The Power of Noise: Redefining Retrieval for RAG Systems

Florin Cuconasu
cuconasu@diag.uniroma1.it
Sapienza University of Rome
Rome, Italy

Giovanni Trappolini
trappolini@diag.uniroma1.it
Sapienza University of Rome
Rome, Italy

Federico Siciliano
siciliano@diag.uniroma1.it
Sapienza University of Rome
Rome, Italy

Simone Filice
simone.filice@tii.ae
Technology Innovation Institute
Haifa, Israel

Cesare Campagnano
campagnano@di.uniroma1.it
Sapienza University of Rome
Rome, Italy

Yoelle Maarek
yoelle.maarek@tii.ae
Technology Innovation Institute
Haifa, Israel

Nicola Tonellotto
nicola.tonellotto@unipi.it
University of Pisa
Pisa, Italy

Fabrizio Silvestri
fsilvestri@diag.uniroma1.it
Sapienza University of Rome
Rome, Italy

ABSTRACT

Retrieval-Augmented Generation (RAG) systems represent a significant advancement over traditional Large Language Models (LLMs). RAG systems enhance their generation ability by incorporating external data retrieved through an Information Retrieval (IR) phase, overcoming the limitations of standard LLMs, which are restricted to their pre-trained knowledge and limited context window. Most research in this area has predominantly concentrated on the generative aspect of LLMs within RAG systems. Our study fills this gap by thoroughly and critically analyzing the influence of IR components on RAG systems. This paper analyzes which characteristics a retriever should possess for an effective RAG's prompt formulation, focusing on the type of documents that should be retrieved. We evaluate various elements, such as the relevance of the documents to the prompt, their position, and the number included in the context. Our findings reveal, among other insights, that including irrelevant documents can unexpectedly enhance performance by more than 30% in accuracy, contradicting our initial assumption of diminished quality. These results underscore the need for developing specialized strategies to integrate retrieval with language generation models, thereby laying the groundwork for future research in this field.[1]

CCS CONCEPTS

• **Information systems** → *Retrieval models and ranking*; *Novelty in information retrieval*; *Retrieval models and ranking*.

[1]The code and data are available at github.com/florin-git/The-Power-of-Noise

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

KEYWORDS

RAG, LLM, Information Retrieval

ACM Reference Format:
Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2018. The Power of Noise: Redefining Retrieval for RAG Systems. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 11 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Large Language Models (LLMs) [5] have demonstrated unprecedented proficiency in various tasks, ranging from text generation and complex question-answering [3], to information retrieval (IR) tasks [14, 44]. However, LLMs are limited in handling large contexts [40], a constraint that leads to an increased reliance on their pre-trained knowledge. This limitation not only confines their ability to effectively manage extended discourse, such as in books or long conversations but also increases the probability of generating hallucinations, instances where the model produces factually incorrect or nonsensical information [33]. To improve the accuracy of responses generated by LLMs, Retrieval-Augmented Generation (RAG) systems have emerged as a promising solution [19]. These systems are primarily designed to improve factual accuracy by providing the model access to external information instead of solely depending on the knowledge infused by the pre-training phase, which may also be limited or outdated. A key advantage of RAG systems is their ability to increase the effective context size for LLMs. They achieve this by incorporating an IR component that dynamically sources relevant external information during the response generation process. This approach significantly expands the range of data accessible to the model, extending its context window beyond the initial input. At their core, RAG systems are made of two fundamental components: retriever and generation. The IR component is responsible for sourcing external information to enrich the input for the generation module. In contrast, the generation component leverages the power of LLMs to produce coherent and contextually relevant text. This study concentrates on the IR aspect of RAG systems, and we pose the following research question: *"What essential characteristics are required in a retriever to optimize prompt construction for RAG systems? Are current retrievers ideal?"*. We focus on the three main types of documents that a retriever can fetch: *relevant*, *related*, and *irrelevant*. *Relevant* documents contain directly pertinent information to the query, offering gold-standard data that directly answers or informs the query. *Related* documents, while not directly answering the query, are semantically or contextually linked to the topic. For instance, if one asks for the color of Napoleon's horse, a document expressing the color of Napoleon's wife's horse, while not containing the right information, would be highly related. *Irrelevant* documents, on the other hand, are unrelated to the query, representing a kind of informational noise within the retrieval process. Our analysis finds that related documents are more harmful than unrelated ones in RAG systems. Even more surprisingly, we discover that noisy documents are beneficial and cause an improvement by up to 35% in terms of accuracy. These results are in contrast with the standard customer-facing usage of IR systems, where related documents are typically perceived as more acceptable than unrelated ones. The analysis suggests that conventional retrieval techniques may not be optimal in this new paradigm, necessitating the development of specialized approaches tailored to the specific demands of integrating retrieval with language generation models. These insights highlight the potential for novel research in the field, paving the way for a systematic rethinking and advancement of IR strategies in the context of RAG systems. Summing up, our contributions are: **(a)** We carry out the first comprehensive study focusing on how the retrieved documents impact the RAG frameworks. We aim to understand the characteristics required in a retriever to optimize prompt construction for a RAG system; **(b)** This study finds that related documents are more harmful than irrelevant in RAG systems. Indeed, contrary to conventional wisdom, we found that noisy (irrelevant) documents can improve performance by up to 35% in terms of accuracy; **(c)** We propose strategies to take advantage of this phenomenon. At the same time, we highlight the need to reconsider information retrieval strategies, paving the way for future research efforts.

2 RELATED WORKS

2.1 Generative Language Models

The inception of the modern LLM era can be traced back to the seminal paper titled "Attention Is All You Need" [40]. This work introduced the transformer architecture, a framework that adopts an attention mechanism instead of recurrent layers, enabling the model to capture global dependencies within the data. Following this innovation, 2018 witnessed the introduction of BERT (Bidirectional Encoder Representations from Transformers)[14]. BERT represented a significant advancement in the domain of context understanding within text, utilizing a deeply bidirectional, unsupervised language representation. The evolution of transformer-based models continued with the development of the Generative Pre-trained Transformer (GPT) [27]. Its successor, GPT-2 [28], expanded upon this foundation with a larger scale model and demonstrated improved performance across a variety of language tasks without task-specific training. The subsequent iteration, GPT-3 [5], represented a further enhancement in model scale and capabilities, particularly in the realm of few-shot learning. Concurrently, a proliferation of specialized models and variants built upon and extended the transformer architecture. Notably, Google's T5 (Text-to-Text Transfer Transformer) [29] proposed a unified framework for NLP tasks by reframing them as a text-to-text problem. XLNet [43], developed by Google and CMU in the same year, surpassed BERT's performance by employing a permutation-based training approach. Finally, recent times have seen a surge in the production of large, publicly available language models. Several actors have released their models, most notably, Llama [38, 39], Falcon [1], Mosaic MPT [37], and Phi [10, 20].

