بسم الله الرحمن الرحیم

Retrieval Augmented Generation

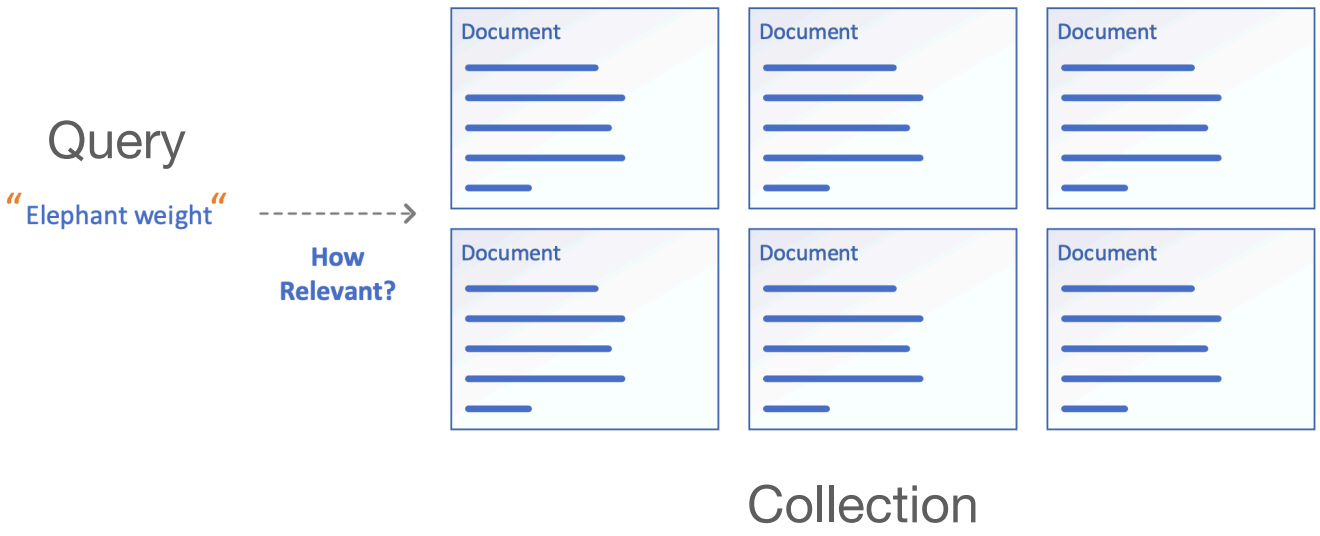
Ehsaneddin Asgari

Dec. 19th 2023

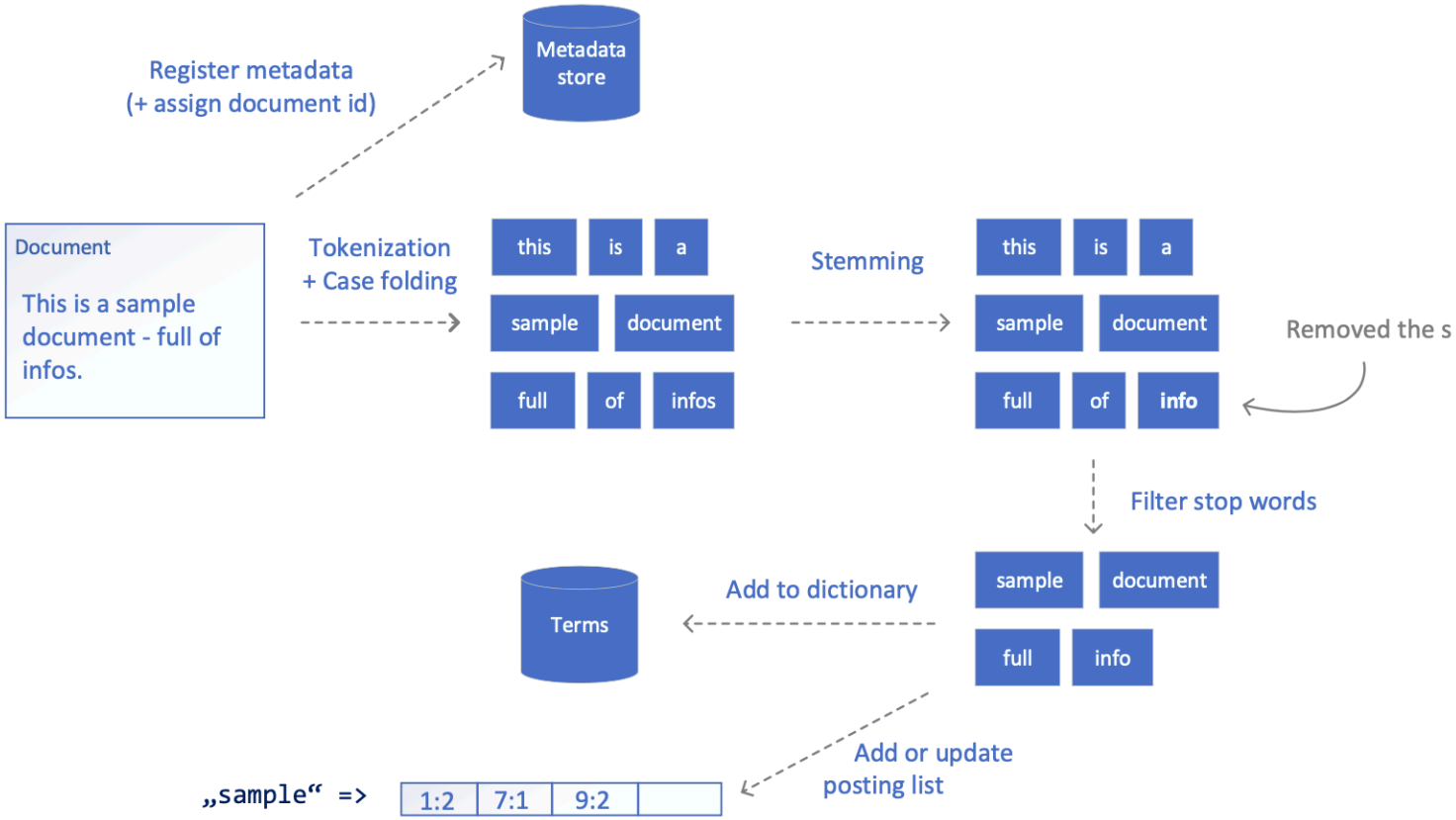
Artificial Intelligence Group
Computer Engineering Department, SUT

