A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models

Wenqi Fan
wenqifan03@gmail.com
The Hong Kong Polytechnic University, HK SAR

Yujuan Ding*
dingyujuan385@gmail.com
The Hong Kong Polytechnic University, HK SAR

Liangbo Ning
BigLemon1123@gmail.com
The Hong Kong Polytechnic University, HK SAR

Shijie Wang
shijie.wang1999@outlook.com
The Hong Kong Polytechnic University, HK SAR

Hengyun Li
neilhengyun.li@polyu.edu.hk
The Hong Kong Polytechnic University, HK SAR

Dawei Yin
yindawei@acm.org
Baidu Inc, China

Tat-Seng Chua
dcscts@nus.edu.sg
National University of Singapore, Singapore

Qing Li
csqli@comp.polyu.edu.hk
The Hong Kong Polytechnic University, HK SAR

ABSTRACT

As one of the most advanced techniques in AI, Retrieval-Augmented Generation (RAG) can offer reliable and up-to-date external knowledge, providing huge convenience for numerous tasks. Particularly in the era of AI-Generated Content (AIGC), the powerful capacity of retrieval in providing additional knowledge enables RAG to assist existing generative AI in producing high-quality outputs. Recently, Large Language Models (LLMs) have demonstrated revolutionary abilities in language understanding and generation, while still facing inherent limitations such as hallucinations and out-of-date internal knowledge. Given the powerful abilities of RAG in providing the latest and helpful auxiliary information, Retrieval-Augmented Large Language Models (RA-LLMs) have emerged to harness external and authoritative knowledge bases, rather than solely relying on the model's internal knowledge, to augment the quality of the generated content of LLMs. In this survey, we comprehensively review existing research studies in RA-LLMs, covering three primary technical perspectives: architectures, training strategies, and applications. Furthermore, to deliver deeper insights, we discuss current limitations and several promising directions for future research. Updated information about this survey can be found at *https://advanced-recommender-systems.github.io/RAG-Meets-LLMs/*[1].

CCS CONCEPTS

• **General and reference** → **Surveys and overviews**; • **Computing methodologies** → **Natural language generation**; • **Information systems** → **Retrieval models and ranking**.

KEYWORDS

Retrieval Augmented Generation (RAG), Large Language Model (LLM), Pre-training, Fine-tuning, In-context Learning, Prompting

ACM Reference Format:
Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3637528.3671470

1 INTRODUCTION

As one of the most fundamental data mining techniques, retrieval aims to understand the input query and extract relevant information from external data sources [23, 29, 60, 118]. It has found extensive application in various fields [8, 27, 90, 144], such as search, question answering, and recommender systems. For instance, search engines (e.g., Google, Bing, and Baidu) are the most successful applications of retrieval in the industry; they can filter and retrieve the most relevant web pages or documents that can match a user's query [19, 144], enabling users to find the desired information effectively. Meanwhile, retrieval models, through effective data maintenance in external databases, can provide faithful and timely external knowledge, thereby serving vital functions in various knowledge-intensive tasks. Due to their powerful capacities, retrieval techniques have been successfully incorporated into advanced generative models in the era of AI-Generated Content (AIGC) [68, 112, 134]. Notably, the integration of retrieval models with language models has given rise to Retrieval-Augmented Generation (RAG) [66], which has emerged as one of the most representative techniques

*Corresponding author: Yujuan Ding

[1]For the long version of this survey, please refer to https://arxiv.org/abs/2405.06211

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08.
https://doi.org/10.1145/3637528.3671470

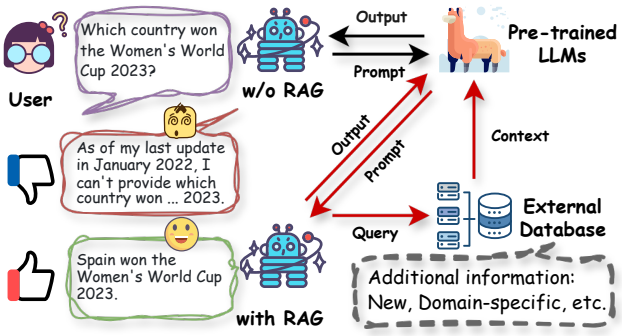

Figure 1: Retrieval-Augmented Generation (RAG) meets Large Language Models (LLMs). When the user's query is out-of-scope, e.g., unseen content in training data or requiring the latest information for the answer, LLMs might show inferior generation performance. With the help of RAG, LLMs can leverage additional relevant information from external database to enhance their text generation capability.

in the field of generative AI, aiming to enhance the quality of the generated text content with retrieved information [6, 66, 68].

More specifically, to facilitate the generation task in the NLP area, RAG incorporates information or knowledge from external data sources, which serves as supplementary reference/instruction for the input query or the generated output [56, 87]. In general, RAG first invokes the retriever to search and extract relevant documents in the external database. These documents are then combined with the original query as the context to enhance the answer generation process [50]. In practice, RAG techniques are feasible and efficient to apply in various generation tasks, by simply adapting the retrieval component and requiring minimal or even no additional training[98]. Recent studies have demonstrated the great potential of RAG not only for knowledge-intensive tasks such as open domain question answering (OpenQA) [6, 44, 92], but also for general language tasks and downstream applications [56, 78, 134, 138].

Recent years have witnessed the rapid development of pre-trained foundation models, particularly Large Language Models (LLMs). These models have demonstrated impressive performance across various tasks [1, 18], such as recommender systems [37], molecule discovery [68], and report generation [26]. Technically, the great success of LLMs can be attributed to the advanced architecture with billion-level parameters pre-training on a huge amount of training corpus from various sources. These technical improvements lead to the emergence of remarkable capabilities of LLMs [37, 157], particularly in language understanding and generation, in-context learning, and other aspects. For instance, GPT-FAR introduces detailed prompts to teach GPT-4 to perform image tagging, statistical analysis and text analysis for multi-modal fashion report generation [26]. LLMs also achieve promising performance in recommender systems by understanding users' preferences towards items [37, 127]. Despite these success, LLMs still suffer from intrinsic limitations [37, 157], such as the lack of domain-specific knowledge, the issue of "hallucination", and the substantial computational resources required for updating the LLMs. These problems are particularly notable in domain-specific fields like medicine and law. For instance, a recent study has demonstrated that legal hallucinations are pervasive and disturbing, with hallucination rates ranging from 69% to 88% in responses to specific legal queries for state-of-the-art LLMs [20]. Moreover, the challenges of tackling the hallucination problem become even harder due to the substantial computational resources required for fine-tuning LLMs with domain-specific or the latest data. This, in turn, significantly hinders the widespread adoption of LLMs in various real-world applications.

To address these limitations, recent efforts have been made to take advantage of RAG to enhance the capabilities of LLMs in various tasks [6, 49, 56, 114], especially those demanding high for the latest and reliable knowledge such as Question Answer (QA), AI4Science, and software engineering. For example, Lozano et al. [80] introduces a scientific QA system based on retrieving scientific literature dynamically. MolReGPT leverages RAG to enhance the in-context learning ability of ChatGPT for molecular discovery [68]. It is also been demonstrated that RAG can effectively reduce hallucinations in conversational tasks [116, 139]. As illustrated in Figure 1, an LLM-based dialog system will not be able to answer well for out-of-scope queries. With the help of RAG to retrieve relevant knowledge from external database and integrate it into the process of generation, the dialog system succeeds in giving correct answers. Given the remarkable progress in advancing LLMs with RAG, there is an imperative need for a systematic review of recent advances in Retrieval-Augmented Large Language Models (**RA-LLMs**).

This survey provides a comprehensive overview of RA-LLMs by summarizing representative methods from the aspects of the architecture, training strategy, and application area respectively. It first review the architecture of existing RA-LLMs from three primary perspectives: retrieval, generation, and augmentation in Section 2. Training techniques are further summarized in Section 3. Subsequently, various RA-LLMs applications are presented in Section 4. In Section 5, key challenges and potential directions for future exploration are further discussed. Due to the page limit, this published version omits a part content including background knowledge of LLMs, details of RA-LLM architectures, visual illustrations, etc. Please refer to the long version for more information [32].

Concurrent to our survey, several related surveys have diverse focuses for RAG and LLMs. For example, Zhao et al. [156] specifically review multi-modal information-based RAG techniques and Zhao et al. [155] discuss the RAG for AIGC. Gao et al. [39] conduct a relatively comprehensive overview of RAG for LLMs. Our survey differs from these surveys in concentrating on technical perspectives and systematically reviewing models according to the architecture and training paradigm in RA-LLMs, as well as application tasks.

2 RETRIEVAL-AUGMENTED LARGE LANGUAGE MODELS (RA-LLMS)

The RAG framework in the era of LLMs consists of several major processes: *retrieval*, *generation*, and *augmentation*. In this section, we will introduce important techniques involved in each process.

2.1 Retrieval

Given the query from the input of LLMs, the retriever is an information provider in RAG, aiming to return relevant knowledge by measuring the distance between the query and documents from

the external knowledge sources. As shown in Figure 2, the retrieval component consists of several compulsory or optional procedures to function as a whole for effective information retrieval. The specific pipeline of the retrieval part is jointly determined by several perspectives of design, such as retriever type and retrieval granularity. In this subsection, we will introduce the existing retrieval methods in RA-LLMs based on these key aspects.

2.1.1 Retriever Type. Retrieval methods can be generally categorized into two types: sparse and dense, based on the information encoding methods. Sparse retrieval is word-based and applied in text retrieval mostly, while dense retrieval embeds queries and external knowledge into vector spaces and can be applied to various data formats.

As a straightforward approach, sparse retrieval, e.g., TF-IDF and BM25 [106, 121], usually relies on inverted index matching along with the raw data input. For example, many studies directly apply BM25 for passage-level retrieval to facilitate their RAG [10, 52, 98, 137, 159, 160], where passages are specifically represented as a bag of words and ranked based on term and inverse document frequencies [50]. On top of offering supplementary to enhance the input of the generator, sparse retrieval has also been used to find demonstrations to function in In-Context Learning (ICL) for RA-LLMs [2, 83, 107, 117, 141]. The main limitation of applying sparse retrieval in RAG is its no-training nature, which makes the retrieval performance heavily rely on the quality of the database and the query. Moreover, such fixed term-based methods only support similarity-based retrieval, while cannot be adapted for other retrieval criteria possibly existing in LLM applications, such as the diversity [30].

Dense retrieval, on the contrary, embeds the query and documents into continuous vector space with certain criteria, for example, semantic similarity [55]. Dense retrieval methods are usually trainable, therefore holding more flexibility and potential in adaptation. As the key component of dense retriever, the embedding models have delicately different designs in existing RAG models. A simple design [56, 64, 136] is to directly use a part of the generation model as the embedding layer of the retriever, which might be able to enhance the alignment between the retrieval and generation processes. BERT-based backbone [24] is widely applied in retrieval models. One common retriever design in RAG is to construct two-stream encoders with the BERT structure (one encoder for the query and the other for the documents), which is also called bi-encoder [114, 135]. Early-stage RAG methods tend to freeze [6, 98] or partially freeze [66] the parameters of the retriever to perform general-level relevant knowledge extraction and pay more attention to the knowledge leveraging and generator fine-tuning. Large-scale specialized pre-training further enhances RAG models to excel in more knowledge-intensive tasks. One typical success is Dense Passage Retriever (DPR) [55], which uses a BERT-based backbone and is pre-trained specifically for the OpenQA task with question-answer pair data. A recent study [103] has also discovered that DPR training decentralizes how knowledge is stored in the network, creating multiple access pathways to the same information. With effective fine-tuning, bi-encoder retrievers are also applied widely in ICL-based RAG [72, 81, 86, 93, 107, 141]. Specifically, they have been more often used for sentence embedding

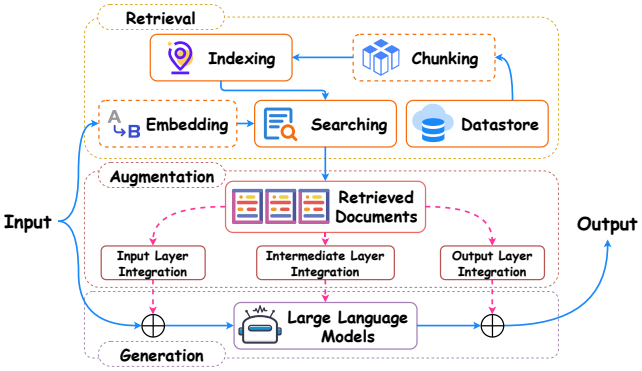

Figure 2: Illustration of the basic Retrieval-Augmented Large Language Models (RA-LLMs) framework, which consists of three major components: *retrieval*, *generation*, and *augmentation*. Retrieval includes different procedures depending on the specific designs. The retrieved documents are further leveraged in generation with the augmentation module, which may be at different integration stages.

similarity-based retrieval, as well as for some special requirement in ICL, such as diverse example retrieval [141]. Another stream of dense retrievers having been widely applied in RA-LLMs uses one encoder only, which may be based on Transformer, BERT or other off-the-shelf sequence modeling backbones. These one-encoder retrievers are generally pre-trained on large-scale unaligned documents by contrastive learning [103], which may therefore excel for their versatility, meaning that they can transfer and generalize better to new domains or tasks. Such general-purpose pre-trained retrievers, e.g., Contriever [40] and Spider [99], would be more flexible to use in LLMs targeting on various tasks and have demonstrated their effectiveness in many RA-LLM methods, such as In-Context RALM [98], Atlas [51] and Self-RAG [5].

2.1.2 Retrieval Granularity. Retrieval granularity denotes the retrieval unit in which the corpus is indexed, e.g., document, passage, token, or other levels like entity. For RAG, the choice of retrieval granularity can significantly impact the overall performance of the model in terms of effectiveness and efficiency as they determine the saving space for the database as well as the computational cost for searching [4]. Early stage retrieval-augmented language models [10] propose to retrieve whole pieces of documents, and then apply a machine comprehension model trained to detect answer spans in the returned documents, which focuses more on language reading and key information locating in the document. In generative language models, **Chunk retrieval** (also called passages in some references [44, 52, 55]) is common, which has been used in both traditional and LLM-based RAG models such as REALM [44], RAG [66] and Atlas [51]. A more fine-grained retrieval, i.e., **token retrieval**, instead can be done with faster searching but will bring more burden for the database saving. Token retrieval is more suitable in cases requiring rare patterns or out-of-domain data [56], meanwhile cooperates well with the every-token retrieval strategy as applied in kNN-LM and other similar work [45, 88, 145]. In

comparison, a text chunk may contain compact and complete information with less redundancy and irrelevancy, therefore becoming the mainstream retrieval text granularity in RAG.

Another major retrieval granularity proposed in RAG is **entity retrieval**. Unlike the above types of granularity, entity retrieval is designed from the perspective of knowledge rather than language. Févry et al. [38] introduce the Entities as Experts (EAE) model, which divides the parameter space of language models according to the entity identity. The proposed EAE model aims to learn entity representations from the text along with other model parameters with the Wikipedia database and represent knowledge with entity memory. At a more fine-grained level, de Jong et al. [21] propose to build the knowledge base by learning and retrieving mention rather than entity. Overall, applying entity or mention-level retrieval in RAG would be more effective for entity-centric tasks, and more efficient in space compared to token-wise retrieval.

2.2 Generation

The design of the generator heavily depends on the downstream tasks. For most text generation tasks, Decoder-only and Encoder-Decoder are two dominant structures [157]. The recent development of commercial closed-sourced large foundation models makes black-box generation models mainstream in RA-LLMs. In this part, we will briefly review studies with these two types of generators: parameter-accessible (white-box) and parameter-inaccessible (black-box).

2.2.1 Parameter-Accessible Generators (White-box). The structure of *Encoder-Decoder* processes the input and the target independently with different sets of parameters, in which a cross-attention component is developed to connect input tokens to target tokens. Representative Encoder-Decoder models include T5 [97] and BART [65]. In comparison, *Decoder-only models* process inputs and targets after concatenation, which makes the representations of the two parts concurrently built layer-by-layer as they propagate up the network. These two types of generators are widely applied in existing RAG work. For example, RAG [66] and Re^2G [42] employ BART; FID [50] and $EMDR^2$ utilize T5. There are other models [6, 73] leveraging Transformer-based Encoder-Decoder architecture but with some customized design. Generators in RAG differ themselves from general ones by incorporating retrieved data to enhance the generation accuracy and relevance. Furthermore, white-box generators allow parameter optimization, which can be trained to adapt to different retrieval and augmentation approaches for a better performance of generation.

2.2.2 Parameter-Inaccessible Generators (Black-box). A certain proportion of LLMs are released without the disclosure of internal structures or the accessibility of parameters, especially those particularly large-scale ones such as GPT series [1], Codex [12] and Claude, which are called black-box generation models. These generators only allow the operations of feeding queries (input) and receiving responses (output) while not allowing the internal structure to be altered or parameters to be updated. From another perspective, LLMs, even those open for fine-tuning, are large in scale and difficult to tune for downstream domain-specific tasks with only a limited amount of data. Black-box RA-LLMs, therefore, focus more on the retrieval and augmentation processes, trying to enhance the generator by augmenting the input (also called prompt in the context of LLMs) with better knowledge, guidance, or examples for the generation. For example, Rubin et al. [107] proposes to train a prompt retriever with the data labeled by language models themselves, which can be used to provide better examples for in-context learning, therefore enhancing the final generation performance. Xu et al. [137] propose to compress the retrieved documents before in-context integration, which can reduce the computational costs and also relieve the burden of LMs to identify relevant information in long retrieved documents.

2.3 Retrieval Integration for Generation Augmentation

Augmentation describes the technical process that integrates retrieval and generation parts, which is the essential part of RA-LLMs. In this subsection, we introduce three main designs of augmentation, which are conducted at the input, output, and intermediate layers of generator respectively, as illustrated in Figure 2.

2.3.1 Input-Layer Integration. A common way to integrate retrieved information/documents is to combine them with the original input/query and jointly pass them to the generator, which is called input-layer integration. For example, In-Context RALM [98] applies input-layer integration by specifically concatenating the original input and all retrieved documents into a single sequence as the new input for the generation model. Despite the effectiveness, such integration is limited to the number of retrieved documents, since the concatenated new input may be too long to be processed by the generation model. In-context RALM specifically alleviates this limitation by removing tokens from the beginning of the new input. To avoid information loss with such a token removing strategy, FID [50] employs a different integration method that processes each retrieved document independently in the encoder. This strategy is scalable to a large number of contexts as it only performs self-attention over one context at a time in the follow-up processing. Atlas [51] and REPLUG [114] apply a similar parallel integration by concatenating the query and one retrieved document at a time. In general, most black-box generation-based RAG methods apply input-layer integration since neither the intermediate layer of the generation model or the output distribution is accessible.

More specially for LLMs, input-layer integration may use the retrieved content as (additional) prompts or demonstrations on top of using it as supplementary to the original input as in traditional RAGs [107]. Prompt retrieval aims to find suitable natural language prompts automatically through retrieval to teach the LLM to learn in context [7] or to induce the LLM to reason[133]. It may boost the zero-shot ability of LLMs without delicate prompt engineering. For example, Cheng et al. [16] propose to learn a prompt retriever based on the input-prompt pair data with score labels resulting from a frozen LLM.

2.3.2 Output-Layer Integration. Another kind of augmentation is post-hoc, i.e., output-layer integration, which joints retrieval and generation results. For example, kNN-LM [56] interpolates two next-token distributions in prediction: one induced by the LM and

the other induced by the nearest neighbors from the retrieval corpus. Output-layer linear integration [43, 159] is flexible to apply since it can be plugged into most generation models without additional training. However, the simplicity of output-layer integration also limits the model's ability to reason about the retrieved text. To tackle this limitation, Yogatama et al. [145] propose to add an extra gating network to post-process the retrieved data and achieve comparatively better performance. For LLMs, output-layer integration is as reasonable and adaptive as input-layer integration. REFEED [148] proposes an answer refining mechanism that applies an LLM to evaluate the retrieved information and adjust the initial answer accordingly to enhance the accuracy of the response. Similarly, Zhang et al. [154] propose the COMBO framework, which matches LLM-generated passages with retrieved counterparts into compatible pairs based on pre-trained discriminators. The passage pairs are then handled by a Fusion-in-Decoder-based [50] to derive a final answer.

2.3.3 Intermediate-Layer Integration. Compared to the above two non-parametric approaches, a more engaging augmentation is to design a semi-parametric module to integrate the retrieved results through the internal layers of the generation model, which is called intermediate-layer integration. Such integration might add additional complexity and is promising to enhance the capability of the generation model with effective training. Typically, a Transformer module is introduced to leverage retrieved information (mostly encoded into dense representations) into the generation model to interact with the representations in the middle stage of the generation. For example, RETRO [6] introduces a Chunked Cross Attention (CCA) layer to process the retrieved chunks in the generator blocks, and Wu et al. [136] introduces the kNN-Augmented Attention Layer. Similarly, EAE [38] and TOME [21] use Entity Memory and MemoryAttention layer to incorporate the retrieved Entity and Entity Mentions, respectively. Such intermediate-layer integration can use many blocks frequently and efficiently to enhance the capability of the whole RAG model. It offers an efficient alternative to incorporate a large number of text chunks frequently retrieved, which are challenging to process with input-layer integration due to the input length limit of LMs [6]. However, it also needs to be noted that intermediate-layer integration requires high access to the generation models, which is not feasible for most LLMs that are accessible through inference APIs [85].

3 RA-LLMS TRAINING

Based on whether training is required or not, existing RAG methods can be categorized into two main classes: **train-free** approaches and **training-based** approaches. Training-free methods usually directly leverage the retrieved knowledge during inference time without introducing extra training by inserting the retrieved text into the prompt, which is computationally efficient. However, one potential challenge is that the retriever and generator components are not specifically optimized for downstream tasks, which could easily lead to suboptimal utilization of the retrieved knowledge. To fully exploit the external knowledge, extensive methods are proposed to fine-tune the retriever and generator, thereby guiding large language models to effectively adapt and integrate retrieved information. According to the training strategies, we categorize these training-based approaches into three classes: 1) **Independent Training** approaches independently train each component in the RAG procedure, 2) **Sequential Training** methods train one module first and freeze the well-trained component to guide the tuning process of the other part, and 3) **Joint Training** approaches train retriever and generator simultaneously. In the following section, we will comprehensively review the training-free, independent training, sequential training, and joint training methods.

3.1 Training-free

With the huge number of parameters, large language models have exhibited human-level intelligence and achieved promising prediction performance on various downstream tasks. However, it is extremely challenging to frequently perform fine-tuning and update the knowledge stored in the model parameters [66] due to the considerable time and computational resources required. Recently, numerous studies have suggested enhancing large language models with retrieval mechanisms, enabling them to dynamically acquire new knowledge from external sources without extra training processes (i.e., *training-free*) [50, 52, 57], instead of relying solely on the implicit knowledge encoded in the model's parameters. These approaches have shown significant performance improvement for various knowledge-intensive tasks, such as open-domain question answering [66] and document summarization [120]. According to the different ways in which large language models utilize retrieved information, we categorize these training-free methods into two categories: 1) **Prompt Engineering-based Methods** integrate retrieved knowledge into the original prompt directly, and 2) **Retrieval-Guided Token Generation Methods** retrieve information to calibrate the token generation process.

3.1.1 Prompt Engineering-based Methods. As the LLMs' generation performance highly depends on the input query, numerous training-free RAG approaches employ external knowledge by refining the original prompts [52, 57, 71]. Specifically, the retrieved texts are usually used as contextual information and combined with the original prompt to guide the generation of large language models [50, 52, 57, 59, 71, 94, 129]. For example, In-Context RALM [98] keeps the large language model parameters unchanged and directly incorporates the retrieved document before the original prompt to augment the generation process. IRCoT [124] interleaves chain-of-thought (CoT) generation and knowledge retrieval steps, enabling the retrieval of more relevant information for the subsequent reasoning compared to standard retrieval methods that rely solely on the question as the query. Instead of retrieving knowledge from a large corpus, GENREAD [147] first prompts a large language model to generate contextual documents for the query, and then based on them to generates answers. SKR [130] proposes guiding LMs to determine whether they can answer a given question based on their internal knowledge, enabling flexible utilization of both internal and external knowledge by selectively calling the retriever. TOC [59] first retrieves relevant knowledge for ambiguous questions and recursively constructs a tree structure by clarifying ambiguous questions into multiple disambiguate questions, which is further aggregated to generate long-form answers.

3.1.2 Retrieval-Guided Token Generation Methods. In addition to directly integrating external knowledge into the original prompt, the auxiliary information can be employed to adjust the token generation process. For example, KNN-KMs [56] first retrieves k most relevant contexts from the datastore based on the given query, and computes a neighbor distribution based on the distance. The output distribution is calibrated by interpolating the neighbor distribution and the original model's output distribution. Rest [46] is proposed to replace the parametric draft model with a non-parametric retrieval datastore, and retrieves relevant tokens based on the current context for speculative decoding [9, 63, 122].

3.2 Independent Training

Independent training refers to training the retriever and large language models (LLMs) as two entirely independent processes, in which there is no interaction between the retriever and the LLMs during the training process [55, 62, 160]. For the training of large language models, the negative loglikelihood loss is the most representative training objective [96, 123], which aims to guide the large language models to generate desired output y based on the given input x, formulated as $-\log P_{LLM}(y|x)$. Regarding the retriever, it can be categorized into two types: 1) Sparse retriever [101, 106], and 2) Dense retriever [55, 62, 160]. The sparse retrievers usually exploit sparse features, e.g., word frequencies, to represent the documents and calculate the relevance scores based on task-specific metrics [68, 101, 106] such as TF-IDF and BM25. As for the dense retrievers, deep neural networks are employed to encode the query and documents into dense representations, and then the inner product is usually used to calculate relevance scores and retrieve the relevant external knowledge. For example, DPR [55] adopts two independent BERT [24] networks to encode the query and passages respectively, and trains these models by utilizing contrastive learning. CoG [62] proposes to train a prefix encoder and a phrase encoder for retrieval and reformulate the text generation as multiple copy-and-paste operations from existing source text collection.

3.3 Sequential Training

Independent training is an efficient approach to exploit the external knowledge during the generation process, since the retriever and generator can be trained offline and any off-the-shelf models can be utilized, avoiding extra training costs. To better enhance the synergy between the retriever and generator, several methods have been proposed to train the retriever and large language models sequentially. In these sequential training methods, the process typically begins with the independent pretraining of either the retriever or the generator, after which the pretrained module is fixed while the other module undergoes training. Note that various existing models (e.g., BERT [24], CLIP [95], T5 [97]) can be directly employed as the fixed retriever and generator, thereby bypassing the first pertaining process. Compared to independent training, sequential training involves coordinated training of the retriever and generator, where the trainable module benefits from the assistance of the fixed module. Based on different training orders, sequential training can be categorized into two classes: 1) **Retriever First** [5, 108, 109, 126], and 2) **LLMs First** [110, 114, 128].

3.3.1 Retriever First. These methods first train the retrieval model and then fix it. Large language models are then trained by utilizing the retrieved knowledge. For instance, RETRO [6] adopts the BERT model that is pretrained independently as the retriever, and an encoder-decoder architecture is trained to integrate retrieval chunks into the model's predictions. RALMs [146] adopts Google Search and the open-source COLBERTV2 [58] as the pretrained retriever and fine-tunes the large language model to effectively leverage the retrieved passages. ITER-RTGEN [105] utilizes the pretrained S-BERT [104] as the retriever and introduces an adaptive hybrid retrieval strategy for retrieving demonstrations. Additionally, it leverages T5 [97] as the generator, which undergoes further fine-tuning based on the target label and input combining the original prompt with retrieved demonstrations. SMALLCAP [102] proposes using the CLIP [95], which is a powerful pretrianed multimodal network, to encode the input image and the textual data of the external datastore and retrieve the most relevant items based on the cosine similarity. A cross-attention layer is trained and GPT-2 [96] is used as the decoder to produce captions.

3.3.2 LLMs First. Similarly, we can also pre-train large language models first, and then tune the retriever under the supervision of the well-trained LLMs. For example, DKRR [49] shows that attention scores from a sequence-to-sequence model can indicate the document's relevance. Therefore, they propose to leverage the attention scores of a reader model to produce synthetic labels to train the retriever. AAR [149] proposes using a small language model to generate the supervised signal for training retrievers. The well-trained retriever can be further leveraged to enhance the performance of large black-box language models. RA-DIT [75] first fine-tune the large language models to enhance their ability to leverage retrieved knowledge, and then train the retriever to better align its output with large language models. UPRISE [16] proposes a lightweight method to enhance the zero-shot performance of LLMs in unseen tasks by introducing a prompt retriever. A frozen LLM is employed to guide the fine-tuning process of the prompt retriever, and this retriever then retrieves prompts for different tasks with various LLMs during inference.

3.4 Joint Training

Joint training methods [17, 47, 54, 70, 159] employ the end-to-end paradigm to optimize the retriever and generator simultaneously. Instead of training each module sequentially, joint training methods effectively enhance the retriever's ability to locate external knowledge for generation and the generator's capacity to effectively leverage the retrieved information. For instance, RAG [66] minimizes the negative loglikelihood to jointly train the retriever and generator. REALM [44] adopts a similar training paradigm to that of RAG [66], and Maximum Inner Product Search (MIPS) [15, 28, 100, 111] technique is used to locate the most relevant documents. To employ MIPS, all external documents are embedded first and a search index is produced for each embedding. An asynchronous index updating strategy [44, 48, 51, 119] is proposed to refresh the index every several hundred training steps to avoid time consumption of re-indexing all documents.

4 APPLICATIONS

In this section, we will introduce some representative applications of retrieval-augmented large language models (RA-LLMs). To provide a clear overview of the applications of RA-LLMs, we will review them from three perspectives: *NLP applications*, *downstream tasks*, and *domain-specific applications*.

4.1 NLP Applications

Due to the intrinsic capability in text generation, RA-LLMs have various applications in the NLP field, such as Question Answer (QA) Systems, ChatBot, and Fact Verification.

4.1.1 QA Systems. QA Systems aim to provide precise answers to user's queries. However, even when trained on extensive data, these systems may lack the latest information or specific domain knowledge that is not included in their training data [50, 79]. To address this limitation, the integration of RA-LLMs has played a crucial role in advancing the capabilities of QA systems by enhancing their ability to retrieve and synthesize relevant information [6, 50]. Specifically, RA-LLMs can provide coherent and contextually relevant answers by leveraging their retrieval component to access a vast knowledge base. For example, REALM [44] integrates a knowledge retriever that can retrieve information from a large corpus during pre-training, fine-tuning, and inference. This approach allows REALM to effectively retrieve from a vast knowledge corpus, thereby improving the accuracy of its responses. Similarly, Fusion-in-Decoder [50] retrieves passages from support documents and then fuses them with questions to generate the answer, achieving higher accuracy. In addition, Borgeaud et al. [6] indicate that the quality of the answers may rely more on the result of retrieval.

4.1.2 ChatBot. ChatBot is designed to interact with users in a natural and conversational manner [76]. Different from the QA system, ChatBot focuses on maintaining a coherent and contextually rich conversation with the user. To enhance these capabilities, recent methods focus on integrating RA-LLMs [54, 61, 152] for its ability to augment the ChatBot with relevant external knowledge, facilitating more engaging and context-rich interactions with users. For example, some studies [14, 41] retrieve relevant knowledge from static databases (e.g., a Wikipedia dump) to augment conversation. Komeili et al. [61] propose retrieving information from the internet search to further augment conversation performance. Considering the dynamic nature of knowledge in the world, another model [125] further accesses large amounts of dynamic information in search engines to generate responses.

4.1.3 Fact Verification. Fact Verification is a critical task in verifying the accuracy and reliability of information. With the need for trusted evidence, RA-LLMs are being utilized to enhance the capabilities of fact verification [51, 66, 66]. Lewis et al. [66] first propose retrieval of external knowledge to augment a range of knowledge-intensive tasks including fact verification. On the other hand, Atlas [51] examines the performance of the RA-LLMs for fact verification under few-shot learning. Recently, Self-RAG [5] has greatly made a notable impression by incorporating a self-reflective mechanism. Specifically, Self-RAG reflects on whether retrieved information is helpful and judges the reliability of retrieved information, thereby greatly improving the verification accuracy.

4.2 Downstream Tasks

RA-LLMs can also be applied to various downstream tasks, such as recommendations and software engineering.

4.2.1 Recommendations. Recommender systems play an important role in modeling users' preferences and providing personalized recommendations [33–35, 127, 153, 158]. Recently, RA-LLMs have demonstrated great potential in providing personalized and contextually relevant recommendations by integrating retrieval and generation processes [25, 82, 134]. For example, Di Palma [25] proposes a simple retrieval-augmented recommendation model, that leverages knowledge from movie or book datasets to enhance recommendations. Additionally, Lu et al. [82] further retrieval from the reviews to enrich item information in recommender systems. CoRAL [134] utilizes reinforcement learning to retrieve collaborative information from the dataset and align it with semantic information for more accurate recommendations.

4.2.2 Software Engineering. The rise of RA-LLMs has influenced many aspects of software engineering [89, 142, 160]. For example, some studies propose the retrieval-augmented generation paradigm for code generation [160] and program repair [89]. Similarly, Parvez et al. [91] retrieve top-ranked codes or summaries from the codebase and aggregate them with input to enhance code generation and summarization. In addition, RA-LLMs show potential in tabular data processing [67, 142] and Text-to-SQL semantic parsing [93, 113].

4.3 Domain-specific Applications

RA-LLMs have been widely adopted for various domain-specific tasks, such as AI for Science and Finance.

4.3.1 AI for Science. RA-LLMs have proven to be beneficial for the realms of science, such as molecular and protein. **Molecules** include identifying the molecule's property and predicting new molecules, thereby favoring drug discovery. Currently, some RA-LLMs have been applied to molecules by integrating retrieval of molecule structure and biomedical entities like protein, molecule, and disease [78, 131, 132, 140], etc. Li et al. [68], Wang et al. [131] propose retrieval-based frameworks by retrieving from the database to guide molecule generation. Liu et al. [78] introduce a multi-modal molecule structure-text model by retrieving textual knowledge from a large-scale dataset for molecular property prediction. In addition, RA-LLMs also significantly influence **Protein** representation and generation. For instance, RSA [84] queries protein sequences associated with a collection of structurally or functionally similar sequences in the database to enhance protein representations. Furthermore, Lozano et al. [80] present a clinical QA system based on retrieving published review articles.

4.3.2 Finance. In the highly data-driven and information-intensive field of finance, RA-LLMs have proved to be a significant technology for enhancing decision-making [69, 143, 151]. For example, Zhang et al. [151] retrieve financial information from external sources, such as news platforms (e.g., Bloomberg and Reuters) and social media platforms (e.g., Twitter, Reddit), to combine with the original query to enhance the precision of financial sentiment analysis. In addition, financial QA is another primary task of financial analysis,

which usually extracts relevant knowledge from financial documents. As professional documents are usually stored in PDF format, Lin [74] introduces a PDF parser combined with RA-LLMs to retrieve knowledge from financial reports. On the other hand, Yepes et al. [143] propose a document chunking method based on structure instead of chunking based on paragraphs, further improving the quality of RA-LLMs outputs.

5 FUTURE CHALLENGES AND OPPORTUNITIES

Since the studies of RA-LLMs are still in the early stage, we present some potential research directions that can be explored in the future in the field of RA-LLMs.

Trustworthy RA-LLMs. The essential objective of developing RAG-empowered LLMs is to enhance the capability of the language models, thereby benefiting users and society by alleviating redundant and meaningless labor, increasing conveniences, and spurring social progress. However, recent research indicates that RA-LLMs can be maliciously and unintentionally manipulated to make unreliable decisions and harm humans [22, 162], which may have serious consequences in safety-critical scenarios [11, 13, 31, 36, 77]. In addition, private retrieval database has a risk of leakage, raising concerns regarding the privacy of RA-LLMs [150]. Therefore, developing trustworthy RA-LLMs is of paramount importance as it can significantly mitigate the potential negative impacts of LLMs technology and provide people with powerful AI models that can be fully trusted. To be specific, the ideal trustworthiness in RA-LLMs systems should possess the following characteristics: 1) **robustness**, 2) **fairness**, 3) **explainability**, and 4) **privacy**. For example, **robustness** means a trustworthy RA-LLMs system should be robust against malicious or inadvertent perturbations introduced by attackers. **Fairness** indicates a trustworthy RA-LLMs system ought to avoid discrimination during the decision-making process. **Explainability** requires a complete understanding of the intrinsic workings of RA-LLMs systems, i.e., the predictions of RA-LLMs systems are explainable and transparent. **Privacy** entails safeguarding the safety of this private information housed within the datastore when establishing trustworthy RA-LLMs systems.

Multi-Lingual RA-LLMs. The ability to leverage knowledge from multiple languages can greatly enhance the capabilities of retrieval-augmented large language models. As the world becomes increasingly interconnected, there is a growing need for AI systems that can understand and communicate across different languages. By incorporating multilingual knowledge retrieval and generation, these models can access and synthesize information from diverse linguistic sources, enabling more comprehensive and nuanced understanding and generation capabilities. Additionally, multilingual models can facilitate cross-cultural communication and knowledge sharing and breaking down language barriers, thereby bringing convenience to people across different regions of the world, especially those in areas with minority languages [53, 71]. For example, users from countries with less prevalent languages can utilize abundant English and Chinese corpora for knowledge retrieval, enhancing the performance of large language models in downstream tasks.

Multimodal RA-LLMs. Multimodal retrieval-augmented generation extends the knowledge sources beyond text to include various data modalities such as images, videos, and audio. By integrating various modalities, LLMs can leverage richer contextual information than single-modal RAG and develop a more comprehensive understanding of users' needs, bringing precise, fine-grained, and high-quality generation. For instance, an image or video can provide valuable visual cues that complement textual information, leading to more precise language generation [47, 161]. By effectively fusing multiple modalities, multimodal RA-LLMs can develop a more comprehensive understanding of the world, leading to more accurate and insightful outputs, benefiting a wide range of domains, including healthcare [161], drug discovery [115], molecular analysis [3, 78, 115], etc.

Quality of External Knowledge. As a commonly used datastore in current RAG systems, Wikipedia [55, 161] serves as a vast repository of external textual knowledge used to augment the generation process, which contains millions of articles covering various disciplines. However, the reliability and accuracy of individual articles within Wikipedia vary significantly, and the introduction of some texts that deviate from facts might even mislead the model's generation process. Therefore, it is crucial to enhance the quality of the external knowledge corpus and mitigate the negative impact of low-quality knowledge on the performance of LLMs. By enhancing the quality of the external knowledge and tailing robust mechanisms by filtering out low-quality or unreliable information, the RA-LLM systems might produce more accurate, reliable outputs, thereby improving their effectiveness in various real-world applications.

6 CONCLUSION

Retrieval-augmented generation (RAG), a cutting-edge AI technique, has achieved remarkable success across various applications, including recommendation, molecule generation, protein representation, and software engineering, owing to the potent capabilities of retrieval in providing supplementary information to enhance generation performance. Recently, increasing efforts have been made to alleviate the limitations of large language models (LLMs), such as hallucination and out-of-date internal knowledge, by leveraging retrieval to provide the latest auxiliary information and teaching LLMs to harness the retrieved external knowledge. With the rapid advancements in retrieval-augmented large language models (RA-LLMs), there is a pressing need for a comprehensive and systematic overview. To bridge this gap, in this paper, we comprehensively review the RA-LLMs from the perspectives of model architecture, training strategy, and application area, providing researchers with an in-depth understanding. Moreover, since the studies of RA-LLMs are still in the early stage, we also discuss the current limitations and several potential research directions for future research.

ACKNOWLEDGMENTS

The research described in this paper has been partly supported by the National Natural Science Foundation of China (project no. 62102335), General Research Funds from the Hong Kong Research Grants Council (project no. PolyU 15200021, 15207322, and 15200023), internal research funds from The Hong Kong Polytechnic University (project no. P0036200, P0042693, P0048625, P0048752, and P0051361), Research Collaborative Project no. P0041282, and SHTM Interdisciplinary Large Grant (project no. P0043302).

REFERENCES

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv:2303.08774* (2023).

[2] Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2023. In-context Examples Selection for Machine Translation. In *ACL (Findings)*. 8857–8873.

[3] Miles C Andrews, Junna Oba, Chang-Jiun Wu, Haifeng Zhu, Tatiana Karpinets, Caitlin A Creasy, Marie-Andrée Forget, Xiaoxing Yu, Xingzhi Song, Xizeng Mao, et al. 2022. Multi-modal molecular programs regulate melanoma cell state. *Nature communications* 13, 1 (2022), 4000.

[4] Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. 2023. Retrieval-based language models and applications. In *ACL (Tutorial)*. 41–46.

[5] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *ICLR*.

[6] 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 *ICML*. 2206–2240.

[7] 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. In *NeurIPS*.

[8] Stefan Buttcher, Charles LA Clarke, and Gordon V Cormack. 2016. *Information retrieval: Implementing and evaluating search engines*. Mit Press.

[9] Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. Accelerating large language model decoding with speculative sampling. *arXiv:2302.01318* (2023).

[10] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *ACL*. 1870–1879.

[11] Jingfan Chen, Wenqi Fan, Guanghui Zhu, Xiangyu Zhao, Chunfeng Yuan, Qing Li, and Yihua Huang. 2022. Knowledge-enhanced Black-box Attacks for Recommendations. In *KDD*. 108–117.

[12] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv:2107.03374* (2021).

[13] Xiao Chen, Wenqi Fan, Jingfan Chen, Haochen Liu, Zitao Liu, Zhaoxiang Zhang, and Qing Li. 2023. Fairly adaptive negative sampling for recommendations. In *WWW*. 3723–3733.

[14] Xiuyi Chen, Fandong Meng, Peng Li, Feilong Chen, Shuang Xu, Bo Xu, and Jie Zhou. 2020. Bridging the gap between prior and posterior knowledge selection for knowledge-grounded dialogue generation. In *EMNLP*. 3426–3437.

[15] Yudong Chen, Zhihui Lai, Yujuan Ding, Kaiyi Lin, and Wai Keung Wong. 2019. Deep supervised hashing with anchor graph. In *ICCV*. 9796–9804.

[16] Daixuan Cheng, Shaohan Huang, Junyu Bi, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Furu Wei, Weiwei Deng, and Qi Zhang. 2023. UPRISE: Universal Prompt Retrieval for Improving Zero-Shot Evaluation. In *EMNLP*. 12318–12337.

[17] Xin Cheng, Di Luo, Xiuying Chen, Lemao Liu, Dongyan Zhao, and Rui Yan. 2024. Lift yourself up: Retrieval-augmented text generation with self-memory. In *NeurIPS*.

[18] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *J Mach Learn Res* 24, 240 (2023), 1–113.

[19] W Bruce Croft, Donald Metzler, and Trevor Strohman. 2010. *Search engines: Information retrieval in practice*. Vol. 520. Addison-Wesley Reading.

[20] Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E Ho. 2024. Large legal fictions: Profiling legal hallucinations in large language models. *arXiv:2401.01301* (2024).

[21] Michiel de Jong, Yury Zemlyanskiy, Nicholas FitzGerald, Fei Sha, and William W. Cohen. 2022. Mention Memory: incorporating textual knowledge into Transformers through entity mention attention. In *ICLR*.

[22] Gelei Deng, Yi Liu, Kailong Wang, Yuekang Li, Tianwei Zhang, and Yang Liu. 2024. Pandora: Jailbreak GPTs by Retrieval Augmented Generation Poisoning. *arXiv:2402.08416* (2024).

[23] Ziqing Deng, Zhihui Lai, Yujuan Ding, Heng Kong, and Xu Wu. 2024. Deep Scaling Factor Quantization Network for Large-scale Image Retrieval. In *ICMR*. 851–859.

[24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*. 4171–4186.

[25] Dario Di Palma. 2023. Retrieval-augmented recommender system: Enhancing recommender systems with large language models. In *RecSys*. 1369–1373.

[26] Yujuan Ding, Yunshan Ma, Wenqi Fan, Yige Yao, Tat-Seng Chua, and Qing Li. 2024. FashionReGen: LLM-Empowered Fashion Report Generation. In *WWW*.

[27] Yujuan Ding, P. Y. Mok, Yunshan Ma, and Yi Bin. 2023. Personalized fashion outfit generation with user coordination preference learning. *IP&M* 60, 5 (2023), 103434.

[28] Yujuan Ding, Wai Keung Wong, Zhihui Lai, and Zheng Zhang. 2020. Bilinear Supervised Hashing Based on 2D Image Features. *IEEE Trans. Circuits Syst. Video Technol.* 30, 2 (2020), 590–602.

[29] Yujuan Ding, Wai Keung Wong, Zhihui Lai, and Zheng Zhang. 2020. Discriminative dual-stream deep hashing for large-scale image retrieval. *IP&M* 57, 6 (2020), 102288.

[30] Andrew Drozdov, Nathanael Schärli, Ekin Akyürek, Nathan Scales, Xinying Song, Xinyun Chen, Olivier Bousquet, and Denny Zhou. 2022. Compositional semantic parsing with large language models. In *ICLR*.

[31] Wenqi Fan, Tyler Derr, Xiangyu Zhao, Yao Ma, Hui Liu, Jianping Wang, Jiliang Tang, and Qing Li. 2021. Attacking black-box recommendations via copying cross-domain user profiles. In *ICDE*. 1583–1594.

[32] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models. *arXiv:2405.06211* (2024).

[33] Wenqi Fan, Xiaorui Liu, Wei Jin, Xiangyu Zhao, Jiliang Tang, and Qing Li. 2022. Graph Trend Filtering Networks for Recommendation. In *SIGIR*. 112–121.

[34] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *WWW*. 417–426.

[35] Wenqi Fan, Yao Ma, Qing Li, Jianping Wang, Guoyong Cai, Jiliang Tang, and Dawei Yin. 2020. A graph neural network framework for social recommendations. *TKDE* (2020).

[36] Wenqi Fan, Xiangyu Zhao, Xiao Chen, Jingran Su, Jingtong Gao, Lin Wang, Qidong Liu, Yiqi Wang, Han Xu, Lei Chen, et al. 2022. A Comprehensive Survey on Trustworthy Recommender Systems. *arXiv:2209.10117* (2022).

[37] Wenqi Fan, Zihuai Zhao, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Jiliang Tang, and Qing Li. 2023. Recommender systems in the era of large language models (llms). *arXiv:2307.02046* (2023).

[38] Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as Experts: Sparse Memory Access with Entity Supervision. In *EMNLP*. 4937–4951.

[39] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv:2312.10997* (2023).

[40] Izacard Gautier, Caron Mathilde, Hosseini Lucas, Riedel Sebastian, Bojanowski Piotr, Joulin Armand, and Grave Edouard. 2022. Unsupervised dense information retrieval with contrastive learning. *J Mach Learn Res* (2022).

[41] Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *AAAI*, Vol. 32.

[42] Michael R. Glass, Gaetano Rossiello, Md. Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2G: Retrieve, Rerank, Generate. In *NAACL-HLT*. 2701–2715.

[43] Edouard Grave, Armand Joulin, and Nicolas Usunier. 2017. Improving Neural Language Models with a Continuous Cache. In *ICLR*.

[44] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *ICML*. 3929–3938.

[45] Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2021. Efficient Nearest Neighbor Language Models. In *EMNLP (1)*. 5703–5714.

[46] Zhenyu He, Zexuan Zhong, Tianle Cai, Jason D Lee, and Di He. 2023. Rest: Retrieval-based speculative decoding. *arXiv:2311.08252* (2023).

[47] Ziniu Hu, Ahmet Iscen, Chen Sun, Zirui Wang, Kai-Wei Chang, Yizhou Sun, Cordelia Schmid, David A Ross, and Alireza Fathi. 2023. Reveal: Retrieval-augmented visual-language pre-training with multi-source multimodal knowledge memory. In *CVPR*. 23369–23379.

[48] Jie Huang, Wei Ping, Peng Xu, Mohammad Shoeybi, Kevin Chen-Chuan Chang, and Bryan Catanzaro. 2023. Raven: In-context learning with retrieval augmented encoder-decoder language models. *arXiv:2308.07922* (2023).

[49] Gautier Izacard and Edouard Grave. 2021. Distilling Knowledge from Reader to Retriever for Question Answering. In *ICLR*.

[50] Gautier Izacard and Edouard Grave. 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In *EACL*. 874–880.

[51] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot Learning with Retrieval Augmented Language Models. *J Mach Learn Res* 24, 251 (2023), 1–43.

[52] Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active Retrieval Augmented Generation. In *EMNLP*. 7969–7992.

[53] Anubha Kabra, Emmy Liu, Simran Khanuja, Alham Fikri Aji, Genta Winata, Samuel Cahyawijaya, Anuoluwapo Aremu, Perez Ogayo, and Graham Neubig. 2023. Multi-lingual and Multi-cultural Figurative Language Understanding. In *ACL*.

[54] Minki Kang, Jin Myung Kwak, Jinheon Baek, and Sung Ju Hwang. 2023. Knowledge graph-augmented language models for knowledge-grounded dialogue generation. *arXiv:2305.18846* (2023).

[55] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *EMNLP*. 6769–6781.

[56] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through Memorization: Nearest Neighbor Language Models. In *ICLR*.

[57] Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *arXiv:2212.14024* (2022).

[58] Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *SIGIR*. 39–48.

[59] Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. 2023. Tree of Clarifications: Answering Ambiguous Questions with Retrieval-Augmented Large Language Models. In *EMNLP*.

[60] Mei Kobayashi and Koichi Takeda. 2000. Information retrieval on the web. *CSUR* 32, 2 (2000), 144–173.

[61] Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-Augmented Dialogue Generation. In *ACL*. 8460–8478.

[62] Tian Lan, Deng Cai, Yan Wang, Heyan Huang, and Xian-Ling Mao. 2022. Copy is All You Need. In *ICLR*.

[63] Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *ICML*. 19274–19286.

[64] Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020. Pre-training via paraphrasing. In *NeurIPS*.

[65] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *ACL*. 7871–7880.

[66] 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. In *NeurIPS*. 9459–9474.

[67] Hongxin Li, Jingran Su, Yuntao Chen, Qing Li, and ZHAO-XIANG ZHANG. 2024. SheetCopilot: Bringing Software Productivity to the Next Level through Large Language Models. In *NeurIPS*.

[68] Jiatong Li, Yunqing Liu, Wenqi Fan, Xiao-Yong Wei, Hui Liu, Jiliang Tang, and Qing Li. 2023. Empowering Molecule Discovery for Molecule-Caption Translation with Large Language Models: A ChatGPT Perspective. *arXiv:2306.06615* (2023).

[69] Xiang Li, Zhenyu Li, Chen Shi, Yong Xu, Qing Du, Mingkui Tan, Jun Huang, and Wei Lin. 2024. AlphaFin: Benchmarking Financial Analysis with Retrieval-Augmented Stock-Chain Framework. *arXiv:2403.12582* (2024).

[70] Xinze Li, Zhenghao Liu, Chenyan Xiong, Shi Yu, Yu Gu, Zhiyuan Liu, and Ge Yu. 2023. Structure-Aware Language Model Pretraining Improves Dense Retrieval on Structured Data. In *ACL*.

[71] Xiaoqian Li, Ercong Nie, and Sheng Liang. 2023. From Classification to Generation: Insights into Crosslingual Retrieval Augmented ICL. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

[72] Xiaonan Li and Xipeng Qiu. 2023. MoT: Memory-of-Thought Enables ChatGPT to Self-Improve. In *EMNLP*. 6354–6374.

[73] Zonglin Li, Ruiqi Guo, and Sanjiv Kumar. 2022. Decoupled context processing for context augmented language modeling. In *NeurIPS*. 21698–21710.

[74] Demiao Lin. 2024. Revolutionizing Retrieval-Augmented Generation with Enhanced PDF Structure Recognition. *arXiv:2401.12599* (2024).

[75] Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, et al. 2023. RA-DIT: Retrieval-Augmented Dual Instruction Tuning. In *ICLR*.

[76] Haochen Liu, Jamell Dacon, Wenqi Fan, Hui Liu, Zitao Liu, and Jiliang Tang. 2020. Does Gender Matter? Towards Fairness in Dialogue Systems. In *ACL*.

[77] Haochen Liu, Yiqi Wang, Wenqi Fan, Xiaorui Liu, Yaxin Li, Shaili Jain, Yunhao Liu, Anil K Jain, and Jiliang Tang. 2021. Trustworthy ai: A computational perspective. *arXiv:2107.06641* (2021).

[78] Shengchao Liu, Weili Nie, Chengpeng Wang, Jiarui Lu, Zhuoran Qiao, Ling Liu, Jian Tang, Chaowei Xiao, and Animashree Anandkumar. 2023. Multi-modal molecule structure–text model for text-based retrieval and editing. *Nature Machine Intelligence* 5, 12 (2023), 1447–1457.

[79] Ye Liu, Semih Yavuz, Rui Meng, Dragomir Radev, Caiming Xiong, and Yingbo Zhou. 2022. Uni-Parser: Unified Semantic Parser for Question Answering on Knowledge Base and Database. In *EMNLP*. 8858–8869.

[80] Alejandro Lozano, Scott L Fleming, Chia-Chun Chiang, and Nigam Shah. 2023. Clinfo. ai: An open-source retrieval-augmented large language model system for answering medical questions using scientific literature. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2024*. 8–23.

[81] Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2023. Dynamic Prompt Learning via Policy Gradient for Semi-structured Mathematical Reasoning. In *ICLR*.

[82] Yu Lu, Junwei Bao, Yan Song, Zichen Ma, Shuguang Cui, Youzheng Wu, and Xiaodong He. 2021. RevCore: Review-Augmented Conversational Recommendation. In *ACL/IJCNLP (Findings)*. 1161–1173.

[83] Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y Zhao. 2023. Dr. icl: Demonstration-retrieved in-context learning. *arXiv:2305.14128* (2023).

[84] Chang Ma, Haiteng Zhao, Lin Zheng, Jiayi Xin, Qintong Li, Lijun Wu, Zhihong Deng, Yang Lu, Qi Liu, and Lingpeng Kong. 2023. Retrieved Sequence Augmentation for Protein Representation Learning. *bioRxiv* (2023), 2023–02.

[85] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models. *arXiv:2305.14283* (2023).

[86] Aristides Milios, Siva Reddy, and Dzmitry Bahdanau. 2023. In-context learning for text classification with many labels. In *Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*. 173–184.

[87] Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering Ambiguous Open-domain Questions. In *EMNLP*.

[88] Sewon Min, Weijia Shi, Mike Lewis, Xilun Chen, Wen-tau Yih, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2023. Nonparametric Masked Language Modeling. In *ACL (Findings)*. 2097–2118.

[89] Noor Nashid, Mifta Sintaha, and Ali Mesbah. 2023. Retrieval-based prompt selection for code-related few-shot learning. In *ICSE*. 2450–2462.

[90] Neil O'Hare, Paloma De Juan, Rossano Schifanella, Yunlong He, Dawei Yin, and Yi Chang. 2016. Leveraging user interaction signals for web image search. In *SIGIR*. 559–568.

[91] Md. Rizwan Parvez, Wasi Uddin Ahmad, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. 2021. Retrieval Augmented Code Generation and Summarization. In *EMNLP (Findings)*. 2719–2734.

[92] Fabio Petroni, Patrick S. H. Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How Context Affects Language Models' Factual Predictions. In *AKBC*.

[93] Gabriel Poesia, Alex Polozov, Vu Le, Ashish Tiwari, Gustavo Soares, Christopher Meek, and Sumit Gulwani. 2022. Synchromesh: Reliable Code Generation from Pre-trained Language Models. In *ICLR*.

[94] Anupam Purwar and Rahul Sundar. 2023. Keyword Augmented Retrieval: Novel framework for Information Retrieval integrated with speech interface. *arXiv:2310.04205* (2023).

[95] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*. 8748–8763.

[96] 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.

[97] 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. *J Mach Learn Res* 21, 140 (2020), 1–67.

[98] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Trans. Assoc. Comput. Linguist.* 11 (2023), 1316–1331.

[99] Ori Ram, Gal Shachaf, Omer Levy, Jonathan Berant, and Amir Globerson. 2022. Learning to Retrieve Passages without Supervision. In *NAACL-HLT*. 2687–2700.

[100] Parikshit Ram and Alexander G Gray. 2012. Maximum inner-product search using cone trees. In *KDD*. 931–939.

[101] Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, Vol. 242. Citeseer, 29–48.

[102] Rita Ramos, Bruno Martins, Desmond Elliott, and Yova Kementchedjhieva. 2023. Smallcap: lightweight image captioning prompted with retrieval augmentation. In *CVPR*. 2840–2849.

[103] Benjamin Z. Reichman and Larry Heck. 2024. Retrieval-Augmented Generation: Is Dense Passage Retrieval Retrieving? *https://arxiv.org/html/2402.11035v1* (2024).

[104] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *EMNLP-IJCNLP*. 3982–3992.

[105] Yubing Ren, Yanan Cao, Ping Guo, Fang Fang, Wei Ma, and Zheng Lin. 2023. Retrieve-and-sample: Document-level event argument extraction via hybrid retrieval augmentation. In *ACL*. 293–306.

[106] 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.

[107] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning To Retrieve Prompts for In-Context Learning. In *NAACL-HLT*. 2655–2671.

[108] Sara Sarto, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. 2022. Retrieval-augmented transformer for image captioning. In *CBMI*. 1–7.

[109] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language models can teach themselves to use tools. In *NeurIPS*.

[110] Zhihong Shao, Yeyun Gong, Minlie Huang, Nan Duan, Weizhu Chen, et al. 2023. Enhancing Retrieval-Augmented Large Language Models with Iterative Retrieval-Generation Synergy. In *EMNLP*.

[111] Fumin Shen, Wei Liu, Shaoting Zhang, Yang Yang, and Heng Tao Shen. 2015. Learning binary codes for maximum inner product search. In *ICCV*. 4148–4156.

[112] Shelly Sheynin, Oron Ashual, Adam Polyak, Uriel Singer, Oran Gafni, Eliya Nachmani, and Yaniv Taigman. 2023. kNN-Diffusion: Image Generation via Large-Scale Retrieval. In *ICLR*.

[113] Peng Shi, Rui Zhang, He Bai, and Jimmy Lin. 2022. XRICL: Cross-lingual Retrieval-Augmented In-Context Learning for Cross-lingual Text-to-SQL Semantic Parsing. In *EMNLP (Findings)*. 5248–5259.

[114] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv:2301.12652* (2023).

[115] Guy Shtar. 2021. Multimodal machine learning for drug knowledge discovery. In *WSDM*. 1115–1116.

[116] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval Augmentation Reduces Hallucination in Conversation. In *EMNLP (Findings)*. 3784–3803.

[117] Suzanna Sia and Kevin Duh. 2023. In-context learning as maintaining coherency: A study of on-the-fly machine translation using large language models. *arXiv:2305.03573* (2023).

[118] Amit Singhal et al. 2001. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.* 24, 4 (2001), 35–43.

[119] Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. *TACL* 11 (2023), 1–17.

[120] Mingyang Song, Yi Feng, and Liping Jing. 2023. Hisum: Hyperbolic interaction model for extractive multi-document summarization. In *WWW*. 1427–1436.

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

[122] Ziteng Sun, Ananda Theertha Suresh, Jae Hun Ro, Ahmad Beirami, Himanshu Jain, and Felix Yu. 2024. Spectr: Fast speculative decoding via optimal transport. In *NeurIPS*.

[123] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288* (2023).

[124] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions. In *ACL*.

[125] Ante Wang, Linfeng Song, Qi Liu, Haitao Mi, Longyue Wang, Zhaopeng Tu, Jinsong Su, and Dong Yu. 2023. Search-engine-augmented dialogue response generation with cheaply supervised query production. *Artificial Intelligence* 319 (2023), 103874.

[126] Boxin Wang, Wei Ping, Peng Xu, Lawrence McAfee, Zihan Liu, Mohammad Shoeybi, Yi Dong, Oleksii Kuchaiev, Bo Li, Chaowei Xiao, et al. 2023. Shall We Pretrain Autoregressive Language Models with Retrieval? A Comprehensive Study. In *EMNLP*. 7763–7786.

[127] Hanbing Wang, Xiaorui Liu, Wenqi Fan, Xiangyu Zhao, Venkataramana Kini, Devendra Yadav, Fei Wang, Zhen Wen, Jiliang Tang, and Hui Liu. 2024. Rethinking Large Language Model Architectures for Sequential Recommendations. *arXiv:2402.09543* (2024).

[128] Liang Wang, Nan Yang, and Furu Wei. 2024. Learning to Retrieve In-Context Examples for Large Language Models. In *EACL*. 1752–1767.

[129] Xintao Wang, Qianwen Yang, Yongting Qiu, Jiaqing Liang, Qianyu He, Zhouhong Gu, Yanghua Xiao, and Wei Wang. 2023. Knowledgpt: Enhancing large language models with retrieval and storage access on knowledge bases. *arXiv:2308.11761* (2023).

[130] Yile Wang, Peng Li, Maosong Sun, and Yang Liu. 2023. Self-Knowledge Guided Retrieval Augmentation for Large Language Models. In *EMNLP*.

[131] Zichao Wang, Weili Nie, Zhuoran Qiao, Chaowei Xiao, Richard G. Baraniuk, and Anima Anandkumar. 2023. Retrieval-based Controllable Molecule Generation. In *ICLR*.

[132] Zifeng Wang, Zichen Wang, Balasubramaniam Srinivasan, Vassilis N Ioannidis, Huzefa Rangwala, and Rishita Anubhai. 2023. BioBridge: Bridging Biomedical Foundation Models via Knowledge Graph. *arXiv:2310.03320* (2023).

[133] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*. 24824–24837.

[134] Junda Wu, Cheng-Chun Chang, Tong Yu, Zhankui He, Jianing Wang, Yupeng Hou, and Julian McAuley. 2024. CoRAL: Collaborative Retrieval-Augmented Large Language Models Improve Long-tail Recommendation. *arXiv:2403.06447* (2024).

[135] Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable Zero-shot Entity Linking with Dense Entity Retrieval. In *EMNLP*. 6397–6407.

[136] Yuhuai Wu, Markus Norman Rabe, DeLesley Hutchins, and Christian Szegedy. 2022. Memorizing Transformers. In *ICLR*.

[137] Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *ICLR*.

[138] Jitao Xu, Josep-Maria Crego, and Jean Senellart. 2020. Boosting neural machine translation with similar translations. In *ACL*. 1570–1579.

[139] Jing Xu, Arthur Szlam, and Jason Weston. 2022. Beyond Goldfish Memory: Long-Term Open-Domain Conversation. In *ACL (1)*. 5180–5197.

[140] Ling Yang, Zhilin Huang, Xiangxin Zhou, Minkai Xu, Wentao Zhang, Yu Wang, Xiawu Zheng, Wenming Yang, Ron O Dror, Shenda Hong, et al. 2023. Prompt-based 3d molecular diffusion models for structure-based drug design. (2023).

[141] Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning. In *ICML*. 39818–39833.

[142] Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. Large Language Models are Versatile Decomposers: Decomposing Evidence and Questions for Table-based Reasoning. In *SIGIR*. 174–184.

[143] Antonio Jimeno Yepes, Yao You, Jan Milczek, Sebastian Laverde, and Leah Li. 2024. Financial Report Chunking for Effective Retrieval Augmented Generation. *arXiv:2402.05131* (2024).

[144] Dawei Yin, Yuening Hu, Jiliang Tang, Tim Daly, Mianwei Zhou, Hua Ouyang, Jianhui Chen, Changsung Kang, Hongbo Deng, Chikashi Nobata, et al. 2016. Ranking relevance in yahoo search. In *KDD*. 323–332.

[145] Dani Yogatama, Cyprien de Masson d'Autume, and Lingpeng Kong. 2021. Adaptive semiparametric language models. *TACL* 9 (2021), 362–373.

[146] Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making Retrieval-Augmented Language Models Robust to Irrelevant Context. In *ICLR*.

[147] Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than Retrieve: Large Language Models are Strong Context Generators. In *ICLR*.

[148] Wenhao Yu, Zhihan Zhang, Zhenwen Liang, Meng Jiang, and Ashish Sabharwal. 2023. Improving language models via plug-and-play retrieval feedback. *arXiv:2305.14002* (2023).

[149] Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. 2023. Augmentation-Adapted Retriever Improves Generalization of Language Models as Generic Plug-In. In *ACL*. 2421–2436.

[150] Shenglai Zeng, Jiankun Zhang, Pengfei He, Yue Xing, Yiding Liu, Han Xu, Jie Ren, Shuaiqiang Wang, Dawei Yin, Yi Chang, et al. 2024. The Good and The Bad: Exploring Privacy Issues in Retrieval-Augmented Generation (RAG). *arXiv:2402.16893* (2024).

[151] Boyu Zhang, Hongyang Yang, Tianyu Zhou, Muhammad Ali Babar, and Xiao-Yang Liu. 2023. Enhancing financial sentiment analysis via retrieval augmented large language models. In *CM International Conference on AI in Finance*. 349–356.

[152] Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2020. Grounded Conversation Generation as Guided Traverses in Commonsense Knowledge Graphs. In *ACL*. 2031–2043.

[153] Jiahao Zhang, Rui Xue, Wenqi Fan, Xin Xu, Qing Li, Jian Pei, and Xiaorui Liu. 2024. Linear-Time Graph Neural Networks for Scalable Recommendations. *arXiv:2402.13973* (2024).

[154] Yunxiang Zhang, Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. 2023. Merging generated and retrieved knowledge for open-domain QA. *arXiv:2310.14393* (2023).

[155] Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, and Bin Cui. 2024. Retrieval-Augmented Generation for AI-Generated Content: A Survey. *arXiv:2402.19473* (2024).

[156] Ruochen Zhao, Hailin Chen, Weishi Wang, Fangkai Jiao, Xuan Long Do, Chengwei Qin, Bosheng Ding, Xiaobao Guo, Minzhi Li, Xingxuan Li, et al. 2023. Retrieving multimodal information for augmented generation: A survey. *arXiv:2303.10868* (2023).

[157] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv:2303.18223* (2023).

[158] Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, et al. 2024. Recommender systems in the era of large language models (llms). *TKDE* (2024).

[159] Zexuan Zhong, Tao Lei, and Danqi Chen. 2022. Training Language Models with Memory Augmentation. In *EMNLP*.

[160] Shuyan Zhou, Uri Alon, Frank F Xu, Zhengbao Jiang, and Graham Neubig. 2022. Docprompting: Generating code by retrieving the docs. In *ICLR*.

[161] Yinghao Zhu, Changyu Ren, Shiyun Xie, Shukai Liu, Hangyuan Ji, Zixiang Wang, Tao Sun, Long He, Zhoujun Li, Xi Zhu, et al. 2024. REALM: RAG-Driven Enhancement of Multimodal Electronic Health Records Analysis via Large Language Models. *arXiv:2402.07016* (2024).

[162] Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. 2024. PoisonedRAG: Knowledge Poisoning Attacks to Retrieval-Augmented Generation of Large Language Models. *arXiv:2402.07867* (2024).