2.2 Information Retrieval

The field of IR originated with a focus on text-based systems. Foundational methodologies, such as the Vector Space Model (VSM) [42] and term frequency-inverse document frequency (TF-IDF) [31], provided a basis for quantifying textual similarity. These sparse retrieval methods, with BM25 being its most famous current iteration [32], characterized by their use of high-dimensional and sparse feature vectors, have been essential in developing early IR systems. A significant evolution in IR is the distinction between sparse and dense retrievers. Sparse retrievers, like VSM and TF-IDF, rely on exact keyword matching and are efficient for large-scale document retrieval due to their interpretability and simplicity. However, they often struggle with understanding semantic relationships between words [22]. In contrast, dense retrievers, emerging from advancements in deep learning, utilize low-dimensional dense vectors for representation. One of the firsts to actually improve on sparse methods was DPR [12], which has been followed by a plethora of other techniques [9, 16].

2.3 Retrieve and Generate

RAG represents a significant shift in machine learning, combining the strengths of both retrieval-based and generative models. The idea had first originated in works such as [6] and [47], but the concept of RAG was popularized in [19], which introduced a model that combined a dense passage retriever with a sequence-to-sequence model, showcasing substantial improvements in knowledge-intensive tasks. Similar methods have almost concurrently emerged, such as [4, 7]. We refer the reader to [23] for a survey on augmented language models. Researchers and practitioners have recently started to explore these RAG systems' inner workings. Notably, [34] analyzed the impact of different types of documents on cascading IR/NLP systems. However, they were not using LLMs, and due to this, they reached almost opposite conclusions. Other works have tried to study how attentive transformers are to their input [15, 21, 30, 36]. In this paper, we want to provide the first comprehensive analysis of the implications of using a retriever module in a RAG system, studying the impact of several key factors, like the position of the documents, their type, and how many of them are used.

3 RAG

RAG offers a powerful framework that can be successfully applied in many problems and downstream tasks. In this paper, we explore the application of RAG in the context of Question Answering, arguably its most popular application.

3.1 Open-Domain Question Answering

Open-Domain Question Answering (OpenQA) refers to the task of developing systems capable of providing accurate and contextually relevant answers to a broad range of questions posed in natural language without limitations to specific domains or predefined datasets. In general, we want to find an answer $\mathcal{A}$ to a query q. To do so, we draw information from a corpus of documents $\mathcal{D} = \{d_1, d_2, \dots, d_n\}$, which is usually assumed to be large in size. A prevalent approach to solving the task effectively involves a two-step architecture, typically comprising a retriever and a reasoner, also known as a generator. This methodology addresses the inherent complexities of OpenQA by dividing the process into distinct phases of information retrieval, finding the correct set of documents, and answer synthesis, which provides the actual response.

3.2 Retriever

The retrieval component plays a critical role in the task of OpenQA. Its goal is to find a sufficiently small subset of documents $\mathcal{D}_r$ that should help the reasoner to answer the query correctly. Among the various retrieval methodologies, the use of a dense retriever has gained prominence due to its effectiveness in handling complex and diverse queries. The core principle of dense retrieval involves transforming textual data into high-dimensional vector representations. This is typically achieved with a neural network, often a transformer-based encoder, like BERT [14]. The dense retriever processes both the query q and potential source documents to generate corresponding embeddings $\vec{q}$ for the query and $\vec{d_i}$ for each document $d_i \in \mathcal{D}$. The embedding process can be represented as:

$$\vec{q} = Encoder_q(q); \ \vec{d_i} = Encoder_d(d_i)$$

where $Encoder_q$ and $Encoder_d$ are neural network-based encoders, potentially sharing weights or architecture, designed to map the textual data into a vector space. Once the embeddings are generated, the retrieval process involves computing the similarity between the query embedding and each document embedding. The most common approach is to use dot product [13], defined as: $s(q, d_i) = \vec{q} \cdot \vec{d_i}$. This score quantifies the relevance of each document to the query by measuring their similarity in the embedded vector space, with higher scores indicating greater relevance. The documents are then ordered based on these scores, and the top-ranked documents are selected for further processing in the generator component.

3.3 Reasoner

The second step involves a generator or answer synthesis component; this is usually a generative LLM in RAG. These models are designed to produce a coherent, contextually relevant, and semantically accurate text in response to a given query, also referred to as prompt. Generative language models operate by predicting the probability distribution of the next token, given the previous tokens. For a given sequence of words $w_1, w_2, \dots, w_n$, a generative language model aims to maximize the likelihood of this sequence, expressed using the chain rule of probability:

$$P(w_1, w_2, \dots, w_n) = \prod_{i=1}^{N} P(w_i | w_1, w_2, \dots, w_{i-1})$$

where $P(w_i | w_1, w_2, \dots, w_{i-1})$ is the conditional probability of the word w_i given the preceding sequence of words $w_1, w_2, \dots, w_{i-1}$. In RAG, the generative language model takes a query q and the retrieved documents $\mathcal{D}_r$ as input and generates a response by sequentially predicting the next token in the sequence. More formally,

$$P_{rag}(y|q) \approx \prod_{i}^{N} \sum_{d \in \mathcal{D}_r} p_\eta(d|q) p_\theta(y_i | q, d, y_{1:i-1}),$$

where $p_\eta(d|q)$ is the retrieval component that provides a (truncated) probability distribution for the top-scoring documents, and $p_\theta(y_i | q, d, y_{1:i-1})$ is a probability distribution parameterized by θ that generates a current token based on the previously generated tokens, the query, and the retrieved document; this role is filled by the LLM. In the case of dense retrieval, the probability distribution for the top-scoring documents may assume a functional form of the kind $p_\eta(d|q) \propto \exp(\vec{q} \cdot \vec{d})$. Given our formalization of the RAG task, we notice how the generative component p_θ depends on a given text, that is the query, and a dynamic text, that is the set of retrieved documents. In this paper, we want to study how changing the set of retrieved documents affects the generative component and, consequently, the whole end-to-end system. In particular, we aim to find the best set of documents $\mathcal{D}_r$ that an effective retriever should supply to this component to maximize the system's performance. In Section 4, we construct an experimental framework that will allow us to study this behavior. In Section 5, we collect the experimental evidence and analyze the results.

