Retrieval-Augmented Generation: Is Dense Passage Retrieval Retrieving?

Anonymous ACL submission

Abstract

Dense passage retrieval (DPR) is the first step in the retrieval augmented generation (RAG) paradigm for improving the performance of large language models (LLM). DPR fine-tunes pre-trained networks to enhance the alignment of the embeddings between queries and relevant textual data. A deeper understanding of DPR fine-tuning will be required to fundamentally unlock the full potential of this approach. In this work, we explore DPR-trained models mechanistically by using a combination of probing, layer activation analysis, and model editing. Our experiments show that DPR training decentralizes how knowledge is stored in the network, creating multiple access pathways to the same information. We also uncover a limitation in this training style: the internal knowledge of the pre-trained model bounds what the retrieval model can retrieve. These findings suggest a few possible directions for dense retrieval: (1) expose the DPR training process to more knowledge so more can be decentralized, (2) inject facts as decentralized representations, (3) model and incorporate knowledge uncertainty in the retrieval process, and (4) directly map internal model knowledge to a knowledge base.

1 Introduction

In just a few years, Large Language Models (LLMs) have emerged from the research labs to become a tool utilized daily by hundreds of millions of people and integrated into a wide variety of businesses. Despite their popularity, these models have been critiqued for frequently hallucinating, confidently outputting incorrect information (Bang et al., 2023). Such inaccuracies not only mislead people but also erode trust in LLMs. Trust in these systems is crucial to their success and rate of adoption.

The retrieval augmented generation (RAG) paradigm is an approach to address hallucinations (Lewis et al., 2020). Unlike traditional LLM interactions where a query directly prompts an output from the model, RAG introduces an intermediary step. Initially, a "retrieval" model processes the query to gather additional information from a knowledge base, such as Wikipedia or the broader internet. This additional information alongside the original query is fed to the LLM, increasing the accuracy of the answers that the LLM generates.

For RAG to be effective, the underlying retrieval model has to excel at finding accurate and relevant information. Typically, model performance is evaluated based on metrics that consider the top-5, top-20, top-50, and top-100 retrieved passages. However, recent studies indicate that LLMs predominantly use information from the top-1 to top-5 passages, underscoring the importance in RAG of not only high recall in retrieval but also precision in ranking (Liu et al., 2023a; Xu et al., 2024). One approach to achieve both high recall and precision involves integrating a "reranking" model, which adjusts the order of retrieved passages to improve the relevance of the top-ranked passages (Nogueira et al., 2019, 2020). However, this approach adds the computational and maintenance cost of an additional model to the pipeline and can also introduce errors. The alternative option is to improve retrieval models to directly retrieve and rank passages well.

Retrieval methods can be broadly categorized into two types: sparse and dense (Zhao et al., 2024). Sparse methods encode queries and passages into sparse vectors, usually based on terms that appear in the queries and passages (Robertson and Zaragoza, 2009; Sparck Jones, 1972). Dense methods employ language models to encode the semantic information in queries and passages into dense vectors (Karpukhin et al., 2020; Huang et al., 2013). Dense methods often share two common properties: (a) the joint training of two or more encoding models – one for embedding a query and the other for embedding a knowledge base, and (b) contrastive training. These commonalities were

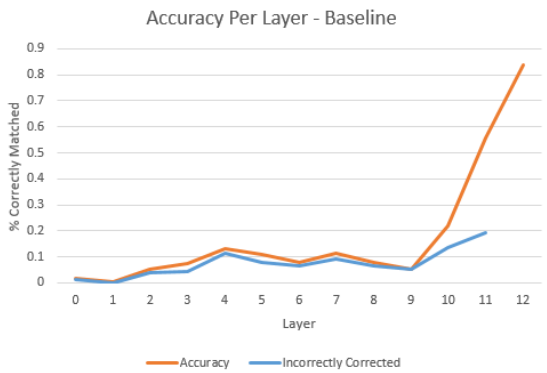

Figure 1: Per layer accuracy of the baseline DPR model and percentage of incorrectly matched output that was correctly matched at layer N.

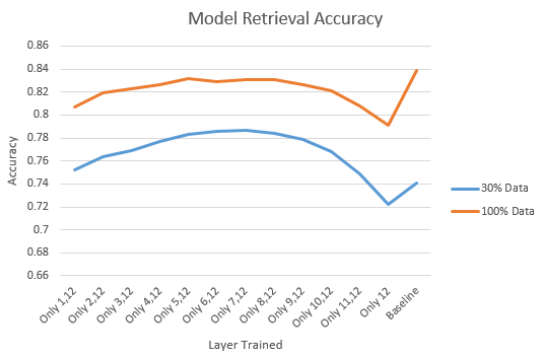

Figure 2: Accuracy of DPR model when only layer N and layer 12 are trained with the full NQ dataset and only 30% of the NQ dataset

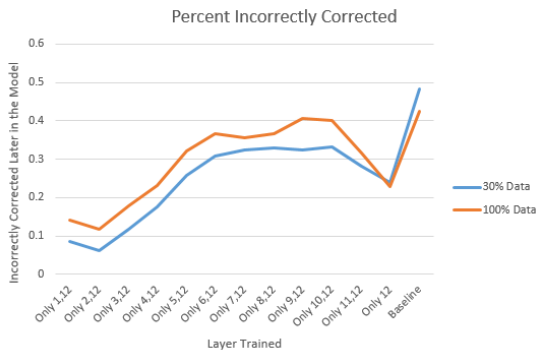

Figure 3: Percent of incorrect matches that intermediary network layers got correct.

introduced in the dense passage retrieval method, inspiring many subsequent methods in the literature.

In this paper, we analyze the original DPR method using the BERT-base backbone. We begin by analyzing the effects of training and using different layers for retrieval (Section 2). Then we continue by probing the model to determine if the features of pre-trained BERT are as discriminative as DPR-BERT in matching a query to the correct hard-negative passage (Section 3). Next, using techniques from the pruning literature, we compare the relative strength and number of activations of the feedforward layers throughout the original pre-trained and DPR-trained models (Section 4). Finally, we add and remove knowledge from the network to investigate how knowledge interacts with DPR training (Section 5). Through these experiments, we analyze DPR from multiple perspectives to understand what is changing in the backbone model during the training process.

