

Dense Passage Retrieval: Is it Retrieving?

Benjamin Reichman

Georgia Institute of Technology
bzs@gatech.edu

Larry Heck

Georgia Institute of Technology
larryheck@gatech.edu

Abstract

Large Language Models (LLMs) internally store repositories of knowledge. However, their access to this repository is imprecise and they frequently hallucinate information that is not true or does not exist. A paradigm called Retrieval Augmented Generation (RAG) promises to fix these issues. Dense passage retrieval (DPR) is the first step in this paradigm. In this paper, we analyze the role of DPR fine-tuning and how it affects the model being trained. DPR fine-tunes pre-trained networks to enhance the alignment of the embeddings between queries and relevant textual data. 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 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, an intermediary step is introduced. 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 this paradigm 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

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 DPR-BERT Query Model	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 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
	Pre-trained BERT DPR-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
3-Passage Probing	Pre-trained BERT 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.

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 introduced in the DPR method, inspiring many subsequent methods in the literature.

In this paper, we analyze the original DPR method using the BERT-base backbone. Through the experiments in each section, we analyze DPR from multiple perspectives to understand what is changing in the backbone model during the training process. We find that:

1. BERT’s ability to discriminate between passages remains unchanged (Section 2).
2. The way that knowledge is structured in BERT changes to be more decentralized after DPR training (Section 3).
3. The knowledge that BERT can retrieve over is an extension of the knowledge contained within it (Section 4).

2 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 probe

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

was trained to classify which passage (between two, three, four, or five passages) is most relevant to a query. The Natural Questions dataset was used in this experiment, which provides labels for which passages are hard-negatives. The probe g_{lN} at layer l received BERT features for the query, true positive passage, and N hard negative passages. f_{lq} represents the features at layer l for the query, f_{ltp} represents the features for the true positive paragraph at layer l , and f_{lhnN} represents the features for the N th 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.

The difference between a true positive passage and a hard-negative passage is usually the presence of 1-2 key distinct facts in the passage. The

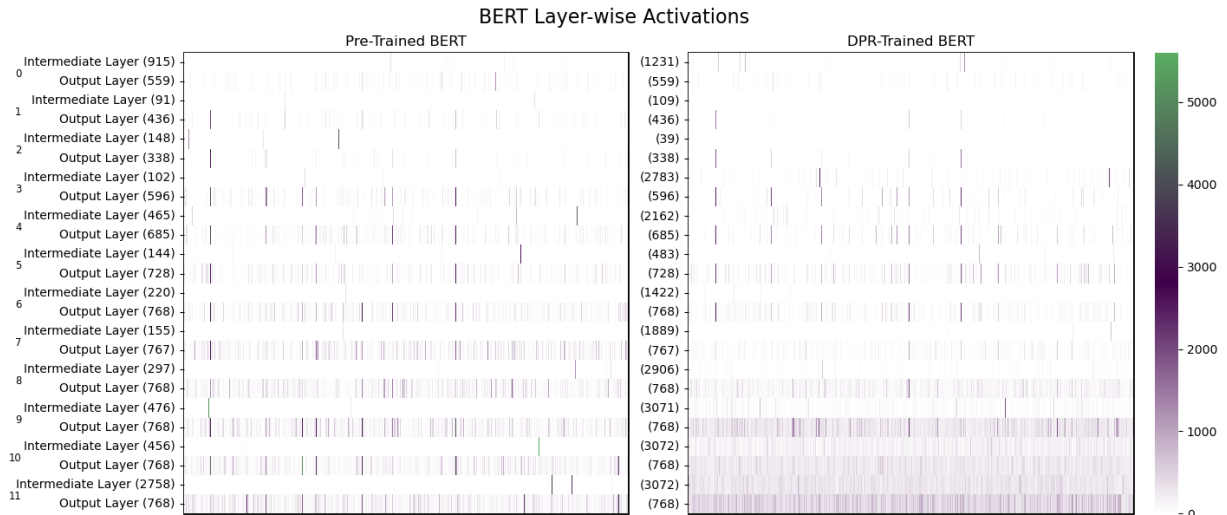


Figure 1: Layerwise activations for pre-trained and DPR-trained BERT. The parenthetical numbers indicate the number of neurons in the layer that are above the attribution threshold for any number of examples.

ability to discriminate between 2-5 of these passages **indicates that the model likely has enough knowledge to know which facts are relevant to the query. This awareness is likely driven by the model’s knowledge of the subject** (as discussed in later sections of this paper). Rather than testing overall retrieval ability, this experiment aims to find how aware/knowledgeable pre-trained BERT’s features are compared to DPR-trained BERT when the difference of knowing or not knowing 1-2 facts can impact the final matching prediction.