4 EXPERIMENTAL METHODOLOGY

In this section, we detail the experimental framework. We will start by describing the data used in the experiments. This sets the stage for examining the type of documents that a retriever can return and pass to the LLM, which will be the main focus of this section.

4.1 Natural Question Dataset

The Natural Questions (NQ) dataset [17] is a large-scale collection of real-world queries derived from Google search data. Each entry in the dataset consists of a user query and the corresponding Wikipedia page containing the answer. Designed to facilitate research in natural language understanding and open-domain question answering, this dataset provides a rich source of real-world questions and contextually relevant answers. The NQ-open dataset [18], a subset of the NQ dataset, differs by removing the restriction of linking answers to specific Wikipedia passages, thereby mimicking a more general information retrieval scenario similar to web searches. This open-domain nature significantly impacts our experimental design, particularly in the selection and categorization of documents. Following the methodology of Lee et al. [18], our primary source for answering queries is the English Wikipedia dump as of 20 December 2018. Consistently with the Dense Passage Retrieval (DPR) approach [13], each Wikipedia article in this dump was segmented into non-overlapping passages of 100 words. A significant challenge in open-domain question answering is the potential temporal mismatch between the Wikipedia dump and the question-answer pairs in the dataset, which can lead to cases where the dump does not contain the answer, as highlighted in the AmbigQA study [24]. To mitigate this, we integrated the gold documents from the original NQ dataset into our Wikipedia document set. Given the open-domain nature of our task, there may be additional documents *relevant* to the query, i.e., containing the answer, but we will *not* consider them as *gold*. The final dataset comprises 21, 035, 236 documents, with 72, 209 queries in the train set and 2, 889 in the test set.

4.2 Types of Documents

In our study, we categorize documents into four distinct types, each represented by a unique symbol, based on their relevance and relationship to the queries:

★ *Gold Document*. The gold document, identified by ★, refers to the original context in the NQ dataset, specifically the passage of a Wikipedia page containing the answer and contextually relevant to a given query.

Relevant Documents. Denoted by , relevant documents are passages that, akin to the gold document, contain the correct answer and are contextually useful for answering the query. They provide additional sources of information that are correct and pertinent to the query. Notably, the gold document is a relevant document.

Related Documents. Symbolized by , related documents are semantically similar to the query but do not contain the correct answer. They serve a crucial role in evaluating the generator's proficiency in discerning between relevant and non-relevant information.

Irrelevant Documents. Indicated by , irrelevant documents are neither related to the query nor contain the answer. They are instrumental in assessing the model's ability to handle completely unrelated information. In practice, in our tests, we will randomly sample these documents from the corpus.

In our analysis, the entire set of documents fetched by the retriever is represented by the symbol . This possibly encompasses all document types — gold, relevant, related, or irrelevant — and serves to discuss the retrieval output in a generalized manner without specifying individual document categories.

4.3 Document Retrieval

Our methodology utilizes a two-step approach in line with a typical RAG setting, as explained in Section 3.2. As the first component, our experiments use *Contriever* [9], a BERT-based dense retriever, as the default retriever. It is trained without supervision using a contrastive loss. To enhance the efficiency of similarity searches within our corpus, comprising about 21 million documents, we also employ the FAISS IndexFlatIP indexing system. The embedding of each document and query is obtained by averaging the hidden state of the last layer of the model.

4.4 LLM Input

Upon receiving a query, the retriever selects the top-k documents from the corpus according to a given similarity measure. These documents, in conjunction with the task instruction and the query, constitute the input for the LLM to generate a response. The NQ-open dataset was structured to include only those queries whose answers consist of no more than five tokens [18]. Consequently, the LLM is tasked with extracting a response, confined to a maximum of five tokens, from the provided documents. The template for this prompt, as depicted in Figure 1, begins with the task instruction, presented in italics for clarity. This is followed by the *context*, which comprises the selected documents. The arrangement, with the query succeeding the documents, aligns with the methodological approach outlined in [21]. While the composition of the context will vary according to the single experiment, the instruction will always be placed at the beginning of the prompt, and the query always at the end.

LLM Input - Only Gold ★

You are given a question and you MUST respond by EXTRACTING the answer (max 5 tokens) from one of the provided documents. If none of the documents contain the answer, respond with NO-RES.
Documents:
Document [3](Title: Millennium Falcon) Han Solo won the Millennium Falcon from Lando Calrissian in the card game sabacc...

Question: who owned the millennium falcon before han solo
Answer: Han Solo

Figure 1: Example LLM input with an erroneous output, highlighted in red. The input consists of an *italicized task instruction*, followed by the context (documents), and the query. The LLM's response is marked under 'Answer'. The gold color highlights both the gold document and the correct answer, "Lando Calrissian", indicating the expected source and content of the accurate response.

4.5 LLMs Tested

We consider several LLMs in our experiments. Consistently across all models, we adopt a greedy generation approach with a maximum response length of 15 tokens. Acknowledging the constraints imposed by memory and computational resources, we have implemented a model quantization strategy, reducing all models to a 4-bit representation[2]. Beyond the prompt described above, the models are not provided with additional exemplars for few-shot learning, which, while of interest and object of vivid research efforts, are outside the scope of this paper. *Llama2*. The 7B parameters version of the Llama2 family [39] shows state-of-the-art performance on

[2]We use the bitsandbytes library integrated into HuggingFace.

most downstream tasks compared to models of the same size. It was trained with a 4096 tokens context window and uses multi-query attention [35]. *Falcon.* Introduced in [1], Falcon is a 7B parameters model, trained on the RefinedWeb dataset [25], a large, filtered and deduplicated corpus. Similarly to Llama2, it uses multi-query attention, with a context length of 2048 tokens. *Phi-2.* This is the smallest model used in this work (2.7B parameters). Despite its modest size, it achieves performance comparable to the other models [10, 20], thanks to its pre-training on "textbook-quality" data. It has a context window of 2048 tokens. *MPT.* This 7B parameters model uses ALiBi attention [26, 37] for a virtually unlimited context length. In our experiments, to leverage the model's full potential, we set the limit to 2048 tokens, i.e., the same used for the model's pre-training.

