e.g., BM25), we also study the behaviors of the models in the case where the document is irrelevant *and* not semantically related to the query. To obtain these additional levels of annotations, we randomly sample documents for each query (excluding assessed ones). Following the terminology used in Neural IR training methodology, we call those documents *easy negatives*, as opposed to the *hard negatives*, labeled as non-relevant [88]. We attribute the relevance level $r = -1$ to these passages and use $r = 0$ for hard negatives. For each dataset listed in Table 6, we consider the highest available level of relevance to be "highly relevant", or $r = 1$, following the annotation in the 4 tracks of TREC-DL. Throughout our work, we focus on these three levels of relevance. In addition to this pre-processing, note that we also applied three different sampling strategies to each dataset's list of query relevance judgments, to meet the requirements of our experiments. These three strategies, called respectively *Top sampling*, *Diverse Sampling* and *BM25 sampling*, are detailed in the Appendix B.2, Figure 17.

We access the datasets using *ir-datasets* [61][7] and managed the experimental plan with the *experimaestro* framework [71, 113]. All the code necessary to reproduce the results is public[8].

4 RQ1: Identifying the contribution of the different components to the prediction of relevance

To unveil the different components involved in modeling and predicting relevance in cross-encoders, we first look at characterizing their respective contributions when performing the IR task. Following Vast et al. [93], we use the Neuron Integrated Gradients approach (or NIG) [26, 58] to identify a reduced set of core neurons with consistently high attributions. In what follows, we begin by describing NIG, how it can be adapted to fit the IR task requirements, and the specificity of this work compared to the original setup.

[5]The BEIR benchmark covers a total of 9 tasks, among which 3 can be considered the closest to the IR task: News Retrieval, Question-Answering (minus HotpotQA [110] which is a multi-hop QA dataset) and Bio-Medical IR.
[6]Note that for Natural Questions, we use the document collection as it was pre-processed in DPR [46] to benefit from their additional filtering.
[7]For TREC-DL 2022, this implies that the version we use has been filtered down to remove all duplicate documents.
[8]Code available at: https://github.com/MathVast/CE-relevance-integration

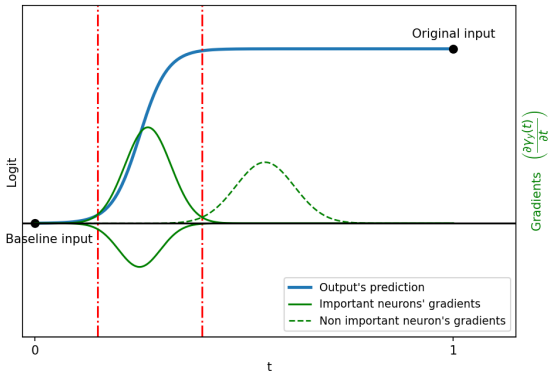


Fig. 1. Illustration of the Neuron Integrated Gradients method. The blue curve (left axis) describes the recovery of the model's logit $z(\cdot)$ (i.e., the input of the sigmoid in P) as we progress along the path $\gamma(t)$. Green curves (right axis) represent the gradients $\frac{\partial \gamma_y(t)}{\partial t}$ of individual neurons along this path: plain curves correspond to important neurons, whose gradient is non-zero where the logit varies (between the two red lines), and the dashed curve to a non-important neuron.

4.1 Background

(Neuron) Integrated Gradients. Originally designed for computing the attributions of the input features (to the output), Integrated Gradients (IG) [84] is based on path integrals. IG requires carefully selecting a baseline input $\mathbf{x}'$ for which the model's prediction is neutral, e.g., in IR $p(relevant|\mathbf{x}') = p(non-relevant|\mathbf{x}')$. IG measures how the decision evolves between $\mathbf{x}$ and $\mathbf{x}'$, and more precisely along the path defined by $\gamma(t) = \mathbf{x}' + t(\mathbf{x} - \mathbf{x}'), t \in [0, 1]$. IG has been generalized into *conductance* [26] to obtain the importance of any neuron y. In this framework, the importance of an individual neuron is given by the Neuron Integrated Gradients formula [82]:

$$NIG^y(\mathbf{x}) = \int_{t=0}^{t=1} \frac{\partial P(\mathbf{x}' + t(\mathbf{x} - \mathbf{x}'))}{\partial \gamma_y(t)} \frac{\partial \gamma_y(t)}{\partial t} dt \tag{1}$$

where P is the classification model (our cross-encoder in Section 3.1), $\gamma_y(t)$ denotes the activation value of a neuron y at the point t of the path γ, i.e., the output y of the network, $P_y(\gamma(t))$. Note that this can be efficiently estimated [82].

To illustrate the intuition behind Neuron Integrated Gradients, let us use Figure 1 that depicts the evolution along the path $\gamma(t)$, between the "Baseline" $\mathbf{x}'$ and the "Original" input $\mathbf{x}$, of the model's logit (in blue, left axis) and the gradients $\frac{\partial \gamma_y(t)}{\partial t}$ of two types of neurons (in green, right axis). In the interval delimited by the two red vertical lines in the figure, a change in the input results in a significant recovery of the original output signal, symbolized by a strong increase in the blue curve. According to NIG, the important neurons are neurons y whose gradient $\frac{\partial \gamma_y(t)}{\partial t}$ evolves with the output P, i.e., whose gradient is non-zero inside this interval (plain green curves) – note that the contribution can be either positive or negative. Conversely, a neuron whose gradient is non-zero only outside of the zone of variation of P (dashed green curve) is not important for the prediction.

Over the years, a series of works have improved the quality of the attributions from Neuron Integrated Gradients. Notably, Integrated Decision Gradients [100] samples mostly in the region where the model decision changes (between the two red vertical lines in Figure 1). We use their algorithm to sample the steps and obtain better attributions.

Neurons. Usually, "neurons" correspond to the activation of any layer/block in the model. Previous works used an even more restricted scope and used "neurons" to refer to outputs of linear transformations (used to compute keys,

queries, values, and inside the feed-forward block) and demonstrated their importance in various mechanisms (e.g., factual recall, performing auxiliary tasks) [13, 24, 37, 90, 101, 103]. While Vast et al. [93] followed the same definition and applied NIG to the outputs of the linear transformations in MonoBERT, we argue that not all of them are equally interesting and can provide meaningful interpretations. As pointed out in [8], activations in transformer-based models do not necessarily possess a *privileged basis*, which makes them hard to interpret. Elhage et al. [30] list "tokens, attention patterns and FFN activations" as "the only vectors with privileged bases" based on their mathematical definitions, but later works discovered that some features in the residual stream are also *basis-aligned* [29]. Motivated by these works, we further restrict the scope of vectors and study the attention patterns (i.e., the attention probabilities obtained after the softmax over the query–key products) and the FFN neurons (see Section 3.1).

4.2 Adapting NIG to identify "task-related" neurons

As IR differs from other domains (CV or NLP) in which NIG has been applied, it requires some adaptations.

Comparisons across datasets. To answer **RQ6** and draw fair comparisons between attributions obtained for different datasets, we aggregate the contribution of each activation over tokens. Since the same neuron is applied to every token of the query-document pair, we sum its conductance over tokens: a "neuron" in our experiments thus includes its corresponding outputs over all tokens. For the FFN neurons, this aggregated conductance can be thought of as the importance of the neuron within the Transformer. For the attention heads, we obtain a score characterizing their importance within the model.

Dependency on the input. As pointed out by Wang et al. [103], gradient-based methods produce results that are input-dependent. As we want to identify "IR task-related" or "skill" neurons, i.e., neurons that are important for any input in the IR task, we average NIG results over multiple samples across multiple datasets and only retain the neurons with the highest mean attribution values.

Baseline. For images, an obvious baseline $\mathbf{x}'$ is a black image. In IR and NLP in general, there is no obvious baseline, and current works that apply NIG to NLP tasks do so at the level of the embeddings. In the IR task, Zhan et al. [111] use Integrated Gradients to study the role of different input parts at different stages of MonoBERT. As a baseline, they replace the part they want to study (for example, the query tokens) with `[PAD]` tokens, compute the corresponding embedding of the baseline, and then proceed with the method. We follow the same procedure to build our baselines.

Better targeting "relevance" neurons. Unlike our previous work [93], where we applied NIG to query-passage pairs and considered positive passages and negative ones separately, and to [111], who used only relevant query-passage pairs, we work with triplets (query, positive passage, negative passage). In practice, this is closely related to the InfoNCE loss [91]. The rationale is that we want to study the discriminative power of cross-encoders and not only their pointwise decision process. Moreover, this simplifies our analysis compared to [93] as we have only a single set of important neurons per dataset instead of two (i.e., one for the positive passages and one for the negatives). We hence use this version of Eq. (1):

$$\mathrm{NIG}^{y}_{\mathrm{InfoNCE}}(\mathbf{x}_+, \mathbf{x}_-) = \int_{t=0}^{t=1} \frac{\partial \sigma\left(z(\mathbf{x}'_+ + t(\mathbf{x}_+ - \mathbf{x}'_+)) - z(\mathbf{x}'_- + t(\mathbf{x}_- - \mathbf{x}'_-))\right)}{\partial \gamma_y(t)} \frac{\partial \gamma_y(t)}{\partial t} dt \tag{2}$$

where $\mathbf{x}_+$ and $\mathbf{x}_-$ respectively represent a query-relevant passage pair and a query-non-relevant passage pair, $\mathbf{x}'_{\{+,-\}}$ their respective baselines, and $z(\cdot)$ the model's logit (see Section 3.1).

4.3 Experiment description

Using the datasets described in Section 3.2 and the *Top sampling* strategy as described in Appendix B.2, Figure 17, we obtain one set of NIG scores per dataset and per model, corresponding to the importance of each neuron. While it is interesting to observe the most important neurons for a given sample or even a specific dataset, we are more interested in identifying the most important neurons *in general* for the IR task. Following Vast et al. [93], we achieve this by aggregating the different sets of NIG scores across multiple datasets with:

- **Intersection**: $\mathrm{I}(t, C) = \bigcap_{c \in C} \mathrm{Top}\%(t, c)$, where C is a pool of datasets and $\mathrm{Top}\%(t, c)$ is the set of the top $t\%$ neurons for a dataset c;
- **Fusion**: $\mathrm{F}(t, C) = \mathrm{Top}\%\left(t, \langle \mathrm{NIG}_c \rangle_{c \in C}\right)$, where $\langle \cdot \rangle_{c \in C}$ corresponds to the average, neuron-wise, of the NIG scores.

The rationale behind these two different ways of aggregating the results across different datasets is that each allows us to observe a different phenomenon. With the Fusion, we can observe *generic* important neurons, i.e., neurons that are important on average over the datasets. In contrast, the Intersection allows us to observe the important neurons *shared* across the pool, i.e., neurons that are important for each dataset in the pool. Using these operations and for $t \in \{0.1\%, 0.5\%, 1\%\}$, we select the most important neurons from the following sets of datasets:

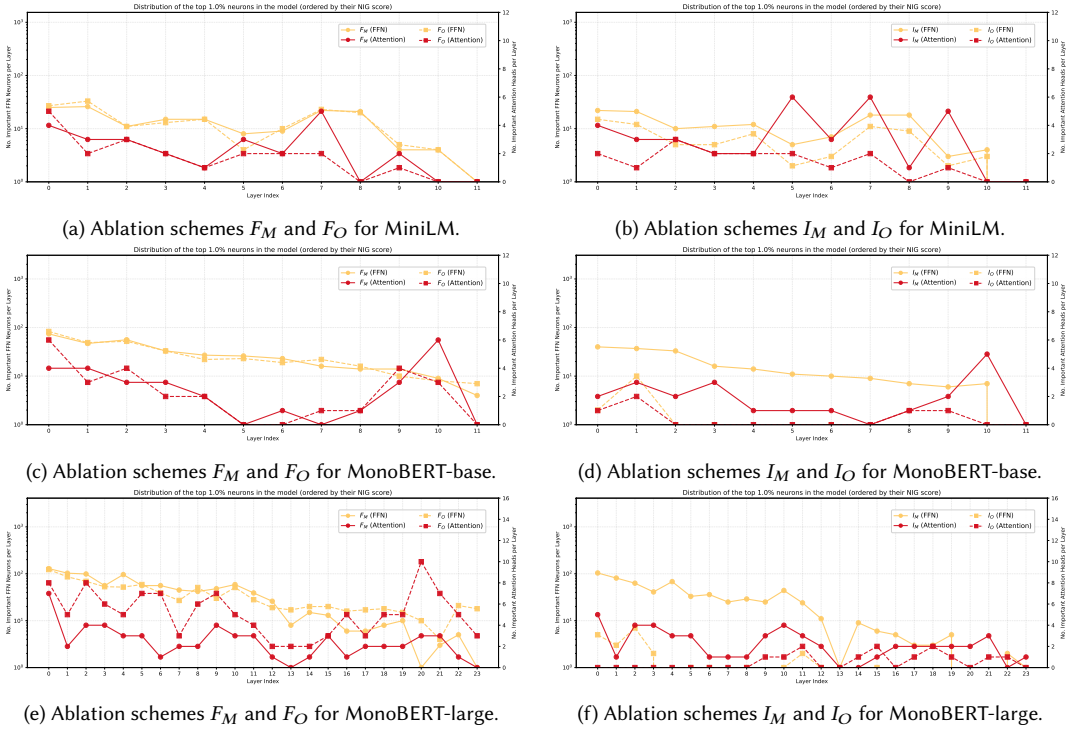
- **M** corresponds to the 4 TREC-DL datasets [20–23] all derived from MS MARCO [66];
- **O** corresponds to the out-of-domain datasets, i.e., all datasets from Table 6 but the 4 TREC-DL datasets;
- **A** corresponds to all the datasets.

By combining a set of datasets, an aggregation operation, and a percentage t, we obtain a list of the most important neurons inside a model. We follow Vast et al. [93] and refer to this list as an *ablation scheme*, even if we do not perform a complementary ablation study. We denote an ablation scheme by its aggregation operation and its pool of datasets, e.g., $F_M = \mathrm{F}(t, M)$, $I_O = \mathrm{I}(t, O)$ or $I_A = \mathrm{I}(t, A)$; unless stated otherwise, $t = 1\%$. We refer to t as the *pruning percentage*.

4.4 Results analysis

■ **Important neurons according to NIG are not located in specific layers or transformer components.** Figure 2 reports the neurons selected with NIG for MiniLM and both versions of MonoBERT. We can see that, in proportion and across ablation schemes, the most important parts of the models tend to be attention heads, but no layer particularly stands out, even though there seems to be a trend where the deeper the layer, the less important its components. Given the known roles of the different layers in transformer-based models (see Section 2), this suggests that relevance prediction involves several consecutive stages, in which attention plays a key role by moving information across input parts.

■ **When exposed to domain shifts, MonoBERT tends to rely on different neurons, whereas MiniLM has learned a more consistent mechanism.** By construction, $F(t, \cdot)$ always contains $t\%$ of the neurons while $I(t, \cdot)$ contains at most $t\%$, so the gap between the I curve and the F curve measures how much the top-$t\%$ set changes from one dataset to the next. In Figure 2, for both MonoBERT models, I_O is nearly empty while F_O is full (see the dashed curves of Figures 2d and 2f, versus the dashed curves of Figures 2c and 2e), so the top neurons differ across OOD datasets; for MiniLM, I_O and F_O have comparable cardinalities at each layer, i.e., the same neurons are among the most important for every dataset. In other words, there are comparatively fewer neurons that are *always* important across different situations in MonoBERT-base and large than in MiniLM, which leads us to the conclusion that MiniLM relies on a more consistent mechanism than MonoBERT.



(a) Ablation schemes F_M and F_O for MiniLM.

(b) Ablation schemes I_M and I_O for MiniLM.

(c) Ablation schemes F_M and F_O for MonoBERT-base.

(d) Ablation schemes I_M and I_O for MonoBERT-base.

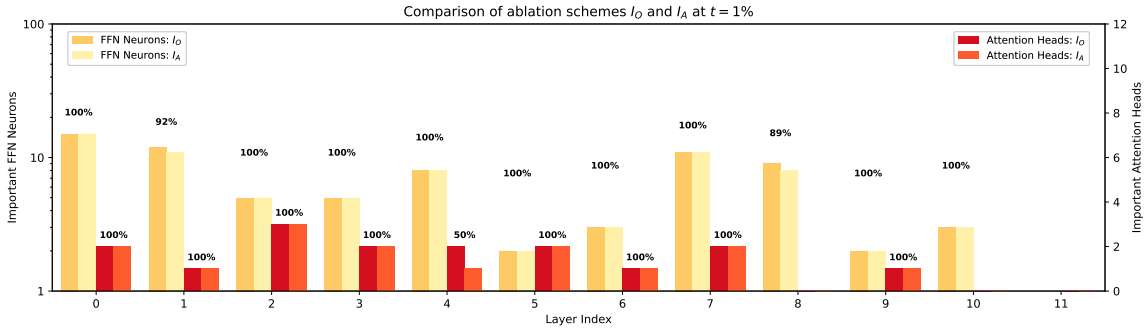
(e) Ablation schemes F_M and F_O for MonoBERT-large.

(f) Ablation schemes I_M and I_O for MonoBERT-large.

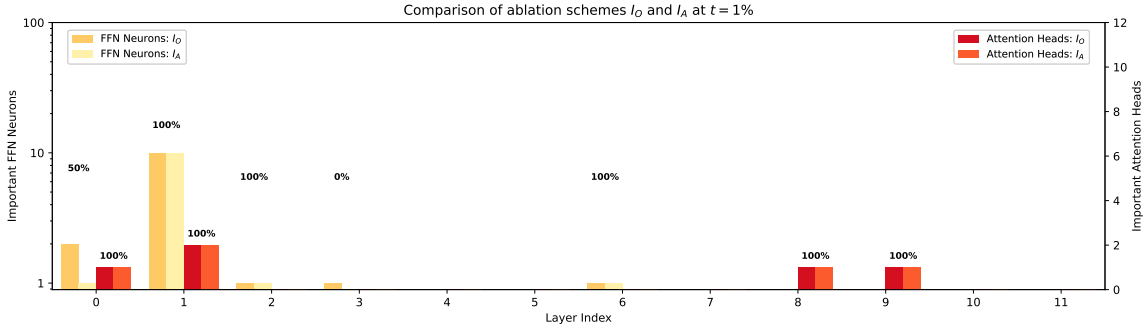
Fig. 2. Number of most important components per layer in the models: attention heads (red, right y-axis) and FFN neurons (yellow, left y-axis), following the ablation schemes obtained from the **Fusion** and the **Intersection** of the NIG scores for the different pools of datasets, at pruning percentage $t = 1\%$. Solid lines correspond to the TREC-DL pool (M), dashed lines to the OOD pool (O).

OOD and ID predictions rely on a disjoint set of neurons. Across models, in Figure 3 we compare the similarities between the sets of the most important neurons shared between the OOD datasets and all the datasets per component (respectively I_O and I_A). Although more visible with MonoBERT-base and MonoBERT-large than for MiniLM (see previous remark), we observe that OOD predictions involve additional important neurons, different from those for ID predictions (i.e., the intersection of the most important neurons across all datasets is a subset of the intersection of the most important neurons across OOD datasets). For MiniLM, this is observable at Layers 1 and 8 for FFN neurons and 4 for attention heads and, for MonoBERT-base, at Layers 0 and 3 and for MonoBERT-large, at Layers 2 and 3 for FFN neurons and 14, 15, 19, 21 and 22 for attention heads. Vast et al. [93] report the same observation solely for MonoBERT-large, but we specify here that this is mostly true for attention heads. Since the OOD-specific components of MonoBERT-large are mostly attention heads in later layers, the differences across domains may lie more in what information is moved (attention) than in how it is transformed (FFN).

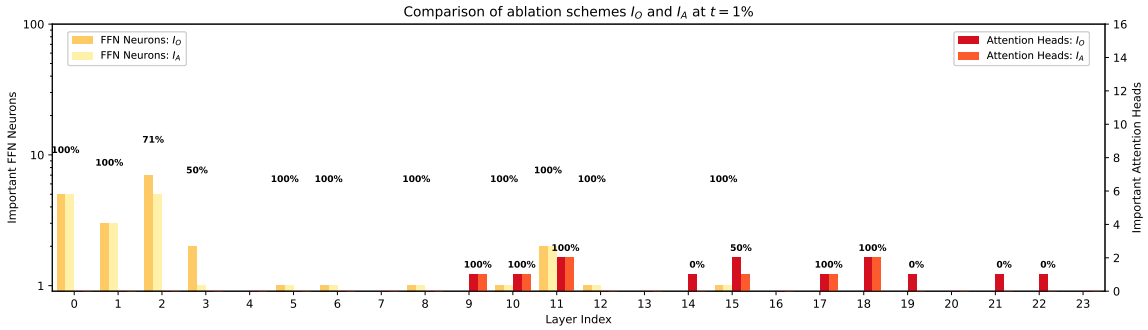
This observation, as well as the identification of other important attention heads, motivates the need to better understand what information is moved between the tokens' representations and how it is moved. In the next section, we focus on characterizing the information flow across the attention heads of each cross-encoder.



(a) Comparison of the most important neurons and attention heads between ablation schemes I_O and I_A for MiniLM.



(b) Comparison of the most important neurons and attention heads between ablation schemes I_O and I_A for MonoBERT-base.



(c) Comparison of the most important neurons and attention heads between ablation schemes I_O and I_A for MonoBERT-large.

Fig. 3. Number of most important FFN neurons (yellow, left y-axis) and attention heads (red, right y-axis), compared between ablation schemes I_O and I_A at pruning percentage $t = 1\%$. Percentage values above the bars correspond to the Jaccard similarity between the two sets of neurons for each model's component at each layer ℓ from the ablation schemes.

5 RQ2: Identifying the flow of information inside the model

5.1 Context and notations

In the previous section, we identified the most important components of the cross-encoders for the IR task. Yet, directly using NIG does not unveil their respective roles in modeling relevance. To tackle this issue, we first look to characterize the information flow between these components. To do so, we consider 5 distinct input parts for each query-passage

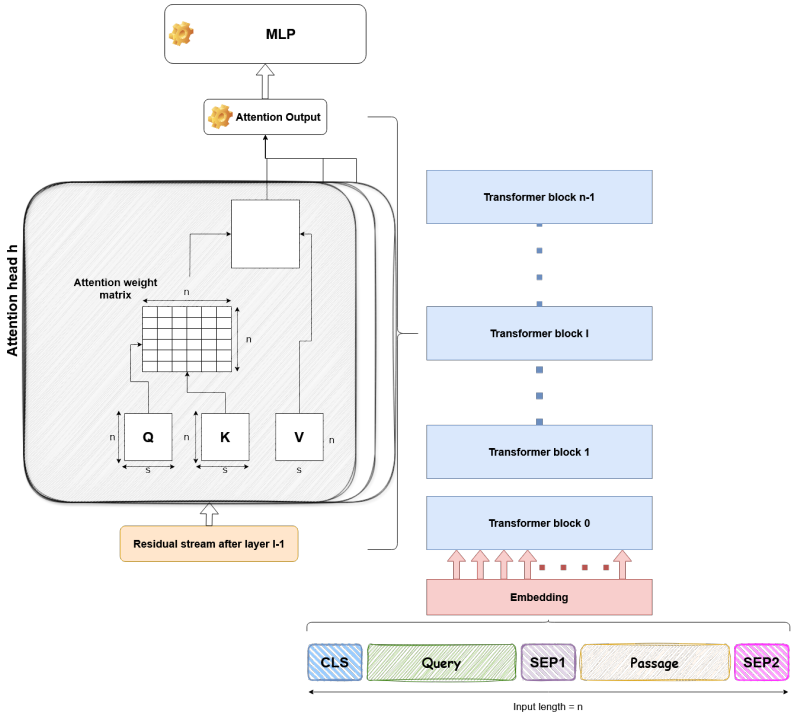


Fig. 4. Description of an attention weight matrix, at the layer ℓ for an attention head h, in the context of a transformer-based model. Each row inside these matrices is a distribution of probabilities over the entire input, relatively to a given token t (the target token), quantifying the amount of information each token (the sources) will send to t.

