International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 8 Issue: 12
Article Received: *25 July 2020* ***Revised:*** *12 September 2020* ***Accepted:*** *30 December 2020*

A Review of Retrieval-Augmented Generation in Natural Language Processing: Architectures, Challenges, and Future Research Directions

Shashank Daram

University at Buffalo, Buffalo, New York, Designation: Student (Master's in computer science and engineering)

Abstract

Pre-trained Large Language Models (LLM) are known to have a number of parameters which implicitly contain world knowledge, and perform well on a wide range of Natural Language Processing (NLP) tasks. But these parameters are not transparent, not dynamic, not cheap to update, and can produce fluent but incorrect text, a phenomenon termed hallucination. Retrieval-Augmented Generation (RAG) bypasses these constraints, using a non-parametric memory, which is a collection of large text bodies, to be retrieved at inference time so that generation can be guided by retrieved evidence as well as by model weights. In this article we review the underpinnings of retrieval-augmented text generation as they evolved in the information retrieval (IR) and NLP literature up to 2021. With a retrieval-centric view, we break down the RAG paradigm into four steps: pre-retrieval, retrieval, post-retrieval, and generation, and explore key techniques for each step, such as approximate nearest neighbour indexing, sparse and dense retrieval, query modification, re-ranking and filtering, and generation conditioned by evidence. We present a taxonomy on representative systems, and analyse retrieval strategies along with algorithmic structure with a description of basic RAG workflow. We present an evaluation methodology review, comparative and meta-analytic tables summarising the field, and discussion of open challenges, retrieval quality, system efficiency, the role and selection of retrieval models, and promising directions for future research. The review brings a systematic categorisation and structure to early RAG studies, as well as a clear understanding of the underlying technology.

Keywords: retrieval-augmented generation, information retrieval, open-domain question answering, dense retrieval, neural language models, knowledge-grounded generation, hallucination.

1 Introduction

Modern NLP is based on neural language models that are pre-trained with impenetrable data on the Transformer [1]. Pre-trained models with a large-scale transformation like contextual encoders (ELMo [2] and BERT [3]) and generative models (GPT-2 [4], GPT-3 [5], T5 [6] and BART [7]) learn a lot of linguistic and factual knowledge in the pre-training process and are successfully applied to downstream tasks. Probing in these models reveals relational facts can be elicited with prompts in such models, which are wasps to some extent like soft knowledge bases [8] and closed book question answering also reveals that a large text-to-text model can answer questions using its parameters alone [9]. However, there are three main challenges with these developments: the problem of a fixed training corpus. Aligning broadly with others, pre-training is considered at a large scale, and broad equals good, meaning that on more specialised fields under-represented in the data, performance drops. Secondly, world knowledge is fixed at training time as well, and the increasing volume of information and the lack of time and cost to re-training hinder models to keep themselves updated. Third, due to their training on vast amounts of data, models can hallucinate and make believable yet factually incorrect claims [10].

One potential solution is a technique called Retrieval-Augmented Generation (RAG), which has the ability to retrieve external data in response to a query, thereby ensuring that the outputs are grounded in the latest, verifiable evidence. The benefit is demonstrated in figure 1: the language model knowledge prior to an event is unreliable; post-retrieval knowledge over an up-to-date language model on the other hand, is both correct and attributable. The concept of RAG was introduced in [11] where a sequence-to-sequence pre-trained generator is combined with a dense retriever over Wikipedia; similar ideas include REALM [12], which learns a retriever during pre-training, and the k-nearest-neighbour language model [13] which infuses generation at a token level with retrieval aspects. These systems sit on top of

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 8 Issue: 12
Article Received: *25 July 2020* ***Revised:*** *12 September 2020* ***Accepted:*** *30 December 2020*

advancements in open domain QA [14,15] and dense retrieval of a passage [16].

Prior to RAG, retrieval and generation were mainly researched in separate communities, using information retrieval for improving the ranking of documents and using parametric decoders for improving the generation of text. RAG brings them together, and as a single method can impact multiple stages of the pipeline, the literature that's being created can be easily confused. Considering its organisation in this review will go by a unified paradigm based on the IR point of view which splits RAG in four phases: pre-retrieval, retrieval, post-retrieval, and generation, allowing us to identify which precise part a technique improves. The emphasis of this survey will be on textual applications, reflecting the emphasis in early work.

Contributions. We present a formalisation of the basic RAG workflow as well as a four-phase paradigm and a taxonomy of representative systems from before 2021. We discuss the main algorithms in each stage: indexing, approximate nearest neighbor search, query manipulation, retrieval and ranking, re-ranking, filtering, and conditioning on evidences for generation, and provide retrieval strategies along with algorithmic structure. (iii) We discuss the methodology of evaluation and integrate the field into comparative and meta-analytic tables. Future directions and open challenges are discussed in (iv). The rest of the article is articulated as follows: Section 2 describes the RAG framework, Sections 3–6 discuss the different stages of the RAG (pre-retrieval, retrieval, post-retrieval, and generation), Section 7 covers evaluation, Section 8 discusses challenges and future directions, and Section 9 concludes the article.

