

Understanding Ranking LLMs: A Mechanistic Analysis for Information Retrieval

Tanya Chowdhury
University of Massachusetts Amherst
MA, USA

Atharva Nijasure
University of Massachusetts Amherst
MA, USA

James Allan
University of Massachusetts Amherst
MA, USA

ABSTRACT

Transformer networks, particularly those achieving performance comparable to GPT models, are well known for their robust feature extraction abilities. However, the nature of these extracted features and their alignment with human-engineered ones remain unexplored. In this work, we investigate the internal mechanisms of state-of-the-art, fine-tuned LLMs for passage reranking. We employ a probing-based analysis to examine neuron activations in ranking LLMs, identifying the presence of known human-engineered and semantic features. Our study spans a broad range of feature categories, including lexical signals, document structure, query-document interactions, and complex semantic representations, to uncover underlying patterns influencing ranking decisions.

Through experiments on four different ranking LLMs, we identify statistical IR features that are prominently encoded in LLM activations, as well as others that are notably missing. Furthermore, we analyze how these models respond to out-of-distribution queries and documents, revealing distinct generalization behaviors. By dissecting the latent representations within LLM activations, we aim to improve both the interpretability and effectiveness of ranking models. Our findings offer crucial insights for developing more transparent and reliable retrieval systems, and we release all necessary scripts and code to support further exploration.

ACM Reference Format:

Tanya Chowdhury, Atharva Nijasure, and James Allan. 2018. Understanding Ranking LLMs: A Mechanistic Analysis for Information Retrieval. In . ACM, New York, NY, USA, 11 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

For many decades, the domain of passage retrieval and reranking has predominantly used algorithms grounded in statistical human-engineered features, typically derived from the query, the document set, or their interactions. In particular, the features in training and evaluation datasets such as the learning-to-rank MSLR dataset [42] are largely derived from features manually developed to support successful statistical ranking systems. However, recent advances have seen state-of-the-art passage retrieval and ranking algorithms increasingly pivot to neural network-based models. The introduction of Large Language Models (LLMs) has notably widened the

performance disparity between neural-based and traditional statistical rankers.

Access to open-source ranking models, particularly those exhibiting performance comparable to GPTs, provides a unique opportunity to explore the inner workings of transformer-based architectures. Neural networks are recognized for their robust feature extraction capabilities; however, the representation of these features remains elusive, making it challenging to discern their nature and potential correlation with features engineered by humans. As a result, despite their efficacy, neural networks present a challenge in terms of transparency: their feature representations are complex and the location of critical features within the network remains obscure. In this context, mechanistic interpretability [44] aims to demystify the internal workings of transformer architectures, showing how LLMs process and learn information differently compared to traditional methods. Our research is motivated by the desire to determine whether the statistical features traditionally valued in algorithms like BM25 and tf*idf are somehow encoded within LLM architectures. This study stands to bridge the gap between neural and statistical methodologies, by using probes to generate hypotheses on the inner workings of the LLMs, offering insights that could enhance the field of information retrieval by enabling more intuitive explanations and better support for model design and analysis.

Summary: This study is built upon four different LLM architectures : Llama2-7b, Llama2-13b, Llama3.1-8b, and Pythia-6.9b. For each of these we use or create a LoRa fine-tuned variant (e.g., RankLlama [35]), optimized for passage reranking tasks using the MS Marco dataset. These point-wise rankers demonstrate substantial accuracy improvements over their non-LLM counterparts. Our analysis concentrates on extracting activations from the MLP unit of each transformer block, which is posited to contain the key feature extractors [24]. We aggregate these activations for each input sequence (query-document pairs) and assign labels to these pairs corresponding to each feature that is a target of our probe. Subsequently, we employ a regression with regularization to correlate these activations with the labels. Our findings reveal a pronounced representation of certain MSLR features within the ranking LLMs, while others are markedly absent. We also observe that distinct LLM architectures capture very similar statistical IR features, when fine-tuned using the same dataset. These insights will allow future researchers to formulate hypotheses concerning the underlying circuitry of ranking LLMs.

Research Questions: In this study, we aim to explore several internal mechanistic aspects of ranking LLMs through probing techniques. Specifically, we seek to determine whether known statistical information retrieval (IR) features are present within the activations of LLMs. We are also interested in identifying groups of features and understanding how they may combine or interact

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.
Conference'17, July 2017, Washington, DC, USA

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
<https://doi.org/XXXXXXX.XXXXXXX>

within the LLM’s activations. Additionally, we investigate whether LLMs contain components that mimic similarity scores from models like BERT or T5. It is also of interest to find out if different LLM architectures encode the same statistical IR features within their activations. Finally, this inquiry extends to examining whether the latent features encoded within activations of LLMs remain consistent or change when the model encounters out-of-distribution queries or documents. By answering these questions, we aim to gain a deeper understanding of the inner workings of ranking LLMs and the extent to which they align with traditional IR methodologies.

1.1 Contributions and Findings

- (1) We discovered that several human-engineered metrics, such as *covered query term number*, *covered query term ratio*, *min of term frequency*, *min and mean of stream length normalized term frequency*, and *variance of $tf*idf$* , are prominently encoded within LLM activations. In contrast, certain features, including *sum of stream length normalized term frequency*, *max of stream length normalized term frequency*, and *BM25*, show no discernible representation in LLM activations.
- (2) We fine-tuned and probed two additional reranking models, Llama3.1-8b and Pythia-6.9b, resulting in RankLlama3-8b and RankPythia, alongside probing RankLlama2-7b and RankLlama2-13b. The results revealed broadly consistent outcomes across all models, suggesting that different LLM architectures, when fine-tuned similarly, encode latent features in a comparable way within their activations.
- (3) We found that the activation patterns of the listed features remain consistent even when the model encounters out-of-distribution queries or documents on RankLlama2-7b, RankLlama3-8b and RankPythia-6.9b. However, they do not remain consistent for certain features of RankLlama2-13b, suggesting potential overfitting in the LLM during fine-tuning.
- (4) We identified specific combinations of MSLR features, like the sum of *covered query term ratio*, *mean stream length normalized term frequency*, and *variance of $tf*idf$* , along with the sum’s square and cube, all of which also exhibit a strong correlation with LLM activations.
- (5) Finally, we show further evidence on the presence of these encoded features using Shapley values. All scripts, datasets, and fine-tuned ranking LLMs developed and/or used in this study are made publicly available.¹

2 BACKGROUND & RELATED WORK

We discuss concepts in Mechanistic Interpretability, in particular sparse probing. Then we touch on inner interpretability approaches in Information retrieval.

2.1 Mechanistic Interpretability & Sparse Probing

Mechanistic Interpretability is the ambitious goal of gaining an algorithmic-level understanding of any deep neural network’s computations [44]. This goal can be achieved in numerous ways, some of them being by studying weights (e.g., weight masking [12, 50],

by continual learning [15]), by studying individual neurons (e.g., excitement based [52], activation patching [37], gradient-based [2], perturbation and ablation based [53]), by studying subnetworks (e.g., sparsity based [39], circuit analysis based [20]), by studying representations (e.g., tokens [33], attention [11], probing [4, 7, 26]), by training sparse auto encoders [13, 31, 38] etc. In this study, we focus on studying the activations of individual neurons and also some groups of neurons with the help of probing regressors. To that end, we use concepts from a host of techniques mentioned above.