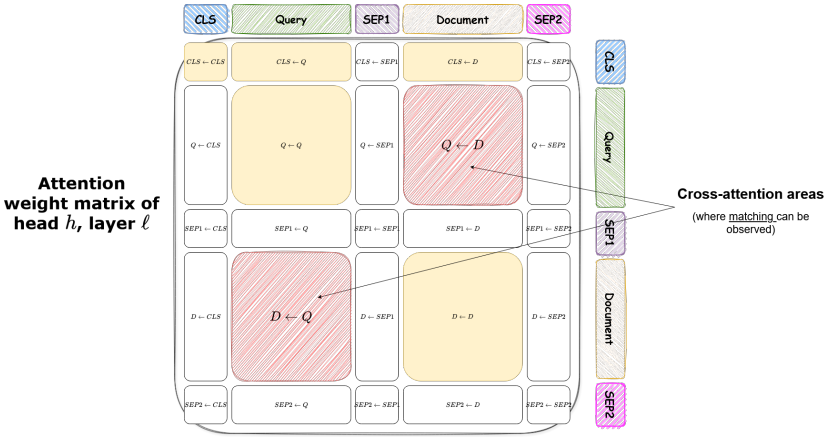


Fig. 5. Segmentation of the attention weight matrices into the 25 parts dictating the information flow between all the input parts $X \leftarrow Y, X, Y \in \{\texttt{[CLS]}, Q, \texttt{[SEP1]}, D, \texttt{[SEP2]}\}$. Highlighted in yellow are the areas we suspect to be the most important for cross-encoders, and in red, the areas where Query and Document exchange information.

pair: the [CLS] token, the query Q, the first separator token [SEP1], the document (or passage) D, and the last separator token [SEP2] (see Figure 4). In transformer-based models, information is moved between token representations thanks to the attention process [54]. As described in Figure 4, the attention weight matrix dictates the effective information transfers. In this section, we focus mostly on these matrices and study the flow of information between input parts. Note that the FFN can also be used as a proxy in this study (see Subsection 5.2), given that important transformations on a part of the input necessarily imply that important signals are available there. We decompose each attention score matrix into different parts, characterizing the information being sent from one input part to another. Figure 5 describes these different parts for a given attention head h at a layer ℓ of a cross-encoder model. Each one of these input parts can attend to itself or any of the others, amounting to 25 possible *directions* for the information flow. We name *direction* the attention of the input's part X to the input's part Y within the attention weight matrix or equivalently, the information transmitted by the input's part Y to the input's part X, also denoted as $X \leftarrow Y$ with $X, Y \in \{\texttt{[CLS]}, Q, \texttt{[SEP1]}, D, \texttt{[SEP2]}\}$. We further introduce the notation $X \nleftarrow Y$ to indicate that the transfer of information from the input part Y to the input part X is blocked.

Throughout this section, we use the following notations to refer to these directions: For a given attention head h at layer ℓ, we denote $p_{h,\ell}(X_i \leftarrow Y_j)$ the probability that X_i, the i^{th} token of X, *attends to* Y_j, the j^{th} token of Y.

5.2 Directly characterizing the flow of information in cross-encoders

In this Section, we analyze the results obtained by applying two different methods to the attention weight matrix framework described above. We begin this section by introducing these two approaches.

5.2.1 Extending the NIG approach to study the information flow. In Section 4, we identified the most important components using NIG. As we monitored the raw importance of each component, we obtained a single score per component (attention heads or FFN neurons) by aggregating along the entire input length (see Subsection 4.2). While this is useful for comparing overall attributions, it discards fine-grained signal about *where* in the input each component's importance is concentrated. We therefore aggregate the conductances (see Subsection 4.1) *per input part* (query, document, [CLS], [SEP]) rather than over the whole input, following the segmentation of Figure 5. By applying the same experimental protocol described in Section 4.3 at this finer granularity, we obtain different ablation schemes, defined by their aggregation operation (Intersection or Fusion), the set of datasets used (M, O or A) and a pruning percentage t.

5.2.2 Studying the information flow with the Information Bottleneck approach. In the X-AI literature, the Information Bottleneck [78] approach is helpful to analyze different components of a model simultaneously. This approach has already been applied to IR and re-ranking [45], but never to directly analyze the directions followed by the information inside a cross-encoder. From a high-level perspective, its objective is to restrict as much as possible the amount of transfers between an input and an output (the bottlenecks), while maintaining the quality of the output.

We provide more background on the theory behind the information bottleneck approach in the Appendix E, and further detail there our experimental setup. We learn the bottlenecks across the whole model at once under the constraint that predictions are preserved. To verify that this constraint holds, we measured the degradation caused when applying the learned masks on the cross-encoders. Table 2 reports the score of the models with their masks on the 4 TREC-DL datasets as well as the OOD datasets. From these results, no statistically significant degradation can be observed across all three models. This validates that the masks we learned do not harm the flow of the information of relevance inside the models.

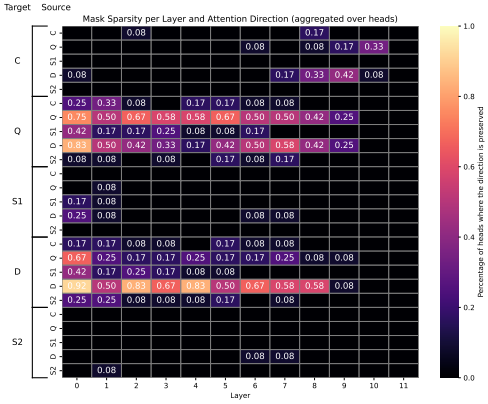
Table 2. Performance (in nDCG@10) of the models evaluated while applying the masks learned with the Information Bottleneck to their attention matrices (see Sec. 5.2.2) on the datasets from TREC-DL and BEIR. No difference with the original model performance is statistically significant at the 5% level under the two-tailed Student's t-test.

Model	T.-DL 19	20	21	22	NFCorpus	FiQA	T.-Covid	Robust-04	NQ	BioASQ
MiniLM	0.74	0.68	0.65	0.49	0.35	0.31	0.74	0.46	0.36	0.50
+ mask	0.71	0.65	0.63	0.47	0.35	0.30	0.75	0.44	0.33	0.50
MonoBERT-base	0.73	0.68	0.64	0.49	0.35	0.30	0.73	0.47	0.36	0.50
+ mask	0.72	0.67	0.63	0.47	0.35	0.29	0.73	0.45	0.34	0.49
MonoBERT-large	0.71	0.68	0.63	0.49	0.35	0.32	0.70	0.45	0.35	0.47
+ mask	0.70	0.66	0.61	0.47	0.35	0.32	0.70	0.44	0.34	0.45

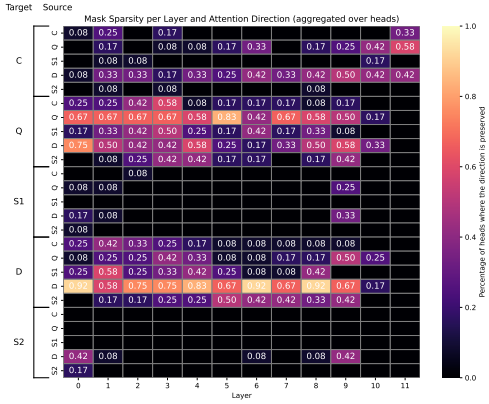
By combining the results from NIG and IB, we can formulate the following observations:

The information flow inside MiniLM is more consistent across domains than the information flow inside MonoBERT models. According to NIG, the most important transfers of information are consistent in all models for ID (see Figure 19 of the Appendix). Indeed, we observe a high similarity between the patterns reported for the most important neurons in general (F_M) and for the most important neurons shared across the four TREC-DL datasets (I_M). Meanwhile in OOD, Figure 7 tells a different story: for MiniLM, important transfers (i.e., with a NIG score in the top $t = 1\%$) in both F_O and I_O appear to be relatively consistent, especially between the query and the document and toward the [CLS] token. However, it is not the case for the two MonoBERT models. This is illustrated by the transfers occurring towards the first [SEP] token in F_O as, surprisingly, [SEP1] $\leftarrow Q$ appears to be important for MonoBERT-base and large, but absent from I_O. Note that the same conclusion can be drawn by comparing the ablation schemes I_A and F_A, detailed in Figure 20 of the Appendix. The information flow inside MiniLM appears to be more robust to domain shift than that of MonoBERT-base and large, both in terms of the attention heads that convey it (see Section 4) and in terms of directions.

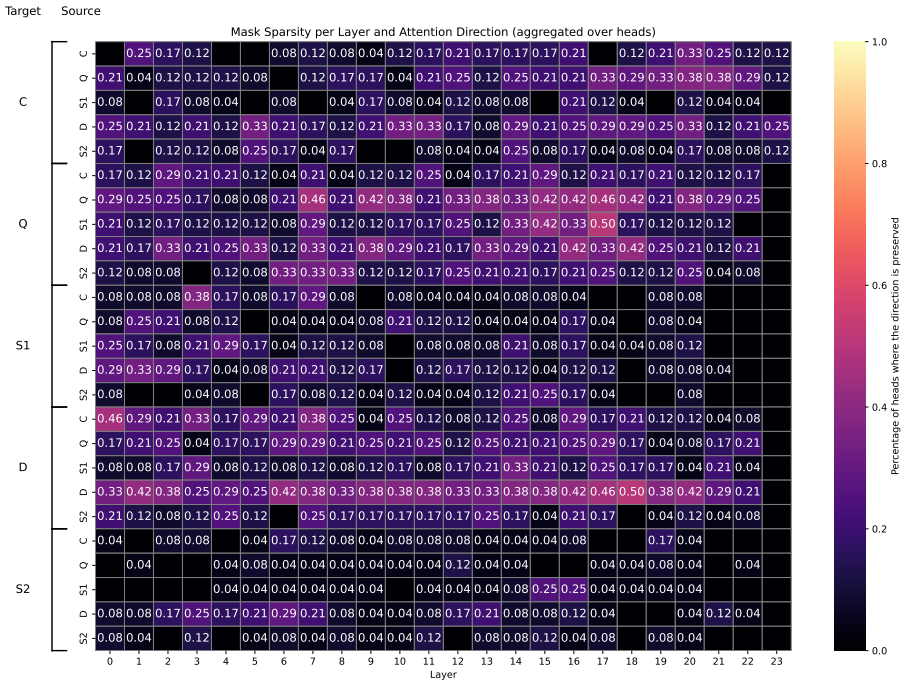
MiniLM integrates relevance information inside the [CLS] for a small set of layers, and marginally refines it once it has been moved. In contrast, MonoBERT-large needs a longer span and refines the information once it has been moved inside the [CLS] representation. Finally, MonoBERT-base exhibits a "hybrid" behavior, with similarities to both MonoBERT-large and MiniLM. In addition to the (surprising) presence of the [SEP] tokens in important directions inside MonoBERT, the other major difference in the information flow compared to MiniLM is in the later layers and how relevance is built inside the [CLS] token. Figures 7, 19 and 20 (NIG) and 6 (IB) show that MiniLM moves important information into the [CLS] in very few layers (1 or 2 according to NIG depending on the scheme, 3 according to IB), and that [CLS] $\leftarrow$ [CLS] plays little role. In contrast, on the same figures, MonoBERT-large needs multiple layers to achieve the same objective, and, in the very last layer, NIG highlights the direction [CLS] $\leftarrow$ [CLS] in some domains (see F_M, F_O and F_A for MonoBERT-large). For MonoBERT-base, NIG and IB suggest a longer integration inside the [CLS], as in MonoBERT-large. Yet the IB masks (Figure 6) also preserve query and document self-attention ($Q \leftarrow Q$ and $D \leftarrow D$) more than other directions, as in MiniLM. In addition, the direction [CLS] $\leftarrow$ [CLS], does not seem to have an important role, just as in MiniLM. A tentative explanation, developed in later sections, is that the [CLS] token integrates *already refined* relevance information from query/document in MiniLM and marginally in MonoBERT-base, whereas relevance information is still *raw signals* in MonoBERT-large. Combined with the previous



(a) Results of the Information Bottleneck on the attention heads of MiniLM.



(b) Results of the Information Bottleneck on the attention heads of MonoBERT-base.

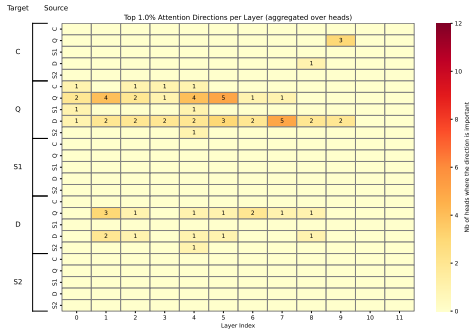


(c) Results of the Information Bottleneck on the attention heads of MonoBERT-large.

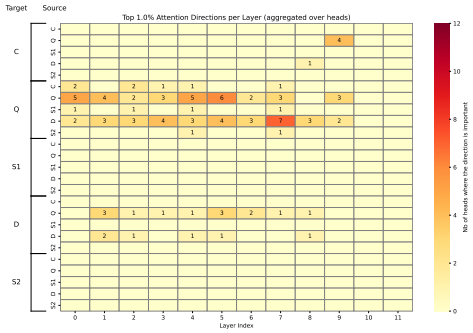
Fig. 6. Overview of the masking of each direction at each layer in the models, averaged over the attention heads.

observation on the greater robustness to domain shifts of MiniLM, this suggests that the three models do not capture relevance through the same mechanism. ■ **The [SEP] tokens do not convey relevance information.**

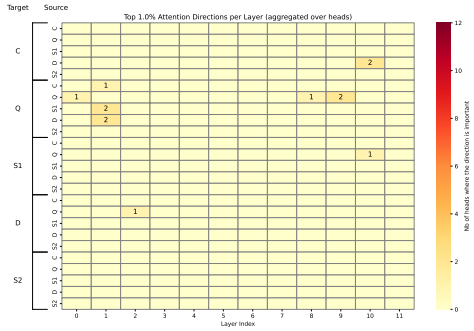
For MiniLM, we can easily conclude on the limited importance of the [SEP] tokens by observing Figures 7 and 6. For both MonoBERT models, however, the [SEP] tokens appear to be involved in a certain number of important



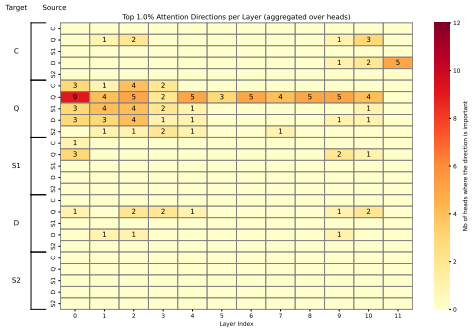
(a) Ablation scheme I_O for MiniLM.



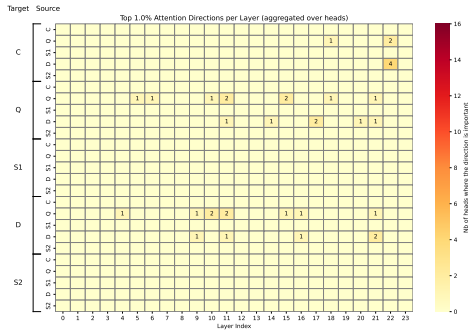
(b) Ablation scheme F_O for MiniLM.



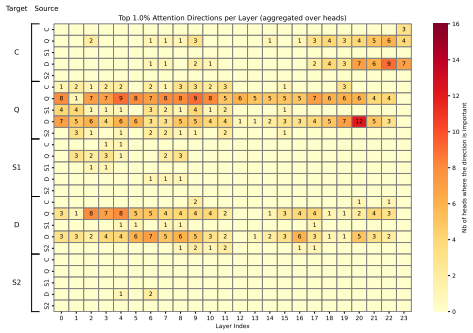
(c) Ablation scheme I_O for MonoBERT-base.



(d) Ablation scheme F_O for MonoBERT-base.



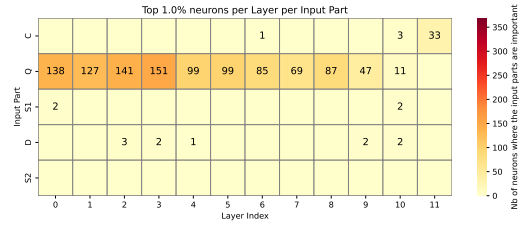
(e) Ablation scheme I_O for MonoBERT-large.



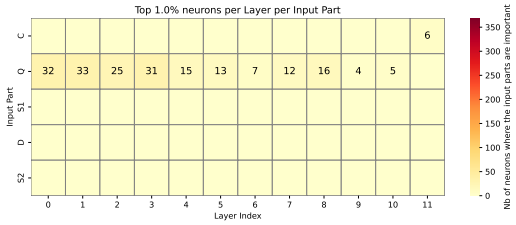
(f) Ablation scheme F_O for MonoBERT-large.

Fig. 7. Distribution of the most important attention heads, relative to each input part, per layer, following the ablation schemes F_O and I_O at pruning percentage $t = 1\%$.

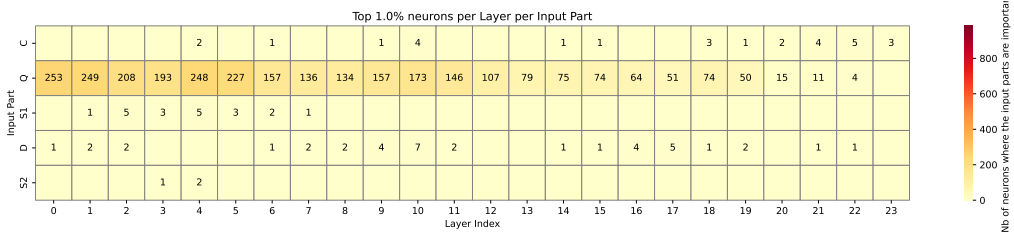
transfers, with no obvious explanation as to why. The major difference between MiniLM and both MonoBERT models is their pre-training, but it seems unlikely that the MLM/NSP strategy used to pre-train BERT-base and BERT-large has led to a unique ability around these tokens in MonoBERT. Previous conclusions point to a higher consistency of the important parameters and of information transfers across domains in MiniLM compared to MonoBERT models. In addition, parameterizing the Information Bottleneck masks proved difficult: it was hard to maintain a comparable level of performance while maximizing sparsity, particularly for MonoBERT-large (see subfigure 6c), and to a lesser extent



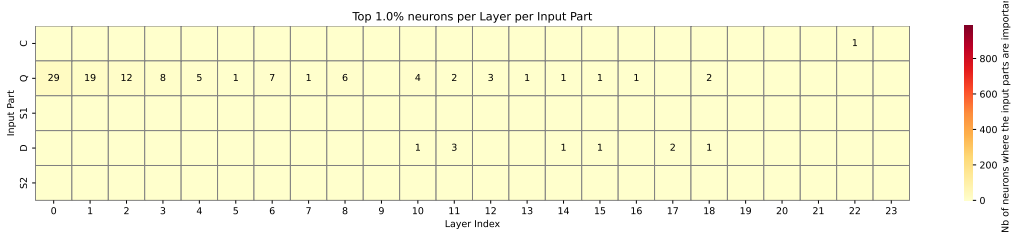
(a) Ablation scheme I_M for MonoBERT-base.



(b) Ablation scheme I_O for MonoBERT-base.



(c) Ablation scheme I_M for MonoBERT-large.



(d) Ablation scheme I_O for MonoBERT-large.

Fig. 8. Distribution of the most important FFN neurons, relatively to each input part, per layer, in MonoBERT-base and MonoBERT-large following the ablation schemes obtained from the **Intersection** of the NIG scores over the TREC-DL and OOD datasets, at pruning percentage $t = 1\%$.

for MonoBERT-base. Together, these observations suggest that MonoBERT may be over-parameterized. With more parameters than necessary, the model may learn several ways of doing the same thing, splitting an otherwise clear signal across concurrent branches. To clarify this point, we further study NIG scores per input part of MonoBERT's FFN neurons in Figure 8. If important information were indeed conveyed through the [SEP] tokens, we would also expect important transformations to happen at their position in subsequent FFN layers. In contrast, we can see that the transformations applied at the [SEP] tokens' position are not considered important compared to the transformations at the levels of the [CLS], query or document tokens. This suggests that the transfers towards the [SEP] are not important for what they convey. This rather points towards their role as *attention sinks*, i.e., to lower attention everywhere else and mitigate other transfers.

This also suggests that NIG does not always identify the neurons that are truly meaningful for modeling relevance. We discuss more in-depth this implication in the Subsection 8.3.

■ **The relevance prediction is decomposed into three consecutive stages, no matter the cross-encoders.** In general, both approaches suggest a similar trend – see Figures 7 for NIG and 6 for IB. Despite the mixed results of using

IB on MonoBERT-large, perhaps because of the learning process, we note that the pattern resembles that obtained with NIG for F_M and F_O. By analyzing the most important directions from NIG and the unmasked entries from IB, we can characterize the following steps of the relevance prediction process:

- **I.** In early to middle layers, the information flow is dominated by transfers between the queries and the documents. In particular, we identify two different types of transfers: self-contextualization, where the query (resp. document) sends information to itself, i.e., $Q \leftarrow Q$ (resp. $D \leftarrow D$), and cross-transfers, where the query (resp. document) sends information to the document (resp. query), i.e., $D \leftarrow Q$ (resp. $Q \leftarrow D$). We associate *self-transfers* with the contextualization mechanism, typical of transformer-based models, and *cross-transfers* with the detection of matching signals. Interestingly, we note that cross-transfers and self-transfers occur at the same layers. This suggests that cross-encoders can detect matching signals *before* the contextualization of the query and document tokens, and can leverage both *lexical* (almost at token-level) and *semantic* matches.
- **II.** In later layers, important information transfers go mostly towards the query (clearly visible for MiniLM, at layers 8 and 9 in Figure 6a, layers 9 and 10 for MonoBERT-base in Figure 6b, and, for all models, in Figure 20).
- **III.** Finally, in the last few layers, the information flows towards the `[CLS]` token, mostly from the query in MiniLM (see Figure 6a), and from both the query and the document in both MonoBERT-base and MonoBERT-large (see Figures 6b and 6c). This difference further suggests that MiniLM integrates into the `[CLS]` representation a signal that has already been refined in the query tokens, while the MonoBERT models further combine signals from the query and the document.

This multi-step process echoes the process hypothesized by Zhan et al. [111], but contradicts it on some key points. Their hypothesis originally describes the following steps:

(1) *At first, query and document contextualize themselves without paying attention to each other.* Our results contradict this as we observe that cross-transfers and self-transfers occur jointly through early to middle layers and link this behavior to the two types of matching signals that cross-encoders can detect (lexical and semantic).
(2) *Then interactions across query and document are important, and the model captures interaction signals to contextualize the tokens.* We confirm that behavior, but our experiments further suggest that the interactions in early-middle layers are not only used for contextualizing the tokens, but also to detect relevance signals (on this, see also the ablation results for $D + Q \nleftarrow D + Q$ in Figure 9, Subsection 5.3).
(3) *Finally, BERT relies heavily on the high-level representation to predict the relevance of the document to the query.* We confirm that behavior, while the importance of the direction $Q \leftarrow D$ in early layers (as well as the work of Lu et al. [56] on the importance of lexical signals for cross-encoders) suggests that interactions across uncontextualized tokens also matter. We further underline that cross-encoders use query tokens to accumulate relevance signal(s) in later layers before transmitting them to the `[CLS]`.

5.3 Analysis of the results from the ablation study

A more direct – but less granular – way to study the information flow inside transformer-based models is to apply a perturbation-based method (see Section 2). Zhan et al. [111] study the *directions* followed by the information between input parts by conducting an ablation study. On MonoBERT-large, they show, for example, the role of the document in the contextualization of the query, or that `[SEP]` tokens play a limited role in the prediction of relevance.

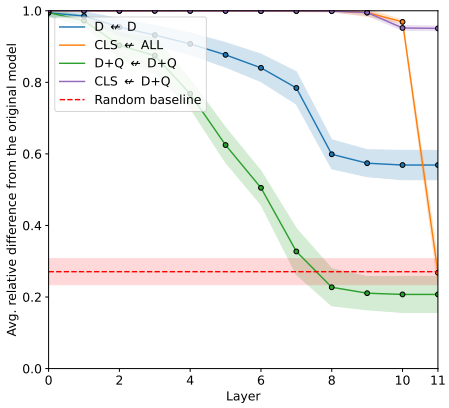
We extend their work by including two additional models and OOD datasets in our analysis. We follow their *Attention Mask* approach, as we perturb specific parts of the attention matrices by *setting the attention probability* to zero (which

implies the attention might not sum to 1 anymore). In particular, we focus our study on the areas that are highlighted in Figure 5. The perturbation is applied on all attention heads, from the first transformer layer to the ℓ-th layer. We report the nDCG@10 result, on our set of datasets (both ID and OOD) and using the pool of queries and passages obtained after the **BM25 sampling** (Figure 17), of the masked models. For clarity, we report the percentage of the original performance that is obtained after ablation. For reference, Table 5 reports the performance of each model and dataset. We also report the expected performance of a random classifier as well as the worst possible one. As the pool of passages for each query in this experiment has been sampled by BM25, we note that this setup closely resembles the re-ranking task for which cross-encoders are used.

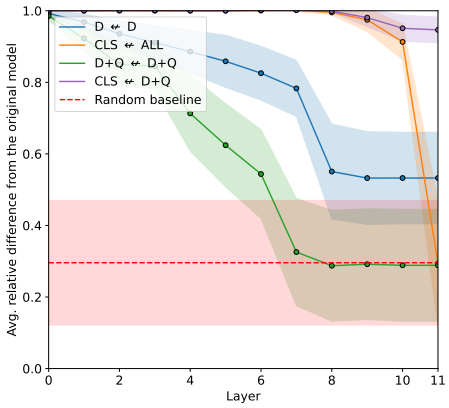
■ **Cross-encoders do not rely on semantic matches only. MonoBERT-base and MonoBERT-large are able to capture matching signals even without explicit query-document interactions.** The results of the ablations of the directions $D \nleftarrow D$, in blue, and $D + Q \nleftarrow D + Q$, in green, are reported in Figure 9 for both MonoBERT versions and MiniLM, on ID as well as OOD. Firstly, we remark that the impact of these two ablations, in both ID and OOD, is similar for the three models. Secondly, when preventing the document contextualization ($D \nleftarrow D$), we can observe a drop in performance around middle layers (7 or 9 for MiniLM and MonoBERT-large respectively, earlier for MonoBERT-base). This suggests that contextualization does not play an important role in the first layers, but gains importance as we progress through the layers. Implicitly, this means that the semantic matches should be captured at this level or after by cross-encoders. Also, we note that while masking $D \nleftarrow D$ throughout the layers of the models negatively impacts their performance, the final result is still far above the level of random rankings. Either cross-encoders do not rely only on semantic matches, or document information still reaches the model through the query tokens (recall the importance of $Q \leftarrow D$ in Subsection 5.2).