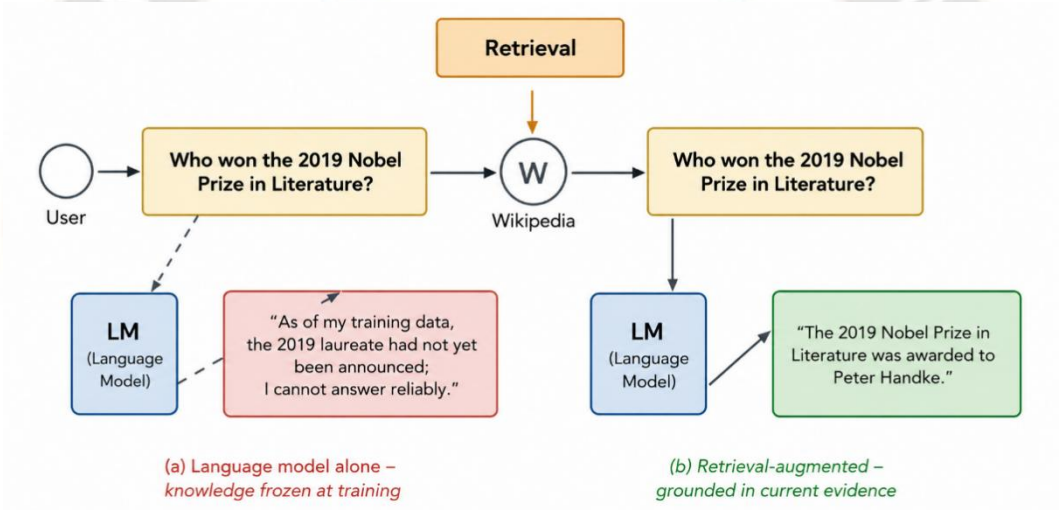

Fig. 1. An example of the benefit of RAG. A language model alone answers from knowledge frozen at training time and fails on facts postdating its corpus, whereas a retrieval-augmented model grounds its answer in current evidence and responds correctly.

2 RAG Framework

This lack of up-to-date information is more generally ascribed to the inability of a language model that is based on a fixed training corpus to handle hallucination. RAG alleviates this by augmenting the model's existing knowledge with up-to-date data from data stores outside the system with the help of a retrieval model, thus improving the accuracy of the results. RAG can also be a more economical solution than frequent re-training, as new data can be added to the RAG corpus and index, instead of retraining the model weights. The ability to ground in real human written documents makes RAG more reliable and easier to verify the outputs and more reliable when it comes to grounding. There are also clear precursors of retrieval augmentation, in memory-augmented neural networks, which provide an addressable external memory, read by attention [17,18,19] and RAG extends this idea to the case where the memory consists of an information-retrieval system and the memory itself is as large as a corpus: a representative example is the k-nearest-neighbour language model [13]. As time went on the paradigm shifted from supplying additional context in a single pass, to accommodation of one or multiple interactions between retrieval and generation; with several passes of retrieval to iteratively enhance the evidence, and hence the output.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 8 Issue: 12
Article Received: *25 July 2020* ***Revised:*** *12 September 2020* ***Accepted:*** *30 December 2020*

2.1 Basic RAG Workflow

Figure 2 shows the common RAG core concepts. The workflow starts with developing an index on external contents and the index is used as a retrieval basis of relevant information by using a retrieval model called "retriever" together with the query, before finally the retrieved relevant information is combined with the query by a type of generation model called "generator". Every RAG system can be broken down into the three pillars of indexing, retrieval, and generation.

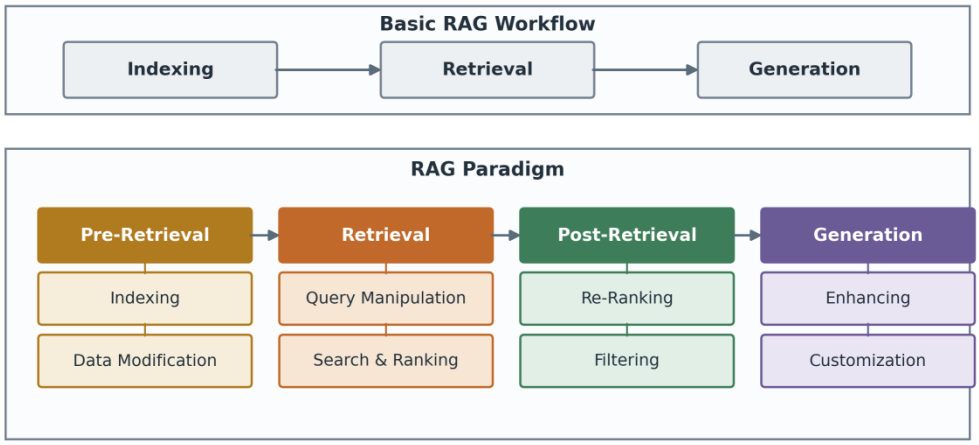

Fig. 2. The unified RAG core concepts. The basic workflow (indexing → retrieval → generation) is refined by the RAG paradigm into four phases: pre-retrieval, retrieval, post-retrieval, and generation, each with two sub-components.

2.1.1 Indexing

A proper indexing is the foundation of efficiency retrieval. In this step, data is pre-processed using techniques like normalisation, splitting documents into passages and, in contemporary systems, semantic vector representations are produced using pre-trained encoders. While sparse term vectors can be stored in a classical IR index and inverted [20], dense vectors as learned embeddings can be stored in a vector index for fast approximate retrieval [16]. Granularity of indexing varies according to the task; summarisation is addressed by document-level indexing while question answering is addressed by passage-level indexing.