While the concept of reverse-engineering specific neurons within large language models (LLMs) is relatively new, existing studies [23, 24] illustrate that the feed-forward layers of transformers, comprising two-thirds of the model’s parameters, function as key-value memories. These layers activate in response to specific inputs, a mechanism we aim to demystify by reverse-engineering the activation function of these neurons. A notable challenge in this endeavor is the phenomenon of superposition, where early layers in LLMs select and store a vast array of features—often exceeding the number of available neurons—in a linear combination across multiple neurons. In contrast, later layers tend to focus on more abstract features, discarding those deemed non-essential [25].

Probing aims to determine if a given representation effectively captures a specific type of information, as discussed by Belinkov [4]. This technique employs transfer learning to test whether embeddings contain information pertinent to a target task. The three essential steps in probing include: (1) obtaining a dataset with examples that exhibit variation in a particular quality of interest, (2) embedding these examples, and (3) training a model on these embeddings to assess if it can learn the quality of interest. This method is versatile as it can utilize any inner representation from any model. However, a limitation of probing methods is that a successful probe does not necessarily mean that the probed model actually utilizes that information about the data [45]. Belinkov [4] provides a comprehensive survey on probing methods for large language models, discussing their advantages, disadvantages, and complexities. Gurnee et al. [25] further introduce sparse probing, where they mine features of interest over groups of representations up to k in size. We build upon sparse probing in this work.

2.2 Inner Interpretability in IR

Most works of interpretability in information retrieval [1] have been model extrinsic [9, 43, 47]. Among model intrinsic interpretability methods, some works focus on using gradient-based methods to identify important neurons [6, 19]. Although gradient based methods give an accurate perspective on the flow of information, they are too fine-grained to give a human-level understanding of the LLM’s inner circuit [1]. A number of works have attempted to examine the inner workings of neural retrievers to understand if they satisfy IR axioms and/or to spot known features. Fan et al. [18] probe fine-tuned BERT models on three different tasks, namely document retrieval, answer retrieval, and passage retrieval, to find out if these different forms of relevance lead to different models. Zhan et al. [51] study the attention patterns of BERT after fine-tuning on the document ranking task and make note of how BERT dumps

¹<https://suppressed.for.review>

Table 1: The Ranking effectiveness of various fine-tuned ranking LLMs, in comparison to the performance of a BM25 reranker. We observe that while LLMs are effecient for reranking, they are opaque and complex to understand.

Model Name	Base Model	DEV MRR@10	DL19 NDCG@10	DL20 NDCG@10
BM25	Lucene [29]	16.7	50.6	48.0
RankLLama2-7b	Llama2 [49]	44.9	75.6	77.4
RankLLama2-13b	Llama2 [49]	45.2	76.0	77.9
RankLlama3-8b	Llama3 [16]	42.8	74.3	72.4
RankPythia-6.9b	Pythia [5]	42.9	76.0	71.5

```

LlamaForSequenceClassification(
  (model): LlamaModel(
    (embed_tokens): Embedding(32000, 4096)
    (layers): ModuleList(
      (0-31): 32 x LlamaDecoderLayer(
        (self_attn): LlamaSdpaAttention(
          (q_proj): Linear(in_features=4096, out_features=4096, bias=False)
          (k_proj): Linear(in_features=4096, out_features=4096, bias=False)
          (v_proj): Linear(in_features=4096, out_features=4096, bias=False)
          (o_proj): Linear(in_features=4096, out_features=4096, bias=False)
          (rotary_emb): LlamaRotaryEmbedding()
        )
        (mlp): LlamaMLP(
          (gate_proj): Linear(in_features=4096, out_features=11008, bias=False)
          (up_proj): Linear(in_features=4096, out_features=11008, bias=False)
          (down_proj): Linear(in_features=11008, out_features=4096, bias=False)
          (act_fn): SiLU()
        )
        (input_layernorm): LlamaRMSNorm()
        (post_attention_layernorm): LlamaRMSNorm()
      )
    )
    (norm): LlamaRMSNorm()
  )
  (score): Linear(in_features=4096, out_features=1, bias=False)
)

```

Figure 1: The RankLlama2-7B internal architecture with 32 layers and 4096 dimensional vectors. Similarly, the 13B models contains 40 layers and 5120 dimensional vectors. Note the score layer, added to the general Llama2 architecture during fine-tuning using LoRa.

duplicate attention weights on high frequency tokens (like periods). Similar to them, Choi et al. [8] study attention maps in BERT and report the discovery of the IDF feature within BERT attention patterns. ColBERT investigations [21, 22] study its term-matching mechanism and conclude that it captures a notion of term importance, which is enhanced by fine-tuning. MacAvaney et al. [36] propose ABNIRML a set of diagnostic probes for neural retrieval models that allow searching for features like writing styles, factuality, sensitivity and word order in models like ColBERT and T5. Recently Parry et al. [41] provide a framework for diagnostic analysis and intervention within Neural Ranking models by generating perturbed inputs followed by activation patching.

These studies, however, have been conducted in the pre-LLM era and hence are much smaller in size. It is unknown if the findings from BERT will carry over to LLMs like Llama, given that BERT is an encoder-only model whereas most modern LLMs are decoder-only models. While most of the above works focus on attention heads and pattern, our work focuses on probing the MLP activation layers, which are now believed to be the primary feature extraction location within the LLM [25]. To the best of our knowledge, this is one of the first works towards study of neuron activations in LLMs for a large scale of statistical and semantic features.

3 IDENTIFYING CONTEXT NEURONS

We describe our probing pipeline to identify context neurons - neurons that are sensitive to or encode desired features - in the LLM architecture.

Ranking Models: For our Ranking LLM interpretability study, we first select RankLlama [35], an open source pointwise reranker that has been fine-tuned on Llama-2 with the MS MARCO dataset using LoRa. Given an input sequence (query Q , document set D), RankLlama reranks them as :

$$\begin{aligned}
 \text{input} &= \text{'query : } \{Q\} \text{ document : } \{D\} < /s > ' \\
 \text{Sim}(Q, D) &= \text{Linear}(\text{Decoder}(\text{input})[-1])
 \end{aligned}$$

where $\text{Linear}(\cdot)$ is a linear projection layer that projects the last layer representation of the end-of sequence token to a scalar. The model is fine-tuned using a contrastive loss function. RankLlama is accessible on Huggingface² and has demonstrated near state-of-the-art performance on the MS MARCO Dev set, as noted by Ma et al. [35]. We experiment on both the 7b and 13b parameter models fine-tuned for passage reranking.