4.6 Accuracy

The NQ-open dataset allows a range of potential answers for each query. Frequently, these answers are different variants of the same concept (e.g., "President D. Roosevelt" or "President Roosevelt"), while in some cases, a single query may admit multiple distinct correct answers. To evaluate the accuracy of responses generated by LLMs, we use an assessment technique in line with [11, 21]. This methodology examines whether at least one of the predefined correct answers is contained within the response produced by the LLM. We measure the correctness of the LLM's responses as either accurate or inaccurate based on the presence of the answer in a binary fashion. Nevertheless, this evaluation strategy is not without challenges. A principal issue arises in determining response correctness, particularly in instances involving date representations or varying phrasings conveying identical meanings. For example, if the LLM generates "Roosevelt" in response to a query where the established correct answer is "President Roosevelt", the response would be deemed incorrect under our current evaluation schema. Recognizing this limitation, we acknowledge the necessity for a more advanced analysis of answer variations, which we leave to future research.

5 RESULTS

Studying the characteristics of optimal prompts for RAG systems corresponds to answering this primary research question **(RQ)**: "*What are the essential characteristics required in a retriever to optimize prompt construction for RAG systems?*" We decompose this question into sub-questions corresponding to testing various prompt combinations, focusing on those components of the prompt that are directly influenced by the retrieval mechanism. This will allow us to assess how different elements of the prompt interact with and are shaped by the retrieval process. To enhance clarity and comprehension of our experimental setup, we will employ a streamlined schema to represent the composition of prompts. This schema is represented as follows: **[I, ★, 🔗, ⛓️‍💥, 🎲, Q]**. In this model, the task instruction (I) and the query (Q) are consistently positioned at the beginning and end, respectively. The middle section varies and represents different contextual elements - in this instance, these are gold, relevant, related, and irrelevant, appearing in that specific sequence. Additionally, the quantity of these contextual documents will be a variable in its own right and will be reported in the results table.

5.1 Impact of Related Documents

LLM Input - Related ⛓️‍💥 and Gold ★

Task Instruction...
Documents:
Document [1](Title: Han Solo) Before the events of the film, he and Chewbacca had lost the "Millennium Falcon" to thieves, but they reclaim the ship after it...
Document [2](Title: Millennium Falcon) The "Falcon" has been depicted many times in the franchise, and ownership has changed several times...
Document [3](Title: Millennium Falcon) Han Solo won the Millennium Falcon from Lando Calrissian in the card game sabacc...

Question: who owned the millennium falcon before han solo
Answer: Han Solo

Figure 2: Example LLM input with an erroneous output, highlighted in red. The context of the prompt is composed of related documents and the gold near the query. The task instruction is as in Figure 1.

We start our inquiry by conducting an experiment using a selection of 10K queries from the training set of the NQ-open dataset. In this first experiment, we assume an oracle setup in which the gold document for the query is known. To this, we add a set of related documents, documents that are assigned a high score by the retriever but do not contain the answer, with the goal of measuring their impact on the system; schematically **[I, ⛓️‍💥, ★, Q]**. Figure 2 shows an example of this setup's visualization. Results of this experiment are seen in Table 1 (far, mid, and near relate to the distance between the gold document and the query; more details in the following sub-section). A critical observation emerging from this investigation is a clear pattern of progressive accuracy degradation across all LLMs tested as the number of related documents included in the context increases, with accuracy deteriorating of more than 0.38 (−67%) in some cases. Even more importantly, adding just one related document causes a sharp reduction in accuracy, with peaks of 0.24 (−25%), as can be seen by comparing the row with 0 related documents (only gold scenario, as seen in Figure 1) with that of 1 related document. This experiment highlights a critical issue for RAG systems, particularly in real-world IR settings where related but non-answer-containing documents are commonplace. Our empirical analysis suggests that introducing semantically aligned yet non-relevant documents adds a layer of complexity, potentially misguiding LLMs away from the correct response. A visual explanation can be seen in Figure 3, which illustrates the attention scores within the prompt's context for a specific example where the LLM incorrectly answers. This figure highlights the model's disproportionate focus on a related document (leftmost) at the expense of the gold document (rightmost), likely contributing to the erroneous

Table 1: Accuracy results of the LLMs when evaluated with prompts composed of the gold document ★ and a varying number of related 🔗 documents. The table illustrates how the inclusion of an increasing number of related documents affects LLM's performance. Scenarios where the prompt exceeded the model's input limit, leading to potential data truncation, are not included (-). All values *not* marked with an asterisk * denote statistically significant changes from the gold-only document scenario [I, ★, Q] (first row), as determined by a Wilcoxon test (p-value < 0.01). Additionally, the closed-book accuracy scores for the models are as follows: Llama2 (0.1123), MPT (0.1205), Phi-2 (0.0488), Falcon (0.1083).

	Far - [I, ★, 🔗, Q]				Mid - [I, 🔗, ★, 🔗, Q]				Near - [I, 🔗, ★, Q]			
# 🔗	Llama2	MPT	Phi-2	Falcon	Llama2	MPT	Phi-2	Falcon	Llama2	MPT	Phi-2	Falcon
0	**0.5642**	0.2148	**0.4438**	**0.4330**	**0.5642**	**0.2148**	**0.4438**	**0.4330**	**0.5642**	**0.2148**	**0.4438**	**0.4330**
1	0.4586	0.1976	0.3585	0.3469	no-mid	no-mid	no-mid	no-mid	0.4283	0.1791	0.4227	0.3602
2	0.3455	0.1913	0.3430	0.3246	0.3322	0.1802	0.3375	0.2823	0.3974	0.2002	0.3975	0.3111
4	0.2745	**0.2209***	0.3019	0.2670	0.2857	0.1775	0.2885	0.2378	0.3795	0.2059*	0.3701	0.2736
6	0.2898	0.2171*	0.2943	0.2392	0.2698	0.1424	0.2625	0.2103	0.3880	0.1892	0.3623	0.2656
8	0.2643	0.2077*	0.2513	0.1878	0.2268	0.1002	0.2360	0.1745	0.3748	0.1944	0.3423	0.2424
10	0.2537	-	-	-	0.2180	-	-	-	0.3716	-	-	-
12	0.2688	-	-	-	0.2382	-	-	-	0.3991	-	-	-
14	0.2583	-	-	-	0.2280	-	-	-	0.4118	-	-	-
16	0.2413	-	-	-	0.2024	-	-	-	0.3889	-	-	-
18	0.2348	-	-	-	0.1795	-	-	-	0.3781	-	-	-

response. The interpretation of our results must account for the differing input token capacities of the LLMs used. Llama2 can process up to 4096 tokens, but other models are limited to 2048 tokens. This led to the exclusion of evaluations with a higher number of related documents (# 🔗 $\geq$ 10), as reflected by the empty values in the tables. These omissions were necessary to prevent input truncation and ensure comparability across models.