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 capability to discern relevant from irrelevant passages, when they are directly presented to the model, are likely already present in pre-trained BERT, and DPR-style training does not substantially enhance these discriminative features.**

3 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; Sun-

dararajan 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:

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

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 neurons¹. 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 in the broader activation patterns. When the model is processing inputs, both "true-positive" and "false-positive" knowledge neurons are activated indiscriminately with the activation strength corresponding to their attribution scores. The primary interest lies in how DPR training influences these activation patterns, rather than the role of specific neurons.

Figure 1 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 model². DPR-trained

¹Appendix A.2 demonstrates that our observations are consistent across a spectrum of thresholds.

²Appendix A.1 shows that the observations made in this section also hold for the context model.

Query	Answer in Top-1?		# Strongly Activated Neurons		Title of Top-5 Retrieval	
	Pre-trained BERT	DPR-BERT	Pre-trained BERT	DPR-BERT	Pre-trained BERT	DPR BERT
where is the most distortion on a robinson projection	✗	✗	220	1323	Circle of latitude, Scale-invariant feature transform, Line moiré, Theil-Sen estimator, Pole splitting	Robinson projection, Robinson projection, Arthur H. Robinson, Robinson projection, Arthur H. Robinson
who is the chief legal advisor to the government	✗	✗	65	831	Jimly Asshiddiqie, Judicial system of Iran, Comptroller General of the State Administration, Jimly Asshiddiqie, Law of Kosovo	Attorney General of India, Attorney general, Attorney General of India, K. K. Venugopal, Attorney General of India
what type of government does kenya have 2018	✓	✗	74	287	Government of Kenya, Abundant Nigeria Renewal Party, 2007-08 Kenyan crisis, Independent Electoral and Boundaries Commission, Kingdom of Kongo	Government of Kenya, Politics of Kenya, Government of Kenya, Government of Kenya, Government of Kenya
are pure metals made of atoms or ions	✓	✗	69	1268	Alloy, Common attributes, Metal, Resonance ionization, Alloy	Properties of metals, metalloids and non-metals, Properties of metals, metalloids and nonmetals, Solid, Metal, Metal
who is the bad guy in lord of the rings	✗	✓	100	533	Millennium Earl, The Sword of Shannara, Eye of Ra, The Enchanted Apples of Oz, Ys I & II	Saruman, Saruman, Sauron, Morgoth, Legolas
when were manatees put on the endangered list	✗	✓	42	1522	Ivory trade, Namib Desert Horse, Endangered Species Act of 1973, Iriomote cat, Bile bear	Manatee conservation, Endangered Species Act of 1973, Endangered Species Act of 1973, Manatee conservation, Endangered Species Act of 1973
when did wesley leave last of the summer wine	✓	✓	38	1024	Naif (band), Aiden, Queensrÿche, Josef Brown, Matthew Stocke	Gordon Wharmby, Gordon Wharmby, Brian Wilde, Cory Monteith, Last of the Summer Wine
when did mozart compose his first piece of music	✓	✓	74	364	Wolfgang Amadeus Mozart, Der Messias, Life of Franz Liszt, Die Entführung aus dem Serail, Quattro versioni originali della Ritirata notturna di Madrid	Wolfgang Amadeus Mozart, Wolfgang Amadeus Mozart, Leopold Mozart, Wolfgang Amadeus Mozart, Wolfgang Amadeus Mozart

Table 2: This table presents example queries alongside the corresponding model retrievals and the count of strongly activated neurons for both pre-trained and DPR-trained BERT. Notably, DPR training consistently increases the number of strongly activated neurons. Additionally, the retrievals, even when DPR does not retrieve the correct passage in the top-1 retrieval, are much more focused and targeted to the asked query. In contrast, pre-trained BERT’s retrievals are much more varied and sporadic. This is likely because in pre-trained BERT each neuron that is activated is responsible for more information and the model has no fine-grained path to follow for specific information like it does after DPR training.