2.1.2 Retrieval

Traditional retrieval methods (BM25) rank documents based on term frequency and document statistics, but ignore semantics [21]. Existing strategies rely on trained encoders with mean-sequence inputs like BERT [3] and compare query and document vectors according to their similarity for returning results similar to user's query [16]. The lexical and semantic signal can be combined to get the ranking, which is precise as well as strong.

2.1.3 Generation

Generation phase is when text relevant to the query and relevant from retrieved evidence is generated. The conventional way is to stick the query and the retrieved passages together and pass them onto a generator [11]. The problem is it will always be a trade-off between being faithful to the source and being fluent and coherent, getting the information retrieved and agreeing with it accurately without being difficult to read [10].

2.2 RAG Paradigm

From a retrieval point of view, pre, retrieval, post and generation can be used to differentiate the process of research within the domain, as illustrated in the taxonomy in Figure 3. Both single-hop and multi-hop (iterative retrieve-generate cycles) have this structure.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 8 Issue: 12
Article Received: *25 July 2020* ***Revised:*** *12 September 2020* ***Accepted:*** *30 December 2020*

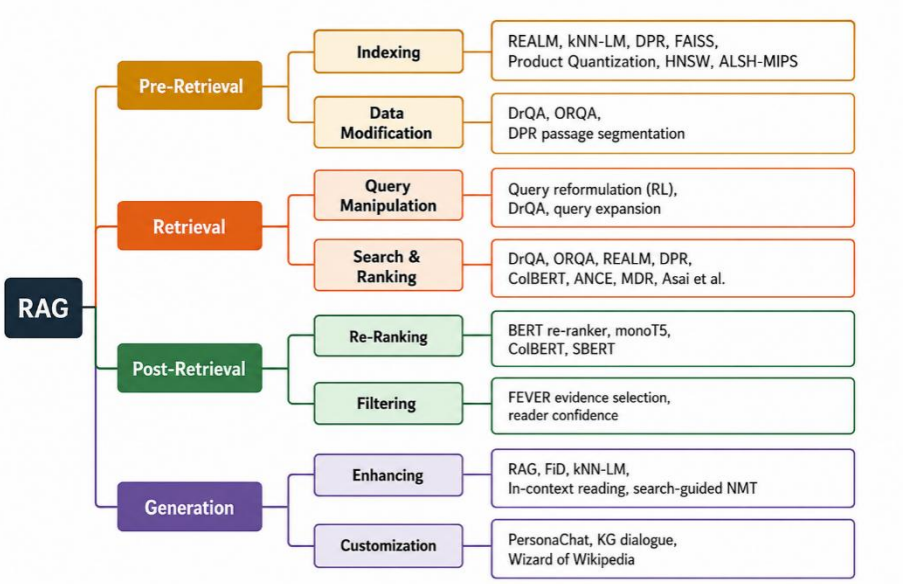

Fig. 3. Taxonomy of RAG core techniques organised by the four phases, with representative pre-2021 systems shown in brackets for each sub-component.

2.2.1 Pre-Retrieval

Indexing. Pre-retrieval phase is the establishment of an organized structure for fast and accurate access. The type of index (either sparse inverted or dense vector) and the granularity of the index is dependent on the task and data type [16,20].

Data Modification. In order to enhance the quality and diversity of content retrievable from such a system, pre-processing operations like the passage decomposition, redundant content removal and addition of metadata have been performed [16,15].

2.2.2 Retrieval

Query Manipulation. Query expansion [20] to introduce related terms, query reformulation [22] to accommodate user intent, and normalisation [21] to cope with spelling and terminology variations are examples of methods that adjust user queries to match better with data indexed on the site.

Search and Ranking. A ranking order the documents in the initial search result according to how well they match the query and it is carried out by search algorithms, based on sparse, dense or hybrid scoring [16,21,23] using the index.

2.2.3 Post-Retrieval

Re-Ranking. Models are retrieved, re-scored and re-ordered by more accurate and expensive models to provide even more relevant evidence across a small set of models [23,24,25].

Filtering. Only high-quality evidence is sent to the generator because documents with low relevance scores, or of other unhelpful document types, are deleted [16,26].

2.2.4 Generation

Enhancing. The information retrieved is incorporated into the query, reorganized to be clear and coherent, and re-expressed to be accurate, drawing on multiple pieces of evidence and logic to give a complete answer [11,27].

Customization. The output is customized based on the user's background and/or preferences by associating the output with a persona, a user history etc. [28,29,30].

3 Pre-Retrieval

3.1 Indexing

An organisational structure for access to information. Although the traditional inverted index is used for mapping words to documents for quick look-up, it is not good at locating semantically close documents using synonyms [20]. This means that when building RAG systems, people use indexing with dense vector embeddings. The problem is that looking for an exact match among countless millions of high dimensional vectors is time-consuming and space consuming; the main idea is therefore Approximate Nearest-Neighbour Search (ANNS) technics, which sacrifice a bit of accuracy for a lot of speed. ANNS methods can be divided into three types: graph-based, quantisation-based and hashing-based, each with its own set of trade-offs in

terms of accuracy, memory consumption and query speed.