Impact of Retrieval with Hard Negatives using ADORE [46]. Our analysis of the impact of related but non-answer-containing documents on LLM performance revealed a significant reduction in accuracy. These documents, obtained from the top-ranked outputs of the Contriever that did not contain the correct answer, led us to hypothesize that this performance degradation is closely linked to the nature of "negative" documents encountered during the retriever's training phase. To explore this hypothesis, we conducted experiments with a retriever trained using "hard negatives"—specifically, documents that closely resemble relevant information but are deliberately designed to be non-relevant. This training approach aims to distinguish between relevant and related information more clearly. For this purpose, instead of Contriever, we employed ADORE [46], a state-of-the-art retriever trained with "dynamic hard negatives", to select the related documents 🔗 in our experimental setup. Results clearly show that ADORE cannot replace Contriever as the IR component, even if ADORE is fine-tuned to discern among closely related documents. In scenarios with 1, 2, and 4 related documents (# 🔗) in the **[I, 🔗, ★, Q]** setting with Llama2, we obtain an accuracy of 0.4068, 0.3815, and 0.3626, respectively. This is significantly lower than the baseline accuracy of 0.5642 where no related documents were included, and than the results obtained with the Contriever in the same settings. Therefore, it is clear that distinguishing between relevant and related information is an open problem in this setting that cannot be challenged by using models like ADORE that are designed to better discern among closely related documents.

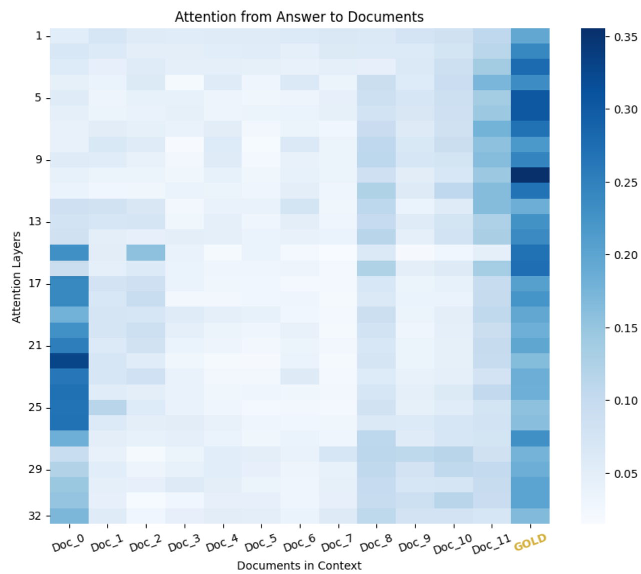

Figure 3: This heatmap depicts the attention distribution across the context documents from the example shown in Figure 2, relative to the answer generated by Llama2-7b in a prompt structured as [I, 🔗, ★, Q]. Cell (i, j) denotes the mean attention that tokens in the generated answer allocate to the tokens of the i-th document within the j-th attention layer. This mean attention for each document is calculated by averaging the attention scores across all its constituent tokens.

Table 2: Accuracy results of the LLMs when evaluated with prompts composed of the gold document ★ and a varying number of irrelevant ⚅ documents. Surprisingly, increasing the number of irrelevant documents in the Near setting improves LLM's performance. Scenarios where the prompt exceeded the model's input limit, leading to potential data truncation, are not included (-). All values *not* marked with an asterisk * denote statistically significant changes from the gold-only document scenario [I, ★, Q] (first row), as determined by a Wilcoxon test (p-value < 0.01). Additionally, the closed-book accuracy scores for the models are as follows: Llama2 (0.1123), MPT (0.1205), Phi-2 (0.0488), Falcon (0.1083).

	Far - [I, ★, ⚅, Q]				**Mid - [I, ⚅, ★, ⚅, Q]**				**Near - [I, ⚅, ★, Q]**			
# ⚅	Llama2	MPT	Phi-2	Falcon	Llama2	MPT	Phi-2	Falcon	Llama2	MPT	Phi-2	Falcon
0	**0.5642**	0.2148	**0.4438**	**0.4330**	**0.5642**	0.2148	**0.4438**	**0.4330**	0.5642	0.2148	0.4438	**0.4330**
1	0.4733	0.2447	0.4329	0.4035	no-mid	no-mid	no-mid	no-mid	0.4862	0.2125*	0.4587	0.4091
2	0.3776	0.2639	0.4249	0.3805	0.3928	**0.2584**	0.4293	0.3612	0.5032	0.2660	**0.4614**	0.3912
4	0.3109	0.2933	0.4091	0.3468	0.3998	0.2577	0.3985	0.3462	0.5221	**0.2930**	0.4311	0.3949
6	0.3547	0.3036	0.4130	0.3250	0.4138	0.2265	0.3891	0.3196	0.5681*	0.2890	0.4388	0.3908
8	0.3106	**0.3039**	0.3812	0.2543	0.3734	0.1566	0.3596	0.2767	0.5609*	0.2911	0.4258	0.3704
10	0.3390	-	-	-	0.3675	-	-	-	0.5579*	-	-	-
12	0.3736	-	-	-	0.3641	-	-	-	0.5836	-	-	-
14	0.3527	-	-	-	0.3372	-	-	-	**0.5859**	-	-	-
16	0.3401	-	-	-	0.3159	-	-	-	0.5722	-	-	-
18	0.3466	-	-	-	0.2982	-	-	-	0.5588*	-	-	-

5.2 Impact of Gold Positioning

We conduct another experiment where we systematically shift the position of the gold document within the context to study its impact on the model's performance. In particular, we define the positions as follows: **Near**, the gold document is placed adjacent to the query in the prompt **[I, ⚡, ★, Q]** (as in Figure 2); **Mid**, the gold document is inserted in the middle of the context **[I, ⚡, ★, ⚡, Q]**; **Far**, the gold document is positioned as far as possible from the query in the context **[I, ★, ⚡, Q]**. Results in this setting partially corroborate evidence from [21]. The accuracy is higher when the gold document is near the query while it is lower when the gold document is furthest from it, and it is the lowest when the gold document is placed in the middle of the context; for instance, Llama2 with 18 related documents obtains accuracies of 0.37, 0.23, and 0.17 respectively. These results are consistent across all models tested in the setting with related documents. However, the situation differs in the setting with irrelevant documents that we are going to explore in the next sub-section.