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.**

Table 2 demonstrates the effects of DPR training through performing retrieval with various queries and the full corpus of 21M Wikipedia passages. Across all instances, DPR training increases the number of strongly activated neurons indicating the existence of more pathways in the network allowing for better access to the information needed to perform retrieval. When examining the titles of the retrieved passages, a marked difference is revealed between pre-trained BERT and DPR BERT. Pre-trained BERT's retrievals are often disparate, aligning with the query in some instances while seemingly unrelated in others. This inconsistency indicates that successful retrievals by pre-trained BERT may hinge on the activation pattern precisely aligning with the relevant article. On the other hand, DPR-BERT, consistently retrieves passages that are topically related to the query, even if they are not the exact best match, reflecting a better ability to hone in on pertinent information. **By having more neurons responsible for each query the model has more fine-grained control to find relevant passages, even if what is found is not the most relevant possible passage.** In the cases where it was not able to navigate to the exact correct passage it is possible that the knowledge needed to discern between the correct and incorrect passage is not in the model.

4 Knowledge Editing Experiments

If DPR rearranges knowledge in the pre-trained BERT model, are these facts discoverable in DPR-BERT? We employed model editing techniques to remove facts from pre-trained BERT to investigate this.

Currently, models struggle to learn and update the knowledge within them. Model editing is a class of techniques that minimizes pre-trained models to add or remove knowledge without disturbing the overall network. The types of methods generally used for model editing can be broadly categorized as fine-tuning with regularization, direct model editing, meta-learning, and architectural (Mazzia et al., 2023).

Owing to the emerging state of this subfield and the variability in results, we employed various model editing techniques. In selecting techniques, we prioritized locality-preserving methods that minimally altered the model. This was to facilitate clearer attributions of our findings to DPR training rather than to potential architectural modifications. This led to TransformerPatch, MalMen, and Mend being chosen to perform the model editing (Huang et al., 2023; Tan et al., 2024; Mitchell et al., 2022). TransformerPatch is an architecture-modifying technique that introduces a single parameter to the last layer for each fact added. The experiments involving TransformerPatch used a learning rate of $3e-5$. MalMen and Mend, on the other hand, are meta-learning techniques. They utilize hypernetworks, a type of network that learns to generate parameters or parameter shifts for another model (Ha et al., 2016). These hypernetworks were trained using a learning rate of $1e-6$ and to only modify the last three layers of the BERT network.

4.1 Knowledge Removal

To select the facts for removal, we identified questions from the NQ dataset that both DPR-BERT and the probed pre-trained BERT correctly answered. For each of the identified questions, we removed one fact from 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.

A total of 284 queries, which both DPR-BERT and the linear probes had accurately matched with their corresponding passages, were randomly se-

Knowledge Editing Technique	DPR Removed
Transformer-Patch	0.87
MalMen	0.81
Mend	1.00

Table 3: Outcomes of the knowledge removal experiments. Of the facts successfully removed from BERT, the "DPR Removed" column is the percentage of the facts that were also absent after DPR training.

lected. 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.

Fact removal success was determined by using the linear probing task from Section 2 on the edited pretrained model. If the probe was able to still pick out the correct passage after fact removal, that indicated that the fact or its associative network was not fully removed. If the network’s features (and hence its internal knowledge) were no longer sensitive enough to complete the task, then the fact was likely removed.

Of the facts successfully removed from the pretrained BERT model by the knowledge editing techniques, Table 3 shows that 81% – 100% were also absent in DPR-BERT. When a fact is removed it and likely part of its associative network is fully removed. As DPR training primarily functions to decentralize knowledge in a network, if the fact does not exist in the network there is nothing to decentralize in the network. DPR training does not appear to add knowledge to the network, once a fact is removed, it cannot be recovered through DPR training.

The knowledge removal experiment demonstrates that **DPR training primarily refines how pre-existing knowledge within BERT is rendered more "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.**

5 Related Works

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). CoBERT 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 author’s 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.