Graph. Graph-based indexes create a proximity graph for structuring the vectors to facilitate search and to minimize distance computation over the entire set of vectors. The hierarchical Navigable Small World graphs can reach nearest neighbours in near-logarithmic time and excellent accuracy-speed trade-offs [31] and libraries like FAISS provide graph and GPU-accelerated billion scale search [32]. An index like that is used in the k-nearest-neighbour language model in which neighbours are retrieved from a large datastore when doing decoding [13].

Product Quantization. Instead of restricting the number of comparisons, one can reduce the cost of each comparison by quantising the vectors into small learned codes, which reduces memory and increases the speed of computing distance [33]. It contributes to resource efficiency which is essential for its large-scale deployment and is often used together with graph-indexes in FAISS [32].

Locality-Sensitive Hashing. Locality-sensitive hashing always "hops" vectors that are similar into the same hash bucket, allowing for efficient look up of all vectors that could be candidates. Under an appropriate transformation [34] maximum inner-product search can be transformed to nearest-neighbour search, due to the dense retrieval that is based on inner product. While graph and quantisation approaches are more popular in RAG, hashing can be beneficial when speed is paramount, albeit it comes with some compromises in accuracy.

3.2 Data Modification

Whether from an external knowledge base or in some designs from the model's own parametric knowledge, the quality and structure of indexed data is of paramount importance for RAG.

Corpus Preparation. Passage segmentation is a key design decision: dense retrievers like DPR also break articles into fixed-size passages; this enables retrieval and reading at a convenient level of granularity [16], while DrQA takes care to preprocess the Wikipedia corpus, using it for the lexical retriever [15]. The precision of retrieval is improved when redundant contents are cleaned, and the text is normalised.

Enrichment. Adding structure or metadata, like titles, section context, and hyperlinks, allows one to further augment the passages for relevance and to enable multi-hop search; one approach that could be done over the Wikipedia hyperlink graph, is the construction of chains of evidence for complex queries [35]. One drawback of using an external database is that if the retriever brings back too much noise or irrelevant sections, then the quality of the generated sentences will suffer.

4 Retrieval

4.1 Query Manipulation

Query manipulation narrows the semantic difference between the way in which a user phrases a query and the way in which some relevant information is expressed in the documents. This is significant in RAG, as good retrieval results will support the quality of the documents generated. There are three strategies in groups as shown below.

Query Expansion. My suggestion was to use the technique of adding synonyms or similar words to the query to find relevant documents, a classical IR technique, described in the literature more than ten years ago [20]. Terms could be retrieved from an initial retrieval pass or from external resources.

Query Reformulation. Reformulation: rewords the query to clarify and/or enhance it. Nogueira and Cho use a neural reformulator that has been trained through reinforcement learning (RL) to maximize recall of relevant documents by rewriting queries with terms chosen to enhance query retrieval [22]. These techniques help direct the re-b sparsely to the information required.

Normalization. Normalisation is used to resolve differences in spelling, casing and terminology for consistent matching and in interactive contexts to resolve context dependent reference, thereby defining a self-contained query to the retriever.

4.2 Search and Ranking

Search and ranking select and prioritise documents improving the output of generator. Sparse lexical models like BM25 do not take much time to calculate, are easy to understand and perform well when queries share vocabulary with relevant documents [21]. For instance, DPR is a dense bi-encoder that takes queries and documents and encodes them as vectors, and matches by taking the inner product, capturing semantics besides lexical overlap and significantly improving open domain QA [16]. For late-interaction models, for example, ColBERT, it is better to keep track of the token-level representations for finer matching at higher index cost [23]. The choice of negative samples is crucial when

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 8 Issue: 12
Article Received: *25 July 2020* ***Revised:*** *12 September 2020* ***Accepted:*** *30 December 2020*

training dense retrievers: ANCE is a late negative mining technique using the growing set of index negatives for more difficult retrieval examples to boost learning [36], and end-to-end learned systems like ORQA and REALM optimise retrieval in conjunction with the downstream task [12,14]. An efficient embedding for large scale similarity search and re-ranking is provided by Sentence-BERT [37]. In reality, sparse and dense signals are complementary, and deployed hybrid retrieval provides a good performance.

4.3 Retrieval Strategy

Different retrieval strategies have been developed to fit different needs and retrieval has evolved from one-step retrieval to dynamically defining when and what to retrieve. The most common RAG framework shown in Figure 4 also has an iterative retrieval strategy, and Table 1 summarizes some of the main retrieval strategies below. Existing retrieval is a key focus of most RAG research, rather than devising new retrieval algorithms.