LLM Input - Random ⚅ and Gold ★

Task instruction...
Documents:
Document [140](Title: Richard Yates (novelist)) For much of his life, Yates's work met almost universal critical acclaim, yet not one of his books sold over 12,000 copies in...
Document [242](Title: Android version history) Code name Version number Initial release date API level Security patches (No codename) 1.0 September 23...
Document [3](Title: Millennium Falcon) Han Solo won the Millennium Falcon from Lando Calrissian in the card game sabacc...

Question: who owned the millennium falcon before han solo
Answer: Lando Calrissian

Figure 4: Example LLM input with a correct output, highlighted in green. The context of the prompt is composed of random documents and the gold near the query. The task instruction is as in Figure 1.

5.3 Impact of Noise

We craft an experimental setting aimed at evaluating the robustness of RAG systems against noise. Specifically, we tested how much the performance of RAG systems deteriorates when random, irrelevant documents are introduced into the context. In practice, we take the gold document and add to it a certain number of documents picked at random from the corpus. Subverting our expectations, the performance did not deteriorate in the presence of noise, as can be seen in Table 2. Instead, we observed an "intriguing" improvement in performance under the best-performing setting (near **[I, ⚅, ★, Q]**), with an improvement of 0.08 (+36%) in the case of MPT. Figure 4 shows an example of this setup. Furthermore, we observe that different models exhibit distinct behaviors. Both Llama2 and Phi-2 showed improvements in this setting when the noise was introduced furthest from the query. However, when the noise was positioned in the far **[I, ★, ⚅, Q]** and mid **[I, ⚅, ★, ⚅, Q]** settings, these models exhibited a decline in performance. Notably, this performance degradation was much less accentuated when compared to the earlier setting with related documents. This suggests that while Llama2 and Phi-2 can effectively handle noise far from the query, their ability to sift through irrelevant information diminishes as the noise is placed closer to it. The MPT model presented a unique

Table 3: Accuracy of Llama2-7b in configurations involving random Wikipedia documents and retrieved documents [SP, ⚃, 🗎, Q]. Rows denote the number of irrelevant documents ⚃ added, and columns show the quantity of retrieved documents 🗎. The left section reports results using the Contriever, and the right section using BM25. Scenarios where the prompt exceeded the model's input limit, leading to potential data truncation, are not included (-). Each value *not* marked with an asterisk * represents a statistically significant change from the base case of retrieved documents only [SP, 🗎, Q] (first row), as determined by a Wilcoxon test (p-value < 0.01).

	Contriever							BM25						
# ⚃ \ # 🗎	1	2	3	4	5	8	10	1	2	3	4	5	8	10
0	0.1620	0.1866	0.1876	0.1866	0.1921	0.2198	0.2108	0.2008	0.2208	0.2084	0.2028	0.2243	0.2492	0.2447
1	0.1308	0.1616	0.1717	0.1893*	0.1987*	0.2153*	0.2146*	0.1568	0.1963	0.1921	0.2115	0.2295*	0.2475*	0.2506*
2	0.1315	0.1644	0.1859*	0.2008	0.2174	0.2156*	0.2368	0.1644	0.1973	0.2080*	0.2281	0.2558	0.2495*	0.2596
3	0.1301	0.1727	0.2008	0.2316	0.2201	0.2198	0.2409	0.1568	0.2063	0.2160	0.2520	0.2579	0.2644	0.2707
5	0.1464	0.2056	0.2233	0.2240	0.2150	**0.2451**	**0.2482**	0.1772	0.2402	0.2437	0.2520	0.2554	0.2804	**0.2866**
8	0.1734	0.2066	0.2336	0.2375	0.2454	0.2416	0.2364	0.1994	0.2451	0.2579	0.2769	0.2817	**0.2859**	0.2777
10	0.1796	0.2174	0.2450	0.2502	**0.2499**	0.2420	-	0.2108	0.2589	0.2734	0.2835	**0.2935**	0.2853	-
15	0.2018	0.2354	0.2551	**0.2530**	-	-	-	0.2243	0.2686	0.2790	**0.2928**	-	-	-
16	0.2032	**0.2471**	**0.2558**	-	-	-	-	0.2323	0.2662	**0.2838**	-	-	-	-
17	0.2039	0.2426	-	-	-	-	-	**0.2326**	**0.2693**	-	-	-	-	-
18	**0.2073**	-	-	-	-	-	-	0.2309	-	-	-	-	-	-

response; it showed an improvement in performance under all settings (near, mid, and far). Standing out from the rest, the Falcon model did not exhibit the improvement in performance as observed in other models with the introduction of noise. Peculiarly enough, it and Llama2 did not exhibit a "lost in the middle" phenomenon consistently, having in some instances better accuracy in the mid than far setting, for instance, in the case with 8 noisy documents added. Motivated by the results obtained so far that subverted our initial expectations, we developed a new set of results that help us shed some light on our research question.

5.4 RAG in Practice

To effectively explore our Research Question (**RQ**) regarding the defining characteristics of an effective RAG retriever, and following the results observed with the inclusion of irrelevant documents, we transition from the oracle setup to a more realistic scenario. In this setting, given a query, we retrieve a set of documents 🗎, noticing that these can either be relevant 🔗 or related ⛓. We then proceed to add irrelevant documents to this set of retrieved ones; schematically: **[I, ⚃, 🗎, Q]**. Results for this experiment, using Llama2, can be seen on the left side of Table 3. These results show that, regardless of the number of retrieved documents, adding irrelevant documents up until the context length is filled is almost always beneficial, with gains in terms of accuracy up to 0.07 (+35%) in the case of 4 retrieved documents.

5.4.1 Testing Sparse Retrievers. In an effort to validate our initial observations, we replicated our experiment using a sparse retrieval approach, specifically BM25. The corresponding results are delineated in the right section of Table 3. Consistent with earlier findings, we observed an enhancement in LLM performance by including irrelevant documents. Notably, the use of BM25 yielded an average increase in accuracy of 3-4 percentage points. This improvement is attributed to the quality of documents retrieved by BM25. We quantitatively evaluated the effectiveness of the retrieval methods by computing the top-k accuracy for varying numbers of retrieved documents 🗎. It is pertinent to note that this heuristic, while indicative, does not capture the full spectrum of relevance. Our evaluation, based on the presence of normalized correct answers within documents, might overlook the context-specific relevance due to potential lexical matches of the answer string in irrelevant documents. Despite this limitation, this method aligns with established computational practices in literature [9, 12]. In our analysis, BM25 demonstrated higher relative top-k accuracies (0.2966, 0.4105, 0.6663 for $k = 1, 2, 10$) compared to those of the Contriever (0.2502, 0.3569, 0.6085 for the same k), underscoring its effectiveness in retrieving more relevant documents for our experimental setup.