6 Conclusion

This paper analyzes the effect of DPR fine-tuning using different analysis methods. Through these analyses, we find that:

1. DPR training alters how knowledge is stored in the model (Section 3).
2. DPR likely does not change how much knowledge is contained in the model (Section 2).
3. DPR’s ability to retrieve knowledge is an extension of the knowledge contained in the model (Section 4)

DPR training refines and restructures how the model stores information in the network transitioning from a centralized pattern of storage to a decentralized pattern. 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. By engaging more neurons, more robustly for each

fact, the model diminishes the uniform reliance on a specific neuron for a specific fact. This allows the model to embed different semantic forms of a query into a vector that will still map to a similar set of relevant passages, instead of requiring a specific ordering of words to trigger a specific neuron in order to get the relevant passage.

Additionally, we find that BERT does not acquire new knowledge through DPR fine-tuning. Knowledge can be removed before DPR fine-tuning to enhance the model, but the training process **does not** seem to change the knowledge content of a model. Facts that are removed stay removed and are unreachable. Being able to discriminate between two pieces of text, like in the linear probing experiment of Section 2, in a classification task is about using the knowledge contained in the network to tell two pieces of text apart. The knowledge to understand the one to two fact difference in pairs (or more) of text is apparent in the features of both pre-trained and DPR-trained BERT. This piece of evidence indicates that DPR’s aim is not to learn new facts, but to align disparate pieces of text in a common embedding space and restructure how knowledge is accessed to facilitate that.

If DPR does not add facts, then we should be able to remove relevant facts that pre-trained BERT knows and not find them again in DPR-trained BERT. Section 4 indicates that this statement is true, suggesting from another perspective, that DPR is dependent on the knowledge already contained in the model to produce satisfactory retrievals. If a fact is removed, it will likely not be added back during DPR training. Thus the existence or absence of necessary facts or webs of knowledge within a model can aid or hamper its ability to retrieve information.

In the most fundamental sense, DPR 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 or on the periphery of what is familiar, then indeed, DPR models ful-

fill 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 the current DPR model 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 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 that can be decentralized. With the knowledge of the purpose of DPR-training more targeted methods can be developed. (3) requires more detailed analysis to determine how a 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).

7 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 not expected 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.

8 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, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. [A multi-task, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity](#). In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718, Nusa Dua, Bali. Association for Computational Linguistics.
- 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.

- 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, 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.
- David Ha, Andrew Dai, and Quoc V Le. 2016. [Hypernetworks](#). [arXiv preprint arXiv:1609.09106](#).
- 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](#).
- 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. [AB-NIRML: Analyzing the behavior of neural IR models](#). [Transactions of the Association for Computational Linguistics](#), 10:224–239.
- Vittorio Mazzia, Alessandro Pedrani, Andrea Caciolai, Kay Rottmann, and Davide Bernardi. 2023. [A survey on knowledge editing of neural networks](#). [arXiv preprint arXiv:2310.19704](#).
- 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 Bezalet, 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.
- 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 Ziemis, 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 2 depicts the activation patterns observed in the context model, mirroring the trends outlined in Section 3. 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 3, 4, 5, 6, and 7 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 3: 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.