Review Basics in Retrieval

Retrieval


Query
“Elephant weight“
How
Relevant?
Document
Document
Document
Document
Document
Document
Collection

Inverted Index



TF–IDF

	W1	W2	...	Wn
Doc 1				
Doc 2			TF-idf	
⋮				
Doc m				

$$\mathrm{tf}(t, D) = \frac{\#(t, D)}{\max_{t' \in D} \#(t', D)}$$

$$\mathrm{idf}(t) = \log \frac{N}{\sum_{D: t \in D} 1}$$

$$\mathrm{tf.idf}(t, D) = \mathrm{tf}(t, D) \cdot \mathrm{idf}(t)$$

$$TF\_IDF(q,d) = \sum_{t \in T_d \cap T_q} log(1 + tf_{t,d}) * log(\frac{|D|}{df_t})$$

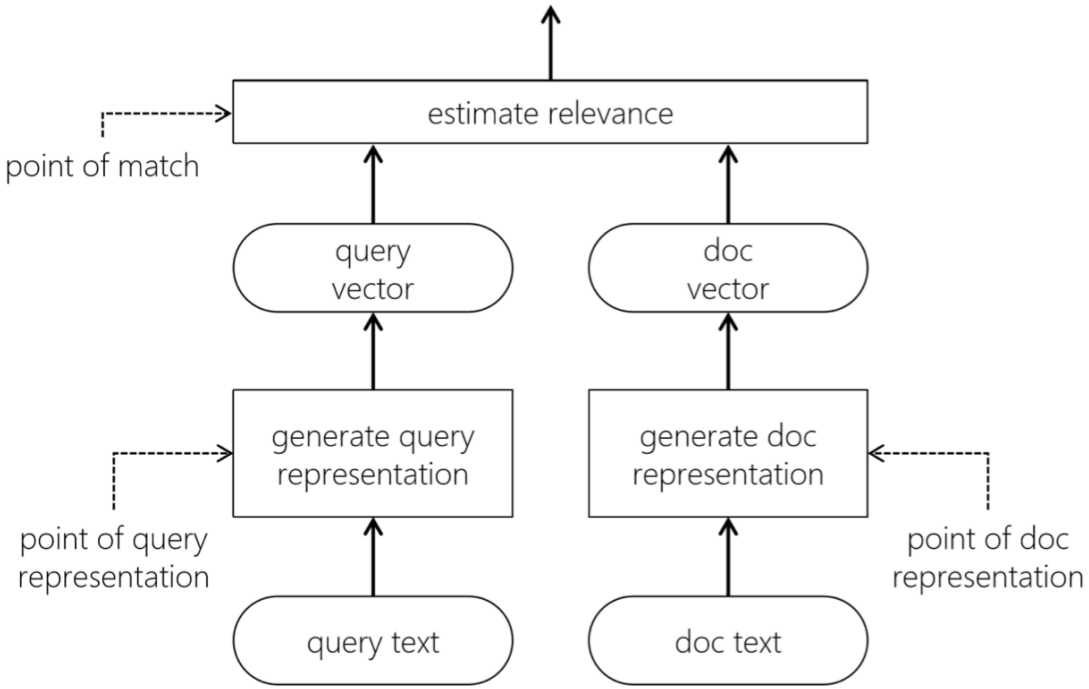
increases with the number of occurrences within a document