5.4.2 Increasing The Randomness. Our previous experiments show that the addition of irrelevant documents improves performance. However, one might argue that these documents are not really irrelevant as they originate from the same corpus (Wikipedia) and that they might induce the LLM to answer in a fashion that is more homogeneous with that particular corpus, not introducing substantial noise. For this reason, we carry out another experiment in which irrelevant documents are drawn from the Reddit Webis-TLDR-17 dataset [41]. We chose this dataset as it represents a stark contrast to Wikipedia in both tone and style. The results are outlined on the left of Table 4. The inclusion of documents from the Reddit corpus not only maintained the observed increase in accuracy but even enhanced it, with an improvement of 0.023 (+9% accuracy) when comparing the two best scores. Pushing the randomness even further, we carry out another test where we consider nonsensical sentences made up of random words as irrelevant documents. Remarkably, even in this scenario, we observed a performance improvement when compared to the base case of Wikipedia irrelevant documents, as documented by the right side of Table 4. This outcome further refutes the notion that the semantic nature or the type

Table 4: Accuracy of Llama2-7b in configurations involving irrelevant documents and retrieved documents by the Contriever [SP, ⚅, 📄, Q]. Rows denote the number of irrelevant documents ⚅ added, and columns show the quantity of retrieved documents 📄. The left section reports results with irrelevant documents from Reddit and the right section with nonsensical sentences made up of random words. Scenarios where the prompt exceeded the model's input limit, leading to potential data truncation, are not included (-). Each value *not* marked with an asterisk * represents a statistically significant change from the base case of retrieved documents only [SP, 📄, Q] (first row), as determined by a Wilcoxon test (p-value < 0.01).

	Irrelevant from Reddit							Random Words						
# ⚅ \ # 📄	1	2	3	4	5	8	10	1	2	3	4	5	8	10
0	0.1620	0.1866	0.1876	0.1866	0.1921	0.2198	0.2108	0.1620	0.1866	0.1876	0.1866	0.1921	0.2198	0.2108
1	0.1693*	0.1931	0.1845*	0.1907	0.2008	0.2084	0.2084	0.1744	0.1924*	0.1969	0.2077	0.2091	0.2139*	0.2073*
2	0.1886	0.2018	0.2101	0.2143	0.2160	0.2222*	0.2219	0.1765	0.1855*	0.2094	0.2122	0.2181	0.2045	0.2084*
3	0.1897	0.2108	0.2212	0.2340	0.2371	0.2326	0.2319	0.1755	0.1990	0.2166	0.2201	0.2288	0.2032	0.2156*
5	0.1897	0.2215	0.2388	0.2468	0.2409	**0.2769**	**0.2451**	0.1862	0.2139	0.2319	0.2367	0.2232	0.2184*	0.2278
8	0.2011	0.2326	0.2354	0.2489	0.2440	0.2568	0.2364	0.1973	0.2274	0.2319	0.2316	0.2305	0.2357	**0.2412**
10	0.2053	0.2326	0.2451	0.2534	**0.2551**	0.2658	-	0.2053	0.2271	0.2340	0.2385	**0.2406**	**0.2499**	-
15	0.2240	0.2489	**0.2689**	**0.2786**	-	-	-	0.2215	0.2416	**0.2589**	**0.2634**	-	-	-
16	0.2240	0.2561	0.2676	-	-	-	-	**0.2219**	0.2437	0.2568	-	-	-	-
17	**0.2243**	**0.2565**	-	-	-	-	-	0.2201	**0.2450**	-	-	-	-	-
18	0.2240	-	-	-	-	-	-	0.2177	-	-	-	-	-	-

of random documents is the driving factor behind the increased accuracy.

Table 5: Accuracy of Falcon-7b on Reddit data in the retrieved + irrelevant setting [SP, ⚅, 📄, Q]. Rows denote the number of irrelevant documents ⚅ added, and columns show the quantity of retrieved documents 📄. Scenarios where the prompt exceeded the model's input limit, leading to potential data truncation, are not included (-). Each value *not* marked with an asterisk * represents a statistically significant change from the base case (first row), as determined by a Wilcoxon test (p-value < 0.05).

# ⚅ \ # 📄	1	2	3	4	5	9
0	0.1568	0.1717	0.1855	0.1938	0.1942	**0.1998**
1	0.1551*	0.1793*	0.1897*	0.1924*	0.1976*	-
2	0.1529*	0.1762*	0.1938*	0.2011*	0.1976*	-
3	0.1599*	0.1727*	0.1911*	0.2021*	**0.2118**	-
4	0.1606*	0.1758*	0.1959	0.2073	0.2108	-
5	0.1627*	0.1762*	0.2000	**0.2108**	-	-
6	0.1651*	0.1848	**0.2004**	-	-	-
7	0.1675	**0.1848**	-	-	-	-
8	**0.1682**	-	-	-	-	-

5.4.3 Falcon. As indicated in Table 2, Falcon does not align with the performance increase pattern noted when random documents are added to the gold document **[I, ⚅, ★, Q]**. In light of this, we extended our examination to assess its performance when presented with retrieved documents to determine whether the introduction of noisy documents would similarly affect Falcon's performance in this new setting. The results of this experiment are detailed in Table 5. We find that the addition of irrelevant documents on top of retrieved documents **[I, ⚅, 📄, Q]** improves the performance of Falcon, too. These results are in contrast with the ones obtained in the oracle setting, where Falcon was robust to noise in the sense that its performance degraded slightly when irrelevant documents were added. This new finding further validates our experimental evidence given that, outside the oracle setting, all the models that were tested showed an improvement when noise was added.

5.5 Retriever Trade-Off

The goal of this paper is to characterize an effective RAG retriever. Starting from the fact that LLMs can handle a finite number of documents due to the limited context size, what documents should a retriever feed the LLM with? Common sense would suggest documents that are semantically close to the query. Conversely, the experimental evidence collected so far highlights a crucial balance between relevant and irrelevant documents. When arranged as described, random documents seem to exert a positive influence on LLM accuracy. However, for the LLM to generate accurate answers, some degree of relevant information must exist in the context. On the other hand, an overabundance of retrieved documents increases the likelihood of including related but not relevant 🔗 information, leading to a sharp decline in performance. While deriving a universal theory remains challenging, our findings imply that optimal trade-off and accuracy are attained when a minimal set of documents is initially retrieved and then supplemented with irrelevant documents until the context limit is reached. For the queries examined in this study, retrieving between 3 and 5 documents is the most effective choice. Adding more than this increases the risk of including too many related but counterproductive documents. There is a pressing need for further research to work out the broader applicability of this rule and, more importantly, to refine our understanding of the retriever's role within a RAG system.