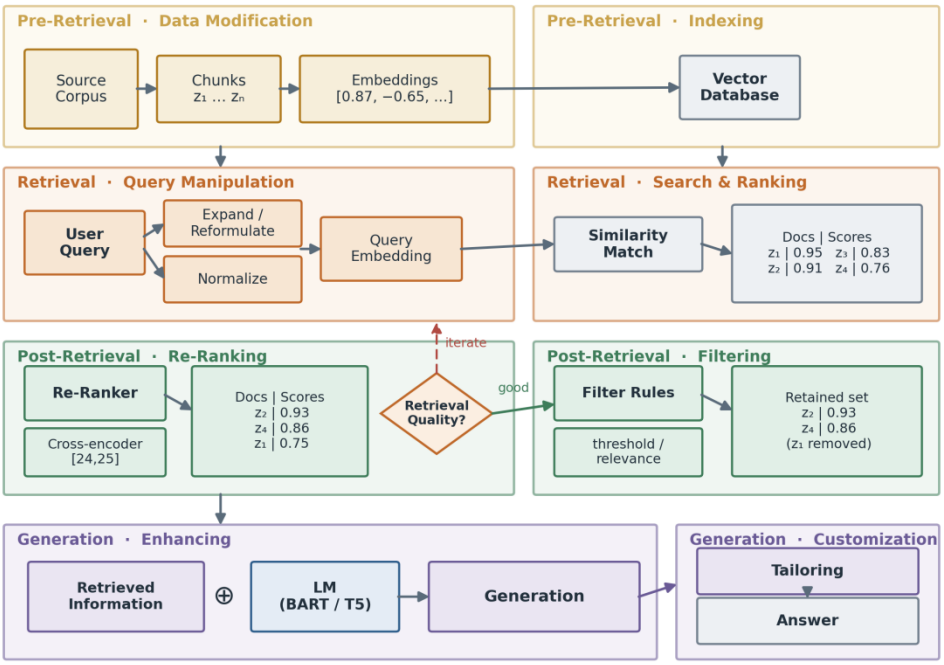

Fig. 4. A common RAG framework with an iterative retrieval strategy. Offline data modification and indexing feed an online loop of query manipulation, search and ranking, re-ranking, and filtering; a retrieval-quality check triggers further iterations before evidence-conditioned generation and customization.

Table 1. Comparison of the principal RAG retrieval strategies.

Strategy	Core mechanism	Primary use case	Complexity	Cost	Key strength
Basic	Retrieve-then-read (single pass)	Simple, factual question answering	Low	Low	Simplicity and speed
Iterative (multi-hop)	Repeated retrieval guided by prior output	Multi-step reasoning; intermediate facts	Medium	Med–high	Explicit step-by-step reasoning paths
Recursive (graph)	Navigates hierarchy / link structure	Structured corpora; connected evidence	High	Medium	Connected, multi-scale evidence

Cost grows with the number of retrieval rounds; the choice of strategy depends on question complexity and the structure of the corpus.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 8 Issue: 12
Article Received: *25 July 2020* ***Revised:*** *12 September 2020* ***Accepted:*** *30 December 2020*

Basic Retrieval. The principal idea of this basic strategy is a linear retrieve-then-read operation, where there are no loops, but only a sequence of operations starting with query, followed by retrieve and, finally, generation. It is appreciated for the simplicity, speed as in [16,15] DrQA and DPR, but a single pass might not suffice for complex questions to capture evidence needed.

Iterative (Multi-hop) Retrieval. Iterative approaches combine multiple retrievals, which provide their own correctness assurance in terms of using previous results to guide subsequent retrievals, with the aim of building the evidence on which multi-step reasoning depends. Recursively, multi-hop dense retrieval is the method that performs well for complex questions, while achieving the best performance in the multihop retrieval setting without using in-corpus hyperlinks [38]. The loop is described in Algorithm 1.

ALGORITHM 1: Iterative (Multi-hop) Retrieval Strategy

```
Require: query q, corpus D, index D_idx, retriever
R, generator G, max hops N
Ensure:  final output y
1:  D_idx <- Index(D)                  // offline: data
modification + indexing
2:  q' <- QueryManipulation(q);  C <- {}     // C:
accumulated context
3:  for i = 1 to N do
4:      D_i  <- R(q', D_idx)             // search and
initial ranking
5:      D_i' <- PostRetrieval(q', D_i)       // re-ranking
and filtering
6:      C    <- C U D_i'                 // accumulate
evidence
7:      y_i  <- G(q', C)                 // generate with
current context
8:      if StopCondition(y_i) then break
9:      q'   <- UpdateQuery(q', y_i)         // refine
query for next hop
10: end for
11: return y <- Synthesize({y_1,...,y_i})
```

Recursive (Graph-structured) Retrieval. Genera to Specifica which utilize hierarchy or linkage structure are recursive. Asai et al. track reasoning paths across a Wikipedia graph, by following hyperlinks to create a series of passages that cumulatively have an answer [35]. This is a very nice property in the case of structured knowledge, but the corpus must be prepared in advance as a navigable structure.

5 Post-Retrieval

5.1 Re-Ranking

The first stage retriever returns a wide set of candidates based on the velocity of retrieval, but the second stage re-ranking puts the retrieved documents in a more accurate order of relevance. Accurate but more expensive models can be used here, due to its relative few candidates that remain.

Unsupervised Re-ranking. Unsupervised re-rankers take advantage of the knowledge of the pre-trained models without any task-specific ranking labels. Sentence-BERT can be used to efficiently re-score candidates [37] and the fine-grained re-ranking signals in ColBERT can be used at the end of interaction [23].

Supervised Re-ranking. Supervised re-rankers are fine-tuned on relevance label for higher precision. Cross-encoders (such as BERT) encode both the query and the document together and output a score of relevance [24] and sequence-to-sequence re-rankers (such as monoT5) predict a relevance token, which is highly effective and has a reasonable out-of-domain generalisation [25]. These models come at a high price and work effectively over a set of candidates for re-ranking.