increases with the rarity of the term in the collection

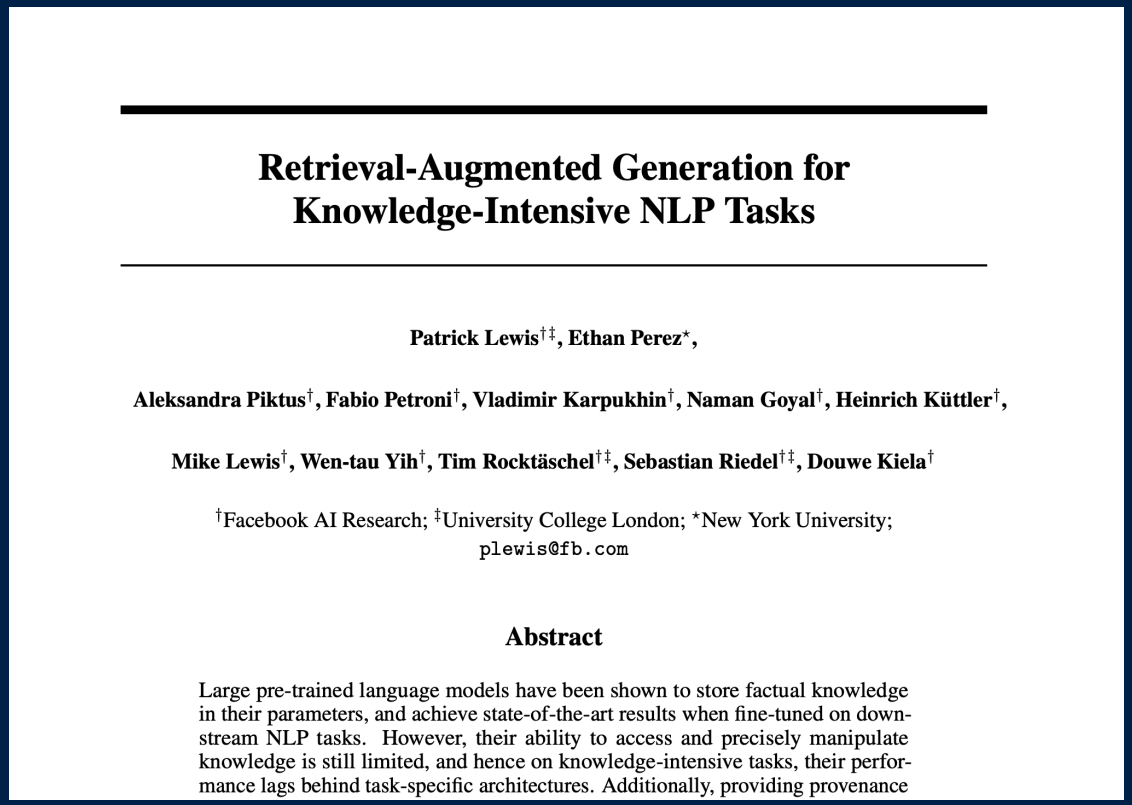
BM25 (as defined by Robertson et al. 2009)

$$BM25(q,d) = \sum_{t \in T_d \cap T_q} \frac{tf_{t,d}}{k_1((1-b) + b\frac{dl_d}{avgdl}) + tf_{t,d}} * log\frac{|D| - df_t + 0.5}{df_t + 0.5}$$

Embedding space for retrieval



Retrieval Augmented Generation



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;
plewis@fb.com

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. Additionally, providing provenance

NeurIPS 2020

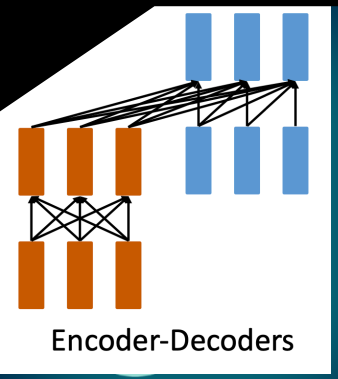
RAG Motivation

Encoder-decoder models are getting powerful

- Common sense/reasoning knowledge in parameters
- Strong results on many tasks
- Applicable for almost everything!

But

- Hallucinate
- Struggle to access and apply knowledge
- Difficult to update



Encoder-Decoders

Externally-retrieved knowledge required in many NLP tasks

- Precise and accurate knowledge access mechanism
- Easy updating at test time
- **Dense retrieval** starting to outperform traditional IR.

But often limited applicability because usually:

- Need retrieval supervision Or “heuristics"-based retrieval
- Need to integrate into downstream models

How can we combine the strengths of
encoder-decoder model and explicit knowledge retrieval?

Retrieval–augmented Generation (RAG)

- **Jointly** learn to **retrieve** and **generate** in end2end.
- **Latent retrieval -** no labels needed for retrieved docs
- **General recipe** for any seq2seq task

Needs 3 things:

- A (pretrained) retriever model P(z|x) e.g. **DPR**
- A (pretrained) generator model P(y|...) e.g. **BART or** T5
- An indexed KB of text documents Z e.g., **Wikipedia**

RAG combines **parametric** and ***non-parametric*** memory work well for ***knowledge intensive tasks***

Retriever: Dense Passage Retriever

$$p_\eta(z \mid x) \propto \exp\left(\mathbf{d}(z)^\top \mathbf{q}(x)\right) \quad \mathbf{d}(z) = \mathrm{BERT}_d(z) \quad \mathbf{q}(x) = \mathrm{BERT}_q(x)$$

Bi-Encoder Architecture | Document Encoder | Query Encoder

1. Get a pretrained **Bi-Encoder**
2. Encode Wikipedia Documents Once with **Document Encoder**
3. Finetune **Query Encoder** end-to-end with RAG

.. (DPR) ..