To facilitate comparison with other ranking architectures, we additionally fine-tune Llama3.1-8b and Pythia-6.9b using an approach similar to the fine-tuning methodology employed by the RankLlama authors. Each of these LLMs is fine-tuned with a LoRa rank of 32 for 1 epoch, and their performance is evaluated on the commonly used TREC DL19 and DL20 datasets. Table 1 compares the ranking performance of all LLMs used in this study, indicating that, once fine-tuned, they achieve similar effectiveness.

Internal Architecture: We refer to the internal architecture of RankLlama2-7b in Figure 1. Each block of the Llama transformer architecture can be divided into two main groups: the Multi-Head self attention layers and the MLP layers. For example, the Llama-2 7b and 13b architectures consist of 32 and 40 such identical blocks, where each component has a dimensionality of 4096 and 5120 respectively. The feed-forward sublayer takes in the output of the multi-head attention layer and performs two linear transformations over it, with an element-wise non-linear function σ in between.

$$f_i^l = \mathbf{W}_v^l \times \sigma(\mathbf{W}_k^l a_i^l + b_k^l) + b_v^l$$

where a_i^l is the output of the MHA block in the l layer, and \mathbf{W}_v^l , \mathbf{W}_k^l , b_k^l , and b_v^l are learned weight and bias matrices.

This MLP layer applies pointwise nonlinearities to each token independently and therefore performs the majority of feature extraction for the LLM. Additionally, the MLP layers account for $\frac{2}{3}$ rd

²<https://huggingface.co/castorini/rankllama-v1-7b-lora-passage>

of total parameters. As a result, our study focuses on the value of the residual stream (also known as hidden state) for token t at layer l , right after applying the MLP layers. Elhage et al. [17] provide further details about the transformer architecture math.

Activation Sourcing: We set up PyTorch transformer forward hooks to mine activations from the ranking LLMs. We tokenize and feed the query-document pairs to the pointwise LLM discussed above, and capture activations corresponding to each input sequence across all layers. The dimension of each layer’s activation is the number of tokens times the number of hidden units in the LLM. This dimension is 4096 in Llama2-7b and 5120 in Llama2-13b. To reduce complexity of the computation, we aggregate activations across tokens in an input sequence within a layer. Following Gurnee et al. [25], we try out mean and max activation aggregation to ultimately obtain a 4096/5120 dimensional activation vector per layer. This corresponds to aggregated activations for 4096/5120 neurons in each layer, which is the focal point of our study. We quantize and store these activation tensors to be used later for further experiments. Intuitively, we have captured the internals of the model at this point, in which we can start searching for desired features.

Target Features: The MSLR dataset [42] provides a comprehensive list of features that have been historically recognized as highly effective for ranking models [27, 28]. Consequently, we aim to search for a subset of these MSLR features within the LLM activations. Specifically, we focus on mining the following 19 features from the MSLR dataset within the LLM (“stream” means the document or passage text): *Min TF*, *Min(TF/L)*, *Min TF*IDF*, *Covered QT Number*, *Covered QT Ratio*, *Mean(TF/L)*, *Variance TF*IDF*, *BM25*, *Mean TF*IDF*, *Variance(TF/L)*, *Mean TF*, *Variance TF*, *Sum TF*, *Max TF*, *Max(TF/L)*, *Stream length*, *Sum TF*, *Max TF*IDF* and *Sum TF*IDF*.

In addition to MSLR features, we mine for known query-document similarity metrics within the LLM architecture. For this we consider traditional query-document similarity scores like *tf*idf cosine scores*, *Euclidian score*, *Manhattan score*, *KL-divergence score*, *Jensen-Shannon divergence score* as well as popular semantic relevance metrics like *BERT* and *T5* scores.

Each feature is probed individually; for instance, the probe for stream length is dedicated solely to that feature without searching for any others. Although many of these features are known to be correlated and may share neurons, this study does not include correlation analysis amongst features.

Probing Datasets: To facilitate probing [46], we need to curate a dataset of input sequences to study the model. We select query-document pairs from the MS MARCO test set [40] for this purpose. For each query, we include documents that are highly relevant, highly irrelevant, and of intermediate relevance. Our input sequences for the ranking LLM consist of these query-document pairs. Note that it is important for each probing dataset to be balanced – i.e., contain a uniform number of samples across the range of values of the feature being probed [4, 25]. For instance, when analyzing the BM25 feature, our dataset included a balanced ratio of query-documents with both low and high BM25 values. Having an unbalanced probing dataset can lead to a biased analysis as the model might have a tendency to overfit the dominant class. For each input sequence, we calculate the expected value of the feature being studied. Consequently, we compute the values for each of the

19 MSLR features and 7 similarity scores for our query-document pairs.

Identifying context neurons: After collecting activations and feature labels across a wide range of input sequences (query-document pairs), we start the process of identifying context neurons. Unlike most prior research that employs classification-based probes [18, 36], we utilize ridge regression-based probes to capture the continuous nature of the features under study. We first split the activations and their corresponding labels into training, validate and testing sets (60:20:20).

Subsequently, we perform a layer-wise analysis. For each layer and feature, we employ a sparse probe by fitting the activation vectors of the training split to the corresponding feature labels. To maintain the sparsity of the context neurons, we use Lasso (L1 regularization) with $\alpha = 0.1$. We perform 5-fold cross validation to avoid overfitting and provide a more reliable estimate. After fitting the activations to the feature’s labels, we compute the R^2 score to measure how well the regression curve explains the variability of the labels. R^2 ranges from 0 to 1, with 1 indicating that the curve perfectly explains the variability in the data. A high R^2 score indicates the presence of a neuron that is sensitive to or activates for the feature being studied, whereas a negative R^2 confirms a negative correlation between the feature and the neuron’s activations.

Experimental Details: For our probing experiments, we select 500 queries from the MS MARCO dev set and retrieve the top 100 documents for each query from the MS MARCO collections corpus using a BM25 ranker. We then compute activations corresponding to each query-document pair using all four models. The results reported in the following section are after mean aggregation and quantization for efficient storage. The size of each probing dataset is greater than 5,000 to ensure incorporation of a wide range of values for the feature being probed. The 100 documents retrieved for a query serve as the query’s corpus for idf calculation. We use the bert-base and t5-base models for BERT and T5 score computations. We use the well-known Okapi implementation of BM25 in our experiments.