If we further prevent query contextualization and query and document from interacting with each other ($D + Q \nleftarrow D + Q$), we note that it always causes more damage to the performance than only blocking document contextualization. When this direction is masked across all the layers, the drop in performance reaches the level of random rankings in ID for all models. In OOD, both MonoBERT models partially maintain their performance, so they capture some matching signal even without explicit query-document interactions, presumably inside the `[CLS]` (the previous section showed that the `[SEP]` tokens are unlikely to convey it). Overall, this confirms the importance of query contextualization and query-document transfers, but also shows that MonoBERT models may have learned to use more intricate mechanisms than MiniLM to capture query-document matches.

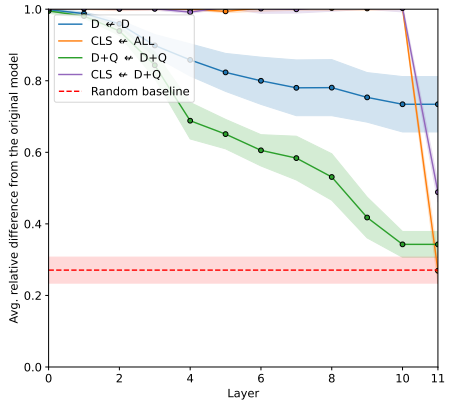
■ **`[SEP]` tokens carry little (MiniLM and MonoBERT-base) to no relevance (MonoBERT-large) information during the relevance prediction process inside cross-encoders.** Looking back at the ablation curve of D+$Q \nleftarrow D$+Q, for ID, in Figure 9 for both MonoBERT cross-encoders, the drop of performance to the random classification level further reinforces the hypothesis that the relevance information does not transit through the tokens of `[SEP]`. However, for MiniLM, we observe that the average percentage of preserved performance in ID and OOD drops even below the level of a random model, suggesting that the special tokens, contrary to MonoBERT, contain a signal that penalizes documents in the absence of relevance signals. To better isolate and understand the effect of the `[SEP]` and `[CLS]` tokens, we prevent the query and document from *directly* sending information to `[CLS]` (`[CLS]` $\nleftarrow D + Q$, in purple). We observe that the results in that case differ significantly between the models. Indeed, we observe that while blocking these transfers of information across the whole model eventually causes the performance of MonoBERT-large to drop even below the level of random rankings (so `[SEP]` and `[CLS]` carry negative signals), this is not the case for MiniLM and MonoBERT-base.



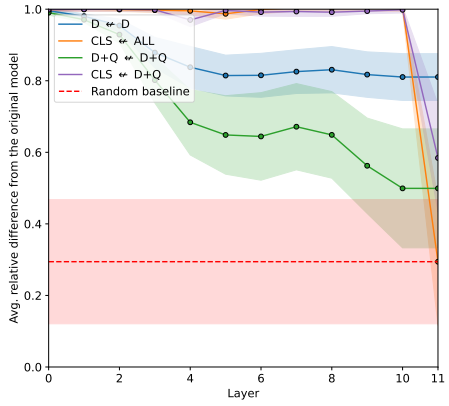
(a) Ablations on the TREC-DL datasets for MiniLM.



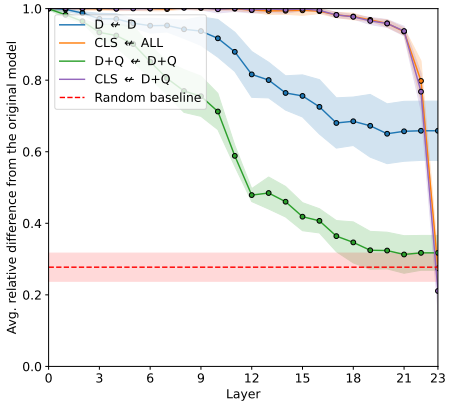
(b) Ablations on the OOD datasets for MiniLM.



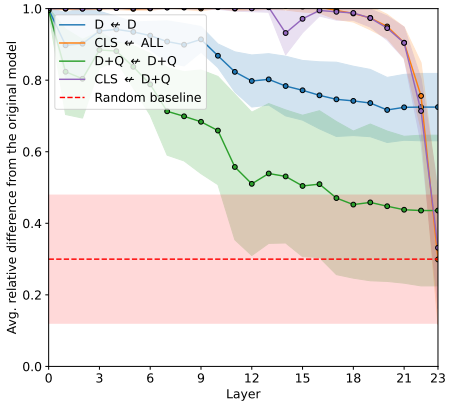
(c) Ablations on the TREC-DL datasets for MonoBERT-base.



(d) Ablations on the OOD datasets for MonoBERT-base.

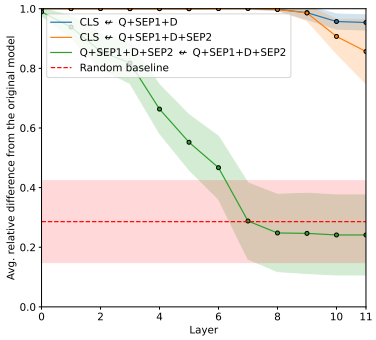


(e) Ablations on the TREC-DL datasets for MonoBERT-large.

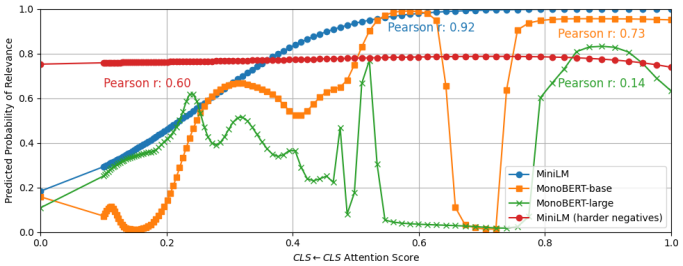


(f) Ablations on the OOD datasets for MonoBERT-large.

Fig. 9. Average percentage of the original performance preserved after the ablations of directions inside the attention heads of the models up to the layer ℓ. The areas around the dotted lines correspond to the variance. Averages and variances are computed over the TREC-DL datasets or the OOD datasets. The red dashed line and red area around it for the "Random baseline" come from Table 5.



(a) Ablations on all datasets for MiniLM.



(b) Evolution of the probability of relevance when masking all of the attention score matrices except the probability [CLS] ← [CLS], and controlling its value.

Fig. 10. Additional experiments to isolate the role of [CLS] ← [CLS] transfers on the prediction of relevance, on MiniLM in particular.

For MiniLM, this ablation highlights that the model might have learned to use [SEP] and/or [CLS] to store relevance signals extracted from the query and document interactions. Although it echoes the presence of the first token [SEP] as an important source of information in the ablation schemes I_M and I_O for MiniLM in Figure 7, this also calls for more research to understand their roles precisely.

Meanwhile, for MonoBERT-base, the analysis is more complex, as the performance of the model still drops significantly when the signal is pruned across all the layers. Yet, unlike its larger counterpart, the remaining transfers from the [SEP] tokens to the [CLS] are able to maintain the performance above the level of a random predictor. In the end, this suggests that MonoBERT-base may have learned to use the [SEP] tokens to store some relevance information, but [CLS] ← [SEP1] + [SEP2] transfers are usually overridden by what is flowing from the query and the documents.

■ **MiniLM may base its prediction on signals of non-relevance, instead of relevance. When fine-tuned with harder negatives, this behavior may even be exacerbated.** Following up on the role of the [SEP] tokens inside MiniLM, we further prevent the [SEP] tokens from sending information to the [CLS] token in Figure 10a ([CLS] ↚ Q + [SEP1] + D, in blue, and [CLS] ↚ Q + [SEP1] + D + [SEP2], in orange). Surprisingly, even if the [CLS] can only receive information from itself, across the entire model's layers, the drop in performance remains limited for MiniLM (around 80% of the original performance is retained across all the datasets). This points to an information leak in our ablation setup (discussed in Section 8.3). It also shows that MiniLM's predictions can be influenced simply by how much attention the [CLS] pays to itself. This observation leads to a series of follow-up questions: is it possible to control the predicted probability of relevance of MiniLM by controlling the probability of attention from the [CLS] to itself? To what extent? What about MonoBERT?

To answer these, we set up a small experiment in which we mask all the attention probability matrices of the cross-encoders, except the probability corresponding to the direction [CLS] ← [CLS]. To understand its impact, we observe the evolution of the predicted probability of relevance when progressively increasing this score, from 0.0 to 1.0, on a query and an unrelated document from MS MARCO. Figure 10b shows that even though all the models originally predict a score close to 0, masking the attention score matrices can heavily impact the prediction. While there does not seem to be a correlation between the attention score of [CLS] ← [CLS] and the probability of relevance predicted by both MonoBERT models (Pearson R correlation of 0.14 for the large version and 0.73 for the base, but non-monotonic curve), for MiniLM the correlation is much stronger (0.92). Note that in a real-world setup, an attention score of 0 for

the direction [CLS] ← [CLS] means the model is capturing information from the rest of the input inside [CLS], while a score of 1.0 means the model is focusing on what is already contained inside the token's representation. Here, we observe that the stronger MiniLM focuses the attention from the [CLS] onto itself (and ignores the rest of the tokens), the more the probability of relevance increases. This experiment suggests that MiniLM does not scan the query and document tokens for "relevance signals" *but rather for "non-relevance signals"*. In their absence, its confidence in the relevance of the document to the query increases.

In Figure 10b, we also report the same behavior but on a MiniLM trained using harder negatives, i.e., negatives sampled by ColBERT-v2 [76]. Interestingly, this model exhibits a third type of behavior: when only [CLS] ← [CLS] is allowed, no matter the weight of the attention, the prediction by default seems to be "relevant". This may indicate that when the model is fine-tuned with harder negatives, it relies even more on the detection of non-relevance signals for its prediction.

■ **[CLS] only absorbs relevance signals in the very last layers of a cross-encoder.** Returning to Figure 9, the last ablation we tested is to prevent each token from sending information to [CLS] ([CLS] ↞ *ALL*, in orange). We can see that when applied from the first layer up to the last layer, we logically end up with a random ranking performance. If we observe where the performance starts to drop, we note that the fall does not occur before the last few layers, especially in the case of MiniLM and MonoBERT-base. This suggests that the [CLS] token only begins to integrate relevance signals from the query and the document tokens at the very end of the process. What is transmitted to the [CLS] and the transformations they undergo through the last FFN layers are unclear at this stage and require more investigation.

With this, we now have a more complete understanding of the information flow across input parts inside different cross-encoders. Furthermore, the multi-stage process of relevance prediction in cross-encoders that we described suggests that they may be able to leverage both *lexical* and *semantic* interaction signals. Indeed, we observed that important interactions between a query and a document are captured by the models both before and after the contextualization of query and document tokens. After answering **RQ1** and **RQ2**, we now aim at characterizing the nature and origin of the signals that contribute to the prediction of relevance inside cross-encoders.

6 RQ3: Characterizing the nature of the signals contributing to the prediction of relevance

Among the attention heads we have found to play a crucial role in the process of modeling and predicting relevance in cross-encoder models, we can *a priori* guess what the role of most of them is. Indeed, as highlighted by prior studies [56, 111] as well as our findings in the previous Section 5, attention heads in cross-encoder models play a key role in 1) contextualizing the query and/or document tokens (in particular for early to middle layers), 2) transferring information between the query tokens and document tokens (in particular for attention heads in early to middle/late layers) and 3) transferring information from the entire input to the [CLS] (late layers). Roles 1) and 3) are as important as 2), but they are better understood. Role 2), usually seen as the *detection* of query-document matches, remains opaque, both in the information it surfaces and in its mechanism.

In their work, Lu et al. [56] use causal tracing [96] to investigate key components inside MiniLM and, in particular, the role of some attention heads in detecting query-document matches. Their approach reveals the causal chain leading to the relevance prediction, but it is hard to scale to multiple datasets, which limits conclusions about generalization (see **RQ5** and **RQ6**). We argue that more direct approaches can lead to comparable results while scaling more easily to multiple datasets. In particular, we believe that studying the attention scores between query and document tokens (red areas in Figure 5), while considering the relevance of the document, can help us identify heads responsible for detecting matches (i.e., when most of the attention weight is focused on a single token) and to characterize their nature (lexical or

semantic). To ensure each level of relevance is represented throughout our datasets, we employ the *Diverse sampling* strategy (see Figure 17).

6.1 Context and notations

In this section, we aim to improve the understanding of the attention heads focusing on transferring information from the document (resp. query) to the query (resp. document), i.e., focusing on cross-attention (as opposed to the self-attention of an input part to itself, like in query/document contextualization).

To do so, we analyze the attention scores from query tokens to document tokens, named "Q/D-attention" (and vice-versa, i.e., "D/Q-attention") in the attention score matrices (see the two red areas on Figure 5) and contrast the amount of cross-attention when a document is relevant versus when it is not relevant for the same query. We intend to differentiate the attention heads that have learned to detect *relevant/important* matching signals from the attention heads that may be attending heavily to unimportant tokens, such as stop words in the context of the no-op operation [18].

Formally, for each attention head h at layer ℓ, relevance level r and a query token Q_i, we compute the maximum attention probability over document tokens D_j as $\max_j p_{h,\ell}(Q_i \leftarrow D_j)$. We then compute the mean for a given query and denote this value $M^{h,\ell}_{Q\leftarrow D/r}$. We define $M^{h,\ell}_{D\leftarrow Q/r}$ for the attention of document tokens on query ones similarly. To get as much contrast as possible, we consider the 3 following levels of relevance: "relevant" ($r = 1$), "hard negative" ($r = 0$), and "easy negative" ($r = -1$), and compare the differences, for each attention head, between $r = 1$ and $r = -1$ as well as between $r = 1$ and $r = 0$, namely: $M^{h,\ell}_{Q\leftarrow D/1} - M^{h,\ell}_{Q\leftarrow D/-1}$ and $M^{h,\ell}_{Q\leftarrow D/1} - M^{h,\ell}_{Q\leftarrow D/0}$.

Other statistics than the maximum of attention can be computed from the same attention distributions. In Section 6.2, we also use the Shannon entropy [80] of the attention of a query token Q_i over the document tokens,

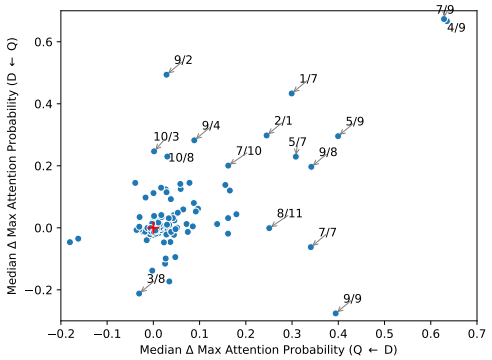
$$E^{h,\ell}(Q_i \leftarrow D) = -\sum_j p_{h,\ell}(Q_i \leftarrow D_j) \log p_{h,\ell}(Q_i \leftarrow D_j),$$

summed over the query tokens and averaged over the queries of a dataset for a given relevance level r, which we denote $E^{h,\ell}_{Q\leftarrow D/r}$ ($E^{h,\ell}_{D\leftarrow Q/r}$ is defined similarly). Note that we thus average entropy values, and not attention distributions. A low entropy indicates that the head concentrates its attention on a few document tokens, whereas a high entropy indicates that its attention is spread over the document. As for the maximum of attention, we compare relevance levels through the differences $E^{h,\ell}_{Q\leftarrow D/1} - E^{h,\ell}_{Q\leftarrow D/-1}$ and $E^{h,\ell}_{Q\leftarrow D/1} - E^{h,\ell}_{Q\leftarrow D/0}$.

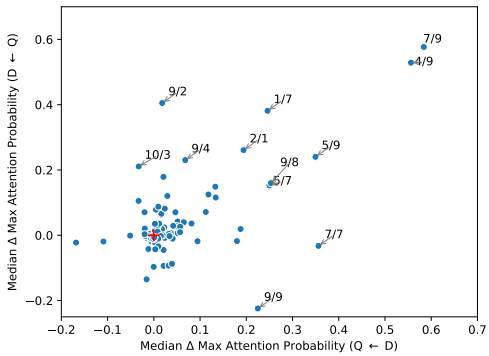
6.2 Analyzing the attention process to find the heads responsible for detecting matches

Using the approach described above and by contrasting the maximum of Q/D-attention and D/Q-attention between two relevance levels across our different pools of datasets for our cross-encoder models, we can see if some attention heads are specialized in detecting *relevant* matching signals. Figure 11 describes the results for the contrast between $r = 1$ and $r = -1$, and Figure 12 for $r = 1$ and $r = 0$.

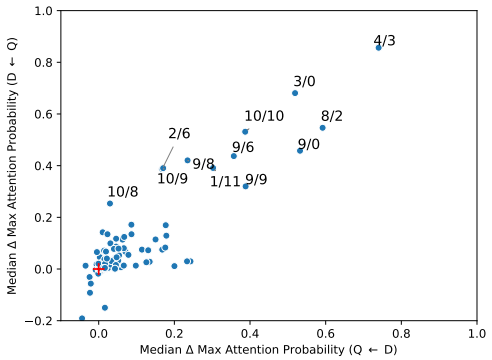
■ **Cross-encoders have attention heads dedicated to detecting matching signals between a query and a relevant passage. For MiniLM and MonoBERT-base, these heads are relatively robust to high lexical proximity but for MonoBERT-large, it can lead to major changes in their matching patterns.** Figure 11 shows that some heads have much higher cross-attention probability levels when the document is relevant compared to when it is not. Interestingly, we note that this is robust to domain shift for every model (**RQ6**, Figures 11b, 11d and 11f). The conclusion we can draw is that some attention heads in cross-encoders are indeed specialized in detecting *relevant* matching signals between the



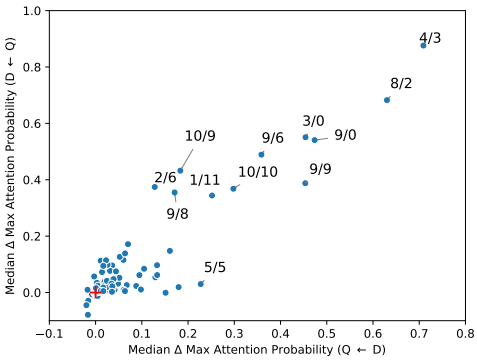
(a) Results over the 4 TREC-DL datasets for MiniLM.



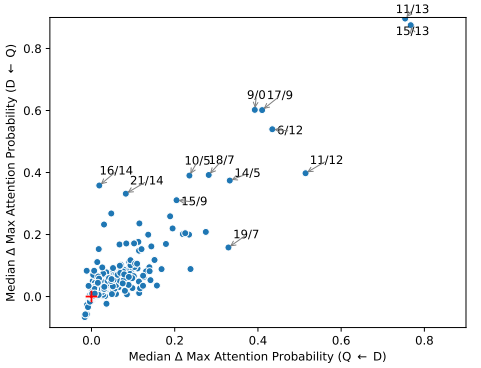
(b) Results over the OOD datasets for MiniLM.



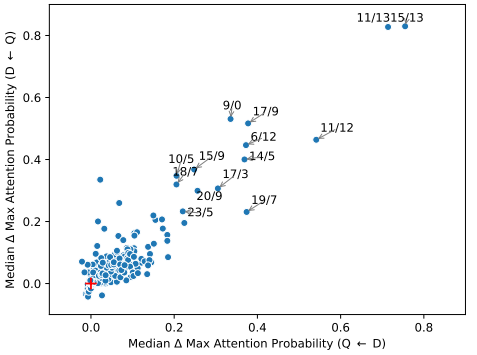
(c) Results over the 4 TREC-DL datasets for MonoBERT-base.



(d) Results over the OOD datasets for MonoBERT-base.

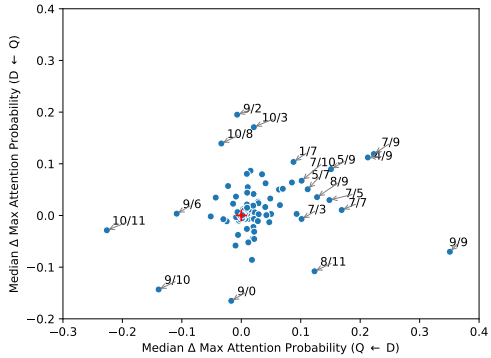


(e) Results over the 4 TREC-DL datasets for MonoBERT-large.

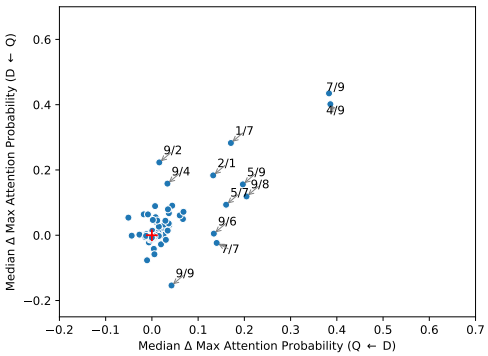


(f) Results over the OOD datasets for MonoBERT-large.

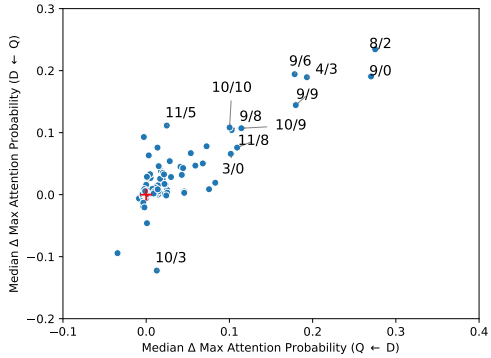
Fig. 11. Relevant vs. easy negatives – Median of the differences in the attention weights $M^{h,\ell}_{Q\leftarrow D/1} - M^{h,\ell}_{Q\leftarrow D/-1}$ and $M^{h,\ell}_{D\leftarrow Q/1} - M^{h,\ell}_{D\leftarrow Q/-1}$ for each attention head of all models. An attention head h at a layer ℓ is denoted as ℓ/h.



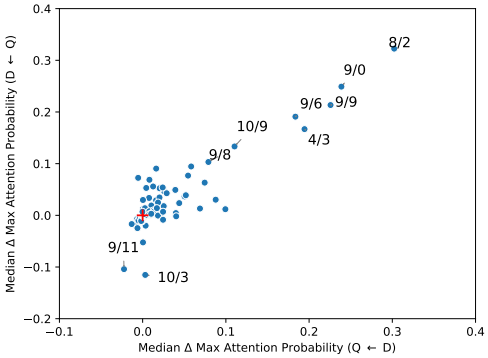
(a) Results over the 4 TREC-DL datasets for MiniLM.



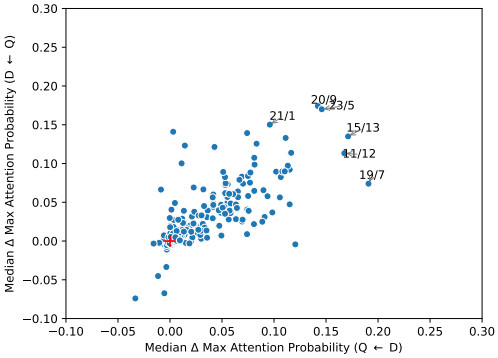
(b) Results over the OOD datasets for MiniLM.



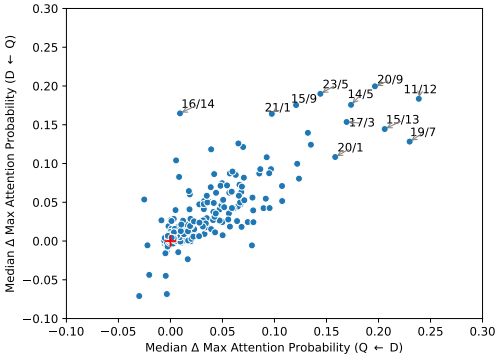
(c) Results over the 4 TREC-DL datasets for MonoBERT-base.



(d) Results over the OOD datasets for MonoBERT-base.



(e) Results over the 4 TREC-DL datasets for MonoBERT-large.



(f) Results over the OOD datasets for MonoBERT-large.

Fig. 12. Relevant vs. hard negatives – Median of the differences in the attention weights $M^{h,\ell}_{Q\leftarrow D/1} - M^{h,\ell}_{Q\leftarrow D/0}$ and $M^{h,\ell}_{D\leftarrow Q/1} - M^{h,\ell}_{D\leftarrow Q/0}$ for each attention head of all models. An attention head h at a layer ℓ is denoted as ℓ/h.