$$L\left(q_i, p_i^+, p_{i,1}^-, \cdots, p_{i,n}^-\right) = -\log \frac{e^{\text{sim}\left(q_i, p_i^+\right)}}{e^{\text{sim}\left(q_i, p_i^+\right)} + \sum_{j=1}^{n} e^{\text{sim}\left(q_i, p_{i,j}^-\right)}}$$

$$\text{sim}(q, p) = E_Q(q)^\top E_P(p)$$

Dense Passage Retrieval for Open-Domain Question Answering

Vladimir Karpukhin, **Barlas Oğuz**, **Sewon Min**†, **Patrick Lewis**, **Ledell Wu**, **Sergey Edunov**, **Danqi Chen**‡, **Wen-tau Yih**
Facebook AI †University of Washington ‡Princeton University
{vladk, barlaso, plewis, ledell, edunov, scottyih}@fb.com
sewon@cs.washington.edu
danqic@cs.princeton.edu

Abstract

Open-domain question answering relies on efficient passage retrieval to select candidate contexts, where traditional sparse vector space models, such as TF-IDF or BM25, are the de facto method. In this work, we show that retrieval can be practically implemented using *dense* representations alone, where embeddings are learned from a small number of questions and passages by a simple dual-encoder framework. When evaluated on a wide range of open-domain QA datasets, our dense retriever outperforms a strong Lucene-BM25 system greatly by 9%-19% absolute in terms of top-20 passage retrieval accuracy, and helps our end-to-end QA system establish new state-of-the-art on multiple open-domain QA benchmarks.[1]

Retrieval in open-domain QA is usually implemented using TF-IDF or BM25 (Robertson and Zaragoza, 2009), which matches keywords efficiently with an inverted index and can be seen as representing the question and context in high-dimensional, sparse vectors (with weighting). Conversely, the *dense*, latent semantic encoding is *complementary* to sparse representations by design. For example, synonyms or paraphrases that consist of completely different tokens may still be mapped to vectors close to each other. Consider the question *"Who is the bad guy in lord of the rings?"*, which can be answered from the context *"Sala Baker is best known for portraying the villain Sauron in the Lord of the Rings trilogy."* A term-based system would have difficulty retrieving such a context, while a dense retrieval system would be able to better

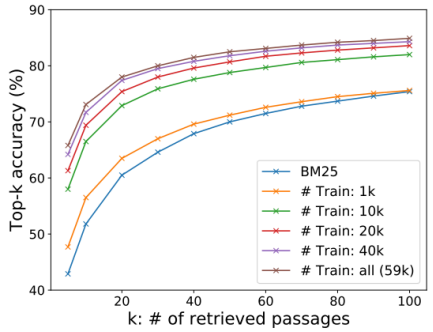


Figure 1: Retriever top-k accuracy with different numbers of training examples used in our dense passage retriever vs BM25. The results are measured on the development set of Natural Questions. Our DPR trained using 1,000 examples already outperforms BM25.

EMNLP 2020

RAG Architecture

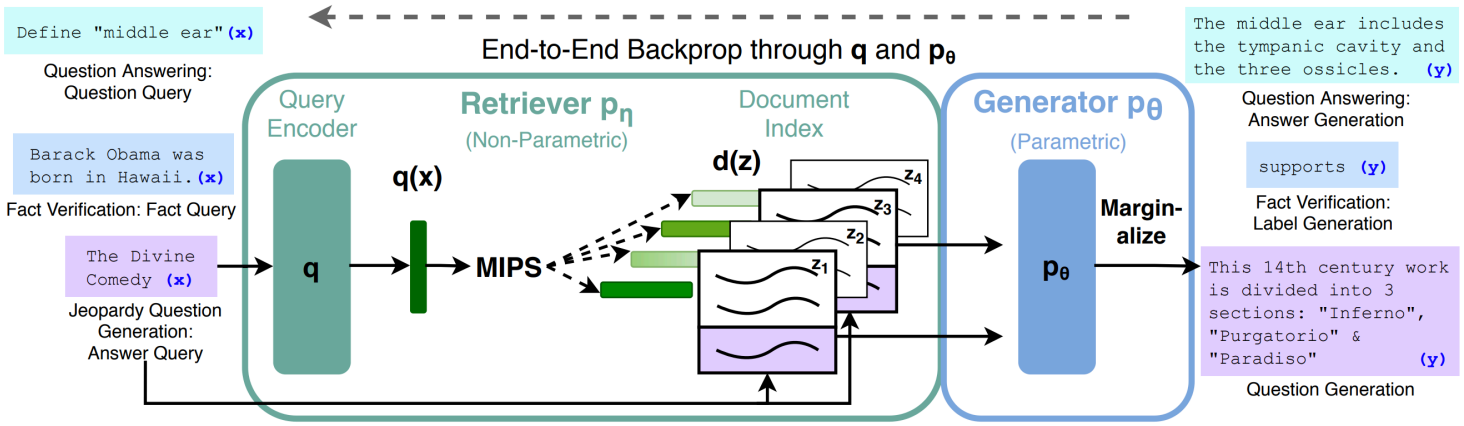


Figure 1: Overview of our approach. We combine a pre-trained retriever (*Query Encoder* + *Document Index*) with a pre-trained seq2seq model (*Generator*) and fine-tune end-to-end. For query x, we use Maximum Inner Product Search (MIPS) to find the top-K documents z_i. For final prediction y, we treat z as a latent variable and marginalize over seq2seq predictions given different documents.