4 RESEARCH QUESTIONS

4.1 Statistical features within RankLlama2-7b

4.1.1 MSLR features. For the first set of experiments, we designed sparse probes for each of the selected MSLR features and conducted experiments on each layer of the RankLlama2-7b architecture. All experiments were run with both mean and max activation aggregation over the input sequence. Our findings (Figure 2) are categorized into two groups: (i) Features that exhibit a strong fit in certain layers, (ii) Features that do not correlate to the LLM activations in any layer

Positives: Features that achieve an R^2 score greater than 0.85 in any particular layer are considered strong positives. This means it is highly likely that the activation has extracted the particular feature within the neuron. We found that seven metrics, namely *Min of TF*, *Min(TF/L)*, *Min of TF*IDF*, *covered QT Number*, *covered QT Ratio*, *Mean(TF/L)*, and *variance of TF*IDF* frequently exhibit low Mean Square Errors and high R^2 scores. This indicates that there exist MLP neurons within the investigated layers that perform feature extractions similar to these MSLR features. Additionally, we

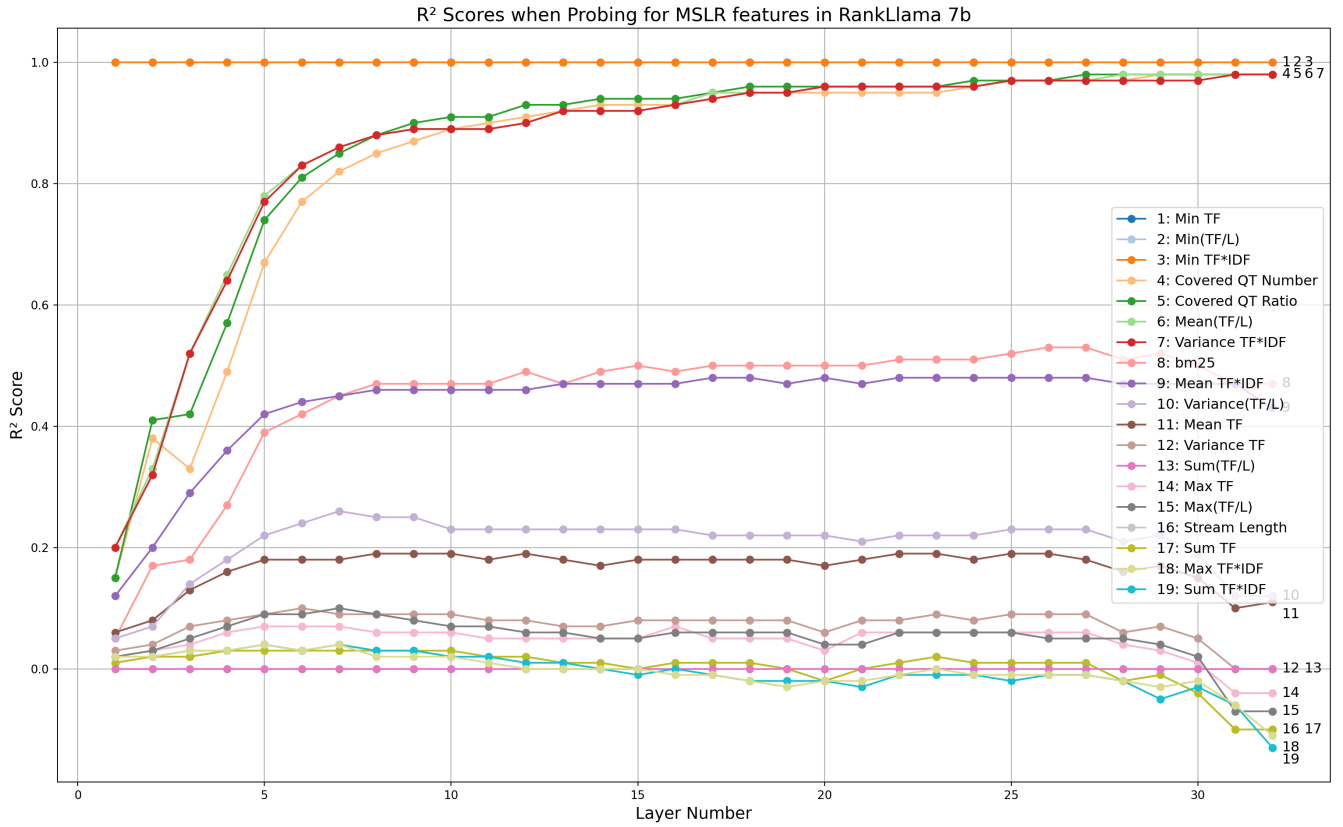


Figure 2: Probing for statistical features from the MSLR dataset in RankLlama2-7b model. Here *QT* stands for Query Term, *TF* stands for Term Frequency and */L* stands for length normalized. The graph lines indicate the presence of a particular feature along the layers of the LLM. Certain features like *Min TF * IDF* show consistent presence across the layers. Other features like *Covered QT Number*, *Covered QT Ratio*, *Mean(TF/L)* and *Variance TF * IDF* show increasing prominence from the 1st layer to the last, ultimately playing an important role in ranking decision making. Other MSLR features like *Sum(TF/L)*, *Max(TF/L)*, and *Sum TF * IDF* show negative correlation with RankLlama decision making.

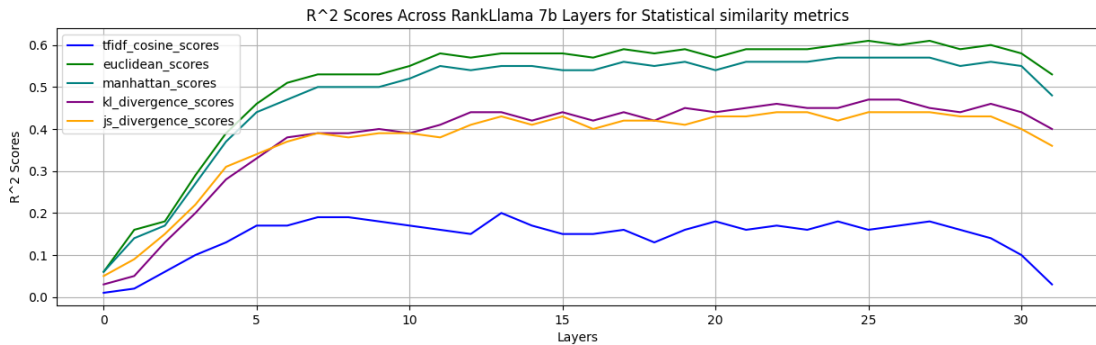


Figure 3: Plot showing R^2 scores of statistical query-document distance metrics when used to probe Rankllama2-7b. Scores indicate that these distance metrics are not encapsulated within Rankllama2-7b as is.

observed that the R^2 scores obtained with mean and max activation aggregation are comparable, except minor exceptions, suggesting either aggregation method is suitable.

