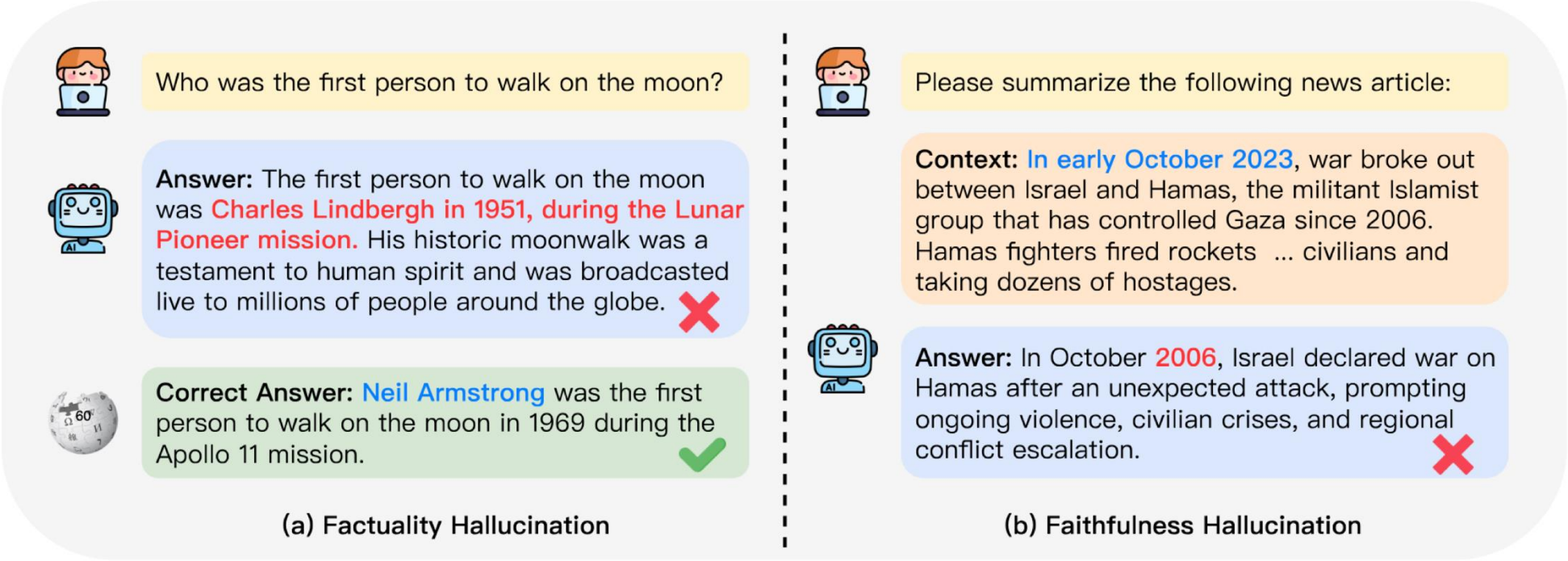
Natural Language Processing

Retrieval-Augmented Generation

Hallucination of LLM

- It is discovered that NLG models often generate text that is nonsensical, or unfaithful to the provided input. Such undesirable generation is referred to Hallucination (Ji et al., 2023).



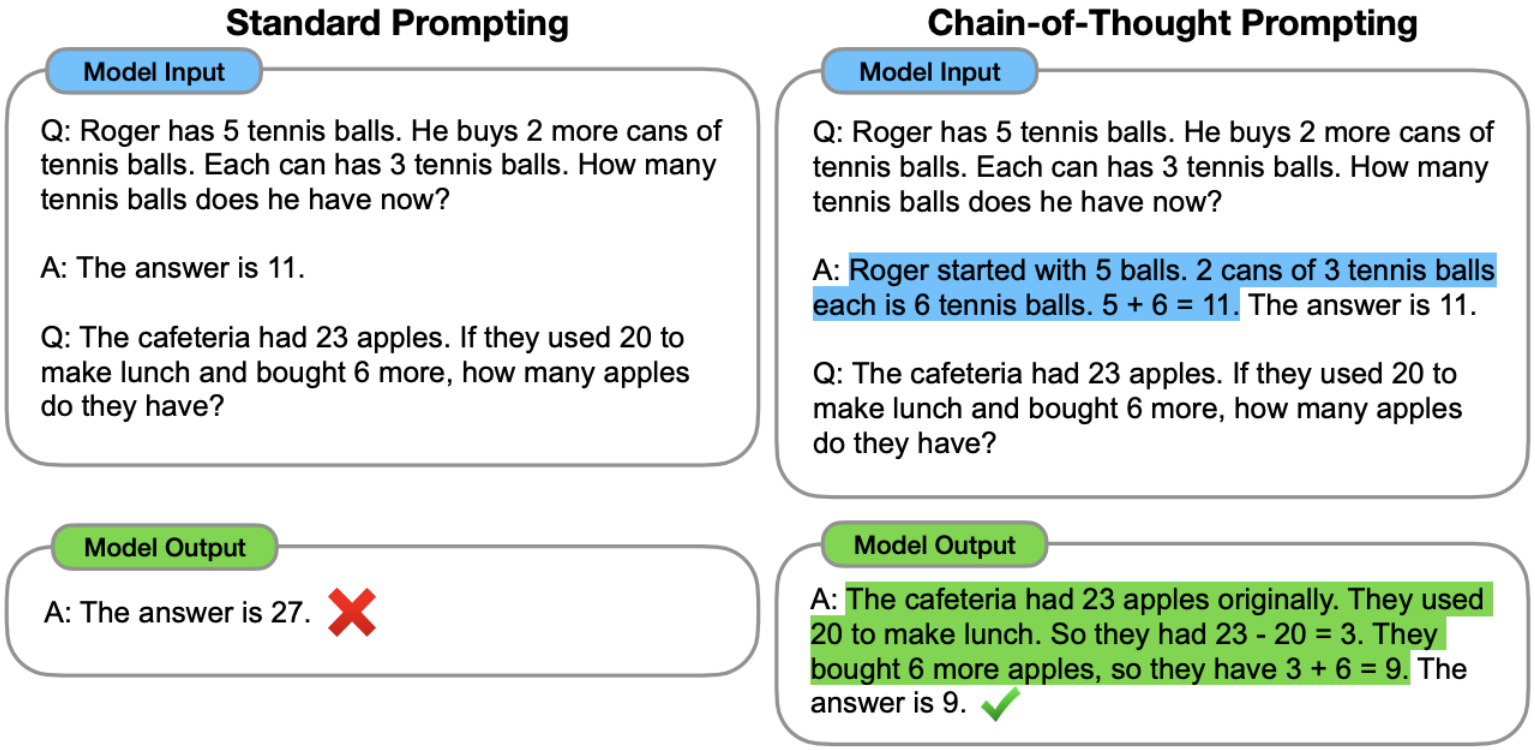
Ji et al. "Survey of hallucination in natural language generation." ACM Computing Surveys 55.12 (2023): 1-38.
Figure source: Munkhdalai, Tsendsuren, Manaal Faruqui, and Siddharth Gopal. "Leave no context behind: Efficient infinite context transformers with infini-attention." arXiv preprint arXiv:2404.07143 (2024).

Solutions to Mitigating Hallucinations

- Chain-of-Thought Prompting (CoT)
- Retrieval-Augmented Generation (RAG)
- …

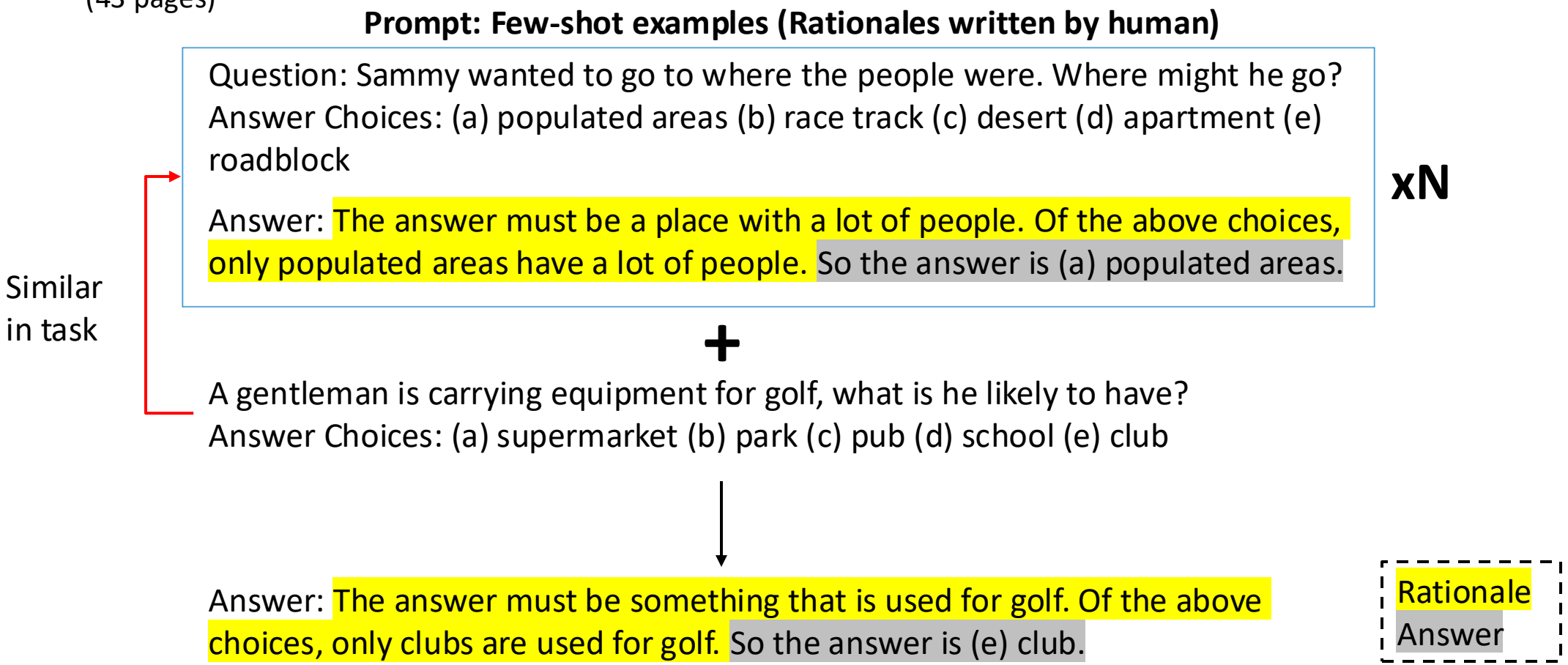
Chain-of-Thought Prompting (CoT)

- Wei et al., Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. NeurIPS 2022. By Google Brain. (43 pages)



Chain-of-Thought Prompting (CoT)

- Wei et al., Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. NeurIPS 2022. By Google Brain. (43 pages)