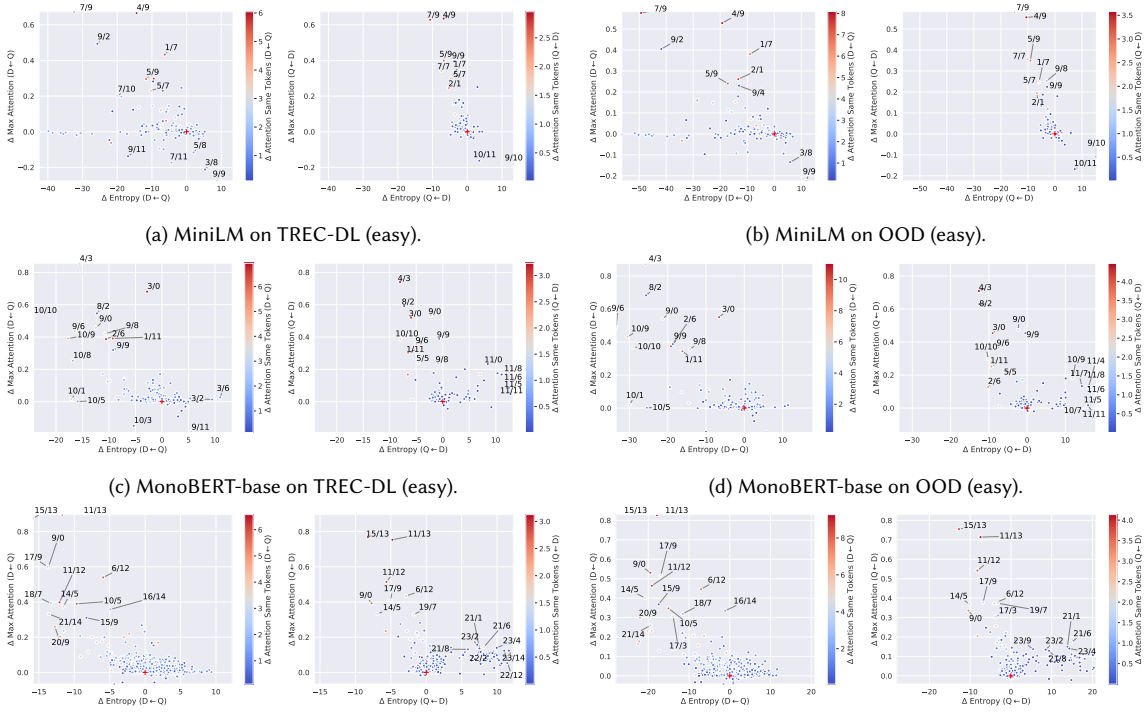
query and the document. We refer to such attention heads as "matching detection" heads. Interestingly, we note that cross-attention is bi-directional, i.e., there is almost a linear relationship between the differences $M^{h,\ell}_{Q\leftarrow D/1} - M^{h,\ell}_{Q\leftarrow D/-1}$ and $M^{h,\ell}_{D\leftarrow Q/1} - M^{h,\ell}_{D\leftarrow Q/-1}$, meaning those heads detect matching tokens in the query (for the document) and in the document (for the query) at the same time. If each model has such heads, we note that they are located in almost all of the MiniLM and MonoBERT-base layers (from layer 1 to 10), compared to mostly middle to later layers (from layer 9 to 23) for MonoBERT-large.

Observing Figure 12, contrasting relevant passages with hard negatives, we can observe similar patterns with lower differences compared to easy negatives (Figure 11). For MiniLM and MonoBERT-base, it seems that the attention heads (7/9 and 4/9 for MiniLM, and 4/3 and 8/2 for MonoBERT-base) that exhibit the most distinctive patterns in Figure 11 are relatively robust to this change, especially in OOD. Meanwhile, for MonoBERT-large, the equivalent two attention heads (11/13 and 15/13) no longer stand out from the others. This comparison between the behaviors of the matching detection heads with different types of negative passages suggests that these heads are tricked by the lexical proximity between a query and a hard negative passage (sampled using BM25). We note that smaller BERT-based models, i.e., MiniLM and MonoBERT-base, seem more resilient to this issue than MonoBERT-large.

■ **Matching detection heads can capture both exact matches and semantic matches.** Beyond these high-level observations, it is difficult to determine the nature of the detected matches. To overcome this limitation, we take into account the source and target tokens of each match to differentiate exact matches (source and target tokens are identical) from semantic matches (source and target tokens are different). As we consider raw tokens and do not apply a pre-processing before studying them, we refer to a match between an identical source and target token as an *exact*, and not a *lexical* match, i.e., a match that would be detected by lexical approaches such as BM25 [1]. Note, however, that we consider exact matches as a proxy for studying lexical matches, and that our approach tends to underestimate these so-called *lexical* matches. Figure 14 provides details on the nature of query-document matches when comparing relevant passages ($r = 1$) to easy negatives[9] ($r = -1$). This figure combines three differences per attention head: in entropy (x-axis), in maximum attention (y-axis, as in Figure 11), and in the probability of attending to a token identical to the source token (color). This last probability is conditioned on the event that a query token attends to the document (and symmetrically for $D \leftarrow Q$), so that it measures the fraction of the *cross-attention* going to identical tokens. A head with a negative entropy difference and a positive maximum-attention difference focuses on a few specific document tokens when the passage is relevant, i.e., it detects matches; the color then tells whether these matches are exact (identical tokens) or semantic (different tokens).

If we first look at the results produced by contrasting easy negatives ($r = -1$) with relevant passages ($r = 1$) in Figure 14, we see relatively similar patterns between ID and OOD situations (**RQ6**). For every model, the attention heads that are standing out in Figure 11, i.e., 4/9 and 7/9 for MiniLM, 4/3 and 3/0 for MonoBERT-base and 15/13 and 11/13 for MonoBERT-large, are all focusing more on single tokens containing relevance signals (lower entropy when the passage is relevant than when it is non-relevant) and with the same identity (higher attention to identical tokens when the passage is relevant than when it is non-relevant). This observation suggests that these heads may be involved, more precisely, in the detection of *exact matches*. At the same time, we also observe other attention heads, with high differences in the attention weights and low entropy, that do not focus on identical tokens like the heads 9/2 for MiniLM, 8/2 or 9/9 for MonoBERT-base and 15/9 or 17/9 for MonoBERT-large. These heads may be involved in the detection of *semantic matches*. In general, we observe a common trend across models and domains, where *semantic*

[9]Figure 23 in the Appendix provides the same comparison between relevant passages and hard negatives ($r = 0$).



(a) MiniLM on TREC-DL (easy).

(b) MiniLM on OOD (easy).

(c) MonoBERT-base on TREC-DL (easy).

(d) MonoBERT-base on OOD (easy).

(e) MonoBERT-large on TREC-DL (easy).

(f) MonoBERT-large on OOD (easy).

Fig. 14. Comparison of the matching patterns of the attention heads over the TREC-DL and OOD datasets, contrasting $r = 1$ with $r = -1$ (easy negatives). Each point is an attention head h at a layer ℓ, denoted as ℓ/h; the left panel corresponds to the $D \leftarrow Q$ direction and the right panel to $Q \leftarrow D$. x-axis: median of the differences in entropy ($E^{h,\ell}_{Q \leftarrow D/1} - E^{h,\ell}_{Q \leftarrow D/-1}$); y-axis: median of the differences in maximum attention ($M^{h,\ell}_{Q \leftarrow D/1} - M^{h,\ell}_{Q \leftarrow D/-1}$); color: median of the differences in the probability of attending to a token identical to the source token. The red cross marks the origin.

matching heads are located in later layers than *exact* matching heads – which corresponds to what is known about transformer-based models. We observe the same patterns when contrasting hard negatives ($r = 0$) to relevant passages ($r = 1$), though less pronounced (see Figure 23 in Appendix).

■ **The aggregation of relevance signals is conveyed from the document tokens through the query tokens before eventually reaching the [CLS] in MonoBERT-base and MonoBERT-large.** In Figures 14 and 23, we observe a very distinct pattern for the direction $Q \leftarrow D$ in both MonoBERT models, where, when the passage is relevant, the attention heads in the very last layers have higher entropy levels and a positive difference in the attention weights. This pattern echoes our description of the last step of the relevance prediction process in the previous Section 5, with query tokens playing a pivotal role in the accumulation of relevance signal(s) from the entire input before transmitting them to the [CLS]. We therefore interpret this difference with MiniLM as follows: both MonoBERT models aggregate *raw relevance signals* from many document tokens into the query tokens, which explains both the higher attention and the higher entropy when the document is relevant.

■ **MiniLM has Q/D-attention heads that are more active when the passage is not relevant to the query. These attention heads are not detecting matching signals but non-relevance signals.** Another difference, visible in

Figures 11, 12, 14 and 23 (Appendix), is that some MiniLM heads cross-attend more when the document is *not* relevant: matching detection heads (3/8 and 9/9 in the first two figures) as well as other heads (9/10 and 10/11 for the direction $Q \leftarrow D$ in the last two). Note that, while less pronounced, MonoBERT-base also partially exhibits this behavior only for the direction $D \leftarrow Q$ (see the heads 10/3 and 9/11), contrary to MonoBERT-large, which does not seem to possess the same mechanism.

From Figures 14 and 23 in the Appendix, focusing on the direction $D \leftarrow Q$, we remark that the attention heads 3/8 and 9/9 with higher cross-attention weights when the passage is non-relevant (in Figures 11 and 12) also have a lower entropy and do not attend to identical tokens. Taken together, and echoing the experiment described in Figure 10b in the previous section, the evidence suggests that these attention heads detect signals of non-relevance between the query and the document. Although we lack a clear understanding of this behavior, it appears both in-domain and out-of-domain, with the same heads consistently involved each time for both $D \leftarrow Q$ and $Q \leftarrow D$ for MiniLM, and solely for $D \leftarrow Q$ with MonoBERT-base.

At the same time, from the Figures 14 and 23, but focusing on the direction $Q \leftarrow D$, we observe two attention heads in MiniLM, 9/10 and 10/11, located in the final layers of the model, exhibiting a comparable behavior to the attention heads in the final layers of MonoBERT for the same direction (see the previous point). However, instead of having higher attention scores for relevant passages, they have higher attention scores when passages are not relevant, but lower entropies. While we lack a clearer understanding of their roles, we consider this observation further evidence that MiniLM uses signals of non-relevance, and not relevance, to make its predictions (see the discussion in the subsection 5.3).

Looking back at Section 4 where we measured the importance of different components inside the cross-encoders using NIG, the question we can ask ourselves is: are the most important attention heads, according to NIG, involved in query-document matches detection? In this section, we identified a small set of attention heads, scattered across the layers of every cross-encoder, whose behavior seems related to the detection of matches. Theoretically, these attention heads should be considered highly important for the IR task, even though their relative importance compared to the attention heads involved in the contextualization of the query and document tokens, or in the transfer of the relevant signal to the `[CLS]` token, is unclear. In Table 3, we observe that most attention heads that exhibit an attention pattern representative of the detection of matching signals in Figures 11a and 11e, are also considered to be important in terms of their NIG score. We further remark the presence of additional attention heads, for which we currently lack a deeper understanding. This underlines the importance of mechanistic interpretability studies or similar works to unveil the role of key components. On this matter, Table 3 further reports the attention heads in MiniLM classified as "Matching heads" in Lu et al. [56]. We observe a high overlap between the most important attention heads we identified with NIG, characterized as detecting query-document matching, and these "Matching heads". This provides further support for our findings and substantiates our claim that lightweight alternatives to causal patching exist.

7 RQ4: Deep dive into the subspace(s) of relevance and the construction of the relevance signal

So far, we have seen that both attention heads and FFN layers play important roles inside cross-encoders (see Section 4). These roles revolve mostly around the query and document tokens, their interactions, and ultimately the `[CLS]` token (see Section 5). In particular, some attention heads contribute to detecting matching signals between query and document tokens while others contextualize tokens or transfer information to `[CLS]` (see Section 6). FFN layers and their "important" neurons process and refine information moved by the attention heads, integrating it into tokens' representation [36, 37]. In cross-encoders, some of these neurons likely operate in a *relevance subspace* where relevance

Table 3. Table detailing if the top 10 attention heads (layer ℓ / head h) in terms of NIG score are involved in detecting matching signals, i.e., if they are highlighted in one of the figures of this section, and if the Information Bottleneck (IB) agrees, i.e., if either $Q \leftarrow D$ or $D \leftarrow Q$ is preserved by the mask for the attention head. The ordering of the attention heads is from the highest NIG score to the lowest and follows the Fusion operation of aggregation.

MiniLM				**MonoBERT-base**			**MonoBERT-large**		
Attention heads	In plots?	In IB?	"Matching heads" in [56]?	Attention heads	In plots?	In IB?	Attention heads	In plots?	In IB?
4/9	✓	✓	✓	3/0	✓	✓	1/9	×	✓
1/7	✓	✓	✓	1/7	×	✓	15/13	✓	✓
2/1	✓	✓	✓	2/6	✓	✓	9/0	✓	✓
5/7	✓	✓	✓	0/9	×	✓	11/13	✓	✓
7/9	✓	✓	✓	0/10	×	✓	11/12	✓	✓
6/5	×	✓	✓	1/6	×	×	16/1	×	✓
5/9	✓	✓	✓	9/5	×	×	14/5	✓	✓
3/1	×	✓	✓	10/3	✓	✓	22/4	×	✓
2/10	×	✓	×	9/11	✓	✓	10/10	×	✓
7/1	×	×	×	0/4	×	✓	2/0	×	×

signals, such as matching signals (Subsection 6.2), are stored to support relevance prediction. This *relevance subspace* is the focus of this section, and as in the previous section, we ensure that each level of relevance is represented by using the *Diverse sampling* strategy (see Figure 17).

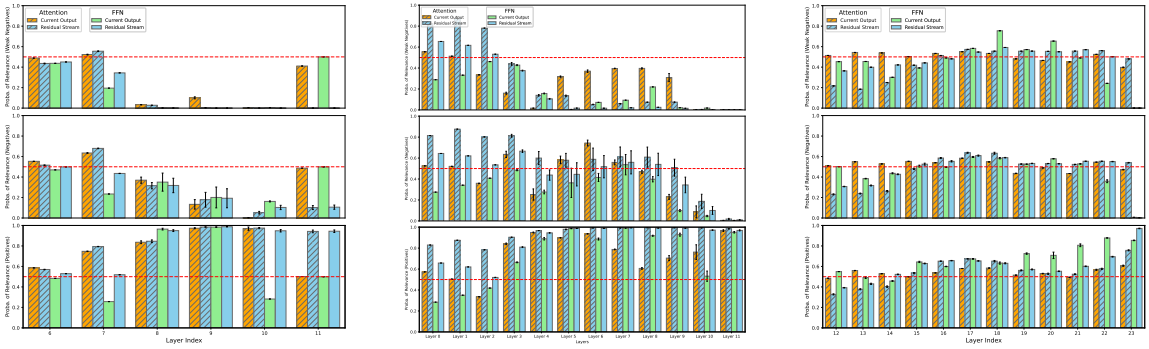
7.1 Context and Notations

To study the *relevance subspace*, we apply a Linear Discriminant Analysis (LDA) to three activations per layer (see Section 3.1 for the notations): $\mathbf{x}_j^{(\ell,a)}$, the attention output after LayerNorm, i.e., the residual stream[10] after attention; $\mathbf{x}_j^{(\ell,\tilde{f})}$, the raw FFN output; and $\mathbf{x}_j^{(\ell)}$, the residual stream after the FFN and its LayerNorm. LDA is a classical method used to identify discriminative subspaces within high-dimensional representation spaces, in our case, the "relevance subspace(s)" of the hidden states representational space. We further compute activations with respect to each input part and the relevance label associated with the sample.[11]

7.2 Results analysis

A relevance subspace does not exist from the start but is constructed by cross-encoders to store and refine relevance signals. From the previous section, we know that matching signals, a necessary piece in the relevance prediction process [56], are detected at different levels in cross-encoder models, from early layers until middle-later layers. From the conclusions we draw in Sections 5 and 6 on the importance of "matching detection" heads, we hypothesize that if a "relevance subspace" exists, the attention mechanism must be gradually directing these matching signals inside it. Our study of the information flow, and specifically its late focus on the `[CLS]` token representation, suggests that this subspace is centered around, or embedded within, the `[CLS]` token's representation space. This, in turn, coupled with our remarks on the probable role in the no-op operation [18] of the `[CLS]` in early layers, implies that the relevance subspace only appears at the end of the processing.

[10]We refer here to the mechanistic interpretability paradigm, where the transformer layers are seen as modules that read and write from a *residual stream* that goes all the way from the embedding layer to the final layer of the model [30].
[11]We provide more details on the different steps we follow to compute the LDA in Appendix G.



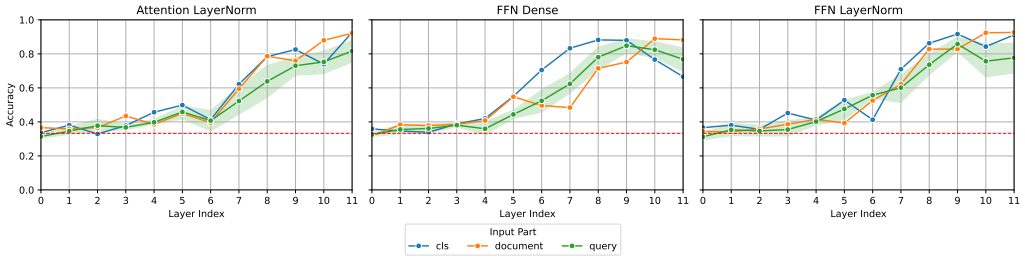
(a) Details for MiniLM. (b) Details for MonoBERT-base. (c) Details for MonoBERT-large.

Fig. 15. Evolution of the probability of relevance based on the [CLS] internal representations, per layers, for $r \in \{-1, 0, 1\}$.

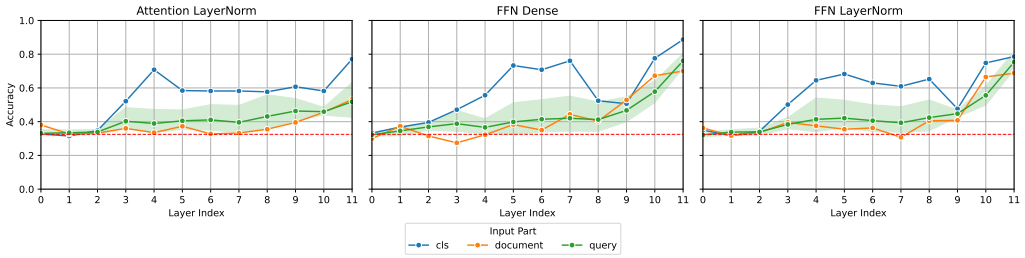
To test this hypothesis, we plot in Figure 15 the evolution of the confidence in the prediction "Relevant" of the three cross-encoders, per layers, based on the [CLS] representation at each step of the processing. We apply a procedure similar to approaches using the unembedding matrix of LLMs on top of the residual stream, where they project an internal representation onto the vocabulary [35]. In our work, we use the classification head of each cross-encoder on the internal representations of the [CLS]. In our case, this projection takes as input the hidden state of the [CLS] token at each layer, a vector of $\mathbb{R}^{d_\mathcal{T}}$, before/after it is written to the residual stream (i.e. $\mathbf{x}_{CLS}^{(\ell,\tilde{a})}$, $\mathbf{x}_{CLS}^{(\ell,a)}$, $\mathbf{x}_{CLS}^{(\ell,\tilde{f})}$ and $\mathbf{x}_{CLS}^{(\ell)}$), and returns as an output a probability of relevance in $[0, 1]$, based on the information currently stored inside it. We argue that, despite its obvious limitations (the classifier is not trained to predict relevance from these different representations), this approximation is reasonable and the trends it exhibits can still be interpreted as a proxy to monitor how much the [CLS] representation points towards relevance vs non relevance.

From these Figures, we first see that, for MiniLM and MonoBERT-large, the predicted probabilities are highly similar across relevance levels for most of the models' layers (up to layer 7 for MiniLM and 18 for MonoBERT-large). This confirms that, before a critical point, the representation of the [CLS] does not contain anything close to what the classification head of each model is expecting to predict the relevance. Meanwhile, we observe a dissimilar trend for MonoBERT-base, as this critical point occurs much earlier, around layer 3. After this layer, we observe that the model is already able to come close to its last layer's prediction, meaning the [CLS] token starts relatively early to aggregate relevance signals.

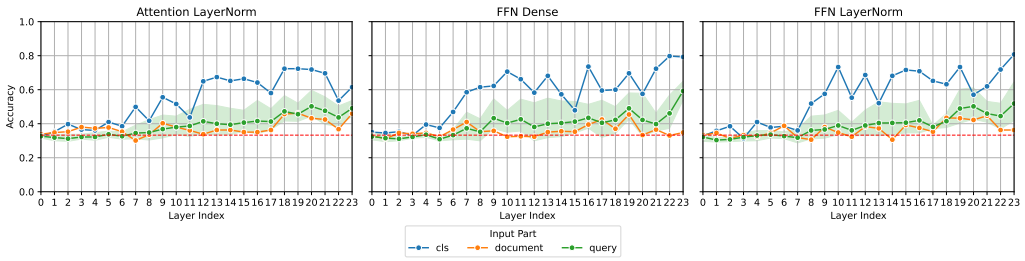
MiniLM and MonoBERT-base refine relevance signals directly inside query and document tokens before passing them to the [CLS], which "composes" them for the final prediction. In contrast, MonoBERT-large stores "raw" signals inside the [CLS] token representation and, after that, refines them in place. This result is suggested first by observing the patterns in Figure 15, past the decision point where each model's behavior starts to change depending on the relevance levels. While the attention process in MiniLM and MonoBERT-base clearly integrates query/document relevance information into the [CLS], MonoBERT-large instead gradually builds this information, firstly inside the residual of the FFN, before transmitting it to the residual stream ($\mathbf{x}_{CLS}^{(\ell,\tilde{f})}$ in green at layers 20-21 first passes below or above 0.5, before $\mathbf{x}_{CLS}^{(\ell)}$, in blue and unhatched, in the next layers). For the smaller cross-encoders, this suggests that the "raw" information transferred to the [CLS] token contains already, at least, a portion of the relevance signal. This could mean that either they can leverage "raw" matching signals, and perhaps other signals such as IDF



(a) Details for MiniLM.



(b) Details for MonoBERT-base.



(c) Details for MonoBERT-large.

Fig. 16. Evolution of the classification accuracy of the activations $\mathbf{x}_p^{(\ell,a)}$, $\mathbf{x}_p^{(\ell,\tilde{f})}$ and $\mathbf{x}_p^{(\ell)}$ projected along the LDA axis, by their original relevance label ($r \in \{-1, 0, 1\}$), per input part $p \in \{\texttt{[CLS]}, Q_1, ..., Q_k, \bar{D}\}$, $k = 6$. LDA and classification are computed over samples from the TREC-DL datasets.

information (already existing in the embedding matrix [14]), as suggested in Lu et al. [56], or that the matching signals (and possibly others) are already processed when they are transferred to the [CLS]. In contrast, MonoBERT-large seems to follow a much more incremental process, where the information that has been transferred to the [CLS] token needs to be refined by the next few FFN layers.

To investigate this observation further and lift the limitations of relying on the classification heads, we study the same phenomenon with an LDA, that further includes the query and document token representations. In Figures 16 and 25 (in Appendix, as both figures depict similar patterns), we compare the evolution of the classification accuracy between the different relevance levels (easy and/or hard negatives and positives) of the internal representations from the [CLS], query and document tokens[12], after they are projected along the first (or first two if the three relevance

[12]For the queries, we consider independently the first $k = 6$ tokens of each query. To get a single vector per document, we average its tokens' representations, per sample.

levels are considered at the same time) axis of the LDA. Firstly, we note that, no matter the domain, the classification accuracy, from any input part, improves as the layer index increases for MiniLM, while the same pattern only happens for the [CLS] token within MonoBERT-base and large. Secondly, if we now focus on the individual patterns of each module for each model, we can see that for MiniLM, the accuracy increases similarly at every level, while we observe differences between the attention and the FFN for MonoBERT-base (accuracy from query and document representations in the FFN increases towards the final layer, while they stay low in the attention). MonoBERT-large exhibits a somewhat similar pattern, but only for the query representations, not the document.

These trends can be seen more clearly if we only use the activations over query-passage pairs with relevance levels $r = -1$ (easy negatives) and $r = 1$ (relevant passage) (see Figures 26, 27 and 28 in the Appendix).

Finally, this observation echoes the patterns observed in the Sections 4 and 5 for the **RQ1** and **RQ2**, and reinforces the hypothesis that the relevance prediction processes of MiniLM, MonoBERT-base and MonoBERT-large are different.

■ **Cross-encoders tend to rely more on exact matches when they encounter OOD samples. This behavior can lead them to be fooled more easily by hard negatives in this context.** In the Appendix G, we provide additional figures detailing the evolution of the classification accuracy with only two relevance levels and over MS MARCO datasets or OOD datasets. As expected, we can clearly see that the classification accuracy is lower for relevance levels $r = 1$ vs. $r = 0$ than for $r = 1$ vs. $r = -1$. Interestingly, we also note that the overall classification accuracy further diminishes if we switch from ID (Figures 26a and 26b, 27a and 27b, and 28a and 28b, all in Appendix) to OOD (Figures 26c and 26d, 27c and 27d, and 28c and 28d, all in Appendix). Furthermore, the characteristic increase in accuracy, observed in every setup, occurs later, in terms of layer index, with hard negatives and in OOD. It seems that intermediate representations in middle layers are more similar between hard negatives, sampled with BM25, and positives (making them harder to differentiate) than between easy negatives, sampled randomly, and positives. Given their respective nature, this confirms that cross-encoders rely heavily on both *lexical* and *semantic* matching, and also confirms that each type of matching is processed at a different stage: early to middle layers for lexical matches versus middle to later layers for semantic ones.