Negatives: Certain features failed to achieve an R^2 score greater than 0.1 in the final layer, often achieving a highly negative R^2 . This indicates that the LLM does not consider these features important

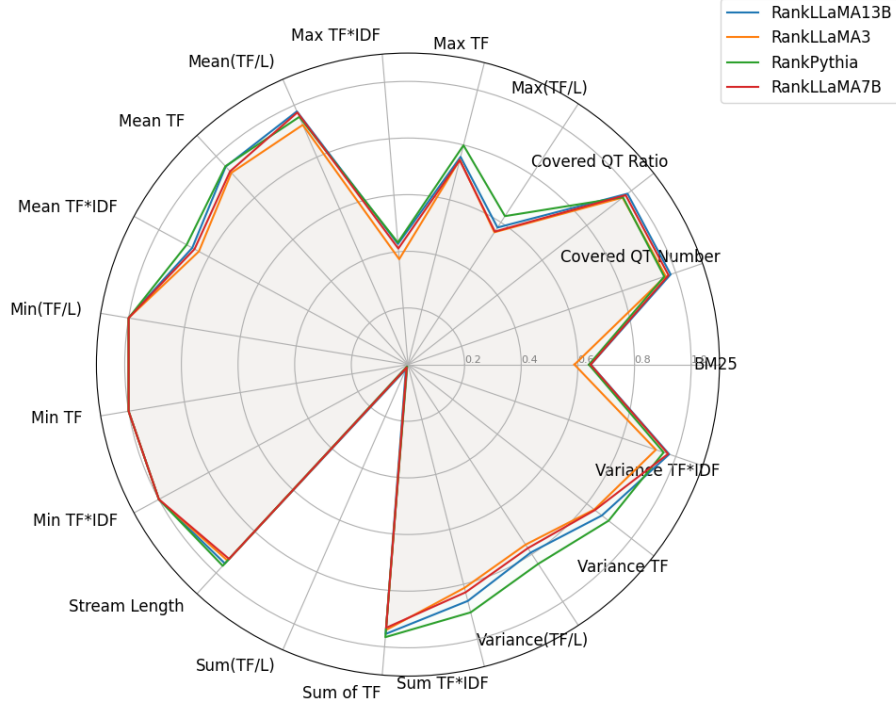


Figure 4: Plot comparing the maximum R^2 score of each query-document IR feature metrics when used to probe different Ranking LLM architectures - RankLlama2-7b and 13b, RankLlama3-8b and RankPythia-6.9b. Here \cdot/L stands for length normalized and QT stands for Query Term. We observe that different LLM architectures, when fine-tuned on the same dataset and loss function, learn very similar ranking related latent features. This suggests, that our probing findings aren't LLM specific, but generalizable.

in their current form. The 10 features from the MSLR set that fell into this category include: *Sum and Max of $TF*IDF$, $Sum(TF/L)$ and $Max(TF/L)$, Max and $variance$ of TF , $stream$ length and Sum of TF* . Thus, while *Mean(TF/L)* is something the LLM highly seeks, it does not seek for the *sum* and *max* of *normalized term frequency* at all!

4.1.2 Traditional similarity metrics. We also probe the LLM architecture for non-semantic statistical query-document distance metrics like *$tf*idf$ cosine score*, *euclidean score*, *manhattan score*, *Kullback-Leibler divergence score* and *Jensen-Shannon divergence score*. Our motive behind this is to identify if the LLMs include components that mimic statistical similarity measures. Results for the probe on RankLlama2-7b are visualized in Figure 3. We observe that statistical score based methods do not correlate well with neuron activations. Out of the feature examined, *euclidean score* shows the highest correlation with R^2 reaching 0.6, dipping before the final layer. Given its well known success in the past, it is ironic that the *$tf*idf$ cosine score* between the query and document shows the least correlation when probed for, within RankLlama activations.

4.1.3 BERT and T5 scores. BERT and T5 are neural models widely used for retrieval and reranking tasks before the advent of LLMs. LLMs are much larger in size compared to BERT models. As a result it is of interest to see if BERT and T5 subnetworks are present within the studied LLM's activations. We design probes to mine for the cosine distance between the BERT and T5 embeddings of the query and the document. Our observations show that both BERT and T5 obtain moderate R^2 scores in our probes, reaching 0.7 and 0.82 on RankLlama2-7b and 13b respectively. The BERT and T5 scores follow each other across the LLM layers. This indicates that the LLM likely does not encode BERT and T5 subnetworks as-is. We compare our findings to related works on BERT probing in Section 5.

4.2 Comparing Ranking LLMs

We next explore whether different ranking LLMs encode similar or distinct Information Retrieval features. To investigate this, we extend our analysis beyond the RankLlama2-7b model and compare it to the two additional fine-tuned LLMs, RankLlama3-8b and

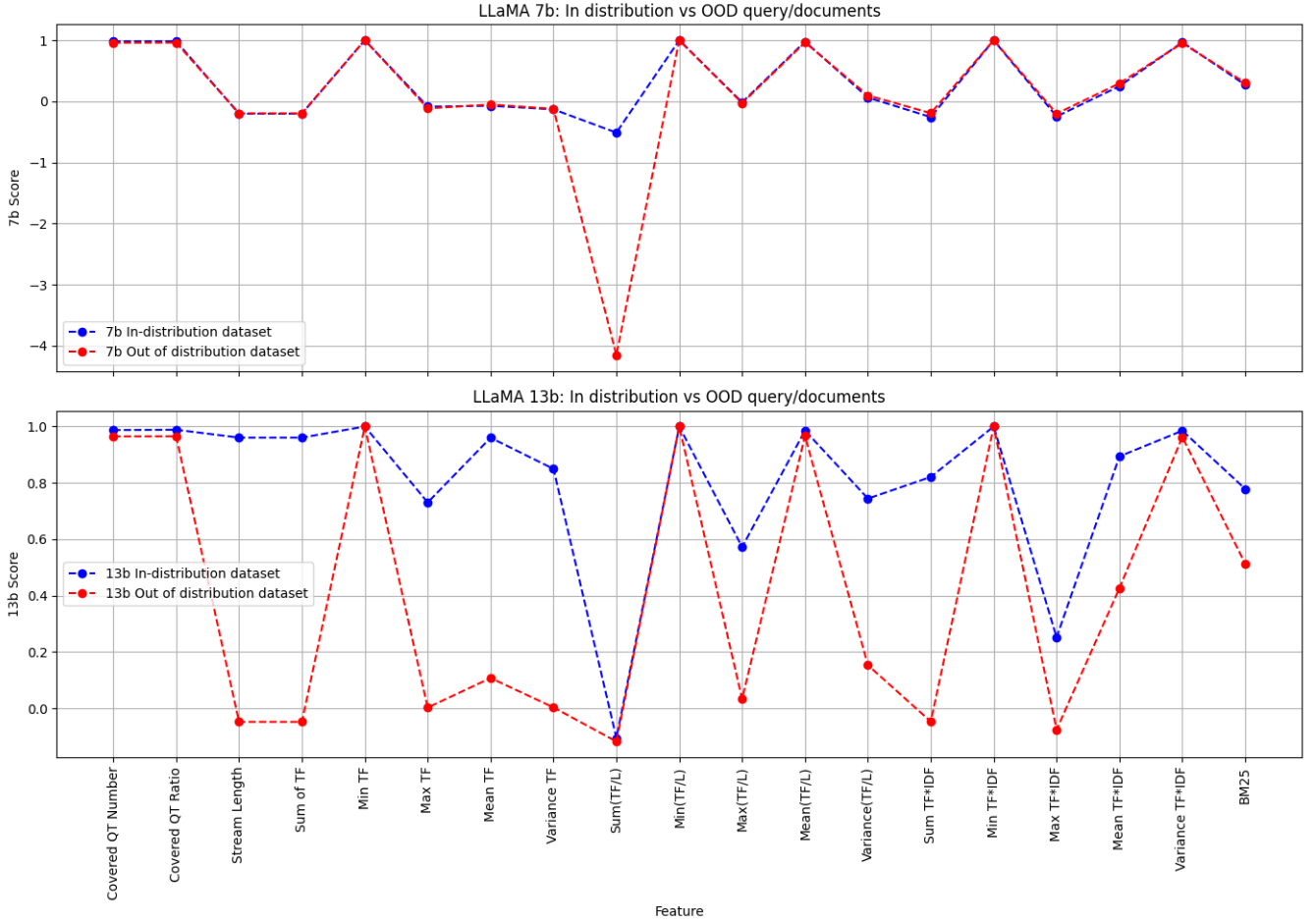


Figure 5: Probing RankLlama2-7b and 13b with in-distribution vs out-of-distribution datasets. We witness that most MSLR features when probed for in RankLlama2-7b, show similar performance with both in-distribution and out-of-distribution datasets. This is however not the case with RankLlama2-13b, where certain features like *stream length* and *sum of term frequency* show a strong presence in the in-distribution dataset probe, even though they are unlikely features to influence a ranking decision. This suggests overfitting on the MS MARCO dataset in RankLlama2-13b.

RankPythia-6.9b, as well as the RankLlama2-13b model. We fine-tune these models under an identical passage reranking framework, with an identical dataset and loss function, to ensure a fair and consistent comparison.

Once the models are fine-tuned, we conduct the same probing experiments to analyze how well each ranking LLM encodes the known human-engineered and semantically meaningful attributes relevant to ranking tasks. The results of these probing experiments are summarized and visualized in a spider chart (Fig. 4), allowing for an intuitive comparison of feature encoding across the four models. It highlights the similarities and differences in the latent features captured by LLMs with varying architectures and sizes, shedding light on the generalizability and architecture-specific tendencies of ranking-focused fine-tuning. The figure depicts the R^2 score for an IR feature, when probed in a particular LLM, and lies between $[0, 1]$.

All four ranking LLMs demonstrate broadly comparable patterns for most features, particularly for the *Min* metrics (e.g., *Min of TF*, *Min of TF*IDF*), where they uniformly show top scores (all 1.0). They also appear closely aligned on coverage-based features—such as Covered Query Term Number and Ratio—with each scoring in the high 0.95–0.98 range. By contrast, the greatest differences emerge in TF-based statistics like *Max of Term Frequency* and *Variance of TF/TF*IDF*, where RankPythia tends to exhibit slightly higher peaks (e.g., a larger R^2 for *Max of TF*) than the RankLLaMA models. Additionally, all LLMs show near zero performance in encoding *Sum of Stream Length Normalized TF*, suggesting similarity in how these models handle term-frequency weighting. Overall, the four models encode IR features in a largely similar manner, indicating that our probing findings are generalizable and not LLM specific.

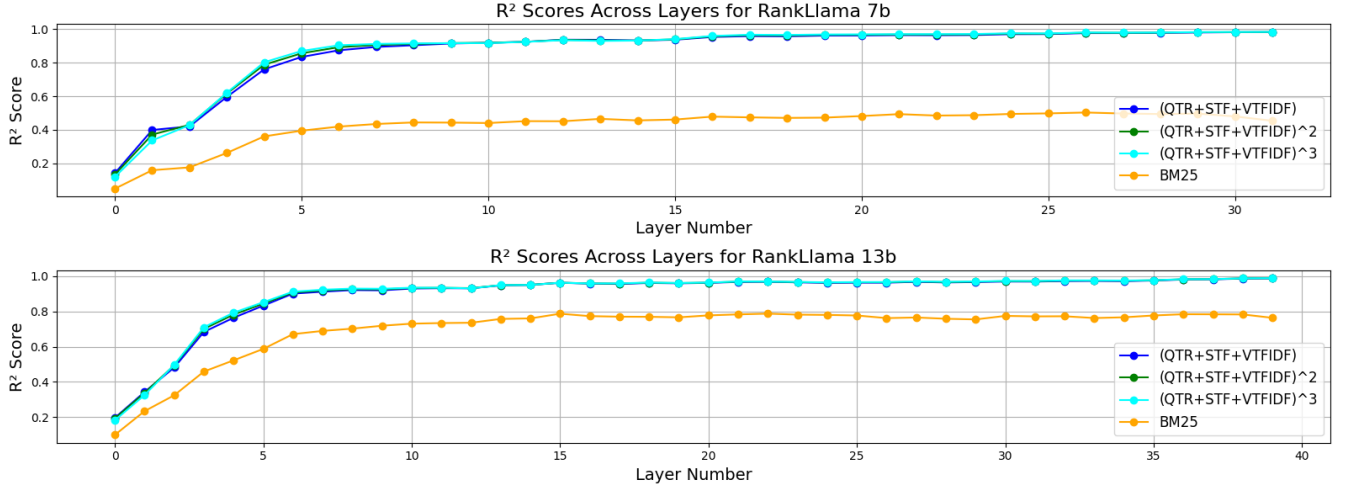


Figure 6: Graph showing R^2 scores of feature groups $(QTR + STF + VTFIDF)$, $(QTR + STF + VTFIDF)^2$ and $(QTR + STF + VTFIDF)^3$ over the layers of the RankLlama2-7b and 13b architectures, where QTR represents covered query term ratio, STF represents mean stream length normalized term frequency and $VTFIDF$ represents variance of $tf*idf$.

4.3 Out-of-distribution queries and documents

All four of the LLMs we study were obtained by fine-tuning the base models using LoRa on the MS MARCO dev set [35]. It is of interest to see if during inference the fine-tuned LLM extracts different features from in-distribution and out-of-distribution query-document pairs. We try probing all the studied LLMs with query-document pairs from two others datasets. First, we use the BEIR Scidocs dataset [48], comprising scientific documents and queries. We select 200 queries and use bm25 to retrieve the top 100 documents for each. We then probe the MS MARCO fine-tuned LLMs with BEIR activations. We repeat the process with the SoDup dataset [32], which contains a question from StackOverflow as query and a list of other relevant questions from StackOverflow as potential duplicate candidates. For each question, we select known duplicate questions from the dataset and treat them as relevant documents. We again pick 200 queries from this dataset for our out-of-distribution probing experiments. We probe for each of the 19 MSLR features on all the fine-tuned ranking LLMs and compare probing results between in-distribution and the out-of-distribution datasets. We show our comparison of the probing results of RankLlama2-7b vs RankLlama2-13b in Figure 5, reporting the R^2 score of each feature in the last layer of the respective LLM. For the feature *mean of tf* we see that our probes cannot find any activations representing the feature in RankLlama 7b, both with in-distribution and out-of-distribution datasets. However, when probing RankLlama 13b, while out-of-distribution activations do not capture this feature, it gets a hit within in-distribution activations.