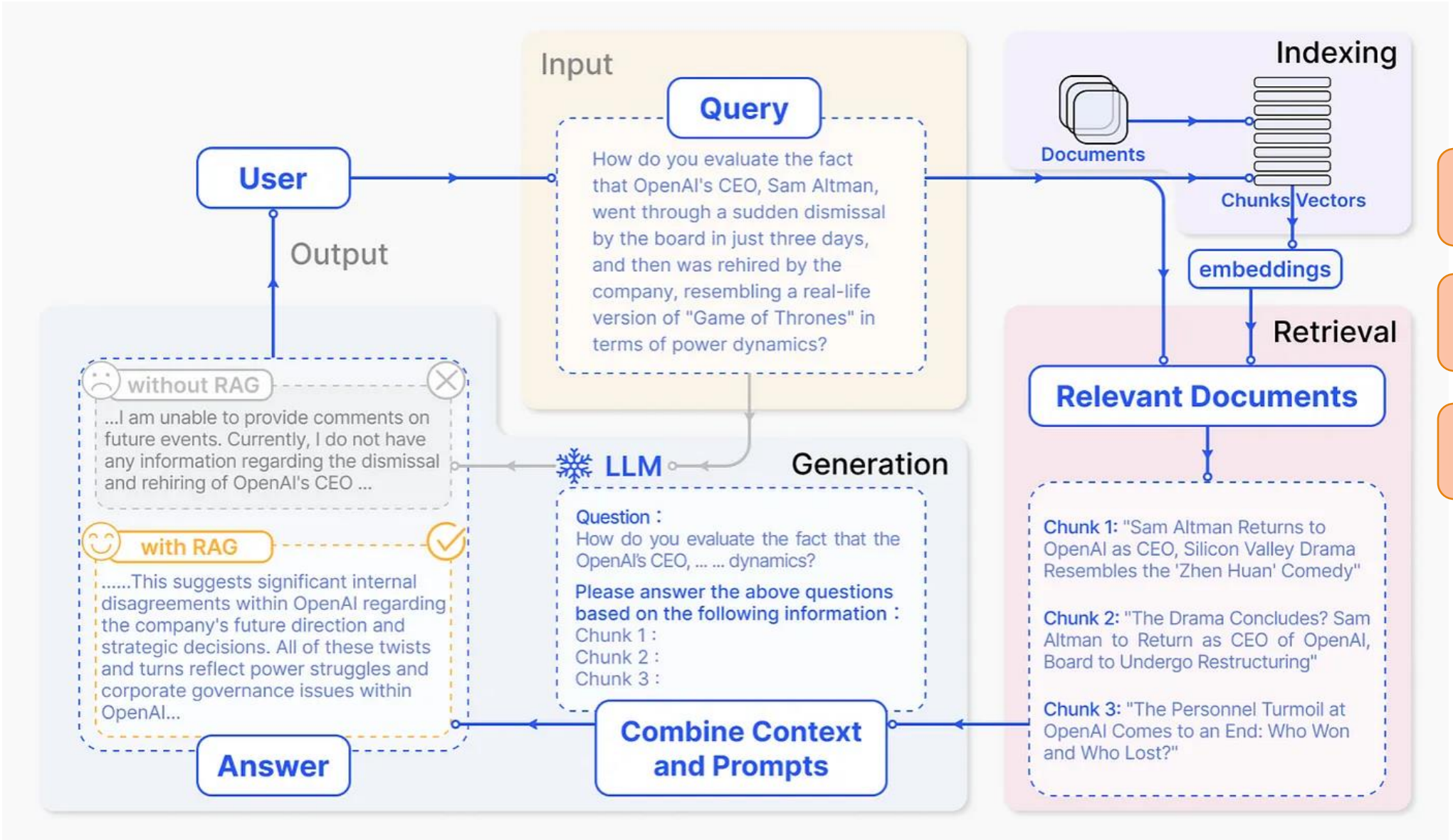
Retrieval-Augmented Generation (RAG)

Information Retrieval

- Retrieval: get relevant information from a pool (like a search engine)

- Retrieval-Augmented Generation (RAG): Perform generation with additional retrieved information

Gao, Yunfan, et al. "Retrieval-augmented generation for large language models: A survey." (arXiv 2023)



What Has Been Improved?

What Problems Remain Unresolved?

What New Problems Have Emerged?

An example of RAG with noise chunks

who won the womens 2017 ncaa basketball tournament?

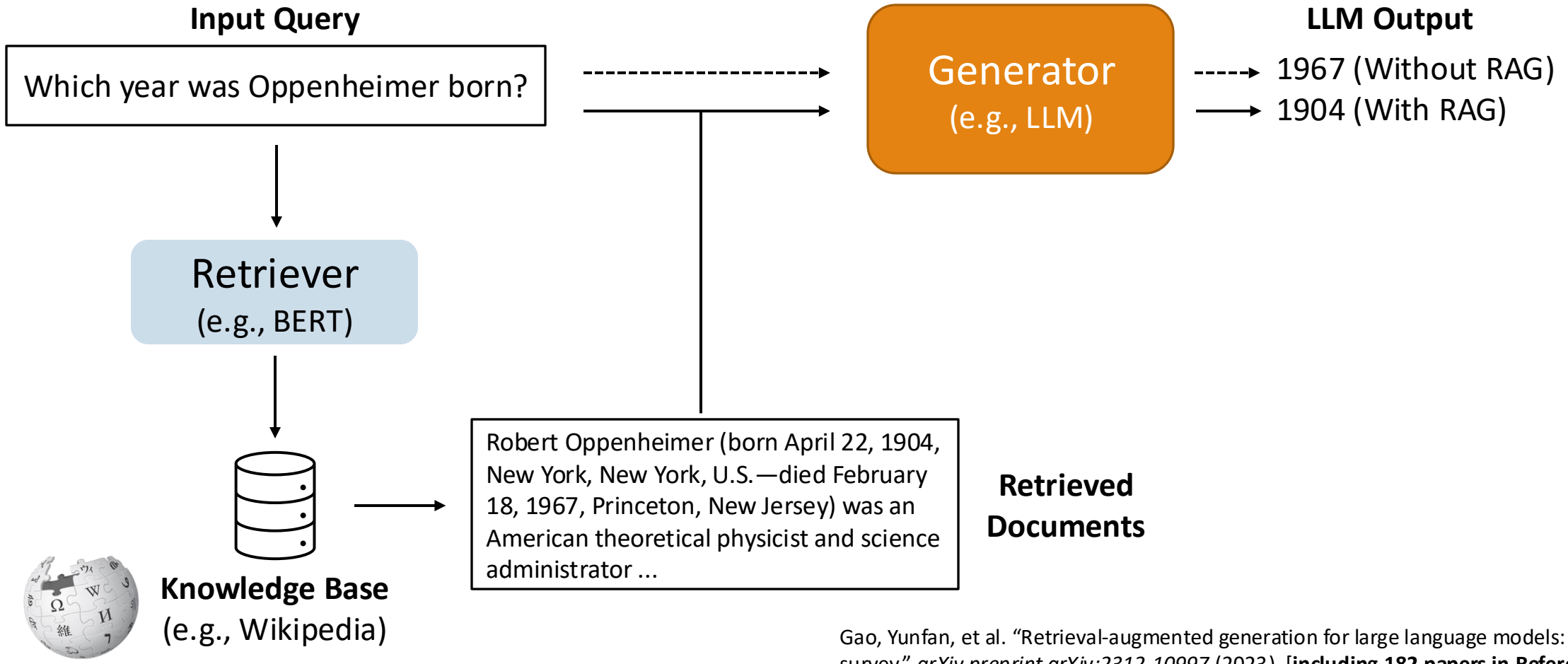
Retrieved text A: ...The **2016-17 NCAA Division I women's basketball season** began on November 11, 2016 and ended with the Final Four title game in Dallas on April 2, 2017, won by **South Carolina**. Practices officially began on September 30, 2016. ...

Retrieved text B: ... **The 2017 NCAA Women's Division I Basketball** Tournament was played from Friday, March 17 to Sunday, April 2, 2017, ...**Notre Dame** defeated Mississippi State to win the championship. ...

Document B states that **Notre Dame** defeated Mississippi State to win the championship. ... Document A is about the 2016-2017 NCAA Division I women's basketball season, which is not relevant to the question. ..., the answer to the question "who won the women's 2017 NCAA basketball tournament?" is **Notre Dame**.

The external argument claims that **South Carolina** won the championship, citing Document A, ...Document B also mentions that **Notre Dame** defeated Mississippi State to win the championship, ...The internal argument also claims that **South Carolina** won the championship, ...The external argument provides a clear and consistent claim that South Carolina won the championship, **supported by Document A and internal knowledge**. Therefore, the correct answer is that **South Carolina** won the 2017 NCAA Division I Women's Basketball Championship.

Retrieval-Augmented Generation (RAG)



Gao, Yunfan, et al. "Retrieval-augmented generation for large language models: A survey." *arXiv preprint arXiv:2312.10997* (2023). [**including 182 papers in Reference**]

Gao, Yunfan, et al. “Retrieval-augmented generation for large language models: A survey.” *arXiv preprint arXiv:2312.10997* (2023). [**including 182 papers in Reference]**

Retrieval-Augmented Generation (RAG)

TABLE I
SUMMARY OF RAG METHODS

Retrieval Source	Retrieval Data Type	Retrieval Granularity	Augmentation Stage	Retrieval process
Wikipedia	Text	Phase	Pre-training	Once
FactoidWiki	Tabular text	Sentence	Inference	Iterative
Dataset-base	KG	Chunk	Tuning	Recursive
Search Engine	Graph	Doc		Adaptive
Synthesized dataset		Entity		Multi-time
Common Crawl		Triplet		
Pre-training Corpus		Sub-graph		
BEIR				
MSMARCO				
Arxiv, Online database				
LLM				

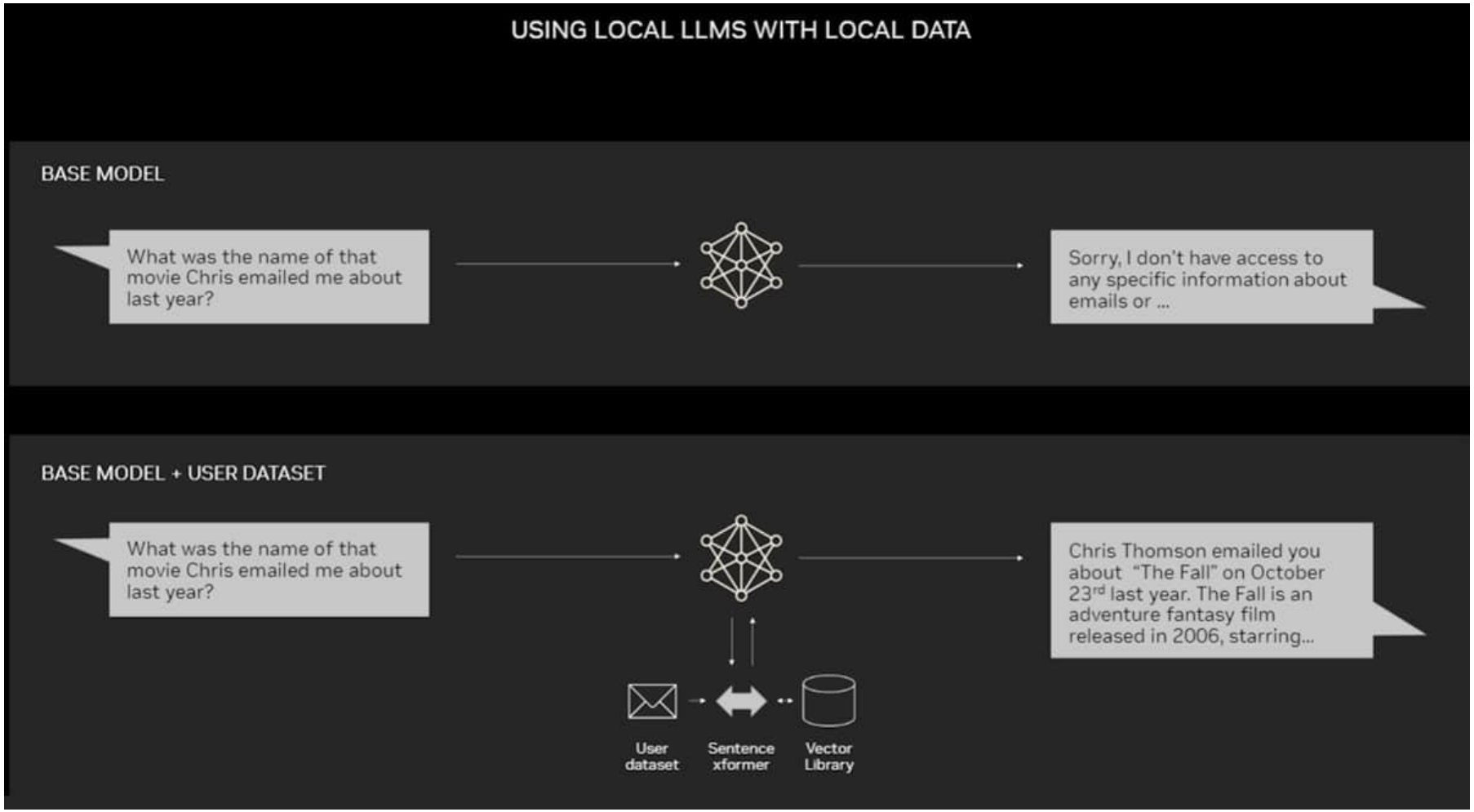
https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/

Why do we need RAG?

- LLMs have profound parameterized knowledge that makes them useful in responding to general prompts.
- However, LLMs are error-prone due to a lack of domain knowledge or outdated information.
- Standalone LLMs do not serve users who want a deeper dive into a current or more specific topic.

https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/

Why do we need RAG?



Retrievers for RAG

- A retriever is aimed at searching relevant documents based on an input query.
- A retriever plays an important role in enhancing the performance of an LLM. Therefore, a good retriever is needed.
- Usually, a retriever produces outputs by computing similarities between query embeddings and document embeddings, which come from other encoder models or the retriever itself.

Embedding Types for Retrieval

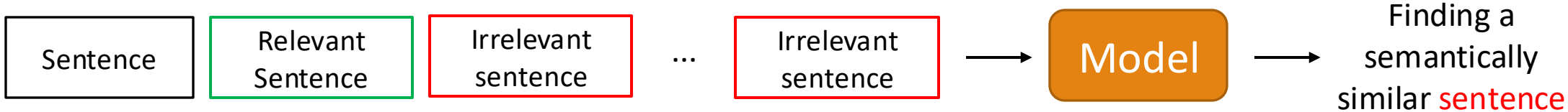
- **Sparse** Embeddings (sparse vectors)
  - E.g., TF-IDF, BM25
- **Dense** Embeddings (dense vectors)
  - E.g., BERT, Sentence-BERT, DPR (Dense Passage Retrieval)

Training Retrievers for RAG

- Task-oriented training:
  - *Retrieval for open-domain question answering (ODQA)*
    - Usually used in RAG because it is more common to search relevant documents.
  - *Sentence embeddings for semantic similarity tasks*

Retrieval for open-domain question answering

Sentence embeddings for semantic similarity tasks



- Both approaches are suitable for retrieval (depends on your task for an LLM).

Sentence embeddings for semantic similarity tasks

Sparse Vectors

- In information retrieval, sparse vectors are vectors with most elements set to zero.
- The advantage of sparse vectors is computational efficiency because most elements are zero and can be ignored.

For an example:
We have a small vocabulary with five words: ["cat", "dog", "fish", "bird", "snake"].
We have a document that only contains the words "cat" and "dog".
The sparse vector for this document would look like this: [1, 1, 0, 0, 0]

The elements represented to "fish", "bird", and "snake" are 0s

Sparse Embeddings: Bag-of-words

```
texts = [
    "This is a book",
    "These are pens and my pen is here"
]
```

- The Bag-of-words approach creates document embeddings.
- The embedding size is equal to the vocabulary size.
- Each value of an embedding is based on frequency counts.

Transform via frequency

Vocabulary size

	a	and	are	book	here	is	my	pen	these	this
sent_0	1	0	0	1	0	1	0	0	0	1
sent_1	0	1	1	0	1	1	1	2	1	0

Since the outputs contain many zeros, this approach is called a sparse embedding method.

Sparse Embeddings: TF-IDF

- TF (Term Frequency)
- IDF (Inverse Document Frequency)

```
texts = [
    "This is a book",
    "These are pens and my pen is here"
]
```

- The **TF-IDF** approach also creates document embeddings.
- The embedding size is equal to the vocabulary size.
- Each value of an embedding is based on **TF x IDF**.

Transform via TF-IDF

Vocabulary size

	a	and	are	book	here	is	my	pen	these	this
sent_0	0.534046	0.000000	0.000000	0.534046	0.000000	0.379978	0.000000	0.000000	0.000000	0.534046
sent_1	0.000000	0.324336	0.324336	0.000000	0.324336	0.230768	0.324336	0.648673	0.324336	0.000000

Since the outputs contain many zeros, this approach is called a sparse embedding method.

https://github.com/tsmatz/nlp-tutorials/blob/master/01_sparse_vector.ipynb

TF-IDF

The mathematical representation of TF-IDF:

$$TF - IDF = TF \times IDF \quad \text{where} \quad TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \qquad IDF_i = lg \frac{|D|}{|\{j : t_i \in d_j\}|}$$

- Where $n_{i,j}$ is the i-th word in j-th text in the dataset.

TF (Term Frequency)

- Represents the "frequency" of a term appearing in a text.

IDF (Inverse Document Frequency)

- Aims for terms to have higher specificity, meaning the fewer texts in the dataset contain the term, the better.

BM25 (an improved version of TF-IDF)

Best **M**atching

$$score = \frac{(k_1 + 1)TF}{TF + k_1 * (1 - b + b * \frac{|D|}{avgD})} * IDF,\ b \in [0,1]$$

k_1 : A term frequency saturation hyper-parameter. For best performance, the value of k_1 should be between 0 and 3.

- This reduces the effect of high-frequency terms so that they don't overpower the score excessively.

b : A document length normalization parameter, this hyper-parameter controls the influence of sequence length.

- This allows shorter and longer documents to compete more equally in retrieval relevance.

k_1=0, b=0 → term frequency is ignored

k_1 increases → the weight of term frequency increases

Robertson, Stephen, and Hugo Zaragoza. "The probabilistic relevance framework: BM25 and beyond." *Foundations and Trends® in Information Retrieval* 3.4 (2009): 333-389.

Dense Vectors

- In NLP, dense vectors comprise compact numerical values representing semantic features of text.
- “Dense” is concept contrary to “sparse.”
- Word2vec is also an approach for creating dense embeddings.

Sparse vector

Dense vector

Is using **CLS** in BERT enough?

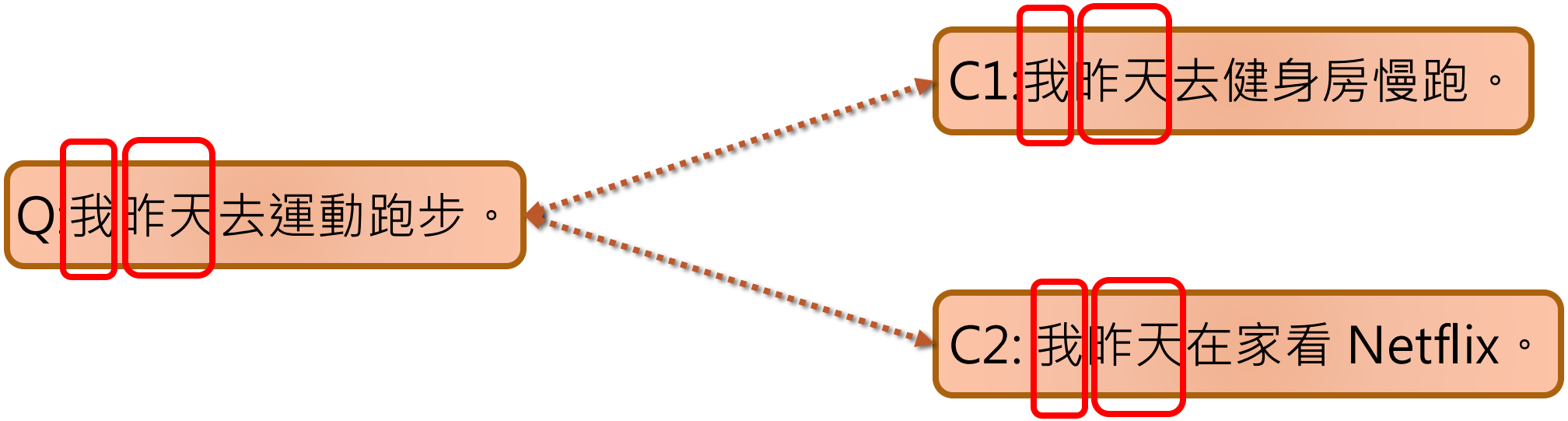
Sim (The man is sitting on the chair. / A person is seated on a chair.) **<** **Sim** (A dog is chasing a ball. / The stock market went up today.)

The pre-training task is NLM, not for sentence semantics.

CLS is not designed for sentence embedding.

CLS is heavily affected by context and positional encoding, making it semantically unstable.

Is using **CLS** in BERT enough?

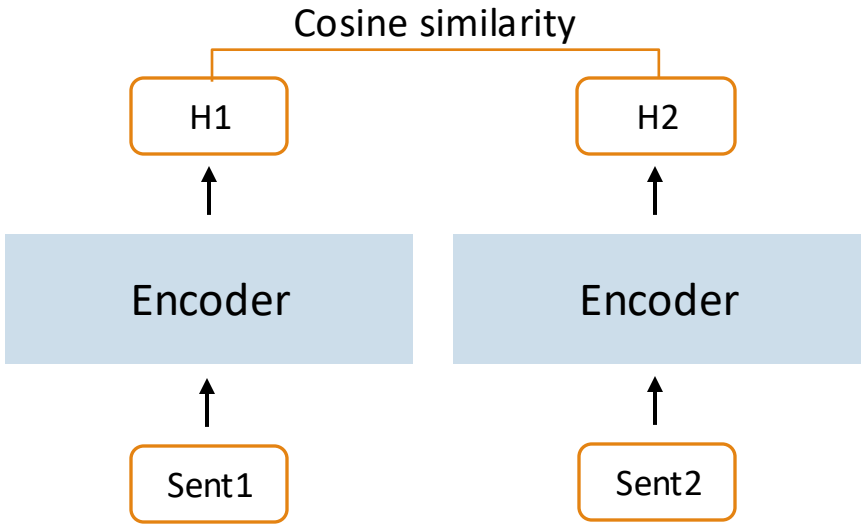


Sim (Q, C1) ≈ Sim (Q, C2)

Approach for Dense Vectors: Dual Encoder

- Also called **bi-encoder**, **Siamese network.**
- Structure:
  - Two identical/similar encoders
  - Processes two inputs independently
  - Outputs separate vectors for each input
- Tasks:

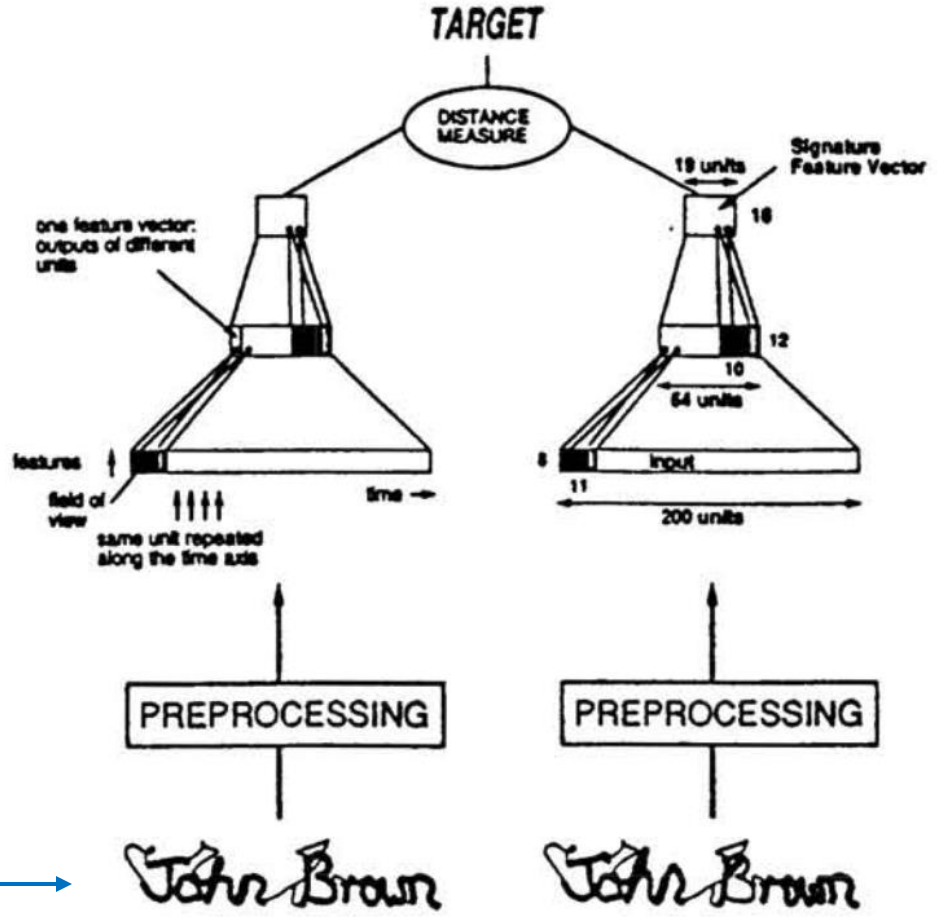
Task	Inputs
Information Retrieval	Document and query
Semantic similarity (or any sentence pair classification)	Two sentences



The First Siamese Network

- Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., & Shah, R. (1993). Signature verification using a" siamese" time delay neural network. NeurIPS.
- Siamese" neural network consists of **two identical** sub-networks joined at their outputs.

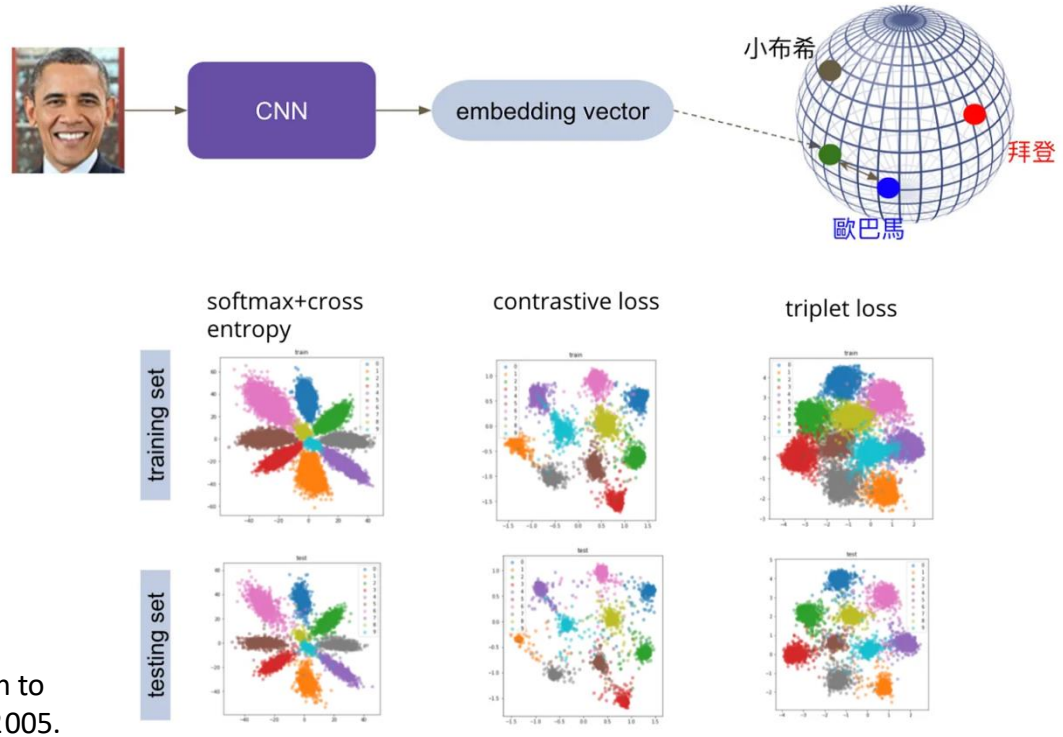
Signature Verification



(Figure source: Bromley et al., 1993)

A Brief History of Siamese Networks

- Signature Verification (Bromley et al., 1993)[1]
- Face Verification (Chopra et al., 2005)[2]
- Image similarity (Koch et al., 2015)[3]
- Text Similarity (Neculoiu et al., 2016)[4]

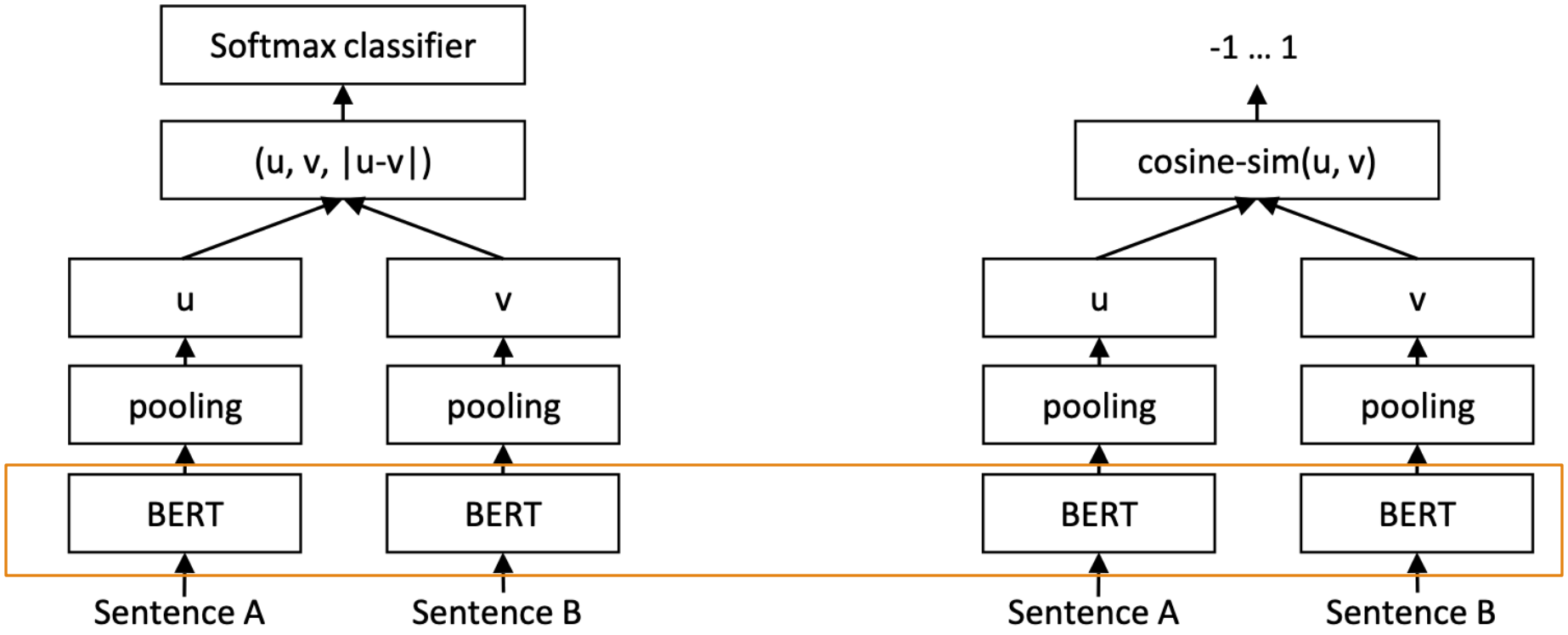


[1] Bromley, Jane, et al. "Signature verification using a" siamese" time delay neural network." NeurIPS 1993.
[2] Chopra, Sumit, Raia Hadsell, and Yann LeCun. "Learning a similarity metric discriminatively, with application to face verification." 2005 IEEE computer society conference on computer vision and pattern recognition. CVPR 2005.
[3] Koch, Gregory, Richard Zemel, and Ruslan Salakhutdinov. "Siamese neural networks for one-shot image recognition." *ICML deep learning workshop*. 2015.
[4] Neculoiu, Paul, Maarten Versteegh, and Mihai Rotaru. "Learning text similarity with siamese recurrent networks." Proceedings of the 1st Workshop on Representation Learning for NLP. 2016.

https://arxiv.org/abs/1908.10084

Sentence-BERT

```
# Pseudo code for Dual Encoder
query_vector = encoder(query)          # [0.1, 0.2, 0.3]
document_vector = encoder(document)    # [0.2, 0.2, 0.4]
similarity = cosine_similarity(query_vector, document_vector)
```



Siamese Network
(weights shared)

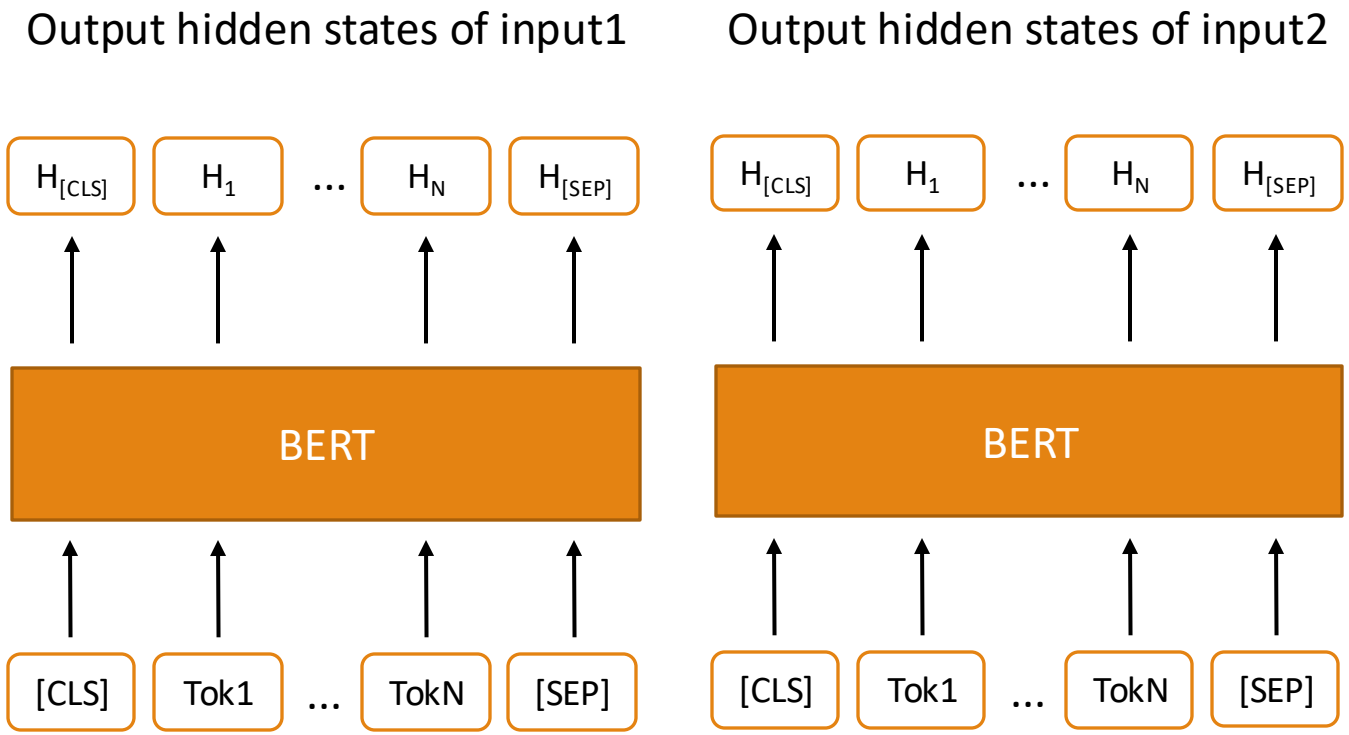
Training time

Inference time

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *(EMNLP-IJCNLP 2019)*

Why does Sentence-BERT need pooling?

- BERT produces embeddings (hidden states from the final layer) for each token.
- We need a single fixed-size vector for the entire sentence.



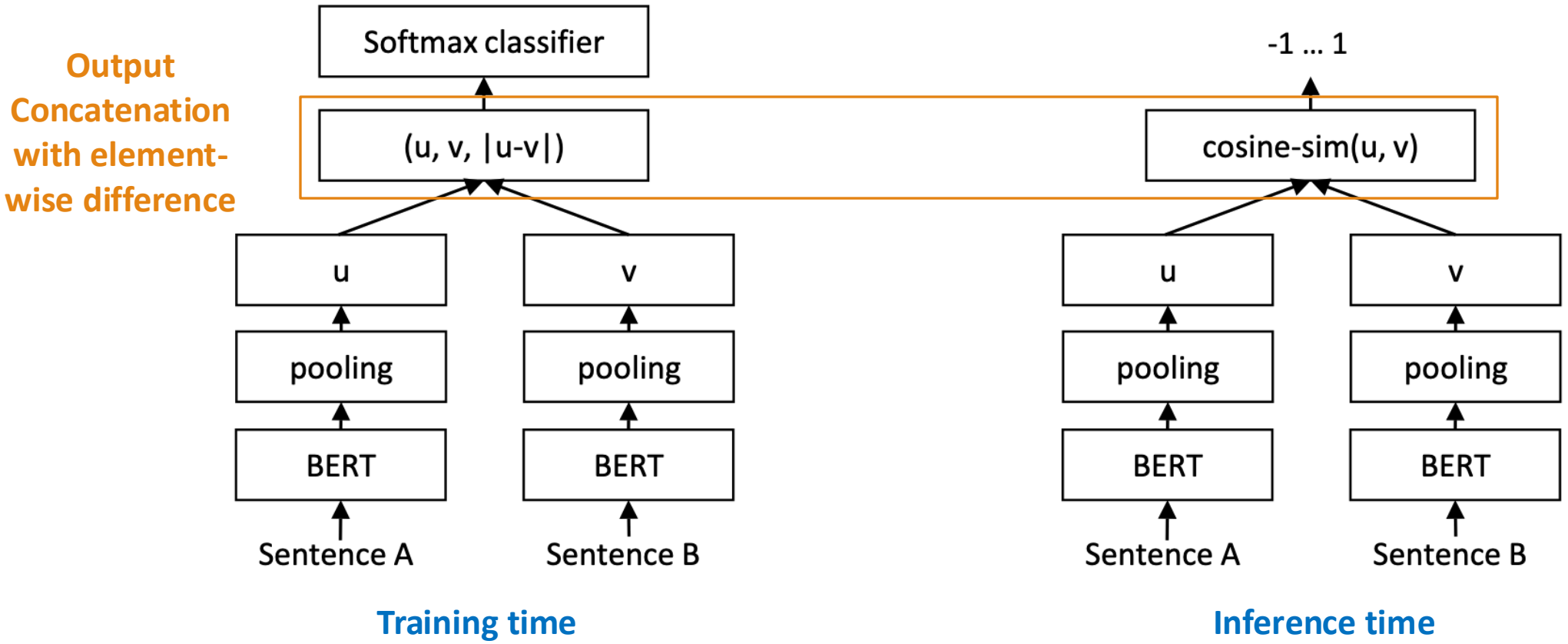
Pooling of Sentence-BERT

We need a single fixed-size vector for the entire sentence.

- **CLS: Use the [CLS] token**
  - This is the default setting in original BERT.
- **MEAN: the mean of all output vectors**
  - Averages all token embeddings.
- **MAX: max-over-time of the output vectors**
  - Takes maximum value across each dimension.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *(EMNLP-IJCNLP 2019)*

Sentence-BERT(Dual encoder)



Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *(EMNLP-IJCNLP 2019)*

Performance comparison of Pooling and Concatenation

- Indeed, [CLS] token can directly be used for representing the entire sentence.
- But pooling may bring better performance.

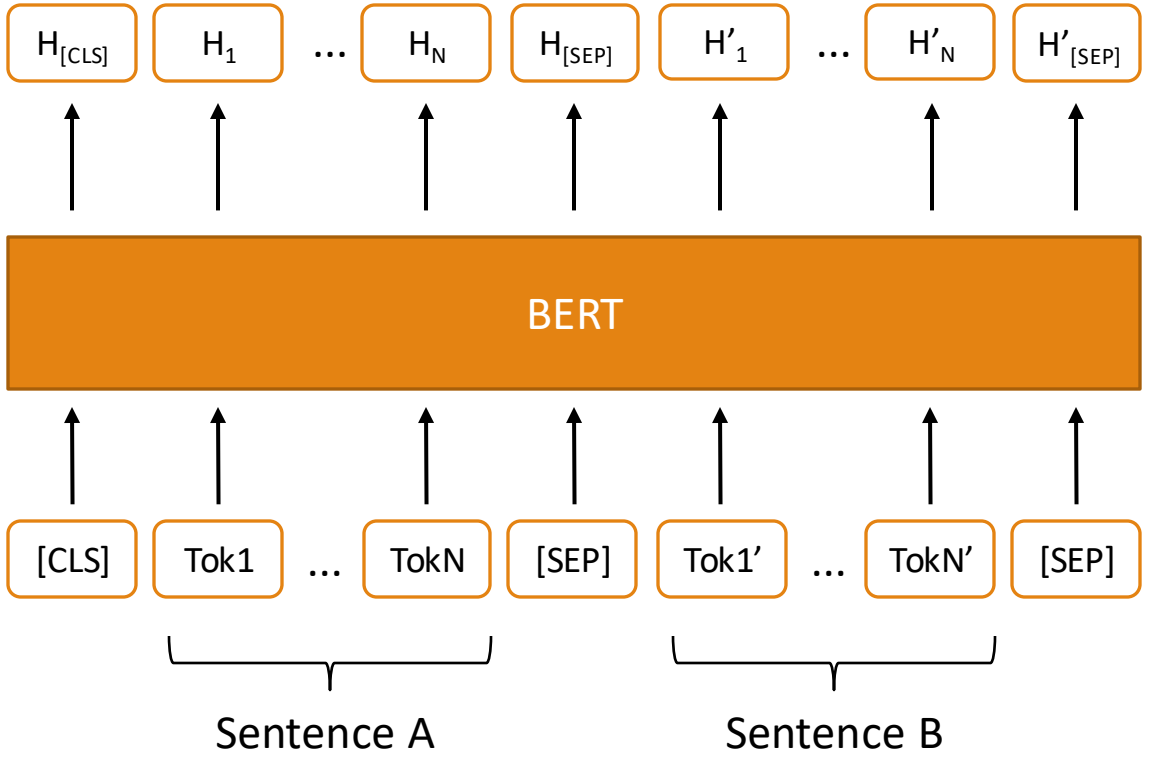
- **Experiment** shows using element-wise difference is better than the other settings.
- Note that the concatenation mode is only used for training.

	NLI	STSb
Pooling Strategy		
MEAN	**80.78**	**87.44**
MAX	79.07	69.92
CLS	79.80	86.62
Concatenation		
(u, v)	66.04	-
$(\|u - v\|)$	69.78	-
$(u * v)$	70.54	-
$(\|u - v\|, u * v)$	78.37	-
$(u, v, u * v)$	77.44	-
$(u, v, \|u - v\|)$	**80.78**	-
$(u, v, \|u - v\|, u * v)$	80.44	-

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *(EMNLP-IJCNLP 2019)*

BERT as a Cross Encoder

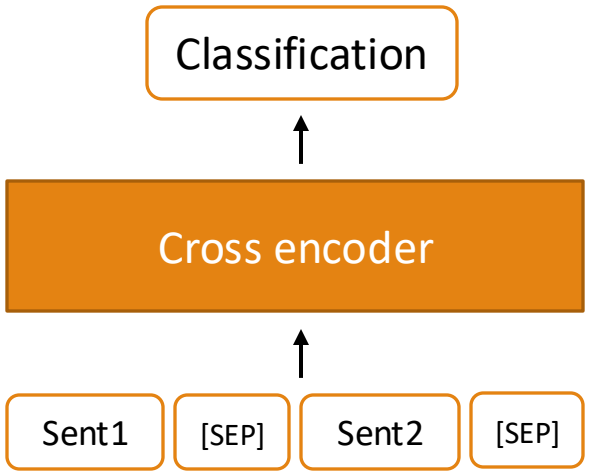
- For a cross encoder, representations of two input sentences are attended with each other.
- The hidden state of [CLS] represents the relationship between the two input sentences.

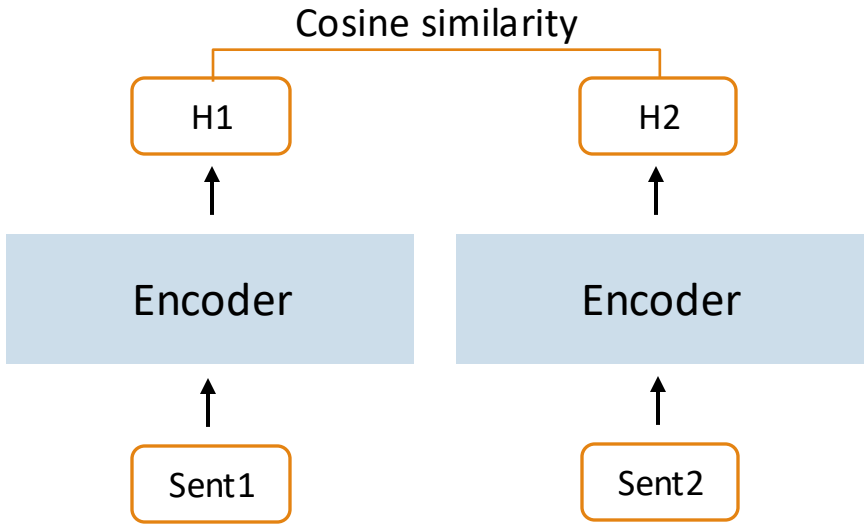


Devlin, J., Chang, M.W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL 2019.

Computation time for Bi-encoders and Cross encoders

- For 10,000 sentence pairs:
  - Cross encoders: $n \cdot (n-1)/2 =$ 49,995,000 inference times
  - Bi-encoders: 10,000 * 2 inference times (can be parallel) with cosine similarity calculation





Performance comparison for Bi-encoders and Cross encoders

Model	Spearman
Not trained for STS	
Avg. GloVe embeddings	58.02
Avg. BERT embeddings	46.35
InferSent - GloVe	68.03
Universal Sentence Encoder	74.92
SBERT-NLI-base	77.03
SBERT-NLI-large	79.23
Trained on STS benchmark dataset	
BERT-STSb-base	84.30 ± 0.76
SBERT-STSb-base	84.67 ± 0.19
SRoBERTa-STSb-base	**84.92** ± 0.34
BERT-STSb-large	**85.64** ± 0.81
SBERT-STSb-large	84.45 ± 0.43
SRoBERTa-STSb-large	85.02 ± 0.76
Trained on NLI data + STS benchmark data	
BERT-NLI-STSb-base	**88.33** ± 0.19
SBERT-NLI-STSb-base	85.35 ± 0.17
SRoBERTa-NLI-STSb-base	84.79 ± 0.38
BERT-NLI-STSb-large	**88.77** ± 0.46
SBERT-NLI-STSb-large	86.10 ± 0.13
SRoBERTa-NLI-STSb-large	86.15 ± 0.35

BERT: Cross encoder

SBERT / SRoBERTa: Bi-encoders

- Generally, the difference in performance between bi-encoders and cross encoders is not large.
- However, bi-encoders are much faster with respect to computation time.
  - If >1M documents in a database, the difference in computation time will be extremely huge.

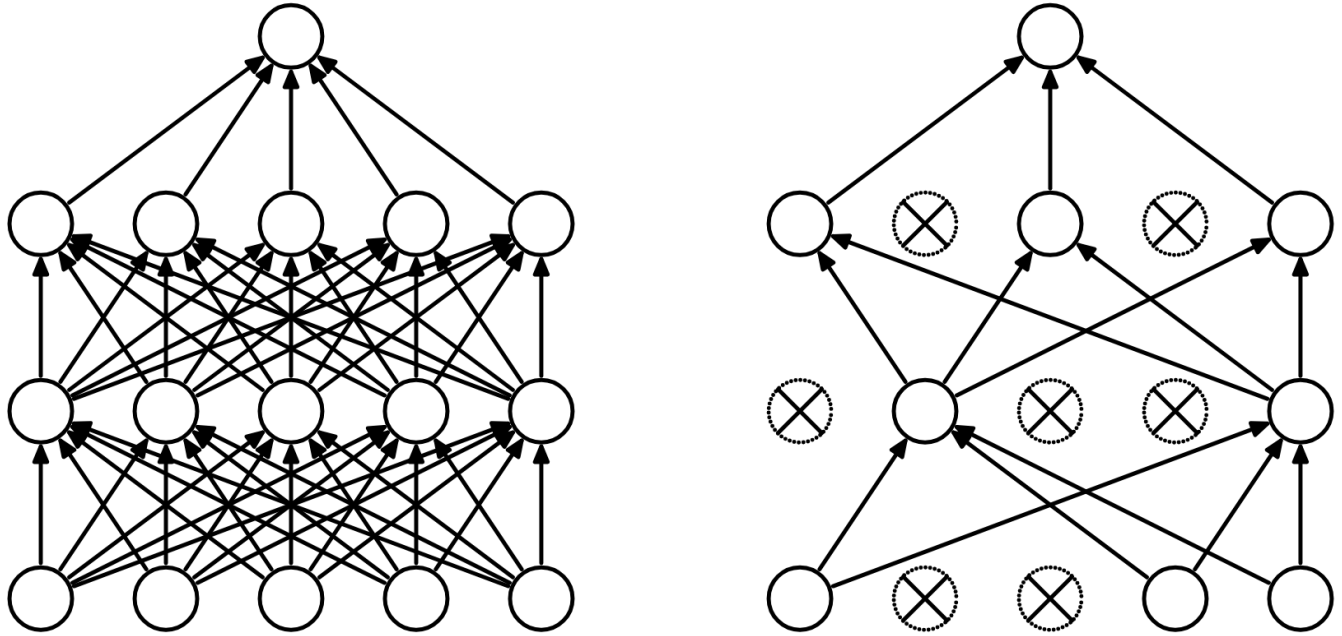
SimCSE (Dual encoder)

- SimCSE: a simple contrastive sentence embedding framework
- Both unsupervised and supervised training approaches were proposed in SimCSE:
  - **Unsupervised** training of SimCSE
    - Relying on **Dropout**
  - **Supervised** training of SimCSE
    - Relying on **labels in a dataset**

Gao, Tianyu, Xingcheng Yao, and Danqi Chen. "SimCSE: Simple Contrastive Learning of Sentence Embeddings." EMNLP 2021.

Dropout

- Dropout randomly drop units (along with their connections) from the neural network during training. This approach usually brings regularization and reduces overfitting.

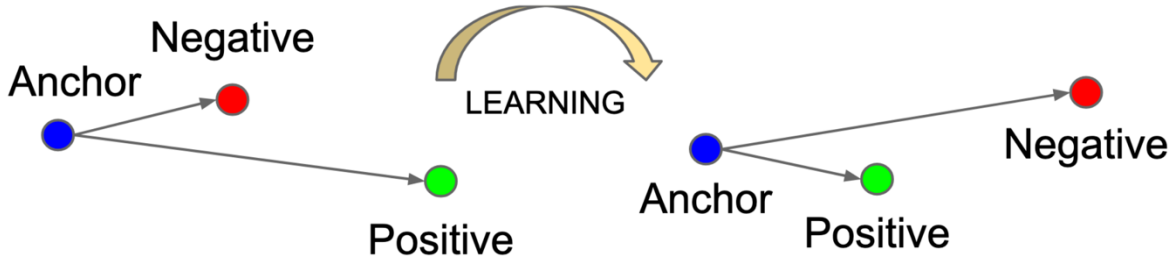


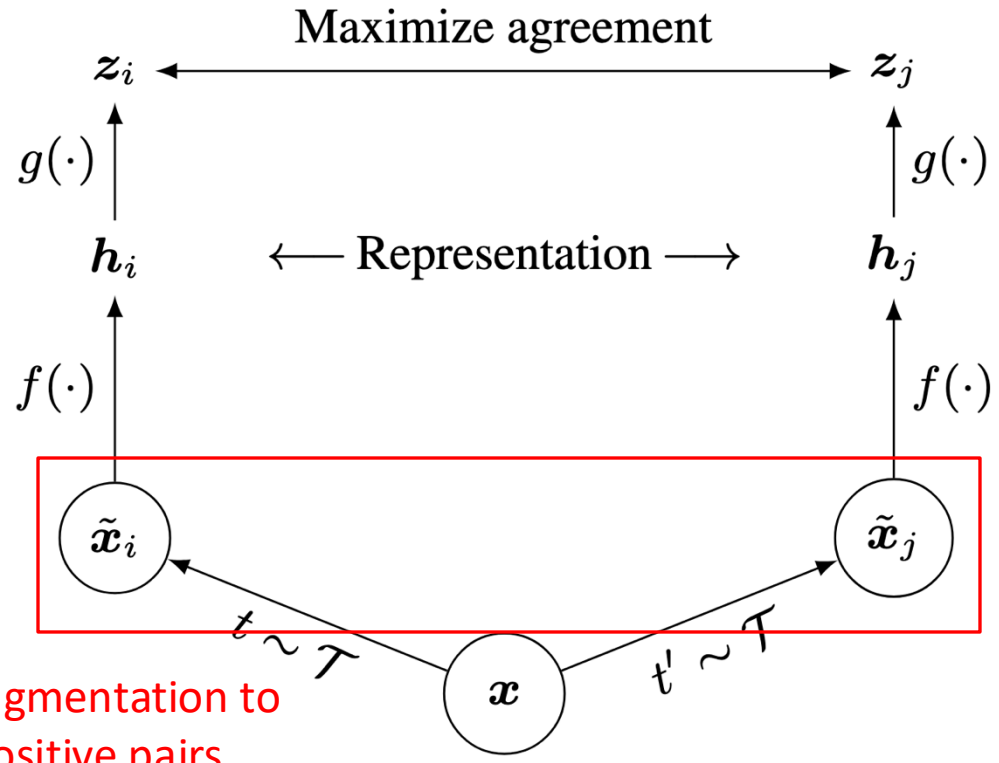
(a) Standard Neural Net (b) After applying dropout.

Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." The journal of machine learning research 15.1 (2014): 1929-1958.

Contrastive Learning

f (·): encoder network
g(·): projection head
z: output logits



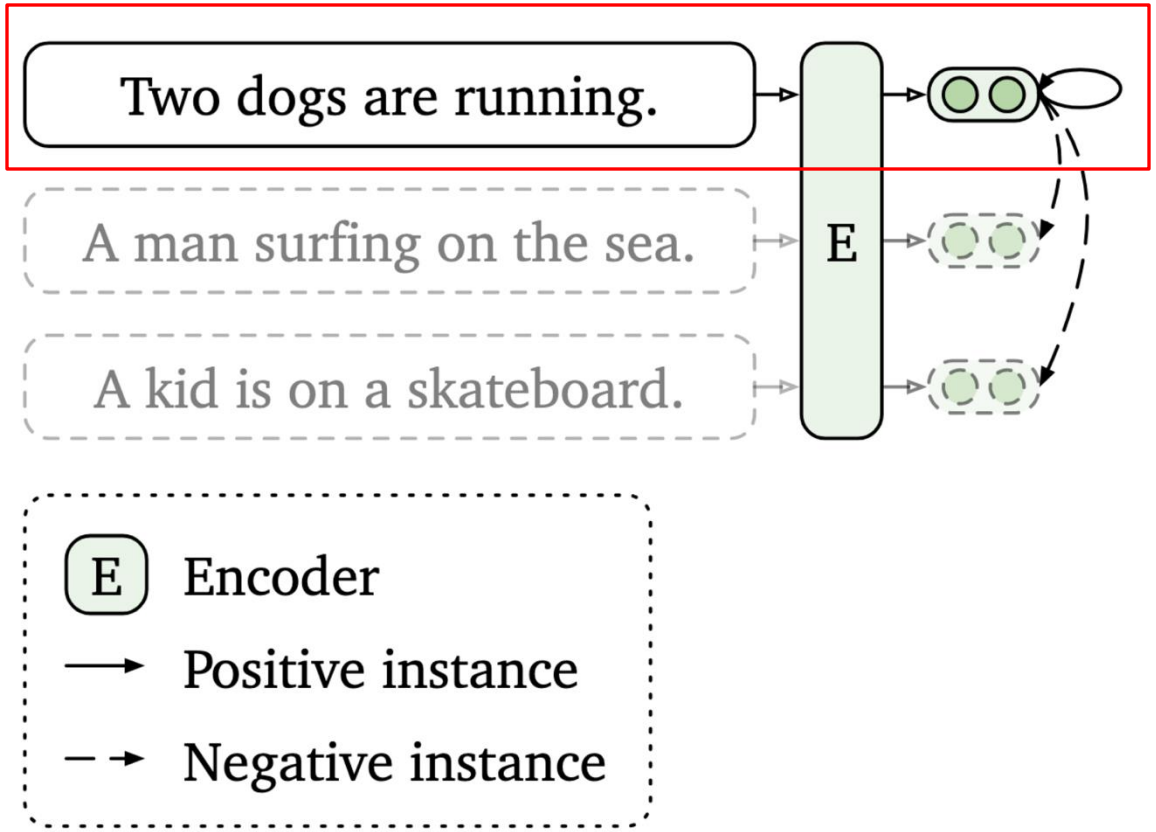


Use data augmentation to create positive pairs

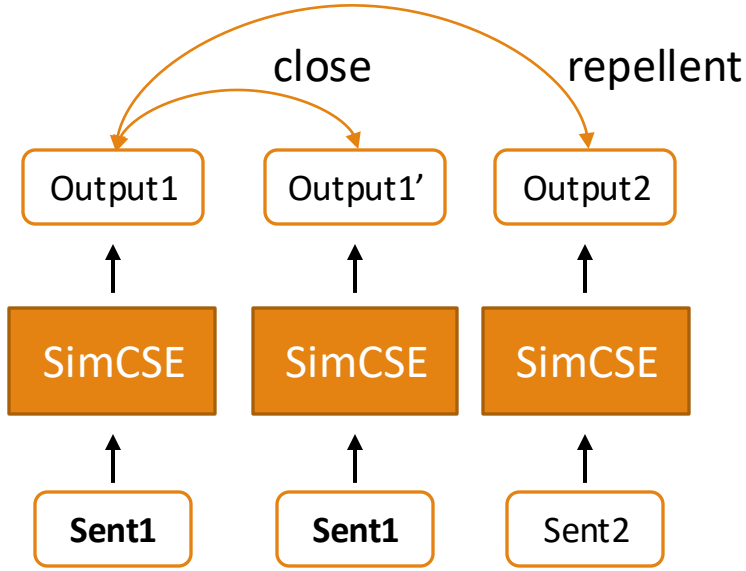
Left Figure source: Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." CVPR 2015.

Right Figure source: Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICLR 2020.

Unsupervised training of SimCSE



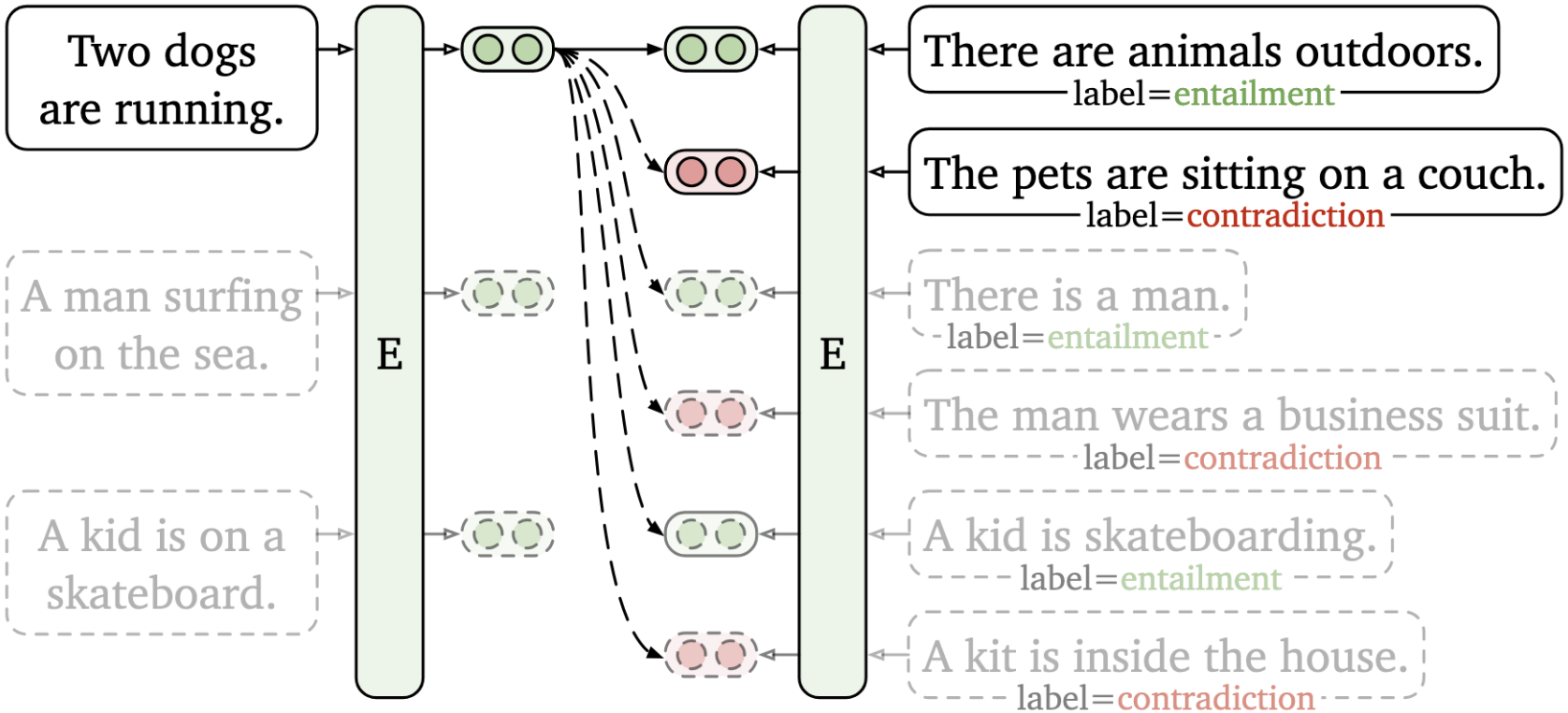
Input a sentence twice with a dropout mask
(Dropout as data augmentation)



Gao, Tianyu, Xingcheng Yao, and Danqi Chen. "SimCSE: Simple Contrastive Learning of Sentence Embeddings." EMNLP 2021.

Supervised training of SimCSE

- Supervised training of SimCSE relies on labels in a dataset to define positives and negatives.



Gao, Tianyu, Xingcheng Yao, and Danqi Chen. "SimCSE: Simple Contrastive Learning of Sentence Embeddings." EMNLP 2021.

SimCSE outperforms Sentence-BERT

- SimCSE outperforms Sentence-BERT on semantic similarity tasks.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Supervised models								
SRoBERTa$_{\texttt{base}}$ ♣	71.54	72.49	70.80	78.74	73.69	77.77	74.46	74.21
SRoBERTa$_{\texttt{base}}$-whitening	70.46	77.07	74.46	81.64	76.43	79.49	76.65	76.60
∗ SimCSE-RoBERTa$_{\texttt{base}}$	**76.53**	**85.21**	**80.95**	**86.03**	**82.57**	**85.83**	**80.50**	**82.52**
∗ SimCSE-RoBERTa$_{\texttt{large}}$	**77.46**	**87.27**	**82.36**	**86.66**	**83.93**	**86.70**	**81.95**	**83.76**

Retrieval for open-domain question answering (ODQA)

Open-domain question answering (ODQA)

- Given a question x such as "What is the currency of the UK?", a model must output the correct answer string y, "pound".
- The "open" part of ODQA refers to the fact that the model does not receive a pre-identified document that is known to contain the answer.
- ODQA is like Reading comprehension (RC) tasks, such as SQuAD, but no relevant articles provided.

(Example of SQuAD)

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?
gravity

Guu, Kelvin, et al. "Retrieval augmented language model pre-training." ICML 2020.
Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. EMNLP 2016.

https://aclanthology.org/2020.emnlp-main.550/

Dense Passage Retrieval (DPR)

- Dense retrieval focuses on **semantic similarity**
- Passages and questions are embedded into dense vectors
- Dense vectors enable better matching for related words or phrases

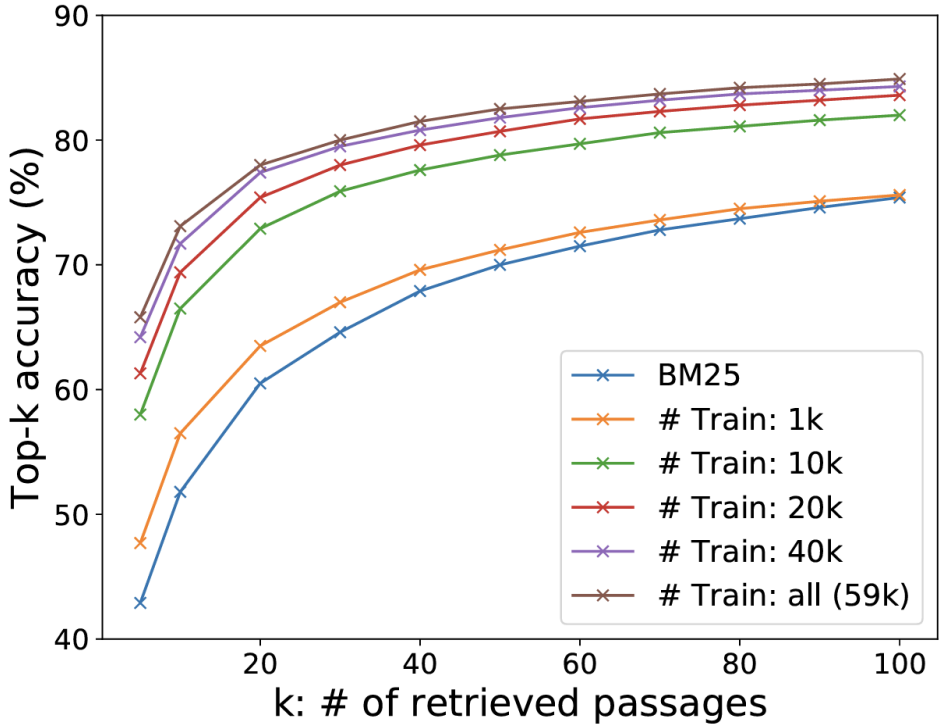
  (e.g., “the body of water” matched with “sea”)

Question	Passage received by BM25	Passage retrieved by DPR
What is the body of water between England and Ireland?	Title:British Cycling . . . **England** is not recognised as a region by the UCI, and there is no English cycling team outside the Commonwealth Games. For those occasions, British Cycling selects and supports the **England** team. Cycling is represented on the Isle of Man by the Isle of Man Cycling Association. Cycling in Northern **Ireland** is organised under Cycling Ulster, part of the all-Ireland governing **body** Cycling **Ireland**. Until 2006, a rival governing **body** existed, . . .	Title: Irish Sea . . . Annual traffic between Great Britain and **Ireland** amounts to over 12 million passengers and of traded goods. **The Irish Sea** is connected to the North Atlantic at both its northern and southern ends. To the north, the connection is through the North Channel between Scotland and Northern **Ireland** and the Malin Sea. The southern end is linked to the Atlantic through the St George’s Channel between Ireland and Pembrokeshire, and the Celtic Sea. . . .

Karpukhin et al., 2020. Dense Passage Retrieval for Open-Domain Question Answering, EMNLP 2020.

Dense Passage Retrieval (DPR)

- Outperforms **BM25** using only 1000 training data!



Dense Passage Retrieval (DPR)

- After training, two **BERT-based encoders** can **independently** encode question (**q**) and passage (**p**) into dense vectors.

- **Similarity** between question and passage = **dot product** between their embeddings

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

q : question text
p : passage text
E_Q : BERT model that outputs question representation
E_p : BERT model that outputs passage representation

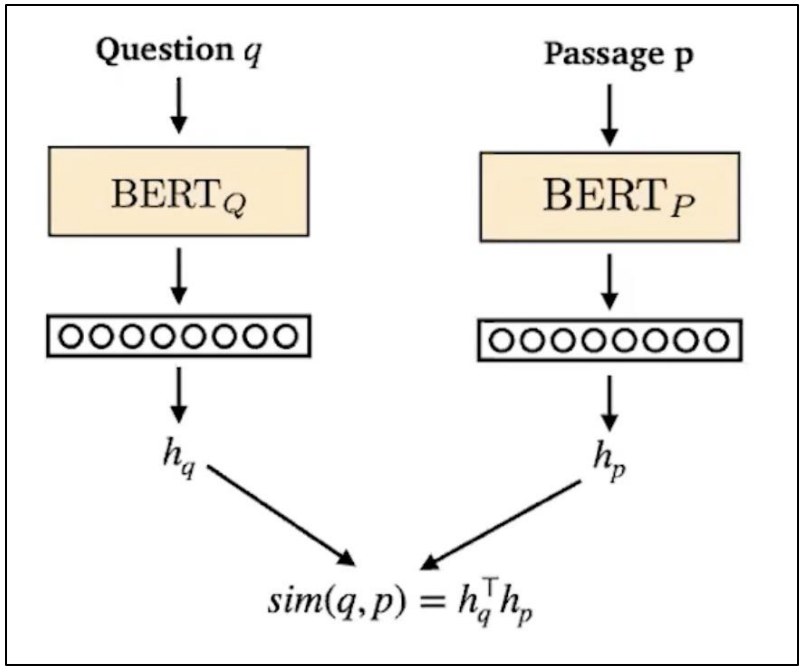


Image source: Stanford CS224N Lecture 12 - Question Answering

Karpukhin et al., 2020. Dense Passage Retrieval for Open-Domain Question Answering