To conclude this Section and answer **RQ4**, it seems that MonoBERT-large progressively builds a single relevance signal within the representational space of the [CLS] token by gathering information from the other tokens and then refining it. Meanwhile, MiniLM appears to be refining relevance information in place, across its query and document tokens, before transferring it to the [CLS] token. Finally, MonoBERT-base exhibits a *hybrid* behavior, where the [CLS] starts to accumulate information on relevance relatively early (similarly to MonoBERT-large), but at the same time, the query and document tokens also seem to be refining additional signals in place (similarly to MiniLM). For MiniLM and MonoBERT-base, our results alone cannot tell whether the information sent to the [CLS] in the last layers only needs to be filtered or weighted, or whether it is further refined inside the [CLS]. Yet, from Lu et al. [56] we know that it is possible to approximate, with a high correlation (80%), the predictions of MiniLM by linearly combining signals that have been extracted and stored inside query tokens by so-called "Relevance Scoring heads". This insight corroborates our hypothesis about MiniLM and confirms that, contrary to MonoBERT-large, MiniLM, at least, has learned to integrate within its [CLS] representation already refined relevance signals. For MonoBERT-base, we can suppose the model has already refined some signals inside the [CLS] (see previous points), prior to gathering additional signals from the query and/or document tokens.

8 Discussion

Before concluding, in this section we further discuss **RQ5** and **RQ6**, reflecting on the issues raised so far as well as discussing the limitations we identified in our study.

8.1 RQ5: On the consistency of our findings across cross-encoders

In our work, we focused on three cross-encoders, MiniLM [102] and MonoBERT [67], in both its base and large versions, that differ mostly by their pre-training regime (deep self-attention distillation [102] for MiniLM versus Masked Language Modeling and Next Sentence Prediction [25] for MonoBERT) and their size (see Table 4). Their comparable performance on the IR task (see Table 5) makes them interesting candidates to verify whether there is a single relevance prediction process towards which all cross-encoders have converged or not. Along with our first four Research Questions, we unveiled different insights suggesting that these cross-encoders possess their singularities. To answer **RQ5**: the stages of the relevance prediction process (contextualization and lexical matching, then semantic matching, then gathering of the relevance signal and construction of the relevance subspace) are similar across the studied models. This has been confirmed by our work around **RQ2** and **RQ3**, in Sections 5 and 6, where we observed very similar flows of information across the input parts as well as attention heads specialized in detecting matching signals. Where the models differ, is principally in the details of these different steps.

For example, we identified one major difference in the stage of gathering relevance signals and constructing a relevance subspace in Section 7. MonoBERT-large gathers raw relevance signals from the document into the query, then into the `[CLS]`, where it refines them and builds a relevance subspace. MiniLM and MonoBERT-base follow a different strategy. For MiniLM, the relevance signals gathered from the document to the query, are refined at the query-level before being transferred to the `[CLS]` token. Meanwhile, MonoBERT-base exhibits a "hybrid" pattern, as we observe, from a relatively early layer, the `[CLS]` beginning to build a relevance subspace, and in later layers, relevance information also starts to appear in the query and document tokens, suggesting that some level of refinement also happens there.

Finally, and perhaps most importantly, we unveiled evidence in Subsection 5.3 that MiniLM, contrary to MonoBERT, may not base its predictions on relevance signals, but on non-relevance signals. As described in the subsection 5.3, by training another MiniLM cross-encoder using harder negatives sampled from ColBERT-v2 [76], we further observed that this behavior may not be an artifact from the fine-tuning process or from the usage of random or "easy" negatives. Instead, it is perhaps tied to the pre-training regime of MiniLM (as we do not report the same trend for any of the two MonoBERT models), though the nature of the negatives seen during the fine-tuning also has an influence on its form. While early, this result strengthens the fact that not all cross-encoders share a unique relevance prediction process.

8.2 RQ6: On the consistency of our findings across domains

Overall, our findings indicate limited discrepancies in the models' behaviors in OOD compared to ID. Most of our observations on ID also verified themselves in OOD. This makes sense, as cross-encoders are considered to be relatively robust to domain shifts compared to other neural IR models like bi-encoders. However, previous works have suggested that there could exist some differences in MonoBERT's behavior in OOD [93]. In Section 4, we reproduced this finding but found it to be true mostly for MonoBERT (base and large), as MiniLM's most important neurons in both scenarios largely overlap. For MonoBERT, the role of these specific "OOD" neurons and attention heads remains opaque at this point. Through Sections 5 and 6, our observations and conclusions were consistent in both ID and OOD. Finally, in

Section 7, we uncovered that the construction of the relevance subspace(s) is hindered by domain shifts. We do not consider this to be a sign of a different mechanism but rather further evidence that transformer-based models, despite their reliance on lexical signals in information retrieval (IR), remain dependent on their domain knowledge and their semantic ability.

8.3 Design implications

Impact of the query sampling. Throughout this work, we relied on diverse sampling strategies to ensure a controlled setup but also to keep the computational cost of our study reasonable given its scale. We argue that as each sampling strategy is designed to fit the requirements of each experimental setup (reproducing an "IR setup", limiting the query-passage pairs or ensuring some levels of diversity in the qrels present in the final pools), and as we run our experiments on multiple datasets, their impact on our observations remains low, even for individual datasets such as NFCorpus or Natural Questions, which are arguably the most impacted datasets of our pool (see Table 7).

We did not try to use a single sampling strategy across all our experiments, as none of them has the characteristics to fit all their requirements at once. For instance, while the BM25 sampling strategy is the closest to the IR task, and thus, perhaps the most relevant in the context of this study, we would not have been able to control the number of samples per relevance level for the Sections 5, 6 or 7.

On the implications of using NIG to identify important neurons. While NIG attributes importance scores to every neuron in the model *with regard to* the IR task, the roles of neurons with a score close to 0 or even largely negative are unclear at this point. By using triplets and back-propagating gradients from their contrast, we try to better target the core concept of "relevance" inside the model and attribute higher NIG scores to components closely related to it. However, this does not necessarily mean that the rest of the networks could be pruned without any impact on the model's performance. We have not explored the roles of neurons with negative NIG scores, and we do not have an explanation for the meaning of such scores. In our work, we restricted ourselves to exploring the roles of the most important neurons and attention heads, i.e., the ones with the highest NIG scores, and showed that our method allows us to identify key components in the relevance prediction process of cross-encoders.

Information leakage in "Attention Mask" ablations. Indeed, while it is more limited in the context of the other ablations, as we mask *after* the softmax operation, following the Attention Mask procedure [111], the distribution of probabilities of each target token in each attention head is still updated based on the information available at this stage. In the case where we prevent all but one source token, as in the experiment where we blocked every transfer except the `[CLS]` ← `[CLS]` in Section 5.3 from sending information to the target token, its conditional probability of attention significantly impacts the target token. In that sense, although other tokens are hidden, their information can still influence the final results by modifying the conditional probability of attention of the last remaining source token.

Is there a best tool to study the information flow? In Section 5, we use 3 different approaches to study the information flow: the Neuron Integrated Gradients (NIG) [82, 84], the Information Bottleneck (IB) [78] and an ablation study. Beyond the strengths and weaknesses listed in Table 1, we discuss here the insights and the granularity each of them brings. From NIG, we obtain a relatively high-level picture of the information flow throughout the model. In addition, NIG results alone are hard to interpret in the context of the attention (see the analysis on the role of the `[SEP]` in Section 5), given the importance of the softmax operator, which greatly influences the roles of the tokens (attention sinks, no-op, etc.). While IB alleviates this limitation and can account for the interplay between the different components of the

model, the learning procedure makes it harder to study multiple datasets. However, its success depends on finding the proper set of hyper-parameters (we present less accurate results for MonoBERT-large for example, for the same set of hyper-parameters). Finally, ablations can provide more causal results, informing us about the importance of a direction by measuring the impact of its absence on the final predictions. Ablations need no hyper-parameters, but a complete picture of every direction at every stage and across domains requires many targeted ablations, which is computationally intensive, and one must also make sure that no information leaks (see the previous point).

It is hard to state whether one method is fundamentally better. From our study, we find it more interesting to combine their insights to provide a more complete picture and to make our conclusions more robust.

Attention probability matrix and transformer-based model behavior. This question fits into the broader debate of using attention as an explanatory tool [44, 107]. In our case, we use the attention mechanism mostly in Sections 5 and 6. For Section 5, we do not directly analyze the attention scores but instead analyze the flow of information with introspection techniques such as Neuron Integrated Gradients [82, 84], ablations, or the Information Bottleneck approach [78], which provide more meaningful results.

In Section 6, we directly work on attention patterns and found that the results correlate with other results from the rest of our work, including methods that take into account the entire computational graph. For that reason, we believe that our method falls into the scope of the answer to the "Attention is not Explanation" paper [44] from Wiegreffe and Pinter [107] so our results can be regarded as faithful and valuable because they are supported by other approaches.

Furthermore, we attempted to explain the matching behavior of the attention heads with the most distinctive patterns by analyzing the bilinear transformation defined by the Query-Key matrix $\left(W_Q^h\right)^\top W_K^h$ in Section F.3 in a mechanistic interpretability manner. As it captures most of the behaviors of the attention heads we identified, we also consider this experiment further validation of our analysis of the attention patterns.

8.4 On the impact of training strategies and models' depth

In addition to the discussions in Section 8.1 about the consistencies of our findings across the models, we want to discuss more specifically the relationship between these differences and the pre-training strategies, as well as the size of the models. In our work, MonoBERT-base acts as "pivot" model, sharing both the number of layers of MiniLM and the pre-training strategy of MonoBERT-large. By comparing its behavior to those of MiniLM and MonoBERT-large, we can draw hypothesis on the source cause behind some of their traits.

In particular, from our analysis based on the Neuron Integrated Gradients in Sections 4 and 5, we noticed that MiniLM appears to be more robust to domain shift than both MonoBERT models, both in terms of what neurons are involved in the prediction process but also in terms of information flows. For the latter, this difference may be explained by the fact that the model has less "space" in its representations compared to MonoBERT-base or MonoBERT-large, but also by the specific pre-training of MiniLM [102]. As the model has been pre-trained by distilling attention patterns from a teacher model, it is likely that this emphasis has led it to learn more consistent patterns. In contrast, models pre-trained on self-supervised tasks, such as Masked Language Modeling or Next Sentence Prediction, due to mechanisms such as the dropout which tends to add redundancy, may have learned more fuzzy attention patterns, which could have ultimately impacted their robustness. This hypothesis is further reinforced by the observation that MonoBERT-large is seemingly integrating relevance information directly inside the `[CLS]` token, much more at least than MiniLM. This could also be a direct cause of the NSP pre-training, where the `[CLS]` is used to predict if two sentences are consecutive or not. Interestingly, we have seen that MonoBERT-base shares a similar habit of integrating information of relevance inside

the [CLS], but Section 7 also revealed that it also uses the document and query tokens, just as MiniLM. Since depth is the only characteristic these two models share, our hypothesis is that building the relevance signal inside a single token, the [CLS], requires many layers. Consequently, smaller models may have learned to share this task across other tokens, in this case query and/or document tokens, and to only integrate "refined" signals inside the [CLS].

From Section 6, we saw that the behaviors of the attention heads in MiniLM and MonoBERT-base are more robust to harder negatives and domain shifts than the ones of MonoBERT-large. We attribute this to the fact that MonoBERT-large has been fine-tuned with a simpler Binary Cross Entropy loss [67], while the cross-encoder based on MiniLM has been fine-tuned using a listwise loss[13] and our MonoBERT-base with InfoNCE [91]. Figure 10b underlines the impact of the negative sampling strategy on a model's behavior.

At the same time, and while all three models accumulate relevance inside the query tokens in the last few layers, the analysis of the matching patterns revealed distinct patterns between MiniLM and the two MonoBERT models. Indeed, while the difference in the attention weights is more pronounced when the document is relevant for both MonoBERT models (for the direction $Q \leftarrow D$), some attention heads in the last layers of MiniLM instead exhibit higher attention weights when the passage is not relevant. While there is no obvious link between this ability and the pre-training strategy of MiniLM (its most prominent unique trait), we do not have as of today, better explanation.

Some questions remain open: why MonoBERT-base relies less on its [CLS] than MonoBERT-large despite the same pre-training recipe, and whether one strategy is more effective than the other.

8.5 Implications of our findings for future work

In general, interpretability studies constitute solid foundations to improve neural models. By revealing key mechanisms or how specific computations are implemented, it becomes possible to leverage architectural improvements and deduce the importance of design choices. For instance in this work, we confirmed the prevalence of the following three steps in the three cross-encoders we studied, to predict the relevance of a document to a query: (1) Query and document independent contextualization across the first layers ($Q \leftarrow Q$ and $D \leftarrow D$); (2) Query and document interactions, especially through the attention ($Q \leftarrow D$); (3) Transmission of the relevance signals ("raw" or "refined") to the [CLS] token for the final prediction, from the query ([CLS] $\leftarrow Q$).

In a recent work [92], we directly leverage this insight to design an optimized cross-encoder where only the most important directions, i.e., that preserve the previous steps, are kept. Thanks to the present interpretability study, we have been able to correctly identify and remove superfluous interactions from cross-encoders. This led to the improvement of the efficiency/effectiveness trade-off, with much lower latencies and also better OOD performance.

In addition, our work also raised some interesting questions, such as why MiniLM seems to have learned to search for "non-relevance" signals, instead of relevance ones, or how exactly attention heads learn to detect query-document matches. While we leave them unanswered at this stage, we believe they are interesting directions for future work.

9 Limitations

In this study, cross-encoders based on encoder-decoder [68, 112] or decoder-only [59] architectures are not considered, which restricts the scope of our findings to encoder-only, and more specifically BERT-based, cross-encoders. Regarding the latter, and although comparisons would have been of interest, decoder-only models are not primarily designed for the IR task, as suggested by a recent study comparing encoder-only and decoder-only models of equal size across various

[13]The exact loss used to fine-tune the MiniLM cross-encoder is hard to trace back from the documentation (available at https://github.com/huggingface/sentence-transformers/tree/main/examples/cross_encoder/training/ms_marco) but the authors indicate it could have been a LambdaLoss [9].

NLP tasks [105]. Regarding the former, while extending our experiments to encoder-decoder models such as RankT5 [112] would have yielded valuable insights, it would have simultaneously introduced additional layers of complexity into our analysis. First, prompt design is non-trivial and has been shown to heavily influence cross-encoder predictions [7]. Second, the cross-attention heads and the decoder side introduce an additional layer that requires explanation and remains largely under-explored. For these reasons, our efforts were focused on encoder-only cross-encoders.

The extension of this work to more recent encoders, such as DeBERTa-v3 [39], ELECTRA [19], or ModernBERT [104], is left for future work. Given the scale of our experimental setup, the decision was made to restrict our scope to BERT-based cross-encoders. Building on our discussion of the impact of pre-training and layer depth (see section 8.4), we believe the series of steps we identified is shared across encoders beyond BERT. Nevertheless, each model may implement a given step differently, and it therefore cannot be asserted that all encoder-only cross-encoders share a single, uniform behavior.

A further limitation, already noted in the Design Implications section 8.3, concerns the accuracy of the Information Bottleneck results, particularly for the MonoBERT models. Because the method learns masks that depends on a carefully selected threshold to preserve performance while removing as many directions as possible, its results are hard to interpret in isolation. By combining IB with experiments based on NIG and ablations, and given that their results largely overlap, this limitation can be mitigated; however, further work is needed to design a more robust experimental plan specific to the Information Bottleneck.

10 Conclusion

In this work, we focused on better explaining the most effective IR models, namely the cross-encoders. This lack of knowledge is a long-standing issue in the IR community that previous works have already attempted to tackle. After reviewing the current state of knowledge on interpreting cross-encoders, we identified that the main limitations are a restricted scope of study (one mechanism) and of applications (single model and single domain, often MS MARCO). In contrast, we comprehensively study the relevance prediction process inside three cross-encoders over multiple datasets, both in-domain and out-of-domain.

The different steps of this process have been roughly hypothesized in the past [111]. Thanks to multiple experiments and the correlation of their results, we have been able to identify their nature and offer insights into their functioning:

(1) Initially, tokens lack contextualization, so the first mission of cross-encoders, similarly to transformer-based models, is to contextualize them to gain a semantic understanding of the query-passage pair. At this stage, self-contextualization (attention from the query or the document to itself) plays a major role, but we further showed that cross-contextualization also occurs at this stage (see Section 5). By studying the attention process more in-depth (see Section 6), we showed that through early to middle layers, some attention heads are specialized in detecting *lexical* matches between the query and the document.
(2) Moving forward in the relevance prediction process and the model's layers, we discovered other attention heads specialized in detecting query-document matches, but this time *semantic*. This happens later in cross-encoders, as the input tokens need to be contextualized first.
For both types of matching, we proposed an explanation for their detection (see subsection F.3).
(3) Once these matching signals (and perhaps other signals) have been detected, cross-encoders refine and gather them inside a *relevance subspace*. This step is decomposed into two different sub-steps, as cross-encoders first gather relevance signals in the query tokens, before transferring them to the `[CLS]`. Importantly, we noticed that

while MonoBERT models base their predictions on evidence that the passage is "relevant", MiniLM may instead be looking for the opposite, i.e., evidence that the passage is "non-relevant".
At this stage, we noticed distinct strategies across the models, probably due to the different pre-training strategies. While MonoBERT models construct a relevance subspace inside the `[CLS]` token and need multiple layers to accumulate the "relevance" signals and refine them, MiniLM constructs multiple relevance subspaces at the level of the query and document tokens (see Section 7). It suggests that MiniLM refines the "relevance" signals it has detected before sending them to the `[CLS]`, while MonoBERT gathers inside it "raw" signals. Note that unlike its larger counterpart, MonoBERT-base also tends to construct relevance subspaces at the level of the query and document tokens, in a similar fashion to MiniLM. Since both models have the same depth, we hypothesize that the number of layers is a bottleneck for building a relevance subspace inside the `[CLS]` alone, which smaller models avoid by spreading the task across tokens.

(4) Finally, the "relevance" signals are refined/aggregated inside the `[CLS]` before the final relevance prediction.

By comparing three BERT-based cross-encoders, of different sizes (from 33M to 300M) and trained following different strategies, we reinforced the hypothesis that this multi-stage process is shared by most BERT-based cross-encoders (even if they differ in their final processing). In addition, we also showed that this process is consistent across domains, which might explain their better generalization ability compared to other neural IR models. Our work, however, underlined how difficult it is for these models to deal with hard negatives, especially in OOD. It suggests that cross-encoders may over-rely on *lexical* matches when they fail to capture the semantics of a new domain.

Following this in-depth comparison of MiniLM and MonoBERT, we can assert that they have comparable performance, despite major differences in their behaviors. Some of our findings further indicate that MonoBERT (large and base) might be over-parameterized (see Section 4). Given its slightly better performance (see Table 5) and the greater robustness of its mechanism across domains, MiniLM-L12-v2 is a more reasonable choice than MonoBERT-base or large, for both the efficiency-effectiveness trade-off and generalization.

Overall, our work provides a robust foundation for future interpretability studies in IR. We demonstrated the importance of including various datasets and considering more than a single model to put results into perspective and strengthen conclusions. Some of the insights we unveiled, such as the fact that MiniLM may rely on "non-relevance" signals and how some attention heads can detect query-document matches, also constitute interesting directions for future work.

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, this work was granted access to the HPC resources of IDRIS under the allocation 2024-AD011014444R1 made by GENCI.

References

[1] Gianni Amati and Cornelis Joost Van Rijsbergen. 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.* 20, 4 (Oct. 2002), 357–389. doi:10.1145/582415.582416

[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] Avishek Anand, Procheta Sen, Sourav Saha, Manisha Verma, and Mandar Mitra. 2023. Explainable Information Retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Taipei, Taiwan) *(SIGIR '23)*. Association for Computing

Machinery, New York, NY, USA, 3448–3451. doi:10.1145/3539618.3594249

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. arXiv:1607.06450 [stat.ML] https://arxiv.org/abs/1607.06450

[5] Yonatan Belinkov. 2022. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics* 48, 1 (04 2022), 207–219. arXiv:https://direct.mit.edu/coli/article-pdf/48/1/207/2006605/coli_a_00422.pdf doi:10.1162/coli_a_00422

[6] Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. 2016. A Full-Text Learning to Rank Dataset for Medical Information Retrieval. *Proceedings of the 38th European Conference on Information Retrieval.* http://www.cl.uni-heidelberg.de/~riezler/publications/papers/ECIR2016.pdf

[7] Marco Braga, Sean MacAvaney, Craig Macdonald, and Gabriella Pasi. 2026. Revealing MonoT5s Learning Mechanisms via Prompt-Token Adaptation. In *Advances in Information Retrieval: 48th European Conference on Information Retrieval, ECIR 2026, Delft, The Netherlands, March 29 – April 2, 2026, Proceedings, Part I* (Delft, The Netherlands). Springer-Verlag, Berlin, Heidelberg, 450–465. doi:10.1007/978-3-032-21289-4_29

[8] Davis Brown, Nikhil Vyas, and Yamini Bansal. 2023. On Privileged and Convergent Bases in Neural Network Representations. arXiv:2307.12941 [cs.LG] https://arxiv.org/abs/2307.12941

[9] Christopher Burges, Robert Ragno, and Quoc Le. 2006. Learning to Rank with Nonsmooth Cost Functions. In *Advances in Neural Information Processing Systems*, B. Schölkopf, J. Platt, and T. Hoffman (Eds.), Vol. 19. MIT Press. https://proceedings.neurips.cc/paper_files/paper/2006/file/af44c4c56f385c43f2529f9b1b018f6a-Paper.pdf

[10] Arthur Câmara and Claudia Hauff. 2020. Diagnosing BERT with Retrieval Heuristics. In *Advances in Information Retrieval*, Joemon M. Jose, Emine Yilmaz, João Magalhães, Pablo Castells, Nicola Ferro, Mário J. Silva, and Flávio Martins (Eds.). Springer International Publishing, Cham, 605–618.

[11] Nick Cammarata, Shan Carter, Gabriel Goh, Chris Olah, Michael Petrov, Ludwig Schubert, Chelsea Voss, Ben Egan, and Swee Kiat Lim. 2020. Thread: Circuits. *Distill* (2020). doi:10.23915/distill.00024 https://distill.pub/2020/circuits.

[12] 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*. 1401–1410.

[13] Yuheng Chen, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024. Journey to the Center of the Knowledge Neurons: Discoveries of Language-Independent Knowledge Neurons and Degenerate Knowledge Neurons. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 16 (March 2024), 17817–17825. doi:10.1609/aaai.v38i16.29735

[14] Jaekeol Choi, Euna Jung, Sungjun Lim, and Wonjong Rhee. 2022. *Finding Inverse Document Frequency Information in BERT.* arXiv:2202.12191 [cs] doi:10.48550/arXiv.2202.12191

[15] Tanya Chowdhury, Atharva Nijasure, and James Allan. 2025. Probing Ranking LLMs: A Mechanistic Analysis for Information Retrieval. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)* (Padua, Italy) *(ICTIR '25)*. Association for Computing Machinery, New York, NY, USA, 336–346. doi:10.1145/3731120.3744603

[16] Tanya Chowdhury, Razieh Rahimi, and James Allan. 2023. Rank-LIME: Local Model-Agnostic Feature Attribution for Learning to Rank. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval* (Taipei, Taiwan) *(ICTIR '23)*. Association for Computing Machinery, New York, NY, USA, 33–37. doi:10.1145/3578337.3605138

[17] Tanya Chowdhury, Yair Zick, and James Allan. 2025. RankSHAP: Shapley Value Based Feature Attributions for Learning to Rank. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=4011PUI9vm

[18] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What Does BERT Look at? An Analysis of BERT's Attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Tal Linzen, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes (Eds.). Association for Computational Linguistics, Florence, Italy, 276–286. doi:10.18653/v1/W19-4828

[19] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *ICLR*. https://openreview.net/pdf?id=r1xMH1BtvB

[20] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 deep learning track. In *Text REtrieval Conference (TREC)*. TREC. https://www.microsoft.com/en-us/research/publication/overview-of-the-trec-2020-deep-learning-track/

[21] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. 2022. Overview of the TREC 2021 deep learning track. In *Text REtrieval Conference (TREC)*. NIST, TREC. https://www.microsoft.com/en-us/research/publication/overview-of-the-trec-2021-deep-learning-track/

[22] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. 2023. Overview of the TREC 2022 deep learning track. In *Text REtrieval Conference (TREC)*. NIST, TREC. https://www.microsoft.com/en-us/research/publication/overview-of-the-trec-2022-deep-learning-track/

[23] 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/

[24] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge Neurons in Pretrained Transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*. 8493–8502.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North*. Association for Computational Linguistics. doi:10.18653/v1/n19-1423

[26] 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

[27] Jonathan Donnelly and Adam Roegiest. 2019. On Interpretability and Feature Representations: An Analysis of the Sentiment Neuron. In *Advances in Information Retrieval*, Leif Azzopardi, Benno Stein, Norbert Fuhr, Philipp Mayr, Claudia Hauff, and Djoerd Hiemstra (Eds.). Springer International Publishing, Cham, 795–802.

[28] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy Models of Superposition. *Transformer Circuits Thread* (2022). https://transformer-circuits.pub/2022/toy_model/index.html.