We see some variance in results between RankLlama2-7b and RankLlama2-13b in this set of experiments. The probes seem to fare similarly on most features between in-distribution and out-of-distribution data on RankLlama2-7b. The only exception being *sum of stream length normalized tf*, which has a negative R^2 score for both in and out distribution data, making its magnitude insignificant. However, the probes seem to fetch different results between in and

out-of-distribution data for RankLlama2-13b. It in particular finds probes strongly correlated to *stream length*, *sum* and *mean of tf*, and *mean of $tf*idf$* in the in-distribution data, but *not* for the out-of-distribution data. A likely reason for this might be that RankLlama 13b has overfit to the MS MARCO dev set and is hence seeking features like *stream length* which are known to not generalize for reflecting query-document relevance [4].

We compared the probing performance of in-distribution and out-of-distribution query-document pairs in RankLlama3-8b and RankPythia-6.9b, observing similar performance for both types of inputs, consistent with the results observed for RankLlama2-7b in Figure 5a. The graphs depicting these probing results are not included due to space constraints.

4.4 Feature Groups within RankLlama

In the previous subsection, we found that several MSLR features (*covered QT Number*, *covered QT Ratio*, *mean(TF/L)*, *min TF*, *min of $TF*IDF$* , *Min(TF/L)* and *variance of $TF*IDF$*) appear to be modeled within LLM activations, especially within the later layers. These features also seem to be a fair choice to model relevance based on intuition. However, these features might not be present individually but in combination with one another or individually with different exponents. To test for this possibility, we probe for combinations of these features within different layers of the LLM and find various combinations of *covered query term ratio*, *mean of stream length normalized term frequency* and *variance of $tf*idf$* . We find strong indication for their combined presence within the LLM activations of the later layers of the LLM. For example, if QTR represents *covered query term ratio*, STF represents *mean of stream length normalized term frequency* and $VTFIDF$ represents *variance of $tf*idf$* , we find high scores on average for all of $QTR+STF$, $QTR+VTFIDF$, $STF+VTFIDF$, $QTR+STF+VTFIDF$, $QTR*STF$, $STF*VTFIDF$, $QTR*VTFIDF$, and $QTR*STF*VTFIDF$ when probing the last layer of the LLM. This potentially indicates that

the LLM has over the course of layers learned some representation of $(QTR+STF+VTFIDF)^k$. In Figure 6, we show the probing performance of a sum of those three features and their exponents, relative to BM25 in RankLlama2-7b and 13b models. We observe that the sum of this feature group, its square, as well as its cube consistently show a strong correlation to neuron activations within RankLlama. This analysis provides a foundation for future work to uncover and investigate complex groups of features encoded within LLM activations. By highlighting how various ranking architectures capture and represent different IR features, it opens avenues for deeper exploration into latent feature interactions and the mechanistic underpinnings of ranking-focused language models.

5 DISCUSSION

Comparing Notes with Previous Probing Work: A number of studies have probed BERT and T5 with the aim of understanding if they encode concepts like term frequency and inverse document frequency. Formal et al. [21] study the matching process of ColBERT [30] and conclude that the model is able to capture a notion of term importance and relies on exact matches for important terms. This is in agreement with our findings, where we find strong correlations to term matching features like *covered query term number/ratio* and *stream length normalized term frequency* within LLM activations. In a later work, Formal et al. [22] conduct a study to measure the out-of-domain zero-shot capabilities of BERT/T5 models in lexical matching on the BEIR dataset, and show that these models fail to generalize lexical matching in out-of-domain datasets or terms not seen at training time. This finding continues for RankLlama as well, where the model is unable to generalize lexical matching to terms not seen beforehand (Figure 5, RankLlama 13b OOD vs In-distribution probing discrepancies).

Different types of neurons: A monosemantic neuron is a neuron within a neural network that responds to a single, specific feature or concept in the input data [3]. However, in large neural networks the networks often try to extract more features than the number of neurons. In such scenarios, the model has to perform a superposition of features on single neurons in order to compress desired features into the limited number of neurons. This gives rise to polysemantic neurons – i.e., neurons encoding more than one feature. This phenomenon is usually witnessed in the early layers of the LLM architecture and is difficult to disentangle using linear probes. There is unfortunately no known method to identify monosemantic vs polysemantic neurons via probing. Another common phenomena in sparse LLMs is when multiple neurons within a layer together represent a particular human-known context. Among the features probed in our study, 23% were found in a single neuron, 32% were distributed between 2-3 neurons and the other 45% were distributed in 4 or more neurons.

Validating Probing Results While probing techniques are widely used to understand the internal workings of LLMs, there are several limitations of using probing for this purpose. (1) Probing techniques reveal correlations between specific features and neuron activations, but not a causal analysis of the decision making process. (2) The insights gained from probing are heavily dependent on the dataset used for probing. If the dataset is not representative of the model’s typical input, the results may not accurately reflect the

model’s general behavior. (3) LLMs often rely on complex interactions between multiple features. Probing techniques may fail to capture these interactions, leading to an incomplete understanding of how the model makes decisions. (4) Probing can identify features but may mislead regarding their role or significance in the model’s decision-making process. However, many of the limitations of probing can be addressed by employing various techniques, such as creating balanced probing datasets and independently validating probing results through methods like ablation studies and feature attribution analyses.

To validate our probing results, we analyzed neurons in the final layer of RankLLaMA models that exhibited strong correlations ($R^2 > 0.85$) with probed features, as this layer directly influences ranking decisions. Using Shapley values [10, 14, 34] for feature attributions we assessed the contributions of individual neurons and neuron groups to the ranking predictions, leveraging a value function based on changes in NDCG scores for MS MARCO query-document pairs. Probing validation was conducted on 100 query-document pairs for each RankLLaMA model, and we computed average attributions across neuron groups of varying sizes. Results showed that identified neuron groups ranked within the 95th percentile in 79 out of 100 cases for RankLLaMA 7B and 84 out of 100 for RankLLaMA 13B, confirming that these neurons are instrumental in ranking decisions and validating the location of the probes.

6 CONCLUSION AND FUTURE WORK

In this study, we probed four different large language models, each fine-tuned for passage reranking, to look for a subset of features from the MSLR dataset. These human-engineered features, which include query-only, document-only, and query-document interaction features, are recognized as significant in ranking tasks. Using a layer-wise probe, we discovered that the activations in most layers could accurately replicate select features such as *Min TF*IDF*, *covered QT Number/Ratio*, *Min TF*, *Min(TF/L)*, and *Mean(TF/L)*. This suggests that the fine-tuned Llama network deems these features important. Conversely, we found no correspondence for features like *sum* and *max* of *tf*idf* and *sum* and *max* of *tf* in the probed neurons. This indicates the absence of these features in RankLlama’s neural feature extractors. We also reported observations on feature groups that show a strong correlation with RankLlama neurons and hypothesized that some abstract features are mined by the LLM. We also observed that different LLM architectures, when fine-tuned on the same dataset with the same loss function, exhibit similar latent features when probed. Finally, we compared later features encoded between in-distribution and out-of-distribution datasets and encountered a case of the LLM’s apparently overfitting during fine-tuning. These findings enhanced our understanding of ranking LLMs and gave us generalizable insights for the design of more effective and transparent ranking models. The long-term objectives of this endeavor include: (i) identifying potential modifications to existing MSLR features, such that they further align with LLM activations, (ii) deciphering and reverse-engineering segments of the LLM that do not correspond to recognized MSLR features, and (iii) ultimately cataloging all features deemed significant by LLMs and plugging them back into simpler statistical models to enhance the performance and interpretability of statistical ranking

models. These directions will help further explain the underlying mechanisms of ranking LLMs and improve human trust into them.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