5.2 Filtering

Filtering eliminates documents that are irrelevant or of low quality to reduce the set sent to the generator, enhancing the downstream efficiency and correctness. One strategy is to determine a minimum threshold of relevance, or, in other words, to implement a cutoff decision based on the reader's confidence on the weak candidates [16]. For evidence-centric tasks, such as fact verification, after the production of the verdict, only sentences which support or contradict the claim are retained in an explicit evidence-selection step. Filtering is supplementary to re-ranking; filtering discards unhelpful documents, and re-ranking reorders the remaining documents.

6 Generation

6.1 Enhancing

Enhancing methods involve adding back content to enhance quality generated output and improve relevance. They vary in their ways in how they combine, aggregate or refine evidence and can be clustered under 3 categories.

Enhancing with Queries. The most straightforward one is to simply use retrieved passages as additional input to

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 8 Issue: 12
Article Received: *25 July 2020* ***Revised:*** *12 September 2020* ***Accepted:*** *30 December 2020*

accompany a query over leave the connection between the retrieved passages and the query to the attention mechanism of the model, which is a constrained design because of the context-size limit of existing models [11]. Passing a query and retrieved passage together, and marginalising over passages, as in RAG-Sequence and RAG-Token [11] are architectural variants.

Enhancing with Ensemble. If multiple passages are retrieved, information should be integrated from the several passages. Fusion-in-Decoder, in which the encoding is done on each passage separately with the question, and is only fused in the decoder, the generator can accumulate evidence over many passages with linear encoding cost, and accuracy improves with more passages provided [27]. Otherwise, the token-level k-nearest-neighbouring model is a distribution that interpolates the parametric distribution and a distribution derived from the retrieved k-neighbours, also without the need of additional training [13].

Enhancing with Feedback. In more complex queries, a single pass may not be enough as feedback loops are used to improve retrieval and generation in an iterative manner. Interleaving the retrieval with the generation of the results, with intermediate results used to retrieve next, enhances multi-step reasoning [35,38] but requires a longer delay.

6.2 Customization

Beyond generic generation, there is customization which makes outputs to a user's context and preferences. Persona-Grounded dialogue: generates answers based on the profile to ensure consistency with a certain persona [30]; Knowledge-Grounded conversation: fetches facts in order to ensure that the answers are correct and detailed enough [29]. This grounding is clearly stated in the Wizard of Wikipedia setting where answers should be accompanied by snippets from Wikipedia that are located/extracted from the internet. So customization falls somewhere between what is relevant and what is adapted to 'audience and context,' which was somewhat of a "blind spot" in early work on RAGs.

7 Evaluation in RAG

Evaluating RAG assesses how effectively a system produces accurate, relevant, and robust responses using external knowledge. Because a RAG system has two coupled stages, evaluation spans both retrieval and generation, and a complete assessment reports them together. Reading-comprehension datasets such as SQuAD [39] established span-based answer evaluation, which open-domain question answering extends to retrieval over a full corpus. Table 2 summarises the principal aspects, the metrics used, and representative datasets.

Table 2. Evaluation aspects, metrics, and representative datasets for retrieval-augmented generation.

Aspect	Focus	Metrics	Representative datasets
Retrieval quality	Relevance of retrieved passages to the query	Recall@k, MRR, MAP	NQ [40], TriviaQA [41]
Answer correctness	Correctness of the extracted or generated answer	Exact Match, token F1	NQ [40], WebQuestions [42], TriviaQA [41]
Generation quality	Fluency and overlap of free-form text	BLEU [43], ROUGE [44]	ELI5 [45], WoW [28]
Faithfulness	Factual consistency of output with evidence	Human evaluation, factuality [10]	FEVER [26], summarisation [10]
Provenance	Whether correct supporting evidence is returned	R-precision (KILT)	KILT [46]

KILT [46] unifies several knowledge-intensive tasks over a shared Wikipedia corpus and jointly evaluates answer correctness and provenance.

7.1 Retrieval-Based Aspect

Retrieval: whether it is providing passages that are relevant to the query and useful to the generator. The relevance is assessed by rank aware metrics as recall at k, mean reciprocal rank and mean average precision, and if gold evidence is annotated, provenance metrics verify that the relevant supporting document is retrieved [46]. Retrieval is a ceiling for generation, and the retriever can be enhanced, e.g., by using dense models and hard-negative training, to consistently improve the downstream task [16,36].

7.2 Generation-Based Aspect

The quality of the final text is used to assess the generation component. Exact match and F1 evaluate for factoid answers while BLEU [43] and ROUGE [44] evaluate for free-form answers and summarisation.

However, overlap metrics do not necessarily reflect factual correctness [10] and motivated faithfulness evaluation and provenance-aware benchmarks like KILT which reward retrieving the correct evidence instead of just providing the correct answer [46].

8 Challenges and Future Directions

8.1 The Role and Selection of Retrieval Models

Table 3 provides an overview of the different studies covered by this paper prior to 2021, including the type of retrieval (multi-hop vs. training) and the phases of the paradigm targeted. Analysis reveals that most systems rely on external data to augment context, dense retrieval is more preferred for the generation of higher quality candidates, and most effort is directed at retrieval, whereas comparatively little effort is given to customization in generation, offering a definite opportunity for future work. Each study used a retriever/generator indicated in table 4.