On The Unreasonable Effectiveness Of Random Documents. We cannot close this paper without attempting to explain the results shown up to this point. We refer back to our RAG formulation, particularly the conditioned function $p_\theta(y|\cdot, d)$. In hindsight, we can now state that by adding random documents to the context, we are better conditioning this function, inducing enhanced accuracy. Previous research [2, 8], particularly [45], hints that there might be cases in which a pathologically low attention entropy causes the LLM to generate degenerate outputs with a sharp decrease in performance. These episodes are named entropy collapse. Following this line of research, we measure the entropy of the attention scores in the case where only the gold document is supplied **[I, ★, Q]** against the case in which random documents are added **[I, ⚅, ★, Q]**. We find that when we introduce random documents, the entropy of the systems increases by a factor of x3. Although these experiments show a pattern, we cannot clearly answer this question definitely. It is, thus, a very important open problem to understand why the LLM shows this behavior. However, as we stated above, our main focus is on the IR component and not on the LLM component, so, for the moment, we limit ourselves to reporting this phenomenon and highlighting it. Future studies should aim to elucidate why this noisy state is more advantageous and identify the characteristics that contribute to its effectiveness.

6 CONCLUSIONS

In this paper, we have carried out the first comprehensive study that focuses on how the retrieved documents impact the RAG frameworks, aiming to understand the traits required in a retriever to optimize prompt construction for a RAG system. This study led to several important findings. Relevant information should be placed near the query; otherwise, the model seriously struggles to attend to it. Related documents are extremely harmful to RAG systems. Subverting our expectations, on the other hand, irrelevant, noisy documents are actually helpful in driving up the accuracy of these systems when placed correctly. While we have proposed strategies to exploit these findings, further research is needed both to uncover the inner mechanisms behind this behavior and to develop a new generation of information retrieval techniques that are more fit to interact with the generative component.

REFERENCES

[1] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The Falcon Series of Open Language Models. arXiv:2311.16867 [cs.CL]

[2] Giuseppe Attanasio, Debora Nozza, Dirk Hovy, and Elena Baralis. 2022. Entropy-based attention regularization frees unintended bias mitigation from lists.

[3] Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open LLM Leaderboard. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard.

[4] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*. PMLR, Baltimora, 2206–2240.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[6] Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2021. UnitedQA: A hybrid approach for open domain question answering.

[7] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*. PMLR, Vienna, 3929–3938.

[8] David T Hoffmann, Simon Schrodi, Nadine Behrmann, Volker Fischer, and Thomas Brox. 2023. Eureka-Moments in Transformers: Multi-Step Tasks Reveal Softmax Induced Optimization Problems.

[9] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning.

[10] Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al. 2023. Phi-2: The surprising power of small language models.

[11] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*. JMLR.org, Honolulu, Hawaii, USA, Article 641, 12 pages.

[12] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering.

[13] 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)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

[14] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, Vol. 1. Association for Computational Linguistic, Minneapolis, 2.

[15] Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context.

[16] Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. ACM, Xi'an, 39–48.

[17] 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 (2019), 452–466. https://doi.org/10.1162/tacl_a_00276

[18] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, Anna Korhonen, David R. Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, Florence, 6086–6096. https://doi.org/10.18653/V1/P19-1612

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

[20] Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: phi-1.5 technical report.

[21] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts.

[22] C Manning, P Raghavan, and H Schutze. 2008. *Term weighting, and the vector space model.* Cambridge University Press Cambridge, Cambridge. 109–133 pages.

[23] Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, et al. 2023. Augmented language models: a survey.

[24] Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering Ambiguous Open-domain Questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 5783–5797. https://doi.org/10.18653/v1/2020.emnlp-main.466

[25] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only. arXiv:2306.01116 [cs.CL]

[26] Ofir Press, Noah Smith, and Mike Lewis. 2022. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. https://openreview.net/forum?id=R8sQPpGCv0

[27] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

[28] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[29] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.

[30] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models.

[31] Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. , 29–48 pages.

[32] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[33] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for Building an Open-Domain Chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty (Eds.). Association for Computational Linguistics, Online, 300–325. https://doi.org/10.18653/v1/2021.eacl-main.24

[34] Artsiom Sauchuk, James Thorne, Alon Halevy, Nicola Tonellotto, and Fabrizio Silvestri. 2022. On the Role of Relevance in Natural Language Processing Tasks. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Madrid, 1785–1789.

[35] Noam Shazeer. 2019. Fast Transformer Decoding: One Write-Head is All You Need. arXiv:1911.02150 [cs.NE]

[36] Simeng Sun, Kalpesh Krishna, Andrew Mattarella-Micke, and Mohit Iyyer. 2021. Do long-range language models actually use long-range context?

[37] MosaicML NLP Team et al. 2023. Introducing mpt-7b: A new standard for open-source, ly usable llms.

[38] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models.

[39] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc., Long Beach. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[41] Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. TL;DR: Mining Reddit to Learn Automatic Summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, Lu Wang, Jackie Chi Kit Cheung, Giuseppe Carenini, and Fei Liu (Eds.). Association for Computational Linguistics, Copenhagen, Denmark, 59–63. https://doi.org/10.18653/v1/W17-4508

[42] SK Michael Wong, Wojciech Ziarko, and Patrick CN Wong. 1985. Generalized vector spaces model in information retrieval. In *Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, Montreal, 18–25.

[43] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding.

[44] Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. 2021. Pretrained Transformers for Text Ranking: BERT and Beyond. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*, Greg Kondrak, Kalina Bontcheva, and Dan Gillick (Eds.). Association for Computational Linguistics, Online, 1–4. https://doi.org/10.18653/v1/2021.naacl-tutorials.1

[45] Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Joshua M Susskind. 2023. Stabilizing transformer training by preventing attention entropy collapse. In *International Conference on Machine Learning*. PMLR, PMLR, Hawaii, 40770–40803.

[46] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing Dense Retrieval Model Training with Hard Negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (<conf-loc>, <city>Virtual Event</city>, <country>Canada</country>, </conf-loc>) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 1503–1512. https://doi.org/10.1145/3404835.3462880

[47] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities.