Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni,
Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis,
Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, Douwe Kiela

Facebook AI Research, University College London, New York University

ABSTRACT

Large pre-trained language models have been shown to store factual knowledge in their parameters, and achieve state-of-the-art results when fine-tuned on downstream NLP tasks. However, their ability to access and precisely manipulate knowledge is still limited, and hence on knowledge-intensive tasks, their performance lags behind task-specific architectures.

We introduce Retrieval-Augmented Generation (RAG), a general-purpose fine-tuning recipe that combines pre-trained parametric and non-parametric memory for language generation. RAG models use a pre-trained seq2seq model as the parametric memory and a dense vector index of Wikipedia as non-parametric memory, accessed with a pre-trained neural retriever.

We compare two RAG formulations: one which conditions on the same retrieved passages across the whole generated sequence, and another which can use different passages per token. We fine-tune and evaluate our models on a wide range of knowledge-intensive NLP tasks and set the state-of-the-art on three open-domain QA tasks, outperforming parametric seq2seq models and task-specific retrieve-and-extract architectures.

For language generation tasks, we find that RAG models generate more specific, diverse and factual language than a state-of-the-art parametric-only seq2seq baseline.

1. INTRODUCTION

Pre-trained neural language models have been shown to learn a substantial amount of in-depth knowledge about the world. This knowledge is stored implicitly in the parameters of neural networks, and can be used to answer factual questions, perform common sense reasoning, and generate fluent natural language.

However, the ability of these models to access and manipulate knowledge is still limited. They struggle to precisely recall factual knowledge and to expand their knowledge over time. When fine-tuned on downstream tasks, they can be effective, but their performance on knowledge-intensive tasks still lags behind approaches that use explicit non-parametric memory.

Task-specific architectures for knowledge-intensive tasks typically use a two-stage retrieve-then-generate pipeline. First, a retriever identifies relevant documents from a large corpus. Then, a reader extracts or generates an answer from those documents. While

effective, these approaches require substantial task-specific engineering and cannot easily transfer across tasks.

We propose Retrieval-Augmented Generation (RAG), a flexible fine-tuning approach that combines the strengths of parametric and non-parametric memory. RAG models use a pre-trained seq2seq model as the parametric component and a dense vector index as the non-parametric component. The retriever and generator are trained jointly end-to-end.

2. METHODS

2.1 Models

RAG models use a pre-trained retriever and a pre-trained seq2seq generator. Given an input query x, the retriever retrieves a set of documents z from a knowledge source. The generator then produces an output y conditioned on both x and z.

We explore two formulations:

RAG-Sequence uses the same retrieved documents for the entire generated sequence. This is appropriate when the entire answer should be derived from the same supporting documents.

RAG-Token can use different retrieved documents for each token in the output. This allows the model to combine information from multiple documents for different parts of the answer.

2.2 Retrieval

We use Maximum Inner Product Search (MIPS) to retrieve documents. The query and documents are embedded into the same vector space using BERT-based encoders. At training time, we update the query encoder but keep the document encoder frozen for efficiency.

The document index consists of 21 million Wikipedia passages, each approximately 100 words. Documents are pre-computed and indexed offline using FAISS for efficient retrieval.

2.3 Generation

The generator is a pre-trained BART model. It takes the concatenation of the input query and retrieved documents as context and generates the output autoregressively.

During training, we marginalize over retrieved documents, treating them as latent variables. This allows the model to learn which documents are useful for which queries without explicit supervision.

3. EXPERIMENTS

3.1 Tasks

We evaluate on four types of knowledge-intensive tasks:

Open-domain QA: Natural Questions, TriviaQA, WebQuestions
Abstractive QA: MSMARCO
Fact verification: FEVER
Jeopardy question generation

3.2 Results

RAG models achieve state-of-the-art results on Natural Questions (44.5% exact match), TriviaQA (56.8% exact match), and WebQuestions (45.2% exact match), outperforming previous approaches by 3-5%.

Compared to BART alone (parametric baseline), RAG provides gains of 5-10% on open-domain QA tasks. The improvement is particularly large on tasks requiring specific factual knowledge.

On abstractive QA and fact verification, RAG also outperforms strong baselines, though the gains are smaller. This suggests RAG is most beneficial when precise factual recall is critical.

3.3 Analysis

We analyze what makes RAG effective:

Retrieval quality: When we provide ground-truth documents instead of retrieved documents, performance increases substantially. This indicates retrieval is a bottleneck and better retrievers would improve results.

Number of documents: Performance improves as we retrieve more documents (up to ~10), then plateaus. This suggests the model can effectively use information from multiple sources.

RAG-Token vs RAG-Sequence: RAG-Token performs slightly better on open-domain QA, while RAG-Sequence is better for generation tasks. The optimal choice depends on whether answers require information from single or multiple documents.

Factuality: Human evaluation shows RAG generates more factual outputs than BART alone. Retrieval grounds generation in real documents, reducing hallucination.

4. RELATED WORK

Our work builds on several research directions:

Pre-trained language models (BERT, GPT, T5) have shown strong performance on NLP tasks by learning from large text corpora.

Open-domain QA systems traditionally use retrieve-then-read pipelines with separate retriever and reader components.

Memory-augmented neural networks use external memory to supplement model parameters, but typically with small, task-specific memory.

Dense retrieval uses learned dense embeddings rather than sparse features like TF-IDF or BM25 for document retrieval.

5. CONCLUSION

We introduced Retrieval-Augmented Generation, a flexible approach that combines parametric and non-parametric memory for knowledge-intensive NLP tasks. RAG models can be fine-tuned end-to-end and achieve state-of-the-art results on multiple benchmarks.

By treating retrieval as a latent variable, RAG learns to retrieve and use information without explicit supervision. This makes the approach general-purpose and applicable to many tasks.

Future work could explore:
- Iterative retrieval where the model retrieves multiple times during generation
- Better integration of retrieval and generation through cross-attention
- Scaling to larger document collections and longer contexts
- Multi-hop reasoning over retrieved documents

RAG represents a promising direction for building more knowledgeable and factual language models.

REFERENCES

[1] J. Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers. 2019.
[2] M. Lewis et al. BART: Denoising Sequence-to-Sequence Pre-training. 2020.
[3] C. Raffel et al. T5: Text-to-Text Transfer Transformer. 2020.
[4] V. Karpukhin et al. Dense Passage Retrieval for Open-Domain QA. 2020.
[5] K. Guu et al. REALM: Retrieval-Augmented Language Model Pre-Training. 2020.