2 Early Passage Match Capability

Multilayer transformers are typically used for DPR. These models perform layer-to-layer updates to refine and promote concepts within the embedding space (Geva et al., 2022). Such iterative refinement can reach a point of saturation, beyond which additional updates in later layers contribute marginally to model performance. This saturation suggests that there may be enough information early in the model to accurately match a query to a passage.

In this section, we analyze model update saturation to identify: (a) which layers contribute most significantly to performance improvements, and (b) aside from the final layer, which layers are crucial for training. Our analysis employs the training methodology outlined in the earlier DPR paper (Karpukhin et al., 2020). The BERT model was fine-tuned in this style on the Natural Questions (NQ) dataset (Kwiatkowski et al., 2019). To assess the DPR model's accuracy at layer N, we conducted a maximum inner product search using the hidden state of layer N.

Due to the number of different training configurations and layers that were tested, it was not possible to re-index the 21M passage Wikipedia knowledge base that is commonly used with NQ. Therefore, we focused on assessing in-batch match accuracy. Each batch comprised 64 queries and 832 passages. The performance metrics reported in this section reflect the accuracy rate, namely the proportion of queries that were successfully matched to a corresponding passage within the batch.

Figure 1 illustrates that peak performance is achieved at the network's final layer. Notably, the most substantial performance increases occur between layers 3 and 4, 9 and 10, and 10 and 11. Moreover, Figure 1 suggests that in some cases, lay-

Task	Model	Layer 0	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6	Layer 7	Layer 8	Layer 9	Layer 10	Layer 11	Layer 12
2-Passage Probing	Pre-trained BERT – Untrained Probe	0.50	0.50	0.51	0.48	0.50	0.52	0.51	0.51	0.50	0.49	0.50	0.54	0.50
	Pre-trained BERT	0.51	0.69	0.74	0.74	0.77	0.79	0.81	0.81	0.81	0.82	0.83	0.84	0.84
	DPR-BERT Query Model	0.51	0.68	0.74	0.77	0.79	0.80	0.81	0.83	0.82	0.83	0.83	0.82	0.82
	DPR-BERT Context Model	0.51	0.68	0.74	0.77	0.79	0.80	0.81	0.83	0.82	0.83	0.83	0.82	0.82
3-Passage Probing	Pre-trained BERT	0.34	0.53	0.59	0.59	0.65	0.64	0.67	0.67	0.68	0.69	0.69	0.73	0.73
	DPR-BERT	0.34	0.54	0.60	0.63	0.66	0.66	0.66	0.70	0.71	0.69	0.73	0.72	0.71
4-Passage Probing	Pre-trained BERT	0.26	0.43	0.47	0.49	0.53	0.57	0.61	0.60	0.56	0.62	0.64	0.66	0.66
	DPR-BERT	0.26	0.46	0.51	0.54	0.57	0.58	0.60	0.63	0.64	0.63	0.65	0.63	0.63
5-Passage Probing	Pre-trained BERT	0.21	0.35	0.42	0.43	0.43	0.50	0.53	0.53	0.54	0.56	0.57	0.60	0.61
	DPR-BERT	0.21	0.36	0.42	0.48	0.49	0.51	0.54	0.56	0.58	0.58	0.60	0.56	0.56

Table 1: This table presents the outcomes of linear probing, where probes classify 2 to 5 passages to determine the best match for a given query. Due to identical performance metrics, DPR-BERT Query and Context model results are consolidated and displayed only for the 2-Passage Probe. Given that probes without training achieved performance at random chance levels across all passage counts, their results are reported solely for the 2-Passage Probe for comparison.

ers 4, 10, and 11 had the highest occurrence where the intermediary hidden layers correctly identified the relevant passage but a later layer subsequently revised the prediction to an incorrect match.

To further investigate the relationship between individual model layers and DPR performance, a series of experiments were conducted where the DPR model was trained two layers at a time. In these experiments, all layers were frozen except for two: layer N, which progressed from the first to the last layer across subsequent experiments, and the final layer, included due to its superior performance. Training followed the protocol established in the DPR paper, with layers N and 12 being trained independently—resulting in two separate loss calculations per batch, one for the interaction between the hidden states at layer N in the query and passage models and one loss for the embeddings outputted at layer 12 in the query and passage models.

Trained over the Natural Questions (NQ) dataset (Kwiatkowski et al., 2019), the experiment demonstrated optimal performance when layers 5, 7, or 8 were concurrently trained with layer 12, as shown in Figure 2. However, since the performance gain was only slight, the experiment was repeated under a data-restricted regime, utilizing only 30% of the NQ dataset. In this constrained data scenario, the disparity in accuracy when training various layers became more pronounced. Notably, training layer 7 provided the greatest accuracy boost. Moreover, training any layer within the middle segment of the network (layers 5-8) surpassed the performance of layers closer to the beginning or end of the network.

Figures 2 and 3 demonstrate that training just one middle layer in conjunction with layer 12 not only yields the highest performance but also leads to a middling frequency of instances where a previously correct match is incorrectly altered later in the model. According to prior research, it is in these middle layers that the processing of syntactic and semantic features co-occur (Geva et al., 2021). The syntactic features are more akin to sparse retrieval methods whereas the semantic features represent the theme of DPR-style retrieval methods. This suggests that learning the interplay between syntactic and semantic features is likely an important part of what DPR-style training teaches the model.

3 Knowledge Consistency between untrained and trained model

Language models are known to store a vast amount of knowledge with the feedforward layers of the transformer architecture acting as a key-value memory store of knowledge (Geva et al., 2021). This section details experiments conducted to understand the impact of DPR-style training from a model-knowledge perspective.

Linear probing, a method to characterize model features, involves training a linear classifier on the internal activations of a frozen network to execute a simple task (Alain and Bengio, 2017). This reveals the mutual information shared between the model's primary training task and the probing task (Belinkov, 2022). A high degree of probe accuracy indicates that the model's features possess sufficient information to accomplish the probing task.

To evaluate whether DPR training improved BERT's discriminative features, linear probing was employed on both pre-trained and DPR-trained BERT. A classification probe

$$g_{lN}(f_{lq}, f_{ltp}, f_{lhn1}, f_{lhn2}, \dots)$$

was trained for each index of the passage deemed most relevant where l signifies the probed layer, f_{lq} the features at layer l for the query, f_{ltp} the features for the true positive paragraph at layer l, and f_{lhnN} the features for the Nth hard negative passage at the same layer. A distinct probe g_{lN} was trained for each layer of BERT to examine how performance fluctuates across layers and with different numbers of hard-negative passages, thereby assessing how performance is impacted as the task's difficulty increases.

Table 1 shows the result of this experiment. The performance disparity between probes for pre-trained BERT and DPR-trained BERT is relatively minor in the two-passage scenario (1.8%) and interestingly, it is the pre-trained BERT that exhibits a slight advantage. As the number of passages increases, the performance gap widens to approximately 6%, and overall probe efficacy declines. These findings suggest that the inherent capabilities to discern relevant from irrelevant passages are already present in pre-trained BERT, and DPR-style training does not substantially enhance these discriminative features. Moreover, echoing the previous section's findings, the hidden features in the network's middle layers perform nearly as well as those from the final layers.

4 Knowledge Decentralization in DPR-Trained Models

The next perspective examined neuron activation patterns for the pre-trained and DPR-trained models. The knowledge attribution method from (Dai et al., 2022) was employed which was inspired by the pruning literature (Hao et al., 2021; Sundararajan et al., 2017). Our analysis targeted linear layers, as this is where the model stores knowledge according to prior research (Geva et al., 2021).

To calculate an individual neuron's contribution to the output, we varied its weight $w_i^{(l)}$ from 0 to its original value. This can be calculated by:

$$\text{Attr}^{(l)}(w_i) = w_i^{(l)} \int_{\alpha=0}^{1} \frac{\partial P_x(\alpha w_i^{(l)})}{\partial w_i^{(l)}} \mathrm{d}\alpha$$

The Riemann approximation was used due to the intractability of calculating a continuous integral. Following (Dai et al., 2022), a threshold of $0.1 * \max(\text{Attr})$ was applied to identify a coarse set of knowledge neurons. Appendix A.2 demonstrates that our observations are consistent across a spectrum of thresholds. In contrast to (Dai et al., 2022), the coarse set of knowledge neurons was not refined to a fine set of knowledge neurons, as our interest is on the broader activation patterns. When the model operates, it activates both "true-positive" and "false-positive" knowledge neurons indiscriminately according to their attribution scores. The primary interest lies in how DPR training influences these activation patterns, rather than the role of specific neurons.

Figure 4 illustrates the impact of DPR training on BERT's neuron activations, charting the attribution score of every neuron across both the intermediate and output linear layers within each transformer block for the query model. Appendix A.1 shows that the observations made in this section also hold for the context model. DPR-trained BERT has more activated neurons in the intermediate layer of each block. The output layer, on the other hand, maintains a consistent number of activations at each transformer block compared to pre-trained BERT, and in the earlier layers DPR-trained BERT activates fewer neurons in the output layers. Previous studies have conceptualized intermediate layers as "keys" and the output layer as the "value" (Geva et al., 2021). This suggests that DPR training expands the set of "keys" available to access a given volume of semantic knowledge while decreasing the accessible volume of syntactic knowledge, embodying a decentralization strategy for semantic knowledge. Rather than relying on a single, highly precise key to unlock some knowledge, DPR allows for the use of multiple, somewhat less precise keys. This underscores DPR training's primary goal: to modify the model's method of knowledge access without altering the stored knowledge itself. These multiple pathways enable morphologically distinct but semantically related text to trigger the same knowledge or collections of facts, thus making retrieval possible.

5 Adding and Removing Knowledge to Model

If DPR is rearranging knowledge found in pre-trained BERT, would we be able to see facts that

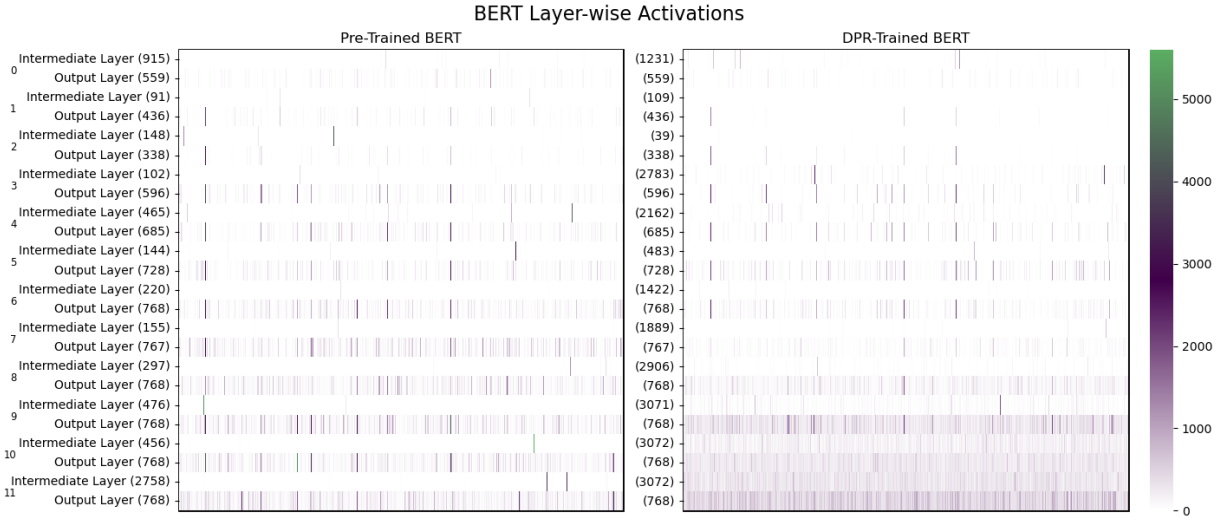

Figure 4: Layerwise activations for pre-trained and DPR-trained BERT. The parenthetical numbers indicate the number of neurons in the layer that are above the attribution threshold for any number of examples.

pre-trained BERT knows reappear in DPR-BERT? To investigate this, we employed model editing techniques to add and remove facts from pre-trained BERT. Owing to the emerging state of this subfield and the variability in results, we employed various model editing techniques. In selecting techniques, we prioritized those that directly manipulated the model's weights or minimally altered the model architecturally. This approach was chosen to facilitate clearer attributions of our findings to DPR training rather than to potential architectural modifications. TransformerPatch, MalMen, and Mend were used to perform the model editing (Huang et al., 2023; Tan et al., 2024; Mitchell et al., 2022). TransformerPatch introduces a single parameter to the last layer for each fact added, whereas MalMen and Mend utilize hypernetworks to add facts by predicting how the model weights would need to be changed.

5.1 Knowledge Addition

The first branch of experiments focused on adding facts to BERT. To select the facts for addition, we identified the questions from the NQ dataset that both DPR-BERT and the probed pre-trained BERT incorrectly answered. For each of the 284 identified questions, we added one fact to BERT, synthesized by transforming each query-answer pair from the NQ dataset into a cohesive sentence with GPT-4. Furthermore, when necessitated by the editing methodology, GPT-4 was employed to generate 10-12 rephrasings of each sentence.

The next step was determining whether the facts had been successfully added to the network. Probing results served as an indicator for this verification. If the probe accurately matched the query associated with a fact, it suggested that the fact was successfully added to the model. Table 2 shows that approximately 54%-57% of the attempted facts were successfully added to the model. The consistency observed across various recently developed methods suggests that this level of performance is representative of current model editing capabilities. We also observed a number of off-target edits; however, this issue was deemed minor, given the primary goal of adding specific facts was achieved. Following the edits, this modified "pre-trained BERT" underwent DPR-style training. Table 2 reveals that DPR-trained BERT accurately recognized 37%-44% of these newly added facts.

The lower-than-expected performance observed does not detract from the results of these experiments. In evaluating these experiments, it is important to note that the facts that were edited are in the test set, while the model is trained with a distinct training set. This discrepancy raises the possibility that the overlap between the facts necessary for training and testing queries might not be sufficiently high. Consequently, the added facts may not have developed a decentralized representation within the model through DPR training. This is consistent with other research that indicates DPR's potential limitations in terms of generalization (Thakur et al., 2021; Gangi Reddy et al., 2022). Additionally, certain queries might require the addition of multiple facts to enable accurate

284 Facts Added	Probing Added	Off-Target Flips - Probing	DPR Added	Off-Target Edits - DPR
Transformer-Patch	0.54	581	0.44	222
MalMen	0.57	592	0.37	236
Mend	0.57	592	0.38	229

Table 2: This table presents the outcomes of the knowledge addition experiments. The "Probing Added" column is the percentage of the total facts that were successfully added to BERT. The "DPR Added" column is the percentage of those facts that were detected after DPR training.

284 Facts Removed	Probing Removed	Off-Target Flips - Probing	DPR Removed	Off-Target Edits - DPR
Transformer-Patch	0.16	689	0.87	183
MalMen	0.11	721	0.81	261
Mend	0.11	722	1.00	252

Table 3: This table presents the outcomes of the knowledge removal experiments. The "Probing Removed" column is the percentage of the total facts that were successfully added to BERT. The "DPR Removed" column is the percentage of those facts that were detected after DPR training.

matching, but our experiments introduced only one fact per query. Given the interconnected and co-dependent nature of facts and knowledge—contrary to being discrete entities—this one fact per query approach might not suffice. Lastly, it is possible that these results simply reflect how new this subfield is. Nevertheless, the reappearance of inserted facts in DPR-BERT underscores the way in which the DPR training process leverages the knowledge of pre-trained BERT to create a model capable of retrieving information.

5.2 Knowledge Removal

The next experiment was the inverse of the previous one: facts were removed from BERT. A total of 284 queries, which both DPR-BERT and the linear probes had accurately matched with their corresponding passages, were randomly selected. Given that the chosen model editing techniques did not provide a direct method to explicitly remove facts from BERT, we employed previously described techniques to "overwrite" BERT's knowledge. To generate factually incorrect statements, the factually correct query-answer pairs were provided to GPT-4, which was prompted to generate new factually incorrect sentences. These new sentences were used by the model editing techniques to overwrite existing knowledge.

Table 3 indicates that merely $11\% - 16\%$ of facts were successfully overwritten. This limited success could stem from the complexity of fully erasing a fact, given that facts are interdependent, exist in multiple logical forms, and are supported by neighboring facts that might compensate for any inaccuracies introduced. This complexity, along with the fact that existing facts are being overwritten rather than new ones being introduced, may contribute to the higher incidence of off-target edits when performing fact removal. Notably, the overwritten facts appear to be more strongly set into BERT. $81\% - 100\%$ of the facts that are overwritten were also incorrectly matched in DPR-BERT, as shown in Table 3. This outcome suggests that once a fact and its interconnected network are overwritten, the ability to train a model to retrieve context that requires that fact becomes significantly compromised. It is unlikely that post-removal the fact remains in the network in a form that can be decentralized in a way that makes it retrievable.

Both the knowledge addition and knowledge removal experiments demonstrate that DPR training primarily refines how pre-existing knowledge within BERT is rendered more "retrievable". Newly added facts to BERT became retrievable, while those that were removed ceased to be retrievable. Thus, it appears that DPR training does not alter the model's inherent knowledge base; instead, it modifies the representation and accessibility of this knowledge.

6 Related Works

Dense passage retrieval (DPR) addresses the challenge of matching a query with the most relevant passages from a knowledge base (Karpukhin et al., 2020). This approach employs dual encoders—one encoder for the passages and another for the query—and utilizes a distance metric, such as the inner product, to identify the passages closest to the query. Inspired by Siamese networks (Bromley et al., 1993), DPR represents the first fully neural architecture to outperform the BM25 algorithm (Robertson and Zaragoza, 2009). Since then, there have been quite a few improvements in how to train DPR-style models. Methods like

RocketQA improve DPR by employing cross-batch negatives and training the network on more difficult hard negatives (Qu et al., 2021). Dragon focuses on novel data augmentation and supervision strategies (Lin et al., 2023). Contriever also employs a greater number of hard-negatives and data-augmentation methods in addition to pre-training the model on the inverse cloze task (Izacard et al., 2022). MVR generates multiple views for each document to allow for multiple diverse representations of each of them (Zhang et al., 2022). ColBERT employs token embeddings for more fine-grained matching (Khattab and Zaharia, 2020). REALM leverages feedback from the reader component to jointly train the retriever with the reader (Guu et al., 2020). Other methods distill knowledge from the reader to the retriever (Izacard and Grave, 2020; Reichman and Heck, 2023). Additionally, efforts in query augmentation or generation aim to better synchronize the query with the document encoder (Ma et al., 2023; Wang et al., 2023; Shao et al., 2023; Gao et al., 2023). Despite these different enhancements, each method builds upon the DPR framework discussed in this paper.

Distinctly, RetroMAE and CoT-MAE pre-train a model using a masked auto-encoder strategy, which they show enhances downstream retrieval performance (Xiao et al., 2022; Wu et al., 2023a; Liu et al., 2023b; Wu et al., 2023b). Following this pre-training phase, both methods subsequently adopt DPR fine-tuning to further refine their models for improved task performance.

Only a few studies have delved into analyzing DPR models. One such study took a holistic look at RAG to see where the pipeline made errors (BehnamGhader et al., 2023). The study found that a similarity-based search during retrieval biased the result in favor of passages similar to the query, even when more relevant but dissimilar passages were available. Another study employed probing techniques to analyze ranking models (MacAvaney et al., 2022). The authors adopted a probing method akin to ours, categorizing passages by specific properties for analysis, in contrast to our approach of random selection among hard negatives. This study explored how query and document characteristics affect ranking outcomes. Another study analyzed the embeddings produced by retrieval models in the vocabulary space (Ram et al., 2023). To do this, they used pre-trained BERT's MLM head on the DPR-trained embeddings' [CLS] token. It was found that DPR implicitly learns the importance of lexical overlap between the query and passage. DPR training causes BERT to retrieve passages that share more tokens with the query as compared to pre-trained BERT. This ties in with our finding where the number of output layer activations in the early part of the model post-DPR training decreased. This may function as a sort of syntactic filter, where many keys can access fewer, but more pertinent, lexical features. However, this filtering can also induce what the authors term "token amnesia". This condition occurs when an encoder fails to correctly retrieve relevant passages because it does not properly encode the relevant token, usually related to a named entity. Unlike previous research, our study adopts a holistic approach, examining model knowledge, activation patterns, and capabilities across different model stages. This analysis approach integrates and makes sense of the different insights from prior works.

7 Conclusion

To reveal possible avenues for improving retrieval augmented generation (RAG) systems, this paper set out to study the purpose served by DPR-style fine-tuning and how DPR-trained BERT operates. We found in Section 2 that middle layers of the LLM impact the performance the most. These layers are where the model is processing a mix of syntactic and semantic features according to (Geva et al., 2021). Through linear probing in Section 3, alongside experiments where we added and removed knowledge from pre-trained BERT in Section 5, we determined that BERT does not appear to acquire new information through DPR fine-tuning. Instead, we observed that the efficacy of retrieval hinges on the activation of shared facts/memories between the BERT models used to encode the query and the context passages. This mechanism implies that incorrect retrieval could occur if a query or context passage inadvertently activates irrelevant or incorrect memories in BERT. Moreover, the absence of necessary facts or webs of knowledge within the model hampers its ability to retrieve information.

However, the crucial insight came in Section 4 from analyzing the changes in BERT's activations before and after DPR-style training. We found that DPR-style training alters the model's internal representation of facts, transitioning from a centralized to a decentralized representation. Pre-trained BERT's representations are very centralized with a select few neurons being activated across a wide ar-

ray of facts and only a few neurons being strongly activated for each fact, suggesting a limited number of pathways for fact or memory activation. The representations in DPR-trained BERT, on the other hand, are a lot less centralized. DPR-trained BERT engages more neurons, more robustly for each fact, and diminishes the uniform reliance on specific neurons across different facts. This decentralization makes it so that each fact/memory has a lot more pathways to get triggered, which in turn allows for more potential inputs to trigger the same set of memories. Such a shift not only underscores the primary objective of DPR training—to diversify the model's retrieval capabilities across an expanded set of queries and passages—but also delineates a crucial mechanism by which these models improve their retrieval performance.

In the most fundamental sense, Dense Passage Retrieval achieves its namesake function—it retrieves, locating and returning relevant context to the user given a query. Yet, as our evidence suggests, DPR models appear constrained to retrieving information based on the knowledge that preexists within their parameters, either innately or through augmentation. This operational boundary delineates a significant caveat: facts must already be encoded within the model for useful context to be accessible by retrieval. Absent these facts or their associative networks, retrieval seems to falter. Thus, if retrieval is understood as the capacity to recall or recognize knowledge already familiar to the model, then indeed, DPR models fulfill this criterion. However, if we extend our definition of retrieval to also encompass the ability to navigate and elucidate concepts previously unknown or unencountered by the model—a capacity akin to how humans research and retrieve information—our findings imply that DPR models fall short of this mark.

Our findings suggest several areas of focus for future work including (1) accelerate knowledge representation decentralization with new unsupervised training methods (2) develop new methods to directly inject facts in a decentralized manner into the network (3) optimize retrieval methods that operate with uncertainty, and (4) map the model's internal knowledge directly to the set of best documents to retrieve.

Current work in optimizing the inverse cloze pre-training task and various data augmentation methods such as (Lin et al., 2023) begin to address (1) by increasing the amount of knowledge that the model is exposed to during fine-tuning and thus the amount of knowledge that can be decentralized. With the knowledge of the purpose of DPR-training more targeted methods can be developed. (3) requires more detailed model analysis to determine how the model processes a query when it is missing key knowledge needed for retrieval. Being aware of when a model is uncertain in its retrieval is crucial. The analysis should reveal methods to more robustly and gracefully handle increased levels of uncertainty. One direction to better leverage a model's knowledge as suggested in (4) is shown in (Tay et al., 2022; Pradeep et al., 2023; Wang et al., 2022; Bevilacqua et al., 2022; Ziems et al., 2023).

8 Limitations

This paper presents a detailed analysis of the DPR formula, specifically focusing on the original DPR training formula utilizing a BERT backbone. We anticipate that our findings will exhibit a degree of generalizability across various DPR implementations, given the underlying commonalities of the core training approach. It is important to recognize that modifications—such as improving hard negatives, different data augmentation techniques, different transformer-based backbones, or leveraging multiple views/vectors from models—while serving to refine and enhance the DPR framework, build upon and amplify the mechanisms of the DPR method. These enhancements, though significant in optimizing performance, are expected not to fundamentally change this analysis. However, it is still a limitation of this paper that we did not repeat our analysis on more DPR-based methods and datasets.

9 Ethics Statements

This work presents an analysis of DPR-style training. Improving DPR-style training would improve RAG pipelines, increasing the factuality of LLMs and decreasing the rate which they hallucinate.

References

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Dai, et al. 2023. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. In *IJCNLP-AACL*, pages 675–718. ACL.

Parishad BehnamGhader, Santiago Miret, and Siva Reddy. 2023. Can retriever-augmented language models reason? the blame game between the retriever and the language model. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15492–15509, Singapore. Association for Computational Linguistics.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive search engines: Generating substrings as document identifiers. *Advances in Neural Information Processing Systems*, 35:31668–31683.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6.

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)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Revanth Gangi Reddy, Vikas Yadav, Md Arafat Sultan, Martin Franz, Vittorio Castelli, Heng Ji, and Avirup Sil. 2022. Towards robust neural retrieval with source domain synthetic pre-finetuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1065–1070, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *EMNLP*, pages 5484–5495, Online and Punta Cana, Dominican Republic. ACL.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *ArXiv*, abs/2002.08909.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2021. Self-attention attribution: Interpreting information interactions inside transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12963–12971.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, CIKM '13, page 2333–2338, New York, NY, USA. Association for Computing Machinery.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023. Transformer-patcher: One mistake worth one neuron. *arXiv preprint arXiv:2301.09785*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Gautier Izacard and Edouard Grave. 2020. Distilling knowledge from reader to retriever for question answering.

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)*, pages 6769–6781, Online. Association for Computational Linguistics.

O. Khattab and Matei A. Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

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*, 33:9459–9474.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. In

Findings of the Association for Computational Linguistics: EMNLP 2023, pages 6385–6400, Singapore. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023a. Lost in the middle: How language models use long contexts. *ArXiv*, abs/2307.03172.

Zheng Liu, Shitao Xiao, Yingxia Shao, and Zhao Cao. 2023b. RetroMAE-2: Duplex masked auto-encoder for pre-training retrieval-oriented language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2635–2648, Toronto, Canada. Association for Computational Linguistics.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

Sean MacAvaney, Sergey Feldman, Nazli Goharian, Doug Downey, and Arman Cohan. 2022. ABNIRML: Analyzing the behavior of neural IR models. *Transactions of the Association for Computational Linguistics*, 10:224–239.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022. Fast model editing at scale. In *International Conference on Learning Representations*.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pre-trained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online. Association for Computational Linguistics.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*.

Ronak Pradeep, Kai Hui, Jai Gupta, Adam Lelkes, Honglei Zhuang, Jimmy Lin, Donald Metzler, and Vinh Tran. 2023. How does generative retrieval scale to millions of passages? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1305–1321, Singapore. Association for Computational Linguistics.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847, Online. Association for Computational Linguistics.

Ori Ram, Liat Bezalel, Adi Zicher, Yonatan Belinkov, Jonathan Berant, and Amir Globerson. 2023. What are you token about? dense retrieval as distributions over the vocabulary. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2481–2498, Toronto, Canada. Association for Computational Linguistics.

Benjamin Reichman and Larry Heck. 2023. Cross-modal dense passage retrieval for outside knowledge visual question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2837–2842.

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

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9248–9274, Singapore. Association for Computational Linguistics.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.

Chenmien Tan, Ge Zhang, and Jie Fu. 2024. Massive editing for large language models via meta learning. In *International Conference on Learning Representations*.

Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. 2022. Transformer memory as a differentiable search index. *Advances in Neural Information Processing Systems*, 35:21831–21843.

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)*.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423, Singapore. Association for Computational Linguistics.

Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, et al. 2022. A neural corpus indexer for document retrieval. *Advances in Neural Information Processing Systems*, 35:25600–25614.

Xing Wu, Guangyuan Ma, Meng Lin, Zijia Lin, Zhongyuan Wang, and Songlin Hu. 2023a. Contextual masked auto-encoder for dense passage retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4738–4746.

Xing Wu, Guangyuan Ma, Peng Wang, Meng Lin, Zijia Lin, Fuzheng Zhang, and Songlin Hu. 2023b. Cot-mae v2: Contextual masked auto-encoder with multi-view modeling for passage retrieval. *ArXiv*, abs/2304.03158.

Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao. 2022. RetroMAE: Pre-training retrieval-oriented language models via masked auto-encoder. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 538–548, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Retrieval meets long context large language models. In *The Twelfth International Conference on Learning Representations*.

Shunyu Zhang, Yaobo Liang, Ming Gong, Daxin Jiang, and Nan Duan. 2022. Multi-view document representation learning for open-domain dense retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5990–6000, Dublin, Ireland. Association for Computational Linguistics.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024. Dense text retrieval based on pretrained language models: A survey. *ACM Trans. Inf. Syst.*, 42(4).

Noah Ziems, Wenhao Yu, et al. 2023. Large language models are built-in autoregressive search engines. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2666–2678, Toronto, Canada. ACL.

A Appendix

A.1 Context Model Activations

Figure 5 depicts the activation patterns observed in the context model, mirroring the trends outlined in Section 4. The only exception occurs in the first intermediate layer of the pre-trained BERT model, where a larger number of neurons are activated as compared to DPR-trained BERT.

A.2 Model Activations at different thresholds

Figures 6, 7, 8, 9, and 10 illustrate neuron activation patterns across varying activation thresholds set at $0.005*\max(Attr)$, $0.01*\max(Attr)$, $0.05*\max(Attr)$, $0.2*\max(Attr)$, and $0.3*\max(Attr)$, respectively. As the threshold increases from 0.005 to 0.3, the visualization narrows down to neurons with stronger activations. This observation reinforces the findings discussed in Section 4: pre-trained BERT shows a trend of fewer but more consistently activated neurons, in contrast to DPR-trained BERT, which exhibits a broader array of neurons activated less frequently.

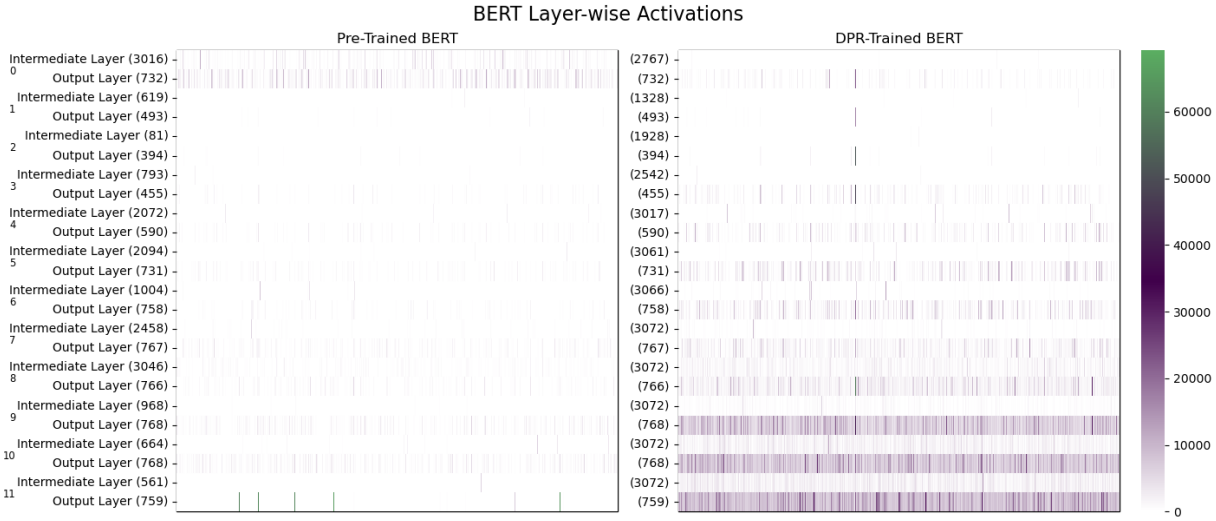

Figure 5: Layerwise activations for pre-trained and DPR-trained BERT - context model. The parenthetical numbers indicate the number of neurons in the layer that are above the attribution threshold for any number of examples.

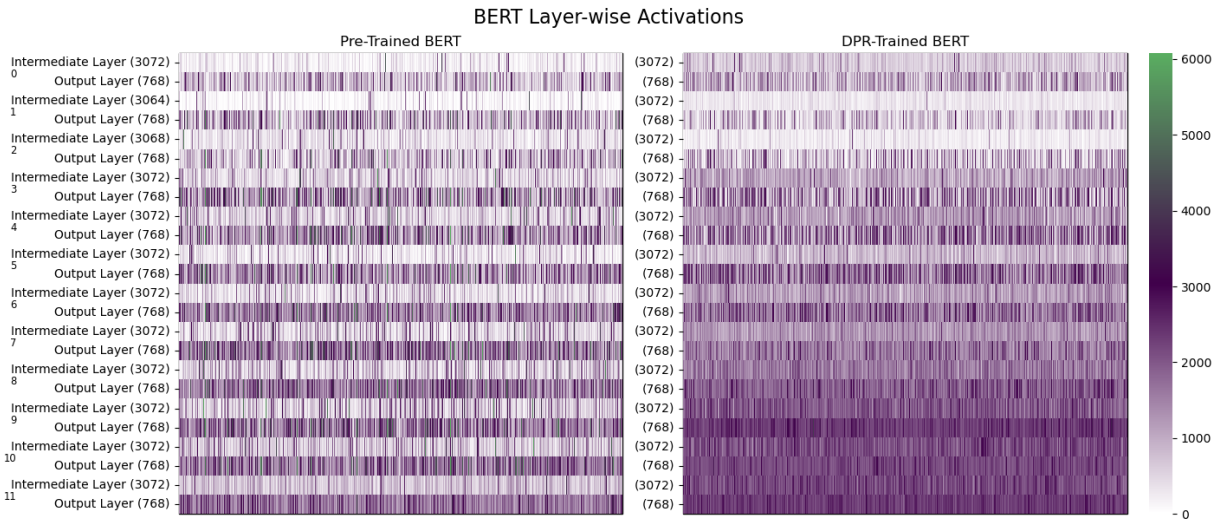

Figure 6: Layerwise activations for pre-trained and DPR-trained BERT with a threshold of 0.005. The parenthetical numbers indicate the number of neurons in the layer that are above the attribution threshold for any number of examples.

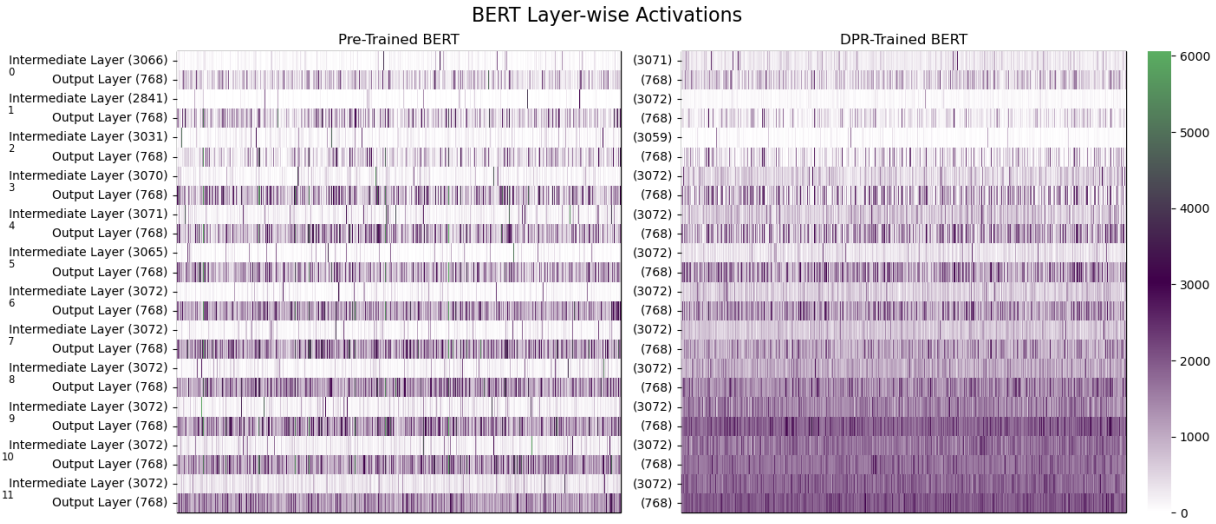

Figure 7: Layerwise activations for pre-trained and DPR-trained BERT with a threshold of 0.01. The parenthetical numbers indicate the number of neurons in the layer that are above the attribution threshold for any number of examples.

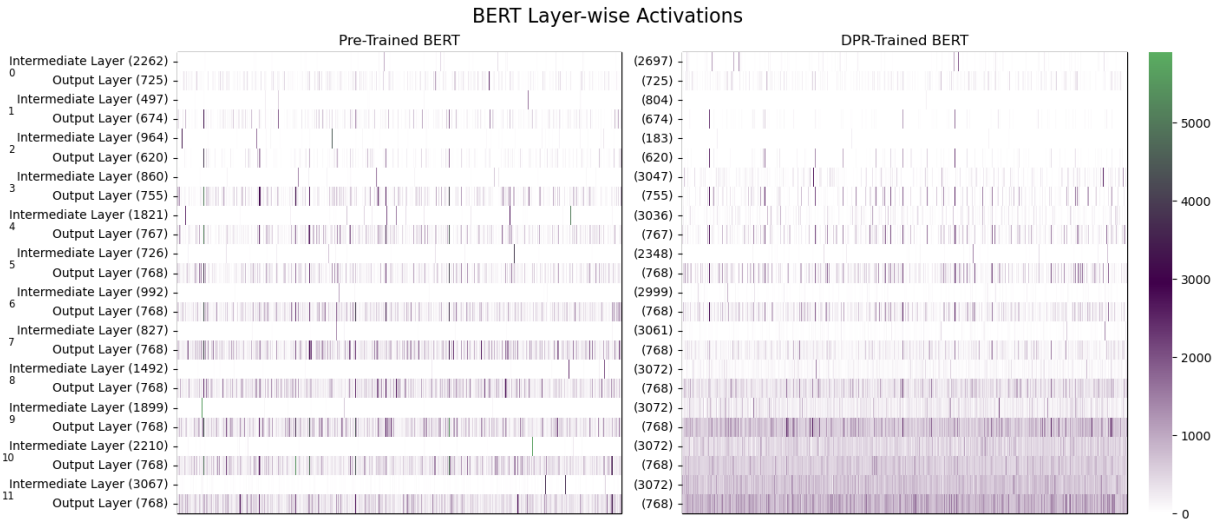

Figure 8: Layerwise activations for pre-trained and DPR-trained BERT with a threshold of 0.05. The parenthetical numbers indicate the number of neurons in the layer that are above the attribution threshold for any number of examples.

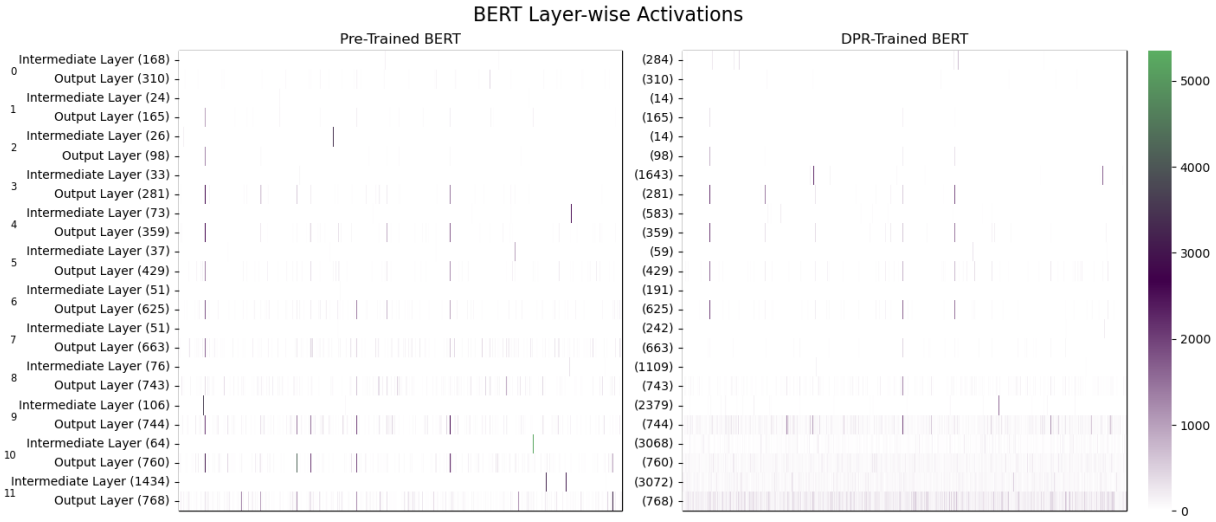

Figure 9: Layerwise activations for pre-trained and DPR-trained BERT with a threshold of 0.2. The parenthetical numbers indicate the number of neurons in the layer that are above the attribution threshold for any number of examples.

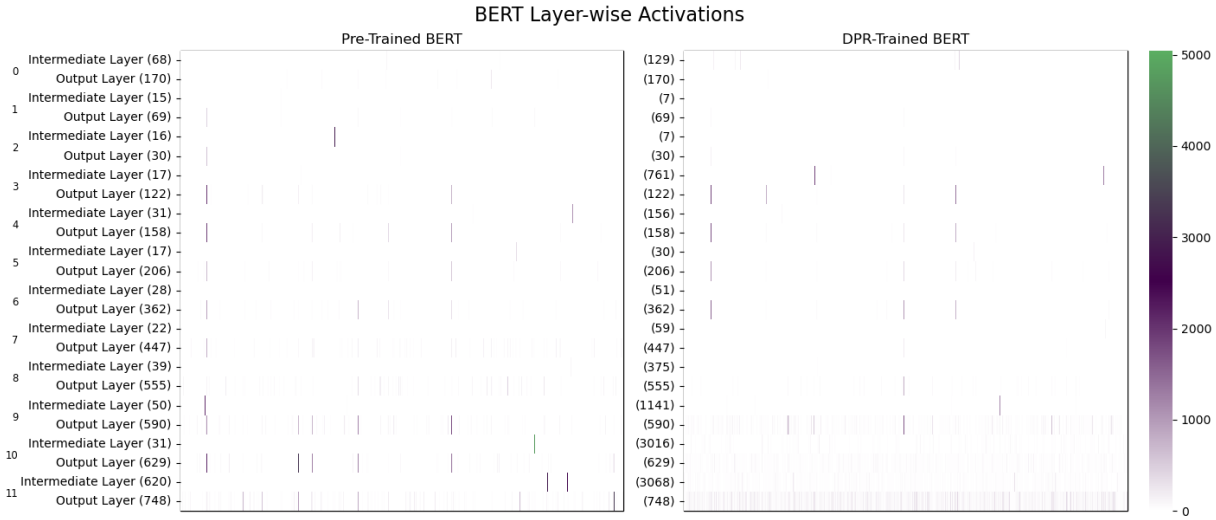

Figure 10: Layerwise activations for pre-trained and DPR-trained BERT with a threshold of 0.3. The parenthetical numbers indicate the number of neurons in the layer that are above the attribution threshold for any number of examples.