[29] Nelson Elhage, Robert Lasenby, and Christopher Olah. 2023. Privileged Bases in the Transformer Residual Stream. https://transformer-circuits.pub/2023/privileged-basis/index.html

[30] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A Mathematical Framework for Transformer Circuits. *Transformer Circuits Thread* (2021). https://transformer-circuits.pub/2021/framework/index.html.

[31] Kawin Ethayarajh. 2019. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 55–65. doi:10.18653/v1/D19-1006

[32] 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

[33] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. ACM. doi:10.1145/3404835.3463098

[34] 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*. Springer-Verlag, Berlin, Heidelberg, 257–263. doi:10.1007/978-3-030-72240-1_23

[35] Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting Recall of Factual Associations in Auto-Regressive Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 12216–12235. doi:10.18653/v1/2023.emnlp-main.751

[36] Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. Transformer Feed-Forward Layers Build Predictions by Promoting Concepts in the Vocabulary Space. *arXiv preprint arXiv:2203.14680* (2022).

[37] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer Feed-Forward Layers Are Key-Value Memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 5484–5495. doi:10.18653/v1/2021.emnlp-main.446

[38] Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. 2023. Localizing model behavior with path patching. *arXiv preprint arXiv:2304.05969* (2023).

[39] Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=sE7-XhLxHA

[40] Shwai He, Guoheng Sun, Zheyu Shen, and Ang Li. 2024. What Matters in Transformers? Not All Attention is Needed. arXiv:2406.15786 [cs.LG] https://arxiv.org/abs/2406.15786

[41] Maria Heuss, Maarten de Rijke, and Avishek Anand. 2024. RankingSHAP–Listwise Feature Attribution Explanations for Ranking Models. *arXiv preprint arXiv:2403.16085* (2024).

[42] John Hewitt and Christopher D. Manning. 2019. A Structural Probe for Finding Syntax in Word Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 4129–4138. doi:10.18653/v1/N19-1419

[43] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 113–122.

[44] Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 3543–3556. doi:10.18653/v1/N19-1357

[45] Zhiying Jiang, Raphael Tang, Ji Xin, and Jimmy Lin. 2021. How Does BERT Rerank Passages? An Attribution Analysis with Information Bottlenecks. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, Jasmijn Bastings, Yonatan Belinkov, Emmanuel Dupoux, Mario Giulianelli, Dieuwke Hupkes, Yuval Pinter, and Hassan Sajjad (Eds.). Association for Computational Linguistics, Punta Cana, Dominican Republic, 496–509. doi:10.18653/v1/2021.blackboxnlp-1.39

[46] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 6769–6781. doi:10.18653/v1/2020.emnlp-main.550

[47] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. 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, 39–48. doi:10.1145/3397271.3401075

[48] János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. 2024. AtP*: An efficient and scalable method for localizing LLM behaviour to components. arXiv:2403.00745 [cs.LG]

[49] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: a Benchmark for Question Answering Research. *Transactions of the Association of Computational Linguistics* (2019).

[50] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[51] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 9459–9474. https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf

[52] Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2026. Qwen3-VL-Embedding and Qwen3-VL-Reranker: A Unified Framework for State-of-the-Art Multimodal Retrieval and Ranking. *arXiv* (2026).

[53] Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open Sesame: Getting inside BERT‘s Linguistic Knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Tal Linzen, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes (Eds.). Association for Computational Linguistics, Florence, Italy, 241–253. doi:10.18653/v1/W19-4825

[54] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A STRUCTURED SELF-ATTENTIVE SENTENCE EMBEDDING. In *International Conference on Learning Representations*. https://openreview.net/forum?id=BJC_jUqxe

[55] Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic Knowledge and Transferability of Contextual Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 1073–1094. doi:10.18653/v1/N19-1112

[56] 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

[57] Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 4768–4777.

[58] Daniel Lundström, Tianjian Huang, and Meisam Razaviyayn. 2022. A Rigorous Study of Integrated Gradients Method and Extensions to Internal Neuron Attributions. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (Eds.). PMLR, 14485–14508. https://proceedings.mlr.press/v162/lundstrom22a.html

[59] Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-Tuning LLaMA for Multi-Stage Text Retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 2421–2425. doi:10.1145/3626772.3657951

[60] Sean MacAvaney, Sergey Feldman, Nazli Goharian, Doug Downey, and Arman Cohan. 2022. ABNIRML: Analyzing the Behavior of Neural IR Models. *Transactions of the Association for Computational Linguistics* 10 (2022), 224–239. doi:10.1162/tacl_a_00457

[61] Sean MacAvaney, Andrew Yates, Sergey Feldman, Doug Downey, Arman Cohan, and Nazli Goharian. 2021. Simplified Data Wrangling with ir_datasets. In *SIGIR*.

[62] Joel Mackenzie, Shengyao Zhuang, and Guido Zuccon. 2023. Exploring the Representation Power of SPLADE Models. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*. 143–147.

[63] Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. WWW'18 Open Challenge: Financial Opinion Mining and Question Answering. In *Companion Proceedings of the The Web Conference 2018* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1941–1942. doi:10.1145/3184558.3192301

[64] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems* 35 (2022), 17359–17372.

[65] Lucas Moeller, Dmitry Nikolaev, and Sebastian Padó. 2023. An Attribution Method for Siamese Encoders. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 15818–15827. doi:10.18653/v1/2023.emnlp-main.980

[66] 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

[67] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[68] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document Ranking with a Pretrained Sequence-to-Sequence Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics. doi:10.18653/v1/2020.findings-emnlp.63

[69] 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, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context Learning and Induction Heads. *Transformer Circuits Thread* (2022). https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

[70] Vitali Petsiuk, Abir Das, and Kate Saenko. 2018. RISE: Randomized Input Sampling for Explanation of Black-box Models. In *Proceedings of the British Machine Vision Conference (BMVC)*.

[71] Benjamin Piwowarski. 2020. Experimaestro and Datamaestro: Experiment and Dataset Managers (for IR). 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, 2173–2176. doi:10.1145/3397271.3401410

[72] Ronak Pradeep, Yuqi Liu, Xinyu Zhang, Yilin Li, Andrew Yates, and Jimmy Lin. 2022. Squeezing Water from a Stone: A Bag of Tricks for Further Improving Cross-Encoder Effectiveness for Reranking. In *Advances in Information Retrieval*, Matthias Hagen, Suzan Verberne, Craig Macdonald, Christin Seifert, Krisztian Balog, Kjetil Nørvåg, and Vinay Setty (Eds.). Springer International Publishing, Cham, 655–670.

[73] Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the Behaviors of BERT in Ranking. arXiv:1904.07531 [cs.IR] https://arxiv.org/abs/1904.07531

[74] 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

[75] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics* 8 (2020), 842–866. doi:10.1162/tacl_a_00349

[76] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (Eds.). Association for Computational Linguistics, Seattle, United States, 3715–3734. doi:10.18653/v1/2022.naacl-main.272

[77] Ferdinand Schlatt, Maik Fröbe, Harrisen Scells, Shengyao Zhuang, Bevan Koopman, Guido Zuccon, Benno Stein, Martin Potthast, and Matthias Hagen. 2025. Rank-DistiLLM: Closing the Effectiveness Gap Between Cross-Encoders and LLMs for Passage Re-ranking. In *Advances in Information Retrieval. 47th European Conference on IR Research (ECIR 2025) (Lecture Notes in Computer Science)*. Springer, Berlin Heidelberg New York. doi:10.1007/978-3-031-88714-7_31

[78] Karl Schulz, Leon Sixt, Federico Tombari, and Tim Landgraf. 2020. Restricting the Flow: Information Bottlenecks for Attribution. In *International Conference on Learning Representations*. https://openreview.net/forum?id=S1xWh1rYwB

[79] 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

[80] C. E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal* 27, 3 (1948), 379–423. doi:10.1002/j.1538-7305.1948.tb01338.x

[81] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning Important Features Through Propagating Activation Differences. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 3145–3153. https://proceedings.mlr.press/v70/shrikumar17a.html

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

[83] Ian Soboroff, Shudong Huang, and Donna K. Harman. 2018. TREC 2018 News Track Overview. In *Text Retrieval Conference*. https://api.semanticscholar.org/CorpusID:197640252

[84] 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.

[85] Aaquib Syed, Can Rager, and Arthur Conmy. 2023. Attribution Patching Outperforms Automated Circuit Discovery. In *NeurIPS Workshop on Attributing Model Behavior at Scale*. https://openreview.net/forum?id=tiLbFR4bJW

[86] 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*, Anna Korhonen, David Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, Florence, Italy, 4593–4601. doi:10.18653/v1/P19-1452

[87] 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 *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. https://openreview.net/forum?id=wCu6T5xFjeJ

[88] Nicola Tonellotto. 2022. *Lecture Notes on Neural Information Retrieval.* arXiv:2207.13443 [cs] http://arxiv.org/abs/2207.13443

[89] George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael Alvers, Dirk Weißenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artieres, Axel-Cyrille Ngonga Ngomo, Norman Heino, Eric Gaussier, Liliana Barrio-Alvers, and Georgios Paliouras. 2015. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics* 16 (04 2015), 138. doi:10.1186/s12859-015-0564-6

[90] Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A. Gers. 2019. How Does BERT Answer Questions?: A Layer-Wise Analysis of Transformer Representations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM '19)*. ACM. doi:10.1145/3357384.3358028

[91] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation Learning with Contrastive Predictive Coding. arXiv:1807.03748 [cs.LG]

[92] Mathias Vast, Victor Morand, Basile van Cooten, Laure Soulier, Josiane Mothe, and Benjamin Piwowarski. 2026. MICE: Minimal Interaction Cross-Encoders for efficient Re-ranking. arXiv:2602.16299 [cs.IR] https://arxiv.org/abs/2602.16299

[93] 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

[94] 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

[95] Manisha Verma and Debasis Ganguly. 2019. LIRME: Locally Interpretable Ranking Model Explanation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 1281–1284. doi:10.1145/3331184.3331377

[96] 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 *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 12388–12401. https://proceedings.neurips.cc/paper_files/paper/2020/file/92650b2e92217715fe312e6fa7b90d82-Paper.pdf

[97] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Anna Korhonen, David Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, Florence, Italy, 5797–5808. doi:10.18653/v1/P19-1580

[98] Ellen Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R.Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. 2021. TREC-COVID: constructing a pandemic information retrieval test collection. (2021). doi:10.1145/3451964.3451965

[99] Ellen M. Voorhees. 2005. The TREC robust retrieval track. *SIGIR Forum* 39, 1 (jun 2005), 11–20. doi:10.1145/1067268.1067272

[100] Chase Walker, Sumit Jha, Kenny Chen, and Rickard Ewetz. 2024. Integrated Decision Gradients: Compute Your Attributions Where the Model Makes Its Decision. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 6 (march 2024), 5289–5297. doi:10.1609/aaai.v38i6.28336

[101] Jonas Wallat, Fabian Beringer, Abhijit Anand, and Avishek Anand. 2023. Probing BERT for Ranking Abilities. In *Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part II* (Dublin, Ireland). Springer-Verlag, Berlin, Heidelberg, 255–273. doi:10.1007/978-3-031-28238-6_17

[102] 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

[103] Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. 2022. Finding Skill Neurons in Pre-trained Transformer-based Language Models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 11132–11152. doi:10.18653/v1/2022.emnlp-main.765

[104] Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, Vienna, Austria, 2526–2547. doi:10.18653/v1/2025.acl-long.127

[105] Orion Weller, Kathryn Ricci, Marc Marone, Antoine Chaffin, Dawn Lawrie, and Benjamin Van Durme. 2025. Seq vs Seq: An Open Suite of Paired Encoders and Decoders. (2025). arXiv:2507.11412 [cs.CL] https://arxiv.org/abs/2507.11412

[106] Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does BERT Make Any Sense? Interpretable Word Sense Disambiguation with Contextualized Embeddings. arXiv:1909.10430 [cs.CL] https://arxiv.org/abs/1909.10430

[107] Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not Explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 11–20. doi:10.18653/v1/D19-1002

[108] Ji Xin, Rodrigo Nogueira, Yaoliang Yu, and Jimmy Lin. 2020. Early Exiting BERT for Efficient Document Ranking. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, Nafise Sadat Moosavi, Angela Fan, Vered Shwartz, Goran Glavaš, Shafiq Joty, Alex Wang, and Thomas Wolf (Eds.). Association for Computational Linguistics, Online, 83–88. doi:10.18653/v1/2020.sustainlp-1.11

[109] Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the Use of Lucene for Information Retrieval Research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 1253–1256. doi:10.1145/3077136.3080721

[110] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[111] 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

[112] Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. 2023. RankT5: Fine-Tuning T5 for Text Ranking with Ranking Losses. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Taipei, Taiwan) *(SIGIR '23)*. Association for Computing Machinery, New York, NY, USA, 2308–2313. doi:10.1145/3539618.3592047

[113] Yuxuan Zong and Benjamin Piwowarski. 2023. XpmIR: A Modular Library for Learning to Rank and Neural IR Experiments. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Taipei, Taiwan) *(SIGIR '23)*. Association for Computing Machinery, New York, NY, USA, 3185–3189. doi:10.1145/3539618.3591818

List of Figures

1 Illustration of the Neuron Integrated Gradients method. The blue curve (left axis) describes the recovery of the model's logit $z(\cdot)$ (i.e., the input of the sigmoid in P) as we progress along the path $\gamma(t)$. Green curves (right axis) represent the gradients $\frac{\partial \gamma_y(t)}{\partial t}$ of individual neurons along this path: plain curves correspond to important neurons, whose gradient is non-zero where the logit varies (between the two red lines), and the dashed curve to a non-important neuron. 11

2 Number of most important components per layer 14

3 Comparison of the ablation schemes I_O and I_A 15

4 Description of an attention weight matrix 16

5 Segmentation of an attention weight matrix between the different input parts 16

6 Information Bottleneck results on the information flow 19

7 NIG results on the information flow (attention heads) 20

8 NIG results on the information flow (FFN) 21

9 Ablation results on the information flow 24

10 Ablations to isolate the role of `[CLS]` ← `[CLS]` transfers on the prediction of relevance 25

11 Contrast of the attention patterns between relevant and easy negatives 28

12 Contrast of the attention patterns between relevant and hard negatives 29

14 Comparison of the matching patterns between relevant and easy negatives 31

15 Evolution of the relevance prediction across the models' layers 34

16 Evolution of the classification accuracy of the relevance subspace contained in the `[CLS]`, query tokens and document tokens for TREC-DL datasets 35

17 Description of the sampling strategies 53

18 Summary of the ablation results across all the datasets 54

19 NIG results on the information flow for I_M and F_M 55

20 NIG results on the information flow for I_A and F_A 56

21 Summary of the matching patterns across all the datasets for relevant and easy negatives 59

22 Summary of the matching patterns across all the datasets for relevant and hard negatives 59

23 Comparison of the matching patterns between relevant and hard negatives 60

24 Mechanistic interpretations of the detection of the matching signals in the attention of BERT-based cross-encoders 61

25 Evolution of the classification accuracy of the relevance subspace contained in the `[CLS]`, query tokens and document tokens for OOD datasets 64

26 Evolution of the binary classification accuracies for MiniLM 66

27 Evolution of the binary classification accuracies for MonoBERT-base 67

28 Evolution of the binary classification accuracies for MonoBERT-large 68

A Models

A.1 MonoBERT-base details

Given that MonoBERT-large and the cross-encoder based on MiniLM do not share the same training loss (binary cross-entropy for MonoBERT-large, and an unknown loss for MiniLM[14]), but also to keep the number of experiments reasonable, we decided to follow recent studies training cross-encoders [72, 77] and to use an InfoNCE loss, with 7 negatives and 1 positive per query. The model's re-ranking performance can be found in Table 5.

A.2 Overview of the models

Table 4. Details of the models in terms of layer numbers and structures, as well as overall parameter counts.

Model	# layers	# attention heads per layer	Hidden size	# params
MiniLM-L12-v2 [102]	12	12	384	33M
MonoBERT-base [67]	12	12	768	110M
MonoBERT-large [67]	24	16	1024	340M

A.3 Re-ranking performance of the models

Table 5 details the performance on the re-ranking task of the cross-encoders with our pool of datasets. We report their performance when re-ranking the top 100 passages retrieved by BM25 (using default values for the hyper-parameters $k_1 = 0.9$ and $b = 0.4$) in *Anserini* [109], along with the performance of BM25.

Furthermore, Table 5 also provides references for the ranking performance of a **Random** classifier and for the **Worst** (possible) classifier, on each dataset. One can observe that in the case of the **Worst** classifier, some datasets have an above 0 nDCG@10 value, in particular TREC-DL19 [23] (0.02), NFCorpus [6] (0.12) and TREC-Covid [98] (0.01). This is mostly due to the composition of the datasets themselves: 1) The unexpectedly high "worst" nDCG@10 value for NFCorpus comes from the distributions of the annotations in the dataset. Table 6 highlights that there are only around 10 passages per query with a relevance level of 0. By construction of the "worst" possible ranking, this leads to a higher proportion of relevant passages in the top-100 passages and potential leaks in the top-10, which then explains the high nDCG@10 value. Similarly, TREC-Covid contains both a very high number of relevant passages per query and very few queries according to Table 6. This could explain that for a few of them, one relevant passage managed to make its way to the top-10 and then explains the above 0 nDCG@10. The same explanation may also apply to TREC-DL19: dividing the number of passages annotated with a relevance level of 1, 2 or 3 by the 43 queries of this track yields around 150 relevant passages per query. Note also that the annotation process of TREC-DL19 was supported by BM25, which further increases the probability that marginally relevant passages could leak into the top-10.

[14] Sentence-transformers, who trained the model, does not detail the loss that was used.

Table 5. Performance across datasets in nDCG@10. **Random** corresponds to the nDCG@10 value obtained after averaging the score over 1000 random rankings. **Worst** corresponds to the nDCG@10 value obtained after re-ordering the top passages retrieved by BM25 in the increasing order of their relevance level (least relevant first).

Model	DL 19	20	21	22	NFCorp.	FiQA	T.-Covid	Rob.-04	NQ	BioASQ	Avg.
BM25	0.51	0.48	0.45	0.27	0.32	0.24	0.59	0.45	0.23	0.52	0.41
MiniLM	0.74	0.68	0.65	0.49	0.35	0.31	0.74	0.46	0.36	0.50	0.53
MonoBERT-base	0.73	0.68	0.64	0.49	0.35	0.30	0.73	0.47	0.36	0.50	0.52
MonoBERT-large	0.71	0.68	0.63	0.49	0.35	0.32	0.70	0.45	0.35	0.47	0.52
Random	0.24	0.17	0.24	0.18	0.23	0.04	0.40	0.19	0.07	0.06	0.18
Worst	0.02	0.00	0.00	0.00	0.12	0.00	0.01	0.00	0.00	0.00	0.02

B Datasets

B.1 Overview

Table 6. Details of the datasets, including the task associated with it, their respective number of topics and the distribution of the annotations, per relevance-level. Relevance levels in bold correspond to the label "relevant" after binarization. "MS MARCO" corresponds to the aggregation of the 4 tracks of TREC-DL.

Datasets	# Topics	Relevance Levels	# Qrels	Total
MS MARCO [20–23, 66] (Web Retrieval)	226	0 (not relevant)	303K	-
		1 (related)	59K	-
		2 (relevant)	51K	-
		3 (perfectly relevant)	4K	417K
FiQA [63] (Question Answering)	648	**1 (relevant)**	57K	57K
Natural Questions (Question Answering)	6.5K	-1 (easy negatives)	326K	-
		0 (hard negatives)	603K	-
		1 (contains the answer, retrieved by BM25)	45K	-
		2 (relevant)	6.5K	980K
BioASQ [89] (BioMedical-IR)	500	**1 (relevant)**	2359	2359
TREC-Covid [98] (BioMedical-IR)	50	0 (not relevant)	43K	-
		1 (partially relevant)	11K	-
		2 (relevant)	16K	69K
NFCorpus [6] (BioMedical-IR)	323	0 (doc. is marginally relevant to a related query)	3.5K	-
		1 (doc. is relevant to a related query)	11.8K	-
		2 (directly relevant)	576	15.8K
TREC-Robust04 [99] (News Retrieval)	249	0 (not relevant)	294K	-
		1 (relevant)	16K	-
		2 (highly relevant)	1K	311K

Table 7. Proportion of the original set of queries remaining after the different sampling strategies per dataset. Note that for Natural Questions, the reported ratios are below one because of the upper bound on the number of queries per dataset, set to 1000 ($\frac{1}{3.4} \approx 0.29$). For NFCorpus, however, the low ratio for Diverse Sampling is due to the filtering of the queries without "relevant" passages after the binarization (see Table 6) or because the model does not recognize them as "relevant".

Datasets	Top Sampling	Diverse Sampling	BM25 Sampling
TREC-DL 19	1	1	1
TREC-DL 20	0.96	1	1
TREC-DL 21	0.96	1	1
TREC-DL 22	0.99	1	1
NFCorpus	0.13	0.37	0.95
FiQA	0.46	1	1
TREC-Covid	0.98	1	1
Robust-04	0.57	1	1
NQ	0.08	0.29	0.29
BioASQ	0.78	1	1

B.2 Pre-processing

To fit our needs in the different experiments that we run in this work, we applied different pre-processing strategies to our datasets. Figure 17 illustrates the different steps of each strategy we performed to 1) limit the maximum number of queries considered and 2) limit the number of passages per query. The first step, shared across all tasks, consists of randomly sampling a maximum of 1000 queries (**Random Query Pooling** in Figure 17) per dataset (only impacts Natural Questions as shown in Table 6). Then, for this subset of queries, we sample the passages depending on the analysis we wish to perform, giving rise to three different **sampling strategies**:

1- **Top sampling** means we sample 1000 query-passage pairs of each dataset. We apply it only to the experiments based on NIG (see Sections 4 and 5) to follow the work done in Vast et al. [93].

2- **Diverse sampling** means we ensure the different relevance levels are present (when available in the annotations) in the pool of passages of each query. For this strategy, we fix both a maximum number of passages per relevance level and a target number of passages per query. If the latter is not satisfied after sampling passages in the annotations, we use BM25[15] [1] and sample hard negatives up to the target number of passages. Once we sampled the target number of passages per query, we sampled easy negatives (see **Easy negatives sampling** on Figure 17). This strategy is used for both LDA (see Section 7) and the analysis of the attention patterns (see Sections 5 and 6) to ensure a certain level of variety in the distribution of the data. Note that to simplify our analysis of the results based on this strategy, we considered a passage as "relevant" (resp. "non-relevant") if, and only if, it is annotated as such in the original datasets *and* the model has classified it as such.

3- **BM25 sampling** means the selection of k (usually 100) passages per query using BM25. We use this strategy for the ablation study (see Section 5) and evaluations, as it mimics a typical IR setup and allows us to get meaningful results before and after the ablations that we can compare.

As we pre-processed each dataset in different ways to fit our needs in this work, we detail in Table 7 the proportion of queries (from the original set) still present after each sampling and filtering strategy.

[15]For pre-processing, BM25 is used with the same default values for each dataset, i.e., $k_1 = 0.9$ and $b = 0.4$.

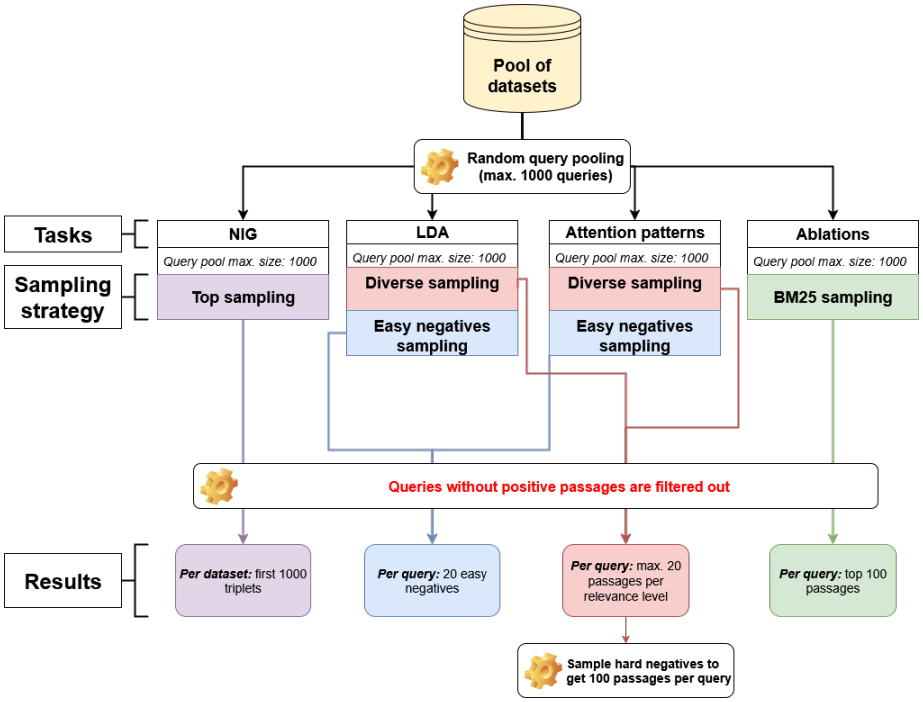
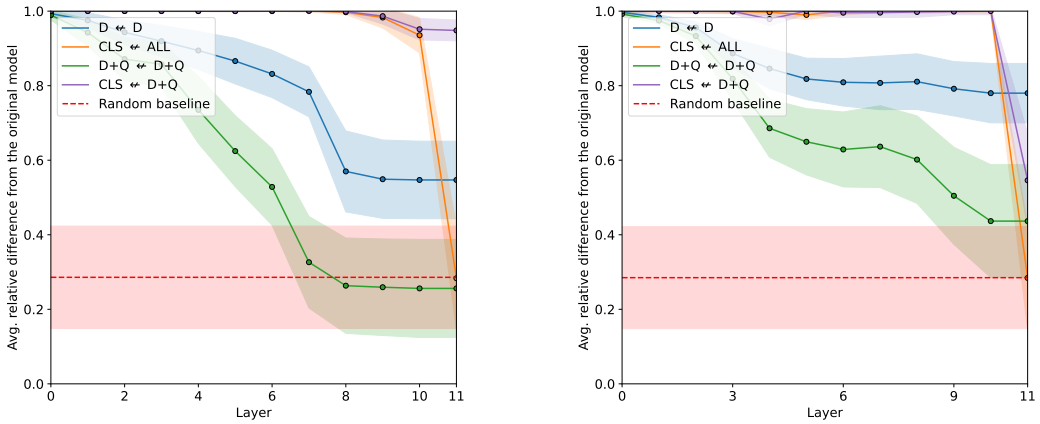


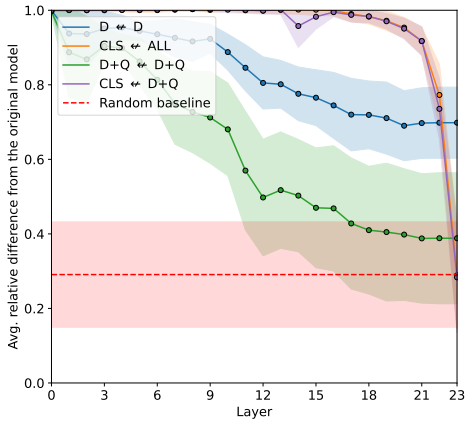
Fig. 17. Description of the different pre-processing steps we follow for each of our experiments. **Easy negatives sampling** indicates that, in addition to the pool of 100 passages for each query, we further randomly sample 20 passages, not already in the pool of assessments for that query, in the dataset to serve as *easy negatives*.

C Ablation study



(a) Average percentage of the original performance preserved on MiniLM.

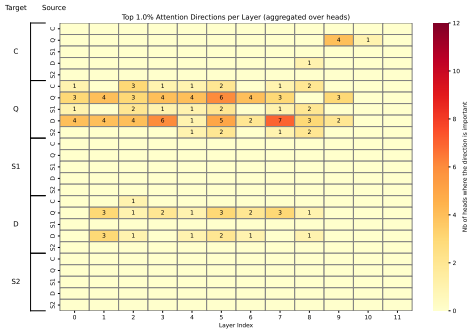
(b) Average percentage of the original performance preserved on MonoBERT-base.



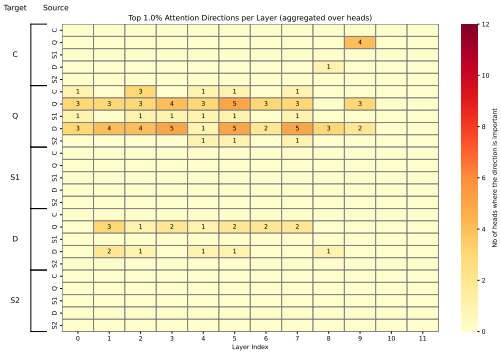
(c) Average percentage of the original performance preserved on MonoBERT-large.

Fig. 18. Average percentage of the original performance preserved after the ablations of directions inside the attention heads of the models up to the layer ℓ. The average is computed over all the datasets.

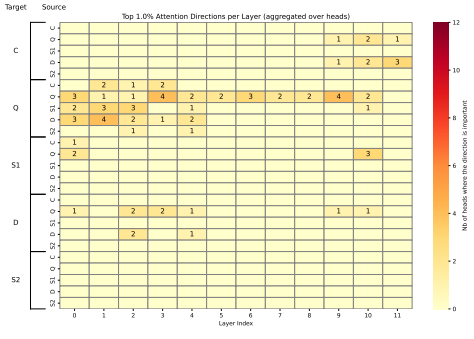
D Neuron Integrated Gradients



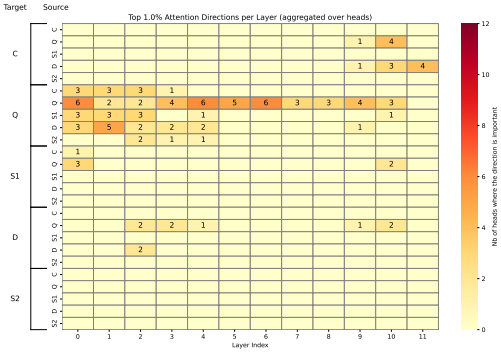
(a) Ablation scheme I_M for MiniLM.



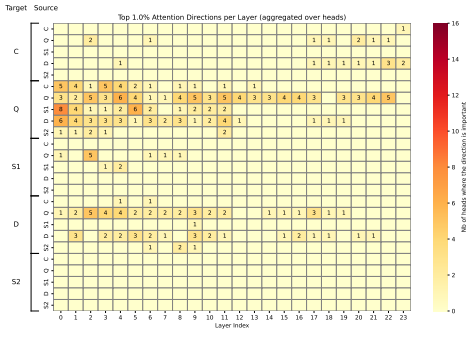
(b) Ablation scheme F_M for MiniLM.



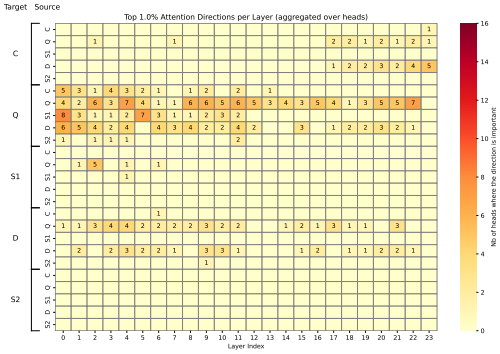
(c) Ablation scheme I_M for MonoBERT-base.



(d) Ablation scheme F_M for MonoBERT-base.

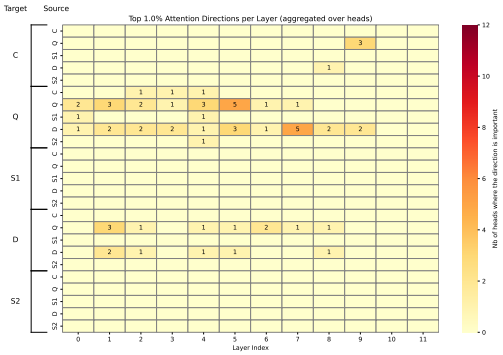


(e) Ablation scheme I_M for MonoBERT-large.

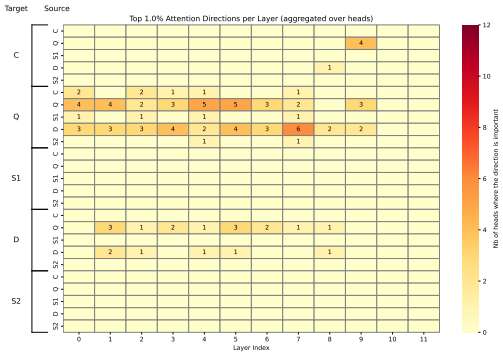


(f) Ablation scheme F_M for MonoBERT-large.

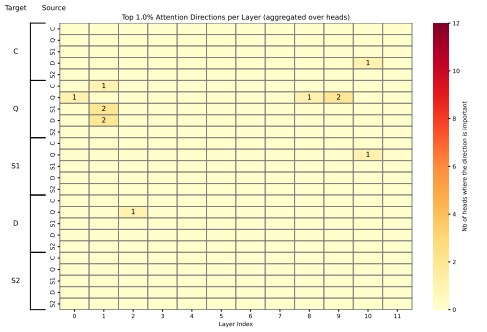
Fig. 19. Distribution of the most important attention heads, relative to each input part, per layer, following the ablation schemes I_M and F_M at pruning percentage $t = 1\%$.



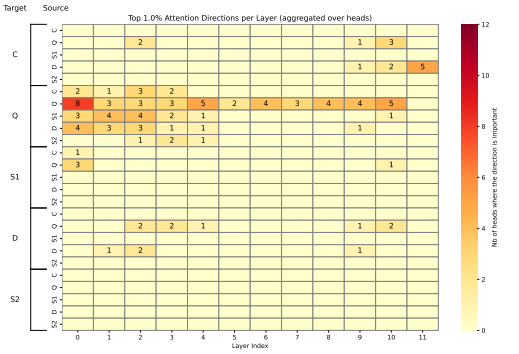
(a) Ablation scheme I_A for MiniLM.



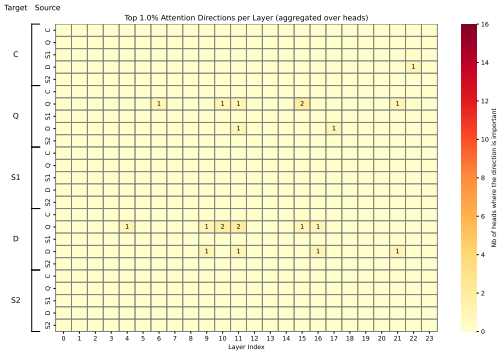
(b) Ablation scheme F_A for MiniLM.



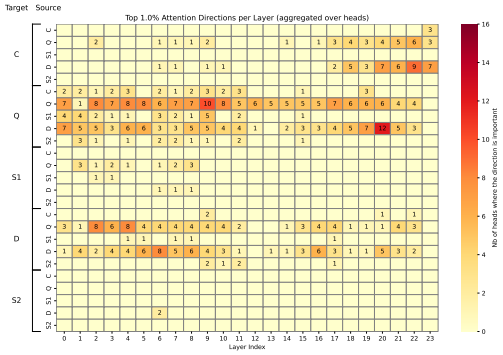
(c) Ablation scheme I_A for MonoBERT-base.



(d) Ablation scheme F_A for MonoBERT-base.



(e) Ablation scheme I_A for MonoBERT-large.



(f) Ablation scheme F_A for MonoBERT-large.

Fig. 20. Distribution of the most important directions across attention heads per layer for ablation schemes F_A and I_A at pruning percentage $t = 1\%$.

E Background on the Information Bottleneck approach

In the Information Bottleneck framework [78], the objective is to restrict as much as possible the flow of information using so-called "bottlenecks" without hindering the performance of the model too much. By design, these bottlenecks are masks, used to prevent the model from using some dimensions from its hidden states. From [45], the information

bottleneck in IR can be summarized as follows:

$$T = \mu \odot X + (\mathbf{1} - \mu) \odot \epsilon \tag{3}$$

where T is the output representation, $X \in \mathbb{R}^o$ represents an output of the model of dimension o, ϵ represents the noise, $\mathbf{1} \in \mathbb{R}^o$ is the all-one vector and $\mu \in \mathbb{R}^o$ is the weight vector controlling the balance between noise and the original signal. They constrain $\mu_i \in [0, 1]$ by setting $\mu_i = \sigma(\alpha_i)$, where σ is the sigmoid function and the α_i are the learned parameters. The training process starts from a frozen re-ranker such as MonoBERT, with α_i initialized to 5 for each dimension i, so that $\mu_i = 0.99$ and $T \approx X$ at the beginning of the training.

E.1 Application to our study

Unlike them, we are not interested in optimizing α for the hidden dimensions but instead for the attention score matrices. Following the setup introduced in the previous parts, we study the different directions between the 5 input parts and try to learn which directions, in every attention head, can be masked without hindering the performance of the models on the IR task. To do so, we reformulate the experimental framework of Jiang et al. [45] to have, for each attention head h at the layer ℓ in the model, a vector $\mu^{h,\ell} \in \mathbb{R}^{25}$, where the 25 rows correspond to the 25 possible directions[16], i.e., $\mu^{h,\ell}_{t \leftarrow s} \in \mathbb{R}$, with s and t respectively the source and target input parts, in an attention score matrix for the IR task (see Figure 5). The equation E.1 below illustrates the relationship that exists between $\mu^{h,\ell}$ and the different sections of the corresponding attention scores matrix $A^{h,\ell}$.

$$A^{h,\ell}_{final} = A^{h,\ell}_{original} \odot \begin{pmatrix} \mu^{h,\ell}_{\texttt{[CLS]}\leftarrow\texttt{[CLS]}} & \mu^{h,\ell}_{\texttt{[CLS]}\leftarrow Q} & \cdots & \mu^{h,\ell}_{\texttt{[CLS]}\leftarrow Q} & \mu^{h,\ell}_{\texttt{[CLS]}\leftarrow\texttt{[SEP1]}} & \mu^{h,\ell}_{\texttt{[CLS]}\leftarrow D} & \cdots & \mu^{h,\ell}_{\texttt{[CLS]}\leftarrow D} & \mu^{h,\ell}_{\texttt{[CLS]}\leftarrow\texttt{[SEP2]}} \\ \mu^{h,\ell}_{Q\leftarrow\texttt{[CLS]}} & \cdots & \ddots & & & & & & \vdots \\ \vdots & & & & & & & & \\ \mu^{h,\ell}_{Q\leftarrow\texttt{[CLS]}} & \cdots & & & & \ddots & & & \\ \vdots & & & & & & & & \vdots \\ & & \cdots & & & & & & \mu^{h,\ell}_{\texttt{[SEP2]}\leftarrow\texttt{[SEP2]}} \end{pmatrix}$$

We fix the maximum input length to 128 tokens in the experiments. Importantly, we do obtain the vector $\mu^{h,\ell}$ by using the Gumbel-Softmax trick on $\alpha^{h,\ell}$. Finally, our loss is composed of a distillation loss term (in our case based on MSE), where the teacher and student are the frozen re-ranker itself, with or without the masks μ, and of a regularization term over the values of α to enforce sparsity:

$$\mathcal{L} = \underbrace{\frac{1}{N} \left\| \mathbf{z} - \mathbf{z}_\mu \right\|_2^2}_{\mathcal{L}_{\text{distill}}} + \lambda \underbrace{\sum_{\ell \in L} \sum_{h \in H} \left\| \boldsymbol{\sigma}(\alpha^{h,\ell}) \right\|_1}_{\mathcal{L}_{\text{reg}}} \tag{4}$$

with z the original score, z_μ the score when the masks are applied, λ the coefficient of the regularization term, and L and H respectively the maximum number of layers in the model and the maximum number of attention heads per layer.

[16] In practice, there is 25 + 1 dimensions in total, to account for [PAD] tokens, but we omit it in our explanations for the sake of simplicity.

E.2 Experimental details

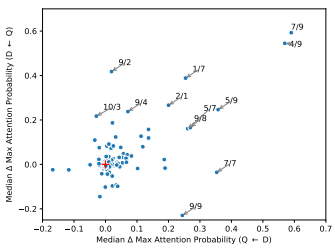
We apply the same set of hyper-parameters to every model. This includes the weight of the regularization term set to $\lambda = 10$ for every model. In particular, for every dimension i, j, $\alpha_{i,j}^{h,\ell}$ is initialized to 3 ($\sigma(3) \approx 0.95$) and the learning rate is 1. The mask $\mu^{h,\ell}$ is computed as $\mu^{h,\ell} = \text{GS}(\alpha^{h,\ell}, \tau)$, with $\tau = 1.0$ [17]. To stay consistent with the dataset used during their fine-tuning, we rely on the train set of MS MARCO [66].

In practice, the learned masks preserve only $\approx$ 9% of the directions for MiniLM, $\approx$ 15% for MonoBERT-base, and $\approx$ 21% for MonoBERT-large.

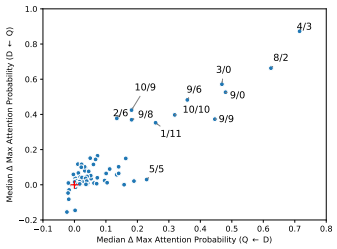
[17] When testing the models, we remove the Gumbel noise and apply a softmax function on the weights $\alpha_{i,j}^{h,\ell}$. To binarize the masks, we manually determined a threshold t above which the direction is kept to maximize the sparsity while conserving most of the original model performance. We set $t = 0.95$, $t = 0.97$ and $t = 0.99$ respectively for MiniLM, MonoBERT-base and MonoBERT-large.

F Attention patterns

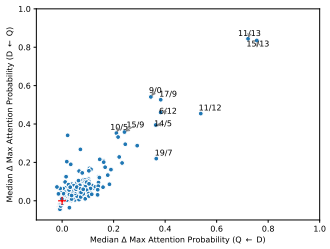
F.1 Attention patterns analysis over all the datasets



(a) Results over all the datasets for MiniLM.

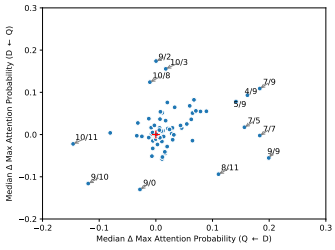


(b) Results over all the datasets for MonoBERT-base.

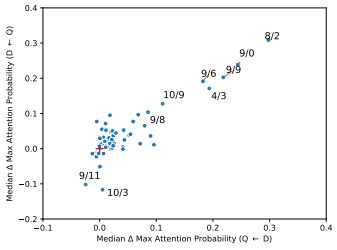


(c) Results over all the datasets for MonoBERT-large.

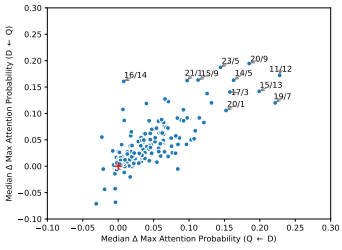
Fig. 21. Median of the differences in the attention weights $M^{h,\ell}_{Q,D/1} - M^{h,\ell}_{Q,D/-1}$ and $M^{h,\ell}_{D,Q/1} - M^{h,\ell}_{D,Q/-1}$ for each attention head.



(a) Results over all the datasets for MiniLM.



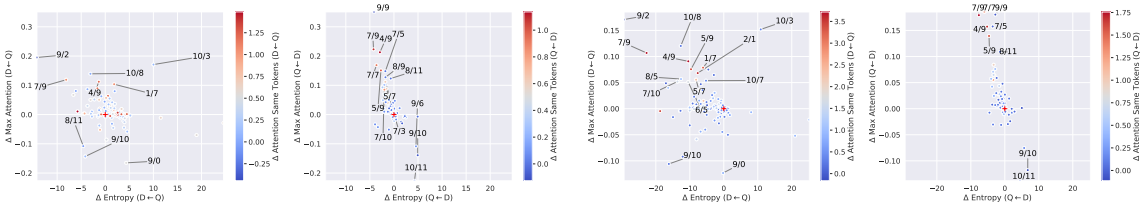
(b) Results over all the datasets for MonoBERT-base.



(c) Results over all the datasets for MonoBERT-large.

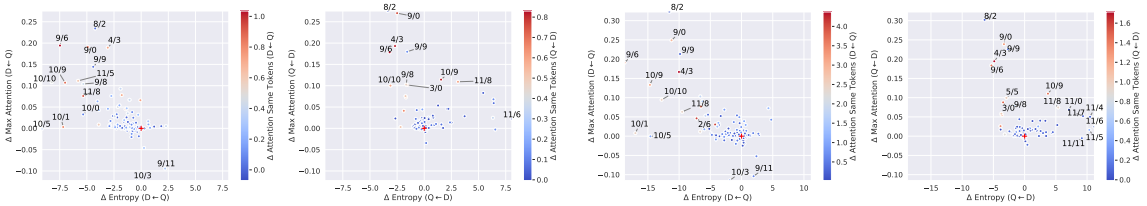
Fig. 22. Median of the differences in the attention weights $M^{h,\ell}_{Q,D/1} - M^{h,\ell}_{Q,D/0}$ and $M^{h,\ell}_{D,Q/1} - M^{h,\ell}_{D,Q/0}$ for each attention head.

F.2 Analysis of the matching signals over hard negatives



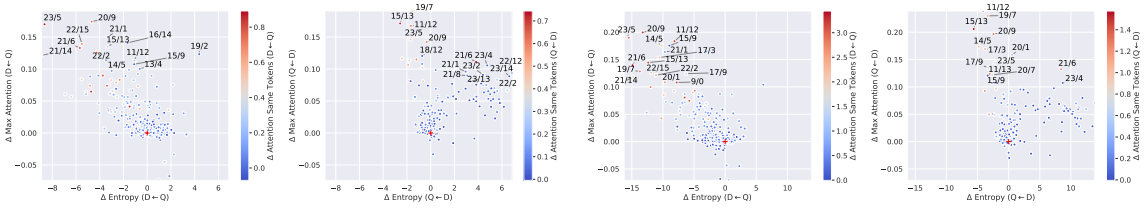
(a) MiniLM on TREC-DL (hard).

(b) MiniLM on OOD (hard).



(c) MonoBERT-base on TREC-DL (hard).

(d) MonoBERT-base on OOD (hard).



(e) MonoBERT-large on TREC-DL (hard).

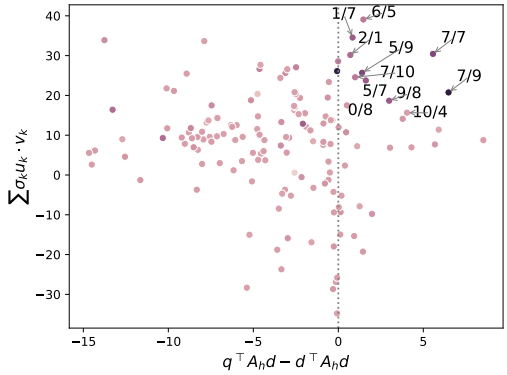
(f) MonoBERT-large on OOD (hard).

Fig. 23. Comparison of the matching patterns of the attention heads over the TREC-DL and OOD datasets, contrasting $r = 1$ with $r = 0$ (hard negatives). Same axes as Figure 14: each point is an attention head ℓ/h; left panel $D \leftarrow Q$, right panel $Q \leftarrow D$; x-axis: median of the differences in entropy; y-axis: median of the difference in maximum attention; color: median of the difference in the probability of attending to a token identical to the source token.

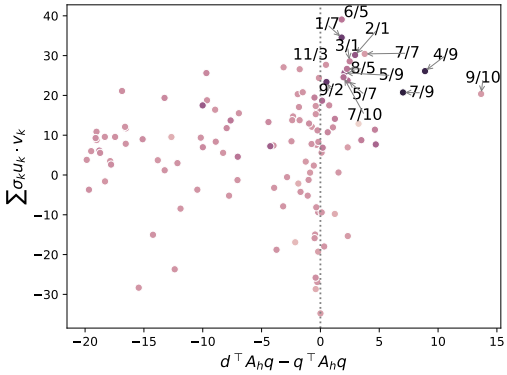
F.3 Unveiling the mechanism behind matching attention heads

From Section 6, we now understand the role of the most important attention heads better, yet the mechanism(s) underlying the detection of matching signals remain elusive. From our analysis on the attention weight matrices, we can hypothesize that for a query token (resp. document token) to be matched with a document token (resp. query token), attention heads' sub-operations need to meet certain criteria. Following the mechanistic interpretability target to break down and explain, in a human-understandable manner, the behavior of each component inside neural networks, we study the mechanisms hidden behind the detection of query-document matching by attention heads. To do so, we follow the work of Elhage et al. [30] and decompose the operations inside each attention head into the Query-Key matrix and the Output-Value matrix.

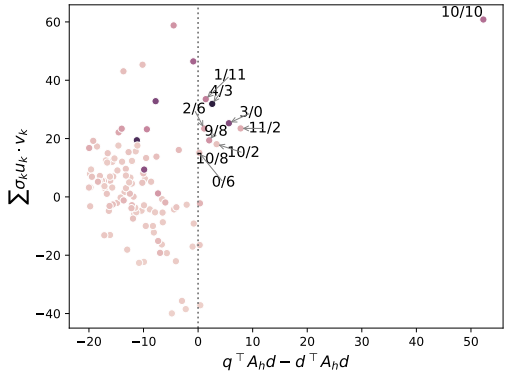
From the self-attention mechanism [54], we get that for any attention head h at a layer ℓ, the attention logits are obtained from the bilinear form defined by the Query-Key matrix $W_{QK}^{h,\ell} = \left(W_Q^h\right)^\top W_K^h$ (see Section 3.1). The logit of the probability that token i in the input attends to token j is $E_i^\top W_{QK}^{h,\ell} E_j$, where E_i and E_j denote the representations



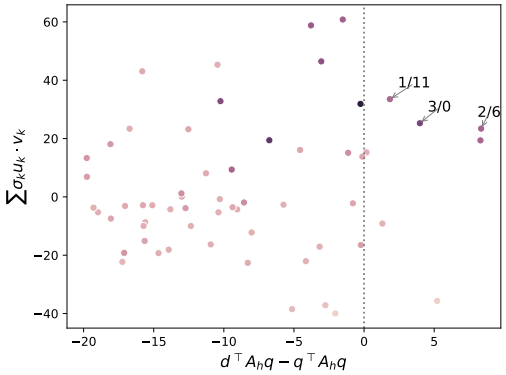
(a) Details for MiniLM from $Q \leftarrow D$.



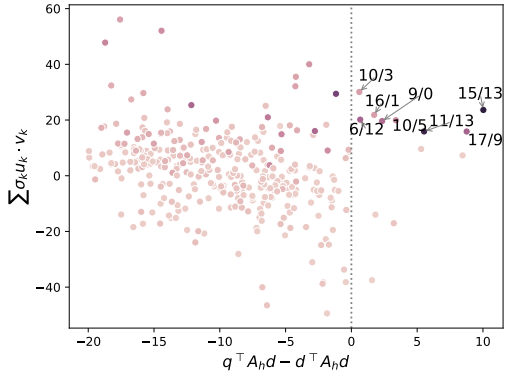
(b) Details for MiniLM from $D \leftarrow Q$.



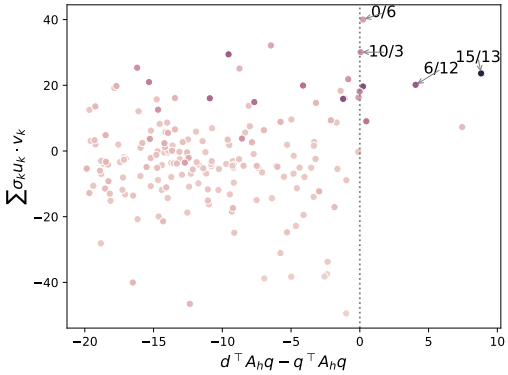
(c) Details for MonoBERT-base from $Q \leftarrow D$.



(d) Details for MonoBERT-base from $D \leftarrow Q$.



(e) Details for MonoBERT-large from $Q \leftarrow D$.



(f) Details for MonoBERT-large from $D \leftarrow Q$.

Fig. 24. Distribution of the attention heads (layer ℓ / head h) based on their Singular Value Decomposition and their ability to do cross-contextualization. Color levels relate to the contrasts between $r = 1$ and $r = -1$ on all the datasets (see Figures 21a and 21c). In the axis labels, A_h stands for the Query-Key matrix W^h_{QK}, and q and d for the mean query and document representations $\bar{q}$ and $\bar{d}$.

of these two tokens at the input of the layer. Suppose now that the representation of the i^{th} token of the query and the j^{th} token of the document can be expressed as $E_i = \bar{q} + x_i + a_i$ and $E_j = \bar{d} + y_j + b_j$, where $\bar{q}$ and $\bar{d}$ respectively represent the part of the embedding of queries and documents shared by every query and document[18], x_i and y_j their semantic/lexical information, and a_i and b_j some residual information. Then, the attention logit for E_i on E_j ($Q \leftarrow D$) can be reformulated as:

$$(E_i)^\top W_{QK}^{h,\ell} E_j = (\bar{q} + x_i + a_i)^\top W_{QK}^{h,\ell} (\bar{d} + y_j + b_j) \tag{5}$$

$$= \bar{q}^\top W_{QK}^{h,\ell} \bar{d} + x_i^\top W_{QK}^{h,\ell} y_j + \epsilon \tag{6}$$

with ϵ, a residual that, hopefully, is close to 0 for "matching detection" heads. Note that we can obtain a similar reformulation for the attention logit E_j on E_i ($D \leftarrow Q$). In the following, we develop our reasoning based on the direction $Q \leftarrow D$, but it also applies to $D \leftarrow Q$.

From that formulation, we hypothesize that for $Q \leftarrow D$ matches to happen, $\bar{q}^\top W_{QK}^h \bar{d} - \bar{d}^\top W_{QK}^h \bar{d} > 0$. To validate this hypothesis, we plot the difference $\bar{q}^\top W_{QK}^h \bar{d} - \bar{d}^\top W_{QK}^h \bar{d}$ (which should be greater than 0 when query-document matching is sought) against the propensity of matching tokens with the same meaning, i.e., where x_i and y_j are close. We evaluate the second condition by measuring the similarity of the left and right singular vectors of the Singular Value Decomposition of W_{QK}^h. Specifically, with $W_{QK}^h = \sum \sigma_k u_k v_k^\top$, we compute $\sum_k \sigma_k u_k \cdot v_k$. From Figure 24, we can see that the heads identified in Table 3 tend to follow our hypothesis, i.e., they favor cross-attention rather than self-attention, and their left and right singular vectors are more aligned.

Yet, we can see that our hypothesis is not able to correctly predict the behavior of all the "matching heads" (not all of them appear in the upper right area of each plot), so more work is needed to refine it.

[18] In practice, $\bar{q}$ and $\bar{d}$ are the average representations of a query and a document, computed over multiple queries and passages.

G Linear Discriminant Analysis

G.1 Description of the computation process

In practice, to compute the LDA, we start by computing the outputs of the attention and FFN modules across the layers of the cross-encoders while distinguishing the sets of query-passage pairs $\mathcal{S}_r$ with $r \in \{-1, 0, 1\}$:

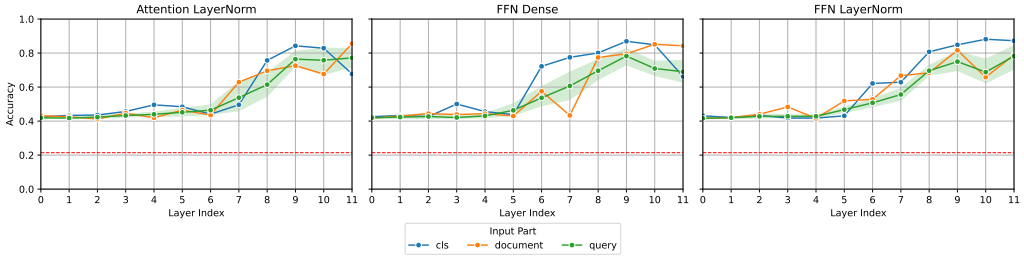
$$\Sigma_r^{\ell,m}(p) = \frac{1}{|\mathcal{S}_r|} \sum_{s \in \mathcal{S}_r} (P(s) - 0.5)\ \mathbf{x}_p^{(\ell,m)}(s)\, \mathbf{x}_p^{(\ell,m)}(s)^\top \tag{7}$$

with $P(s)$ the probability of relevance predicted for the input s, $p \in \{\texttt{[CLS]}, Q, \texttt{[SEP1]}, D, \texttt{[SEP2]}\}$ the input part, $m \in \{a, \tilde{f}, \emptyset\}$ the activation (see Section 3.1), ℓ the layer it belongs to, and $\mathbf{x}_p^{(\ell,m)}(s)$ the value of this activation for the sample s. In the formula, we also re-weight the product by the predicted probability of relevance of each query-passage pair, centered at 0.5. Note that for the query Q, we work at the level of its first $k = 6$ tokens, separately, $Q_1, ..., Q_k$, while for the document D, we work with an averaged representation, per sample, $\bar{D}$, and not directly with their individual token representations.

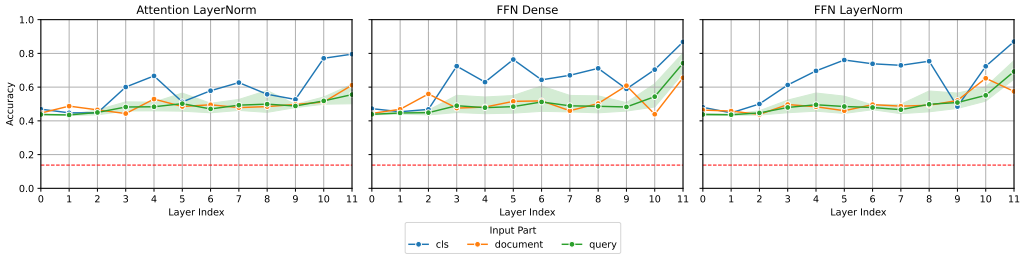
We then compute the between-class and within-class covariance matrices over the relevance levels and use them to solve a generalized version of the eigenvalue problem[19]. As a result, we order the eigenvectors by their corresponding eigenvalues to maximize the class separability. In our case, and for the sake of visualization, we proceed to retain only the first two axes and project onto them the activations previously generated to visualize a potential structure underlying the cross-encoder models' internal representations.

[19] Given that we are trying to solve a generalized version of the eigenvalue decomposition with two matrices A and B, B different from the identity matrix, we use the Scipy implementation which supports this version of the eigenvalue problem: https://docs.scipy.org/doc/scipy/reference/generated/scipy.linalg.eigh.html.

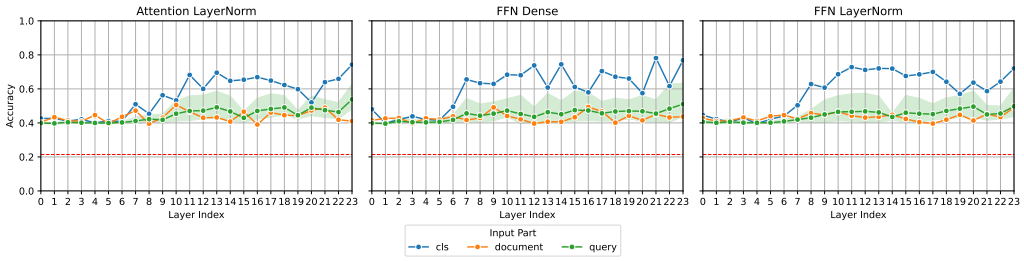
G.2 Evolution of the classification accuracy for OOD datasets



(a) Details for MiniLM.



(b) Details for MonoBERT-base.



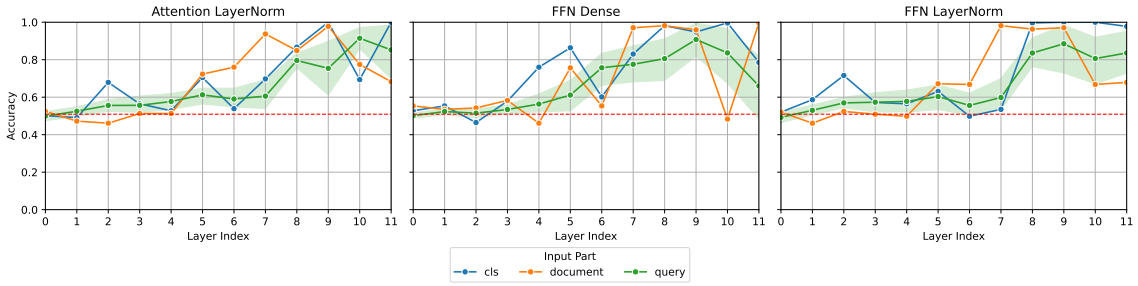
(c) Details for MonoBERT-large.

Fig. 25. Evolution of the classification accuracy of the activations $\mathbf{x}_p^{(\ell,a)}$, $\mathbf{x}_p^{(\ell,\tilde{f})}$ and $\mathbf{x}_p^{(\ell)}$ projected along the LDA axis, by their original relevance label ($r \in \{-1, 0, 1\}$), per input part $p \in \{\texttt{[CLS]}, Q_1, ..., Q_k, \bar{D}\}$, $k = 6$. LDA and classification are computed over samples from the OOD datasets.

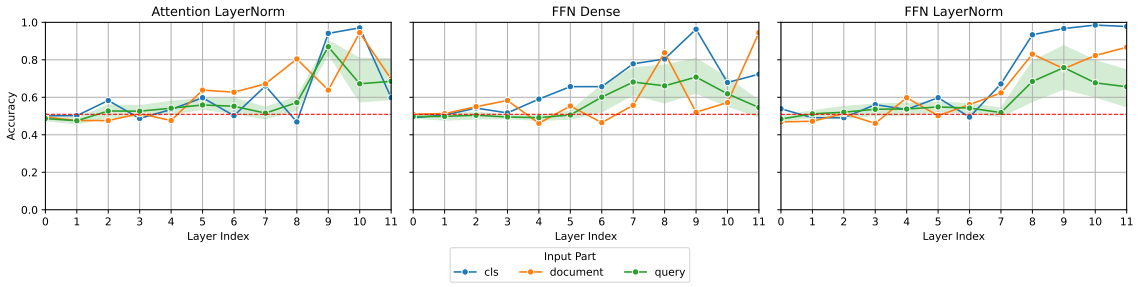
G.3 Additional figures

We provide here, for reference, the evolution of the classification accuracy when we limit the LDA (see Section 7) to two classes: $r = 1$ and either $r = -1$ or $r = 0$ for both MiniLM (Figure 26) and MonoBERT (Figures 27 and 28).

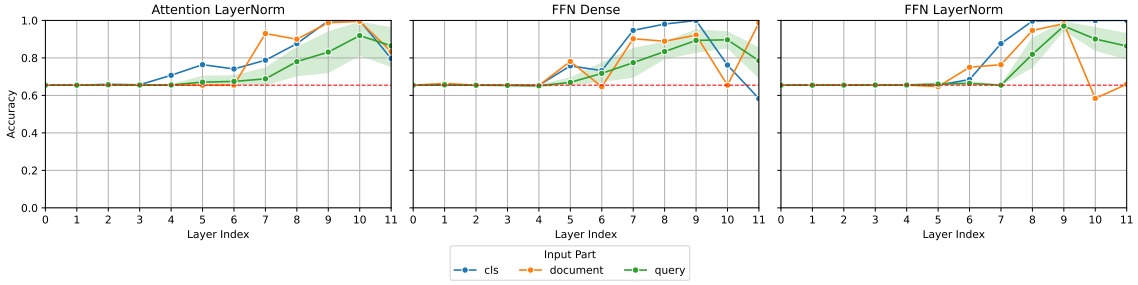
We first observe that, for all three models, distinguishing the activations of query-relevant passage pairs from pairs with easy negatives seems easier than from pairs with hard negatives. In addition, we observe that when the domain is different from the training domain, this difference is exacerbated (Figures 26d and 28d).



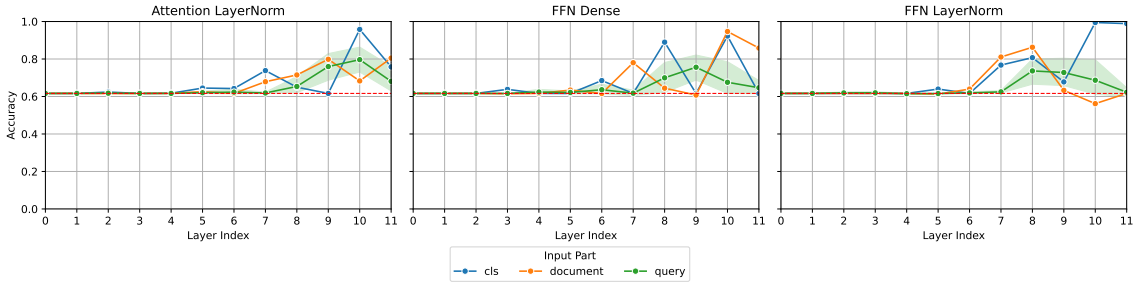
(a) Details on MS MARCO datasets, relevance levels $r = -1$ and $r = 1$.



(b) Details on MS MARCO datasets, relevance levels $r = 0$ and $r = 1$.

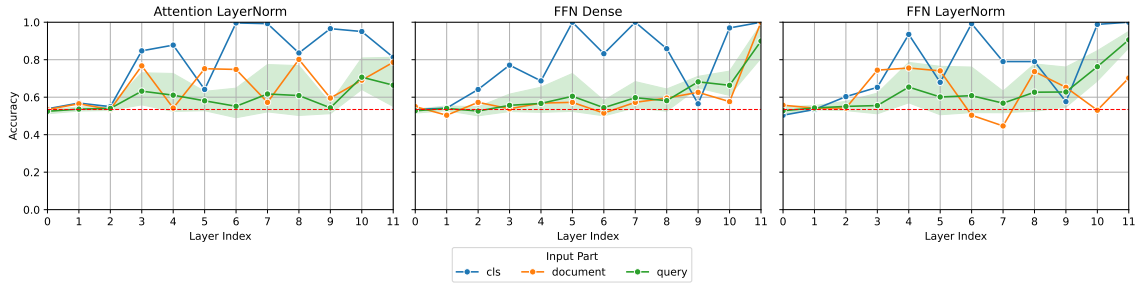


(c) Details on OOD datasets, relevance levels $r = -1$ and $r = 1$.

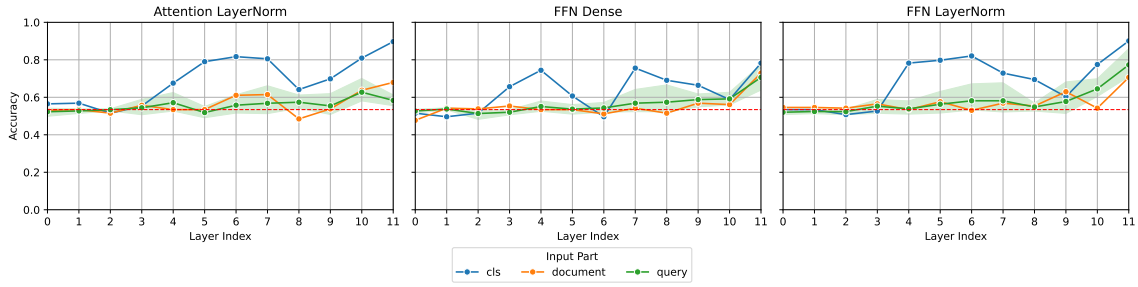


(d) Details on OOD datasets, relevance levels $r = 0$ and $r = 1$.

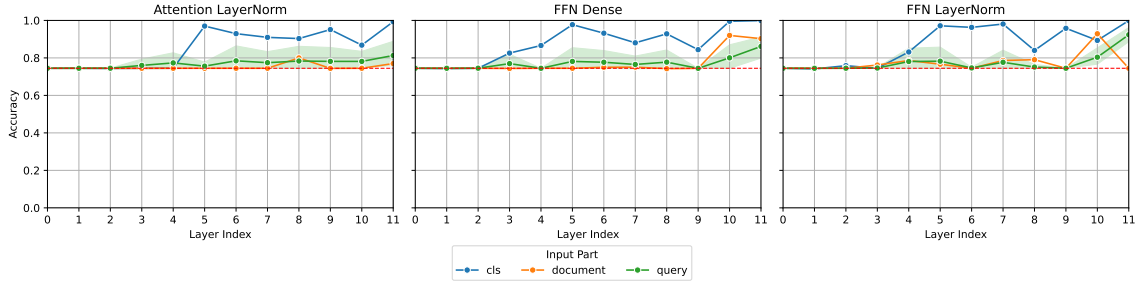
Fig. 26. Evolution of the classification accuracy, per input part and modules of MiniLM, over either the MS MARCO or the OOD datasets, for relevance levels $r = 1$ and $r \in \{-1, 0\}$. On each plot, the red line corresponds to the proportion of the largest class.



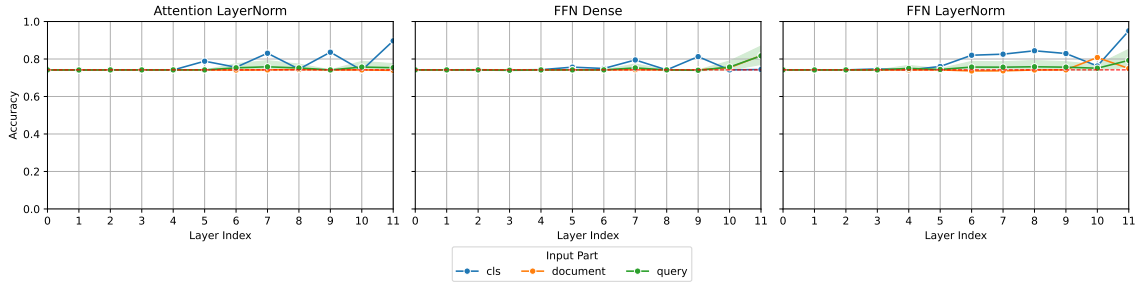
(a) Details on MS MARCO datasets, relevance levels $r = -1$ and $r = 1$.



(b) Details on MS MARCO datasets, relevance levels $r = 0$ and $r = 1$.

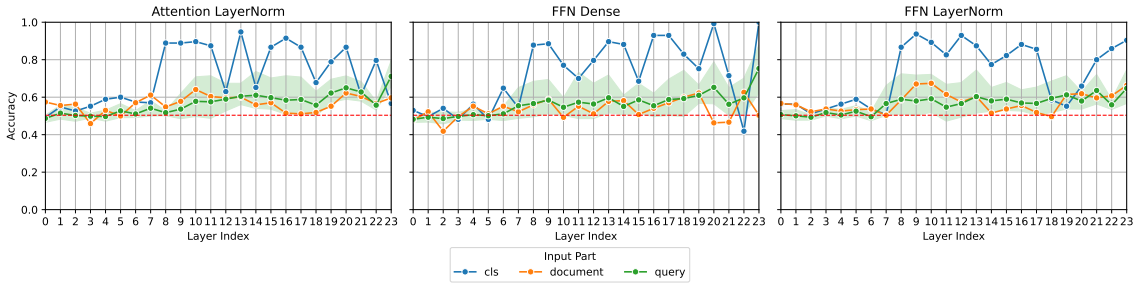


(c) Details on OOD datasets, relevance levels $r = -1$ and $r = 1$.

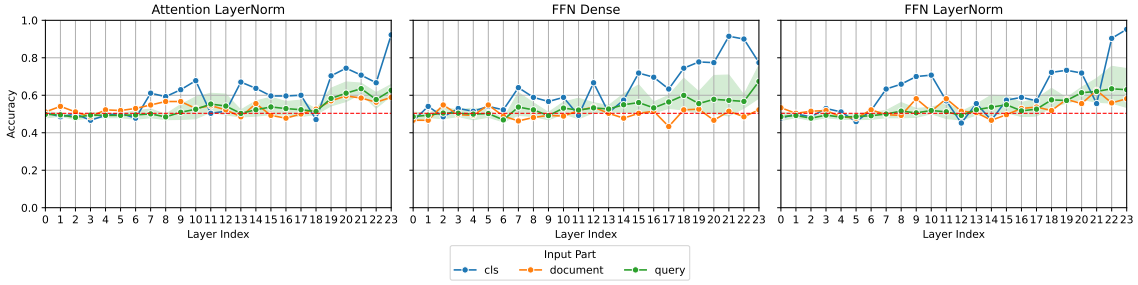


(d) Details on OOD datasets, relevance levels $r = 0$ and $r = 1$.

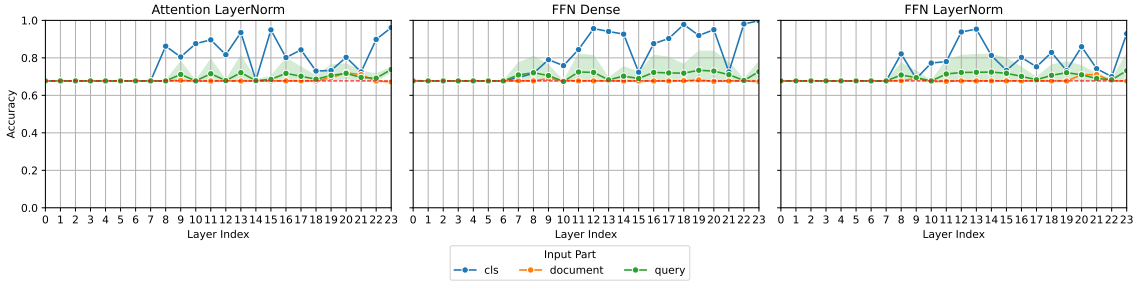
Fig. 27. Evolution of the classification accuracy, per input part and modules of MonoBERT-base, over either the MS MARCO or the OOD datasets, for relevance levels $r = 1$ and $r \in \{-1, 0\}$. On each plot, the red line corresponds to the proportion of the largest class.



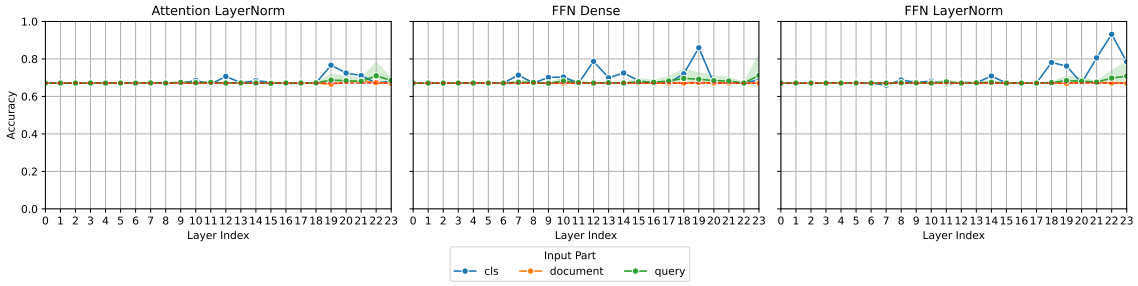
(a) Details on MS MARCO datasets, relevance levels $r = -1$ and $r = 1$.



(b) Details on MS MARCO datasets, relevance levels $r = 0$ and $r = 1$.



(c) Details on OOD datasets, relevance levels $r = -1$ and $r = 1$.



(d) Details on OOD datasets, relevance levels $r = 0$ and $r = 1$.

Fig. 28. Evolution of the classification accuracy, per input part and modules of MonoBERT-large, over either the MS MARCO or the OOD datasets, for relevance levels $r = 1$ and $r \in \{-1, 0\}$. On each plot, the red line corresponds to the proportion of the largest class.