REFERENCES

- [1] 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).
- [2] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2019. Gradient-based attribution methods. *Explainable AI: Interpreting, explaining and visualizing deep learning* (2019), 169–191.
- [3] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. 2020. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences* 117, 48 (2020), 30071–30078.
- [4] Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics* 48, 1 (2022), 207–219.
- [5] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*. PMLR, 2397–2430.
- [6] 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.
- [7] Nuo Chen, Ning Wu, Shining Liang, Ming Gong, Linjun Shou, Dongmei Zhang, and Jia Li. 2023. Beyond Surface: Probing LLaMA Across Scales and Layers. *arXiv preprint arXiv:2312.04333* (2023).
- [8] Jaekel Choi, Euna Jung, Sungjun Lim, and Wonjong Rhee. 2022. Finding Inverse Document Frequency Information in BERT. *arXiv preprint arXiv:2202.12191* (2022).
- [9] 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*. 33–37.
- [10] Tanya Chowdhury, Yair Zick, and James Allan. 2024. RankSHAP: a Gold Standard Feature Attribution Method for the Ranking Task. *arXiv preprint arXiv:2405.01848* (2024).
- [11] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341* (2019).
- [12] Róbert Csordás, Sjoerd van Steenkiste, and Jürgen Schmidhuber. 2020. Are neural nets modular? inspecting functional modularity through differentiable weight masks. *arXiv preprint arXiv:2010.02066* (2020).
- [13] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600* (2023).
- [14] Anupam Datta, Shayak Sen, and Yair Zick. 2016. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE symposium on security and privacy (SP)*. IEEE, 598–617.
- [15] Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnamurthy Kenthapadi, and Adam Tauman Kalai. 2019. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *proceedings of the Conference on Fairness, Accountability, and Transparency*. 120–128.
- [16] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).
- [17] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread* 1 (2021), 1.
- [18] Yixing Fan, Jiafeng Guo, Xinyu Ma, Ruqing Zhang, Yanyan Lan, and Xueqi Cheng. 2021. A linguistic study on relevance modeling in information retrieval. In *Proceedings of the Web Conference 2021*. 1053–1064.
- [19] Zeon Trevor Fernando, Jaspreet Singh, and Avishek Anand. 2019. A study on the Interpretability of Neural Retrieval Models using DeepSHAP. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 1005–1008.
- [20] James Fiocco, Samridhi Choudhary, and Carolyn Rose. 2019. Deep neural model inspection and comparison via functional neuron pathways. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 5754–5764.
- [21] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. A white box analysis of ColBERT. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II* 43. Springer, 257–263.
- [22] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2022. Match your words! a study of lexical matching in neural information retrieval. In *European Conference on Information Retrieval*. Springer, 120–127.
- [23] 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).
- [24] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2020. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913* (2020).
- [25] Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. Finding neurons in a haystack: Case studies with sparse probing. *arXiv preprint arXiv:2305.01610* (2023).
- [26] Wes Gurnee and Max Tegmark. 2023. Language models represent space and time. *arXiv preprint arXiv:2310.02207* (2023).
- [27] Xinzhi Han and Sen Lei. 2018. Feature selection and model comparison on microsoft learning-to-rank data sets. *arXiv preprint arXiv:1803.05127* (2018).
- [28] Yael Hochma, Yuval Feindler, and Mark Last. 2024. Efficient Feature Ranking and Selection using Statistical Moments. *IEEE Access* (2024).
- [29] Georgios Katsimpras and Georgios Paliouras. 2024. GENRA: Enhancing Zero-shot Retrieval with Rank Aggregation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 7566–7577.
- [30] 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*. 39–48.
- [31] Connor Kissane, Robert Krzyzanowski, Arthur Conmy, and Neel Nanda. 2024. Saes (usually) transfer between base and chat models. In *AI Alignment Forum*.
- [32] Nathan Lambert, Lewis Tunstall, Nazneen Rajani, and Tristan Thrush. 2023. *HuggingFace H4 Stack Exchange Preference Dataset*. <https://huggingface.co/datasets/HuggingFaceH4/stack-exchange-preferences>
- [33] Belinda Z Li, Maxwell Nye, and Jacob Andreas. 2021. Implicit representations of meaning in neural language models. *arXiv preprint arXiv:2106.00737* (2021).
- [34] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).
- [35] Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2023. Fine-tuning llama for multi-stage text retrieval. *arXiv preprint arXiv:2310.08319* (2023).
- [36] 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.
- [37] Aleksandar Makelov, Georg Lange, and Neel Nanda. 2023. Is this the subspace you are looking for? an interpretability illusion for subspace activation patching. *arXiv preprint arXiv:2311.17030* (2023).
- [38] Aleksandar Makelov, George Lange, and Neel Nanda. 2024. Towards principled evaluations of sparse autoencoders for interpretability and control. *arXiv preprint arXiv:2405.08366* (2024).
- [39] Clara Meister, Stefan Lazov, Isabelle Augenstein, and Ryan Cotterell. 2021. Is sparse attention more interpretable? *arXiv preprint arXiv:2106.01087* (2021).
- [40] 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. (2016).
- [41] Andrew Parry, Catherine Chen, Carsten Eickhoff, and Sean MacAvaney. 2024. MechIR: A Mechanistic Interpretability Framework for Information Retrieval. (2024).
- [42] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 Datasets. *CoRR abs/1306.2597* (2013). <http://arxiv.org/abs/1306.2597>
- [43] Razieh Rahimi, Youngwoo Kim, Hamed Zamani, and James Allan. 2021. Explaining documents' relevance to search queries. *arXiv preprint arXiv:2111.01314* (2021).
- [44] Tilman Räuher, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. 2023. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. IEEE, 464–483.
- [45] Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. 2020. Probing the probing paradigm: Does probing accuracy entail task relevance? *arXiv preprint arXiv:2005.00719* (2020).
- [46] Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. 2022. Neuron-level interpretation of deep nlp models: A survey. *Transactions of the Association for Computational Linguistics* 10 (2022), 1285–1303.
- [47] Jaspreet Singh and Avishek Anand. 2019. Exs: Explainable search using local model agnostic interpretability. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 770–773.

- [48] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663* (2021).
- [49] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).
- [50] Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. 2020. Supermasks in superposition. *Advances in Neural Information Processing Systems* 33 (2020), 15173–15184.
- [51] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. An analysis of BERT in document ranking. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 1941–1944.
- [52] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2014. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856* (2014).
- [53] Bolei Zhou, Yiyu Sun, David Bau, and Antonio Torralba. 2018. Revisiting the importance of individual units in cnns via ablation. *arXiv preprint arXiv:1806.02891* (2018).