Table 3. Comprehensive summary of representative pre-2021 RAG studies across the four phases (✓ indicates the study addresses that aspect).

Research	Year	Internal	External	Multi-hop	Training	Indexing	Data Mod.	Query Man.	Search/ Rank	Re-Rank	Filter	Enhance	Custom.
DrQA [15]	2017		✓		✓	✓	✓		✓			✓	
ORQA [14]	2019		✓		✓	✓	✓		✓			✓	
REALM [12]	2020	✓	✓		✓	✓	✓		✓			✓	
DPR [16]	2020		✓		✓	✓	✓		✓		✓	✓	
RAG [11]	2020	✓	✓		✓				✓			✓	
FiD [27]	2021		✓		✓				✓			✓	
kNN-LM [13]	2020	✓	✓			✓						✓	
ColBERT [23]	2020		✓		✓	✓			✓	✓			
ANCE [36]	2021		✓		✓	✓			✓				
BERT re-rank [24]	2019		✓		✓					✓			
monoT5 [25]	2020		✓		✓					✓			
Query Reform. [22]	2017		✓		✓			✓	✓				
MDR [38]	2021		✓	✓	✓	✓			✓			✓	
Asai et al. [35]	2020		✓	✓	✓	✓	✓		✓			✓	
SBERT [37]	2019		✓		✓	✓			✓	✓			
Search-NMT [47]	2018		✓		✓				✓			✓	

Persona Chat [30]	2018		✓		✓				✓			✓	✓
KG dialogue [29]	2018		✓		✓				✓			✓	✓
WoW [28]	2019		✓		✓	✓			✓			✓	✓

Internal denotes use of the model's parametric knowledge; External denotes an external corpus. "Training" covers both training and fine-tuning.

Table 4. Summary of retrievers and generators used by representative pre-2021 RAG studies.

Research	Year	Retriever	Generator
DrQA [15]	2017	TF-IDF (sparse)	Multi-layer RNN reader
ORQA [14]	2019	BERT (inverse cloze)	BERT reader
REALM [12]	2020	Learned dense (MIPS)	BERT
DPR [16]	2020	BERT dual-encoder	BERT reader
RAG [11]	2020	DPR	BART
FiD [27]	2021	DPR / BM25	T5
kNN-LM [13]	2020	FAISS over datastore	Transformer LM
ColBERT [23]	2020	BERT (late interaction)	—
ANCE [36]	2021	RoBERTa dual-encoder	—
BERT re-rank [24]	2019	BM25 (first stage)	BERT (cross-encoder)
monoT5 [25]	2020	BM25	T5
MDR [38]	2021	RoBERTa dual-encoder	ELECTRA reader
Asai et al. [35]	2020	BERT (graph paths)	BERT reader
Search-NMT [47]	2018	Search engine	RNN seq2seq
WoW [28]	2019	TF-IDF / Transformer	Transformer

A dash denotes a retrieval or re-ranking method without an associated generation component.

One key decision is the retrieval model itself because that will define the quality of the information that is fed to the generator model. BM25 is a good baseline in domain-specific contexts, where the vocabulary is standardised, and it is still competitive as no vector index is needed and it is easily and quickly obtained. Neural retrievers can learn semantics and find relevant documents that share no synonyms of any query, and can even enhance the open domain retrieval and consequently the quality of generation [16,36]. One emerging space is the building of retrieval models tailored specifically for RAG that harness the benefits of both neural retrieval models and lexical retrieval models.

8.2 Retrieval Quality

Noise Robustness. Noise (irrelevant or misleading passages) during retrieval causes hallucination, but it is found to be reduced when the responses are grounded with retrieved evidences during conversation [48]. This confirms that there is a requirement to provide evidence retrieval and filtering for context awareness, which discriminates useful information from useless information [16,26].

Negative Rejection. Models tend to regardless provide response when retrieval does not find any evidence of relevance, which leads to error rates especially with under-specified queries. To overcome this, query reformulation and query dropping (when confidence is low) are required [22].

Information Integration. Using more than one document to answer a question and synthesizing evidence will result in more complex questions which generate more incomplete responses when the evidence is fragmented or conflicted. Multi-hop and graph-based retrieval facilitates integration by building connected pieces of evidence [35,38] and robustly reconciling conflicting sources is an open question.

8.3 System Efficiency

Retrieval Latency. Along large indexes, the cost of nearest-neighbour search at inference is often the dominant cost; approximate methods and compression

reduce the cost, but trade off some accuracy [32,33,31,34].

Index Maintenance. Realising updatable knowledge requires paying upkeep on a large index when the corpus and retriever evolve; REALM's example of its asynchronous index refresh shows that it is expensive to keep learned representations fresh [12]. The practical issue of efficient incremental indexing.

Context Length. The retrieval of passages can introduce numerous tokens, and the generators may be disoriented by lengthy and cluttered context which is crucial when scaling up to larger generation [27].

8.4 Joint Training and Standardised Evaluation

Joint Training. Retrieving from a downstream objective is challenging because retrieval is inherently discrete and label information of evidence is sparse; marginalisation [11,12], distant supervision [14] and retriever pre-training [14] each attempt to tackle a fragment of this problem; to-date, stable, sample-efficient end-to-end training is ongoing.