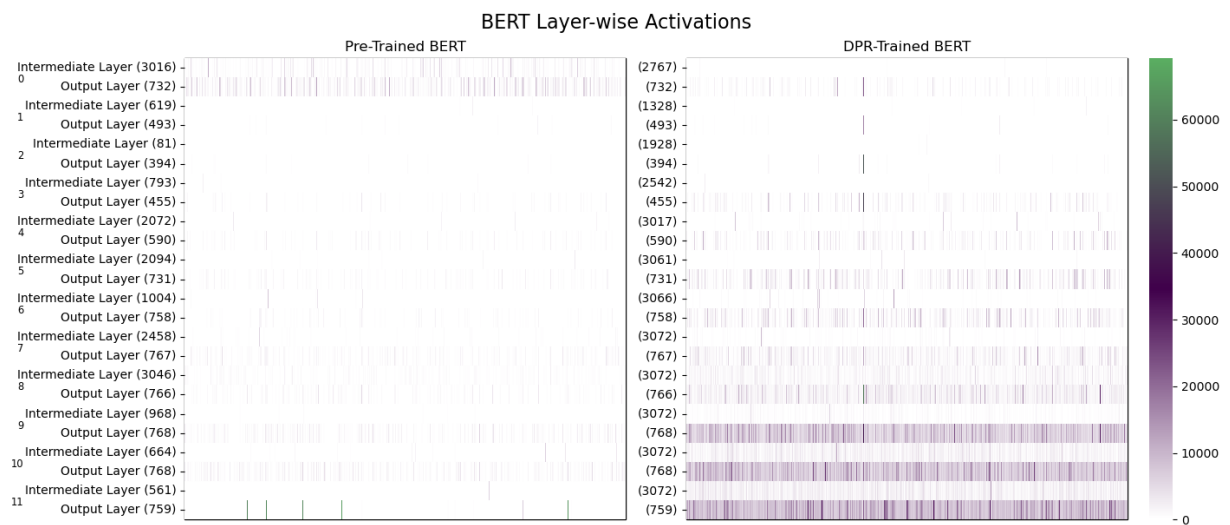


Figure 2: Layerwise activations for pre-trained and DPR-trained BERT - context model. The parenthetical numbers indicate the number of neurons in the layer that are above the attribution threshold for any number of examples.

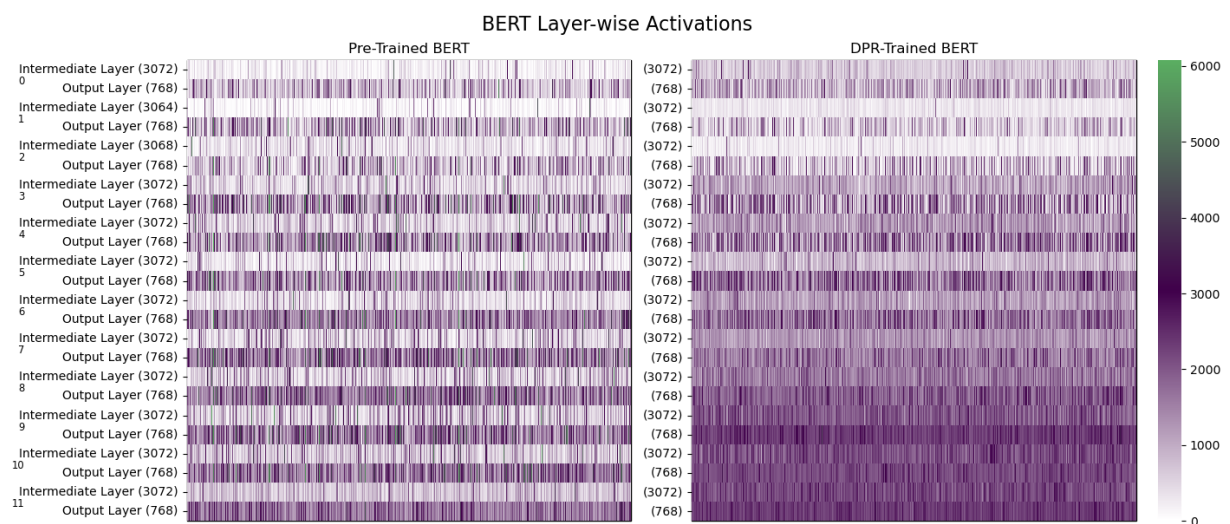


Figure 3: Layerwise activations for pre-trained and DPR-trained BERT with a threshold of 0.005. The parenthetical numbers indicate the number of neurons in the layer that are above the attribution threshold for any number of examples.

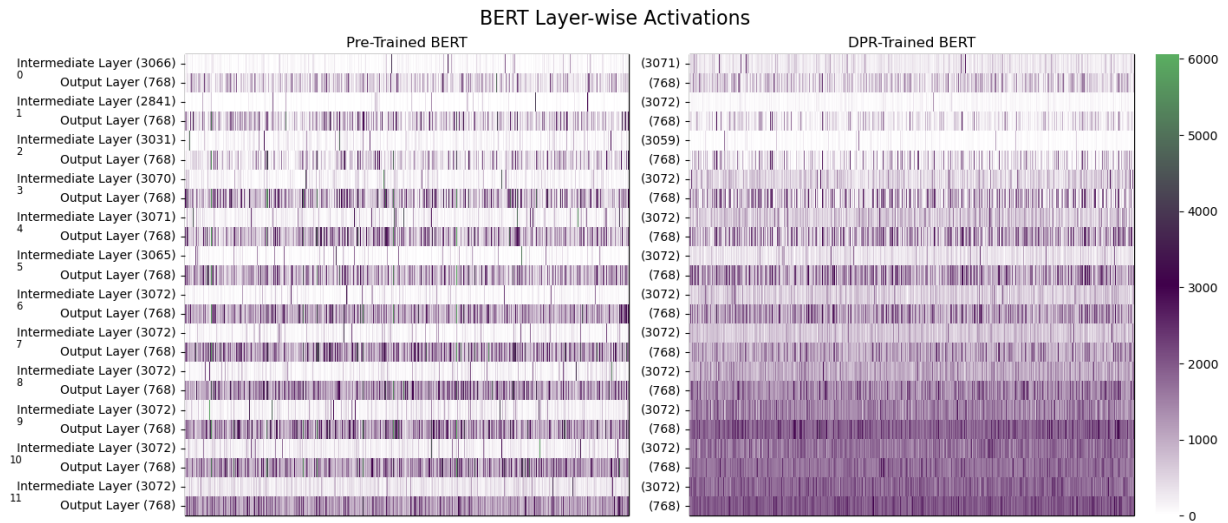


Figure 4: Layerwise activations for pre-trained and DPR-trained BERT with a threshold of 0.01. The parenthetical numbers indicate the number of neurons in the layer that are above the attribution threshold for any number of examples.

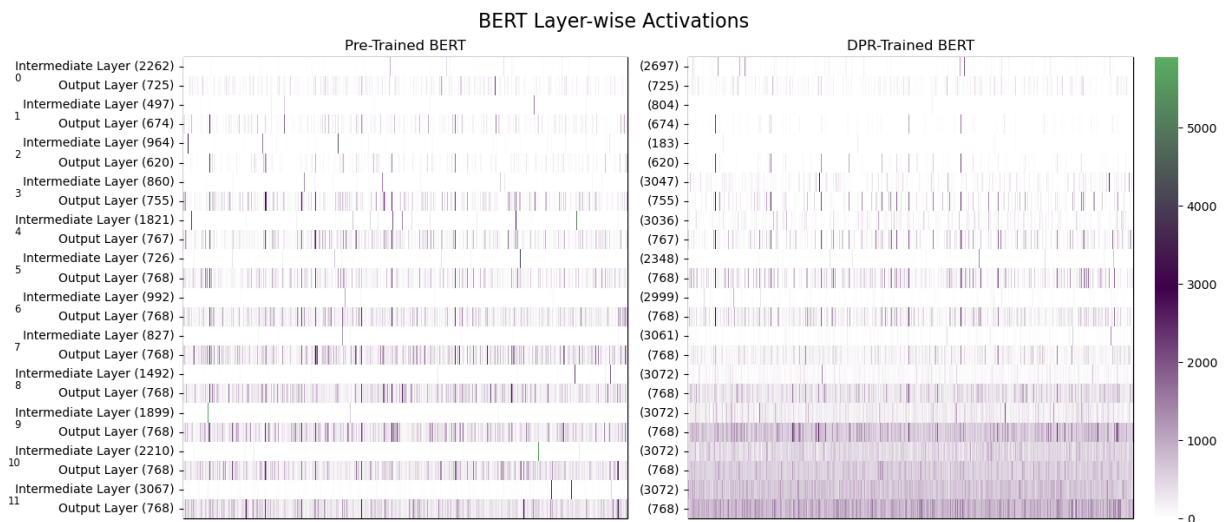


Figure 5: Layerwise activations for pre-trained and DPR-trained BERT with a threshold of 0.05. The parenthetical numbers indicate the number of neurons in the layer that are above the attribution threshold for any number of examples.