RAG-Sequence Model

$$p_{\text{RAG-Sequence}}(y \mid x) \approx \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z \mid x) p_\theta(y \mid x, z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z \mid x) \prod_i^N p_\theta\left(y_i \mid x, z, y_{1:i-1}\right)$$

Retriever

Generator

RAG-Token Model

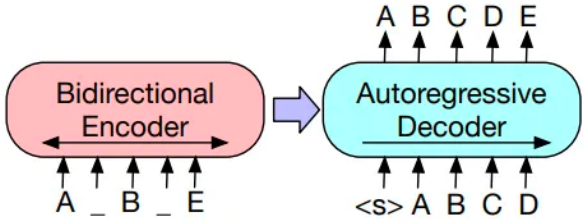
$$p_{\text{RAG-Token}}(y \mid x) \approx \prod_i^N \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta\left(z_i \mid x\right) p_\theta\left(y_i \mid x, z_i, y_{1:i-1}\right)$$

Both trained by directly minimising -log p(ylx)

Bidirectional and Auto-Regressive Transformers (BART)

- A bidirectional encoder and an autoregressive decoder.

- BART achieves the state of the art results in the summarization task.



- **RAG Simply concatenates Latent Document z to Input x**

Decoding from RAG Models

RAG-Token Model

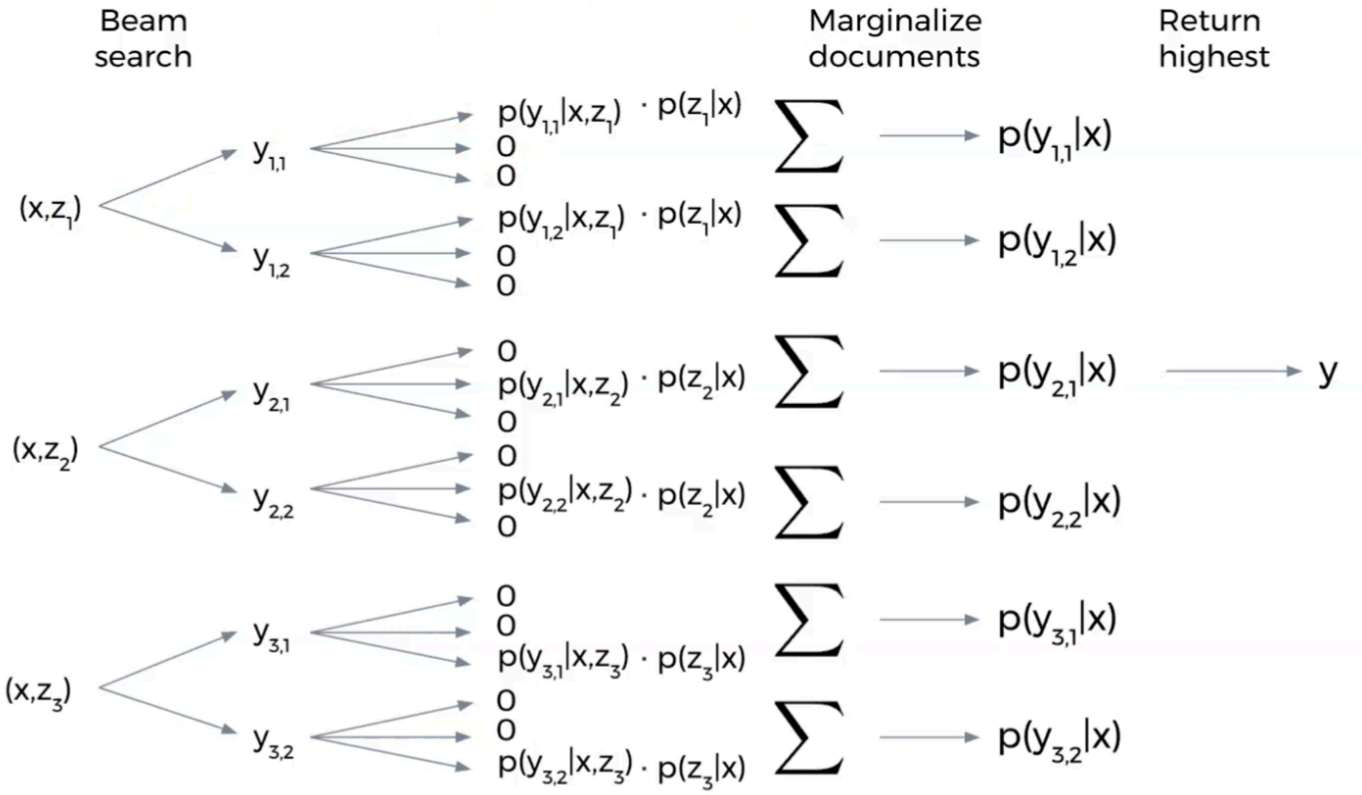
Standard Beam Search with transition probability:

$$p'_\theta\left(y_i \mid x, y_{1:i-1}\right) = \sum_{z \in \text{top}-k(p(\cdot|x))} p_\eta\left(z_i \mid x\right) p_\theta\left(y_i \mid x, z_i, y_{1:i-1}\right)$$

RAG-Sequence Model

...

RAG-Sequence Model Decoding



Experiments

- RAG can be applied to **any task with input and output sequences.**

- Focus on tasks with a clear need for precisely accessing knowledge

Open-domain QA:

Natural Questions, TriviaQA, WebQuestions, CuratedTREC

Abstractive open-domain QA:

“Open” MS MARCO

Question Generation:

Jeopardy questions

Fact Verification:

FEVER

Open-Domain QA

Table 1: Open-Domain QA Test Scores. For TQA, left column uses the standard test set for Open-Domain QA, right column uses the TQA-Wiki test set. See Appendix D for further details.

	Model	NQ	TQA	WQ	CT
Closed Book	T5-11B [52]	34.5	- /50.1	37.4	-
	T5-11B+SSM[52]	36.6	- /60.5	44.7	-
Open Book	REALM [20]	40.4	- / -	40.7	46.8
	DPR [26]	41.5	**57.9**/ -	41.1	50.6
	RAG-Token	44.1	55.2/66.1	**45.5**	50.0
	RAG-Seq.	**44.5**	56.8/**68.0**	45.2	**52.2**

- Strongly outperform “closed-book" models with specialized pretraining
- No span extraction required
- Docs that don’t contain exact answer still contribute to generating correct answer
- Answer questions correctly even when correct answer is not in retrieved docs

Abstractive Open-Domain QA

- Some questions unanswerable without gold passages
- RAG strongly outperforms BART baseline
- . Not so far from SoTA models which use gold passages

Top Retrieved doc:
A typical apple serving weighs 242 grams and provides 126 calories with a moderate content of dietary fiber (table). Otherwise, there is ... is usually not eaten and is discarded.

Input: how many calories in average apple

BART: The average apple contains 1,000 calories in an average apple and 1,200 calories in a medium apple

RAG: There are 126 calories in apple, while an extra large size apple has 172 calories.

GOLD: apple has 80 calories

Table 2: Generation and classification Test Scores. MS-MARCO SotA is [4], FEVER-3 is [68] and FEVER-2 is [57] *Uses gold context/evidence. Best model without gold access underlined.

Model	Jeopardy B-1	Jeopardy QB-1	MSMARCO R-L	MSMARCO B-1	FVR3 Label Acc.	FVR2 Label Acc.
SotA	-	-	**49.8***	**49.9***	**76.8**	**92.2***
BART	15.1	19.7	38.2	41.6	64.0	81.1
RAG-Tok.	**17.3**	**22.2**	40.1	41.5	72.5	89.5
RAG-Seq.	14.7	21.4	40.8	44.2		

Jeopardy Question Generation

Input: Washington

Gold: Florida's in the southeast corner of the 48 contiguous states; this state is in the northwest corner

BART: This state has the largest number of counties in the U.S.

RAG: Its the only U.S. state named for a U.S. President

Input: The Divine Comedy

BART: This epic poem by Dante is divided into three parts: the Inferno, The Purgatorio & the Purgatorio

RAG: This 14th Century work is divided into 3 sections: "inferno", "Purgatorio" & "Paradiso"

Jeopardy Question Generation

- Challenging knowledge intensive generation task
- Unlike other tasks **RAG-Token performs best** here
- Task requires integrating facts from different documents

Table 2: Generation and classification Test Scores. MS-MARCO SotA is [4], FEVER-3 is [68] and FEVER-2 is [57] *Uses gold context/evidence. Best model without gold access underlined.

Model	Jeopardy B-1	Jeopardy QB-1	MSMARCO R-L	MSMARCO B-1	FVR3 Label Acc.	FVR2 Label Acc.
SotA	-	-	**49.8***	**49.9***	**76.8**	**92.2***
BART	15.1	19.7	38.2	41.6	64.0	81.1
RAG-Tok	**17.3**	**22.2**	40.1	41.5	72.5	89.5
RAG-Seq	14.7	21.4	40.8	44.2		

Table 4: Human assessments for the Jeopardy Question Generation Task.

	Factuality	Specificity
BART better	7.1%	16.8%
RAG better	**42.7%**	**37.4%**
Both good	11.7%	11.8%
Both poor	17.7%	6.9%
No majority	20.8%	20.1%

Interaction Between Parametric / Non–Parametric Memory

RAG-Token document probability $p(z|x,y_{1:t-1})$ for input *"Hemingway"*

Document 1: his works are considered classics of American literature ... His wartime experiences formed the basis for his novel **"A Farewell to Arms"** (1929) ...

Document 2: ... artists of the 1920s "Lost Generation" expatriate community. His debut novel, **"The Sun Also Rises"**, was published in 1926.

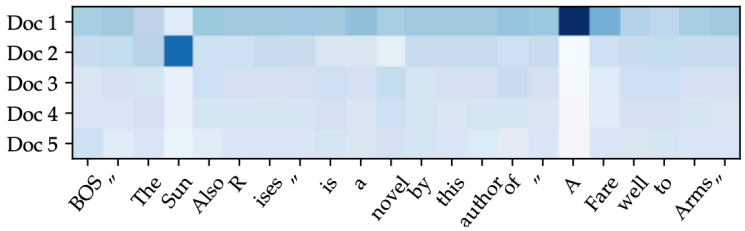


Figure 2: RAG-Token document posterior $p(z_i|x, y_i, y_{-i})$ for each generated token for input "Hemingway" for Jeopardy generation with 5 retrieved documents. The posterior for document 1 is high when generating "A Farewell to Arms" and for document 2 when generating "The Sun Also Rises".

Generation Diversity

Table 5: Ratio of distinct to total tri-grams for generation tasks.

	MSMARCO	Jeopardy QGen
Gold	89.6%	90.0%
BART	70.7%	32.4%
RAG-Token	77.8%	46.8%
RAG-Seq.	83.5%	53.8%

FEVER – Fact checking

- RAG may also be used for classification

- 3-way task
  - o 4.3% behind SOTA
  - o RAG trained only on (claim, label) pairs
  - o SOTA models use complex pipeline and strong retrieval supervision

- 2-way task:
  - o 2.7% RoBERTa model using gold evidence sentence at test time.

Table 2: Generation and classification Test Scores. MS-MARCO SotA is [4], FEVER-3 is [68] and FEVER-2 is [57] *Uses gold context/evidence. Best model without gold access underlined.

Model	Jeopardy B-1	Jeopardy QB-1	MSMARCO R-L	MSMARCO B-1	FVR3 Label Acc.	FVR2 Label Acc.
SotA	-	-	**49.8***	**49.9***	**76.8**	**92.2***
BART	15.1	19.7	38.2	41.6	64.0	81.1
RAG-Tok.	**17.3**	**22.2**	40.1	41.5	72.5	89.5
RAG-Seq.	14.7	21.4	40.8	44.2		

Document Index Hot-Swapping

Update model's memory on the fly by swapping the document index

Compare generated answers using ***2016*** vs. ***2018*** *Wikipedia* index.

Query RAG models `"who is the {position}"?` for world leaders who have changed between **2016** and **2018**

E.g., `"Who is the President of Peru?"`

2016 leaders and **2016** index: 70%
2016 leaders and **2018** index: 12%
2018 leaders and **2016** index: 4%
2018 leaders and **2018** index: 68%

Ablations

Table 6: Ablations on the dev set. As FEVER is a classification task, both RAG models are equivalent.

Model	NQ	TQA	WQ	CT	Jeopardy-QGen		MSMarco		FVR-3	FVR-2
		Exact Match			B-1	QB-1	R-L	B-1	Label Accuracy	
RAG-Token-BM25	29.7	41.5	32.1	33.1	17.5	22.3	55.5	48.4	**75.1**	**91.6**
RAG-Sequence-BM25	31.8	44.1	36.6	33.8	11.1	19.5	56.5	46.9		
RAG-Token-Frozen	37.8	50.1	37.1	51.1	16.7	21.7	55.9	49.4	72.9	89.4
RAG-Sequence-Frozen	41.2	52.1	41.8	52.6	11.8	19.6	56.7	47.3		
RAG-Token	43.5	54.8	**46.5**	51.9	**17.9**	**22.6**	56.2	**49.4**	74.5	90.6
RAG-Sequence	**44.0**	**55.8**	44.9	**53.4**	15.3	21.5	**57.2**	47.5		

- Retrieval-finetuning always helps, even when using supervised
- MIPS retrieval usually outperforms BM25 (FEVER is the exception)

Ablations

Table 6: Ablations on the dev set. As FEVER is a classification task, both RAG models are equivalent.

Model	NQ	TQA	WQ	CT	Jeopardy-QGen		MSMarco		FVR-3	FVR-2
		Exact Match			B-1	QB-1	R-L	B-1	Label Accuracy	
RAG-Token-BM25	29.7	41.5	32.1	33.1	17.5	22.3	55.5	48.4	**75.1**	**91.6**
RAG-Sequence-BM25	31.8	44.1	36.6	33.8	11.1	19.5	56.5	46.9		
RAG-Token-Frozen	37.8	50.1	37.1	51.1	16.7	21.7	55.9	49.4	72.9	89.4
RAG-Sequence-Frozen	41.2	52.1	41.8	52.6	11.8	19.6	56.7	47.3		
RAG-Token	43.5	54.8	**46.5**	51.9	**17.9**	**22.6**	56.2	**49.4**	74.5	90.6
RAG-Sequence	**44.0**	**55.8**	44.9	**53.4**	15.3	21.5	**57.2**	47.5		

- Retrieval-finetuning always helps, even when using supervised
- MIPS retrieval usually outperforms BM25 (FEVER is the exception)

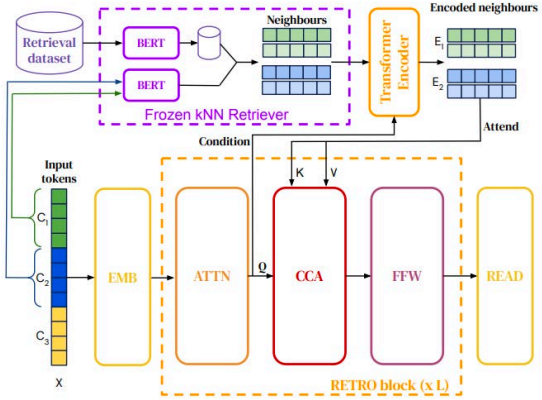
Other Variations: RETRO

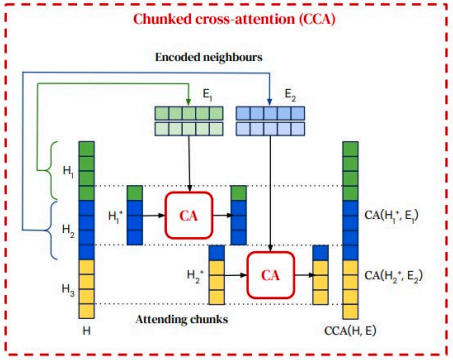
DeepMind

Improving language models by retrieving from trillions of tokens

Sebastian Borgeaud[†], Arthur Mensch[†], Jordan Hoffmann[†], Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae[‡], Erich Elsen[‡] and Laurent Sifre[†,‡]
All authors from DeepMind, [†]Equal contributions, [‡]Equal senior authorship

We enhance auto-regressive language models by conditioning on document chunks retrieved from a large corpus, based on local similarity with preceding tokens. With a 2 trillion token database, our Retrieval-Enhanced Transformer (RETRO) obtains comparable performance to GPT-3 and Jurassic-1 on the Pile, despite using 25× fewer parameters. After fine-tuning, RETRO performance translates to downstream knowledge-intensive tasks such as question answering. RETRO combines a frozen BERT retriever, a differentiable encoder and a chunked cross-attention mechanism to predict tokens based on an order of magnitude more data than what is typically consumed during training. We typically train RETRO from scratch, yet can also rapidly RETROfit pre-trained transformers with retrieval and still achieve good performance. Our work opens up new avenues for improving language models through explicit memory at unprecedented scale.





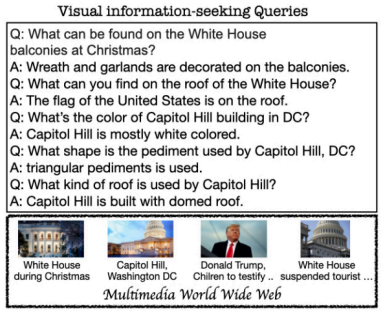
Other Variations: Multimodal

MuRAG: Multimodal Retrieval-Augmented Generator for Open Question Answering over Images and Text

Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, William W. Cohen
Google Research
{wenhuchen,hexiang,patverga,wcohen}@google.com

Abstract

While language Models store a massive amount of world knowledge implicitly in their parameters, even very large models often fail to encode information about rare entities and events, while incurring huge computational costs. Recently, retrieval-augmented models, such as REALM, RAG, and RETRO, have incorporated world knowledge into language generation by leveraging an external non-parametric index and have demonstrated impressive performance with constrained model sizes. However, these methods are restricted to retrieving only textual knowledge, neglect-



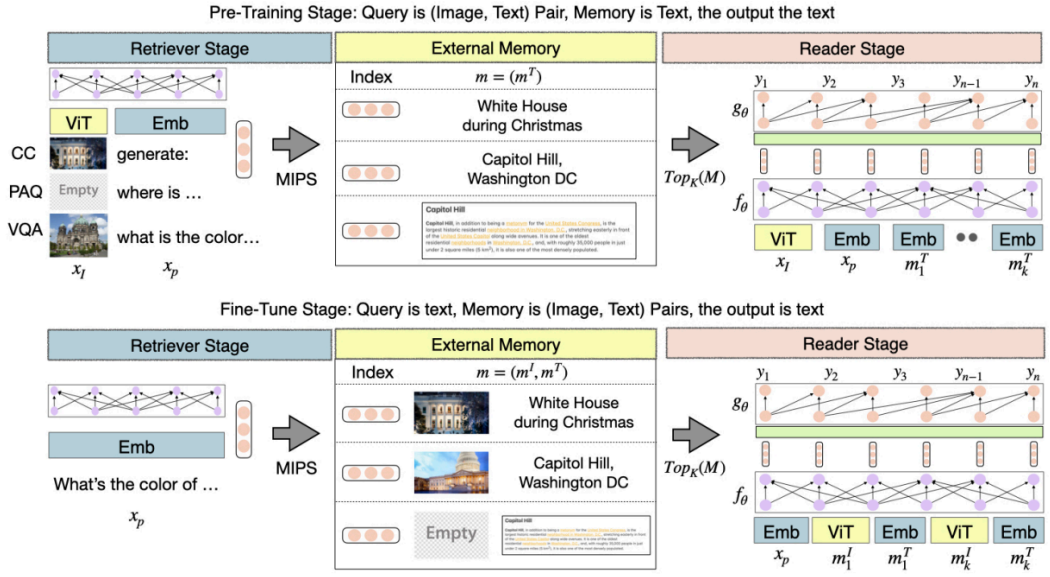


Figure 4: Model Architecture: the model accesses an external memory to obtain multimodal knowledge contained in images or text snippets, which is used to augment the generation. The upper part defines the pre-training implementation, while the lower part defines fine-tuning implementation.

Framework: llama index

https://www.llamaindex.ai/