Evaluation Standardisation. Heterogeneous corpora and metrics hinder comparison; the creation of unified, provenance-aware benchmarks like KILT can provide a step towards comparison [46].

Multi-hop and Multimodal Extensions (MHE). It is straightforward to extend retrieval augmentation to compositional multi-hop reasoning [35,38,49] and to multilingual and multimodal corpora, as well.

9 Conclusions

This article has provided a structured framework of understanding about retrieval-augmented generation and its contribution in improving language models. From the retrieval point of view, we introduced the basic RAG workflow and a four-phase paradigm: pre-retrieval, retrieval, post-retrieval and generation, and analysed the main techniques in each phase, such as approximate nearest neighbour indexing, dense retrieval, re-ranking, filtering and evidence-conditioned generation. We systematically organized representative pre-2021 systems in categories, analysed retrieval strategies and their algorithmic structure, summarized evaluation methodology, and aggregated the field using comparative and meta-analytic tables. The analysis suggests that retrieval augmentation helps boost factual accuracy, promotes parameter efficiency, and mitigates hallucination when compared with purely parametric models, but presents challenges to the quality of retrieval, efficiency of systems, selection of retrievers, joint training, and standardised evaluation. This review aims to clarify the main concepts of RAG from the retrieval perspective and pave the way for deeper exploration and innovation in reliable retrieval and generation of information.

References

1. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 5998–6008.
2. M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in Proc. NAACL-HLT, 2018, pp. 2227–2237.
3. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. NAACL-HLT, 2019, pp. 4171–4186.
4. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” OpenAI Technical Report, 2019.
5. F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller, “Language models as knowledge bases?” in Proc. EMNLP-IJCNLP, 2019, pp. 2463–2473.
6. K. Lee, M.-W. Chang, and K. Toutanova, “Latent retrieval for weakly supervised open domain question answering,” in Proc. ACL, 2019, pp. 6086–6096.
7. D. Chen, A. Fisch, J. Weston, and A. Bordes, “Reading Wikipedia to answer open-domain questions,” in Proc. ACL, 2017, pp. 1870–1879.
8. J. Weston, S. Chopra, and A. Bordes, “Memory networks,” in Proc. Int. Conf. Learning Representations (ICLR), 2015.
9. S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, “End-to-end memory networks,” in Advances in Neural Information Processing Systems (NeurIPS), 2015, pp. 2440–2448.
10. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, et al., “Hybrid computing using a neural network with dynamic external memory,” Nature, vol. 538, no. 7626, pp. 471–476, 2016.
11. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge, U.K.: Cambridge Univ. Press, 2008.
12. S. Robertson and H. Zaragoza, “The probabilistic relevance framework: BM25 and beyond,” Found. Trends Inf. Retr., vol. 3, no. 4, pp. 333–389, 2009.

13. R. Nogueira and K. Cho, "Task-oriented query reformulation with reinforcement learning," in Proc. EMNLP, 2017, pp. 574–583.
14. R. Nogueira and K. Cho, "Passage re-ranking with BERT," arXiv:1901.04085, 2019.
15. J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "FEVER: A large-scale dataset for fact extraction and verification," in Proc. NAACL-HLT, 2018, pp. 809–819.
16. E. Dinan, S. Roller, K. Shuster, A. Fan, M. Auli, and J. Weston, "Wizard of Wikipedia: Knowledge-powered conversational agents," in Proc. Int. Conf. Learning Representations (ICLR), 2019.
17. M. Ghazvininejad, C. Brockett, M.-W. Chang, B. Dolan, J. Gao, W.-t. Yih, and M. Galley, "A knowledge-grounded neural conversation model," in Proc. AAAI Conf. Artificial Intelligence, 2018, pp. 5110–5117.
18. S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston, "Personalizing dialogue agents: I have a dog, do you have pets too?" in Proc. ACL, 2018, pp. 2204–2213.
19. H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 1, pp. 117–128, 2011.
20. Shrivastava and P. Li, "Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS)," in Advances in Neural Information Processing Systems (NeurIPS), 2014, pp. 2321–2329.
21. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proc. EMNLP-IJCNLP, 2019, pp. 3982–3992.
22. P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in Proc. EMNLP, 2016, pp. 2383–2392.
23. T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, et al., "Natural questions: A benchmark for question answering research," Trans. Assoc. Comput. Linguist., vol. 7, pp. 452–466, 2019.
24. M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, "TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension," in Proc. ACL, 2017, pp. 1601–1611.
25. J. Berant, A. Chou, R. Frostig, and P. Liang, "Semantic parsing on Freebase from question-answer pairs," in Proc. EMNLP, 2013, pp. 1533–1544.
26. K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in Proc. ACL, 2002, pp. 311–318.
27. -Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in Text Summarization Branches Out, Proc. ACL Workshop, 2004, pp. 74–81.
28. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, and M. Auli, "ELI5: Long form question answering," in Proc. ACL, 2019, pp. 3558–3567.
29. J. Gu, Y. Wang, K. Cho, and V. O. K. Li, "Search engine guided neural machine translation," in Proc. AAAI Conf. Artificial Intelligence, 2018, pp. 5133–5140.
30. Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning, "HotpotQA: A dataset for diverse, explainable multi-hop question answering," in Proc. EMNLP, 2018, pp. 2369–2380.