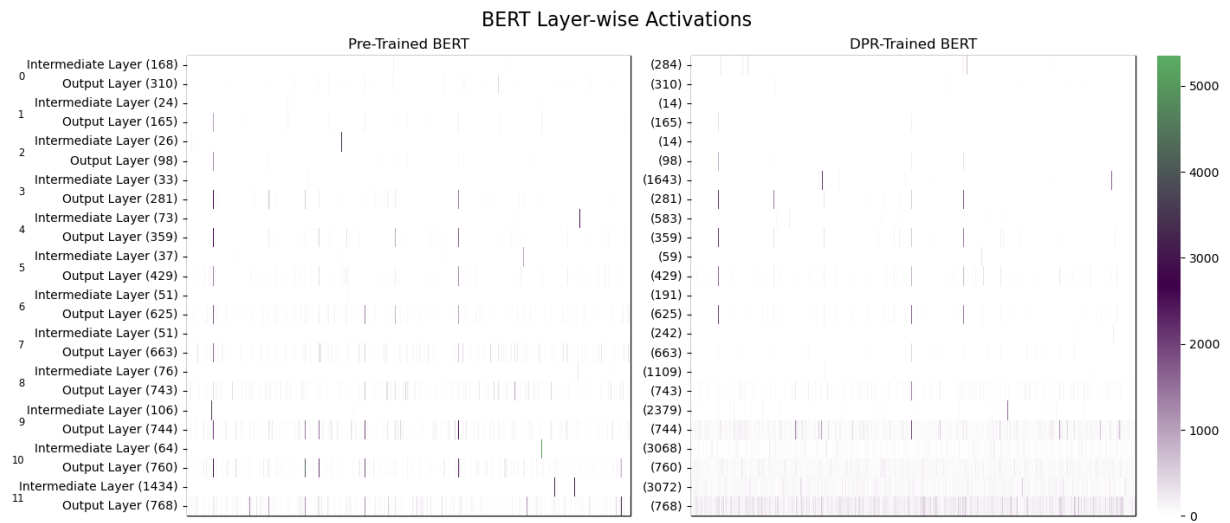


Figure 6: Layerwise activations for pre-trained and DPR-trained BERT with a threshold of 0.2. The parenthetical numbers indicate the number of neurons in the layer that are above the attribution threshold for any number of examples.

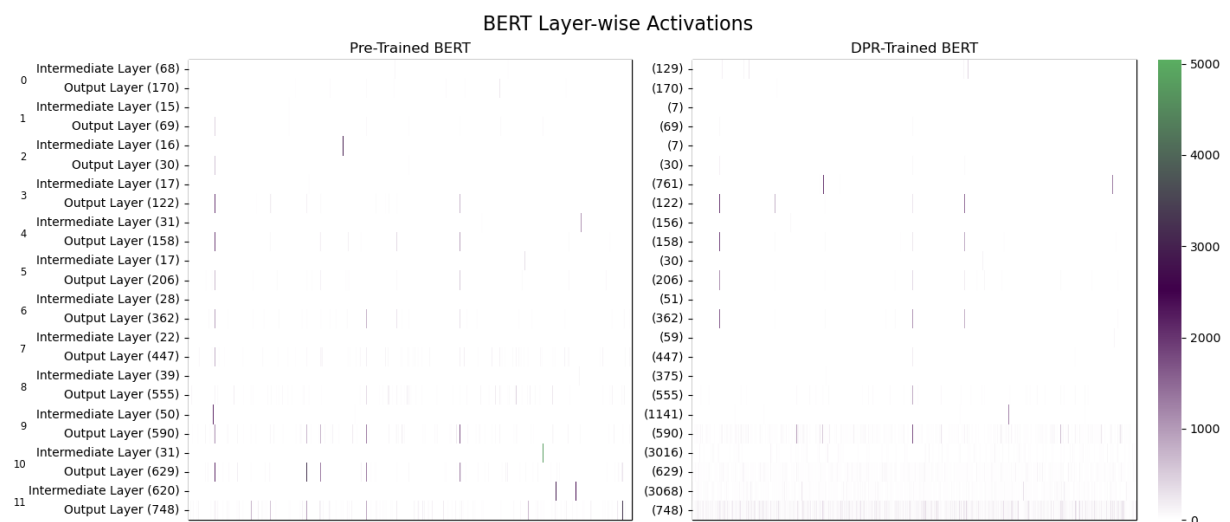


Figure 7: Layerwise activations for pre-trained and DPR-trained BERT with a threshold of 0.3. The parenthetical numbers indicate the number of neurons in the layer that are above the attribution threshold for any number of examples.