Dense Passage Retrieval (DPR)

Training the encoders

- Goal: **Relevant** pairs of questions and passages will have **smaller distance** than the irrelevant ones

- Training data

$$\mathcal{D} = \{\langle q_i, p_i^+, p_{i,1}^-, \cdots, p_{i,n}^- \rangle\}_{i=1}^{m}$$

m training instances

Question

Relevant Passage

n Irrelevant Passages

Karpukhin et al., 2020. Dense Passage Retrieval for Open-Domain Question Answering

Dense Passage Retrieval (DPR)

Training the encoders

- Base model: bert-base-uncased

- Loss function: Negative log-likelihood of the positive passage

$$L(q_i, p_i^+, p_{i,1}^-, \cdots, p_{i,n}^-)$$

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

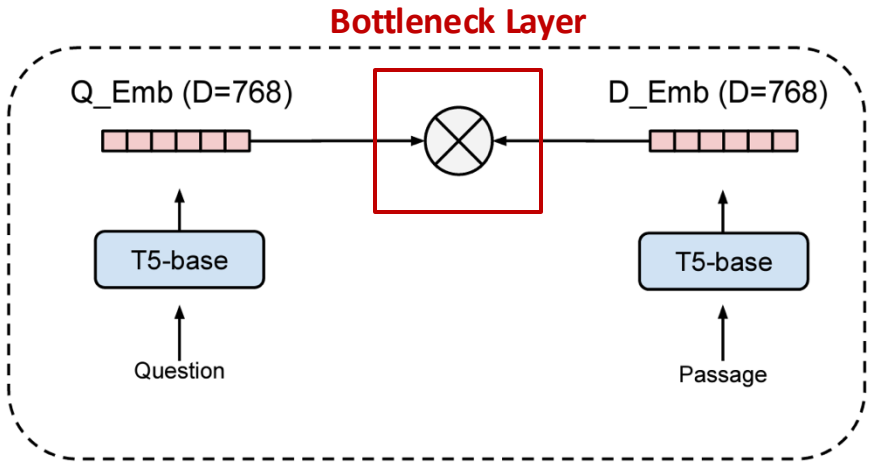
  - **Maximize** the similarity between $\mathbf{q_i}$ and $\mathbf{p_i^+}$
  - **Minimize** the similarity between non-relevant pairs ($\mathbf{q_i}$ and $\mathbf{p_{i,j}^-}$)

Karpukhin et al., 2020. Dense Passage Retrieval for Open-Domain Question Answering

https://aclanthology.org/2022.emnlp-main.669/

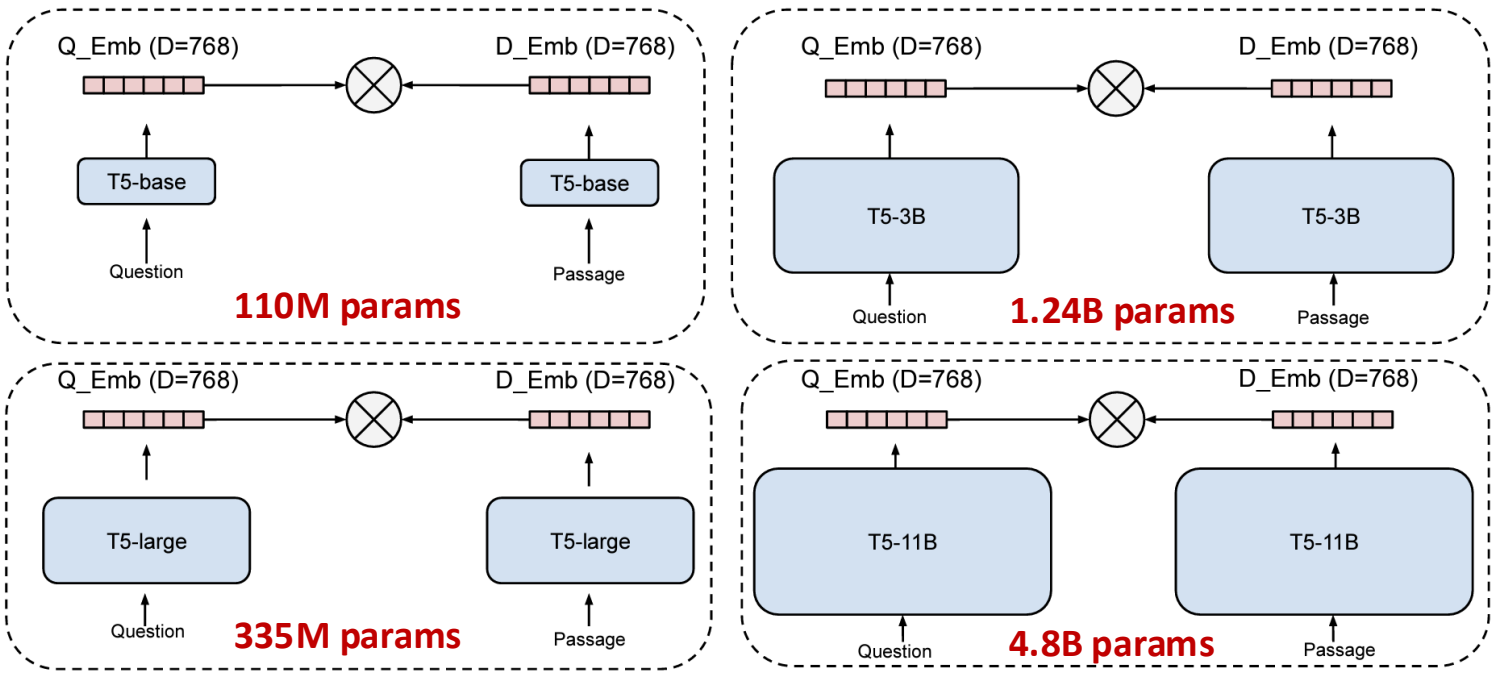
Limitations of Dual Encoders

- Often fail to generalize to other domains for retrieval tasks
- **Bottleneck layer** of dual encoders (simple dot-product or cosine similarity) *might not be powerful enough* to **capture semantic relevance**?



Ni et al., 2022. Large Dual Encoders Are Generalizable Retrievers, EMNLP 2022.

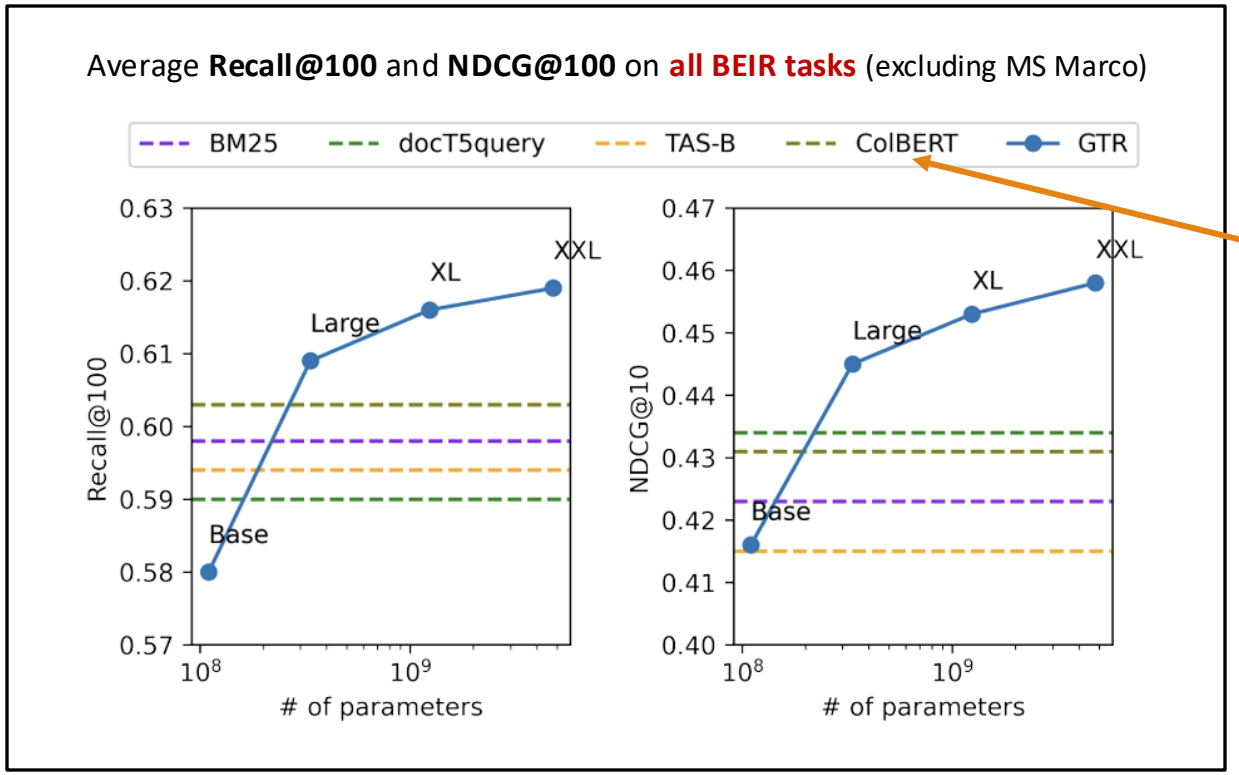
Generalizable T5-based Retriever (GTR)



Can **scaling up** dual encoder model size *improve the retrieval performance*, while keeping the bottleneck layers **fixed**?

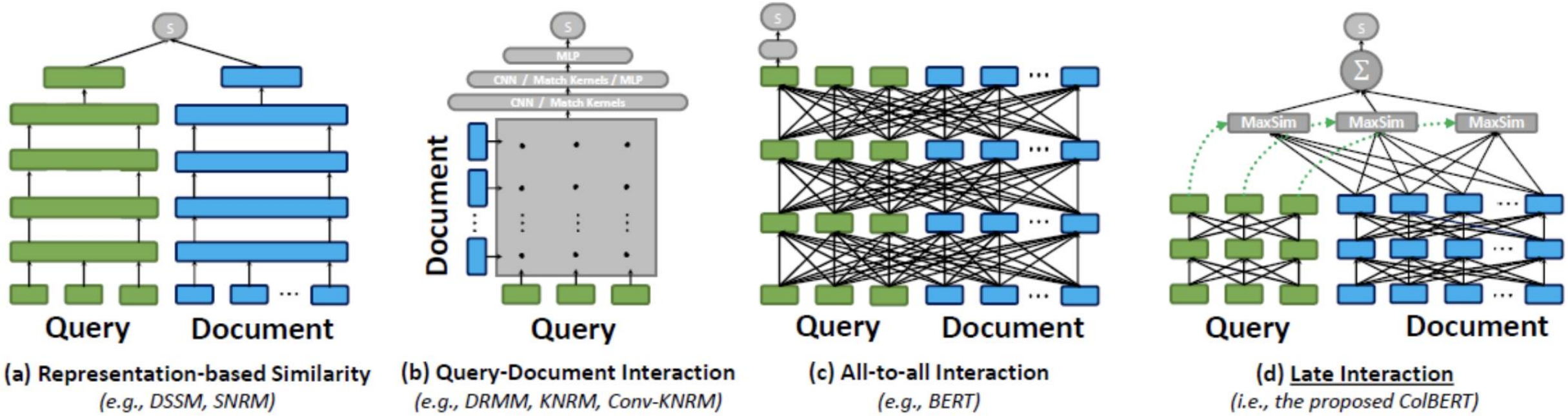
Generalizable T5-based Retriever (GTR)

- Scaling up *consistently improves* dual encoders' **out-of-domain** performance.



ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT, SIGIR 2020

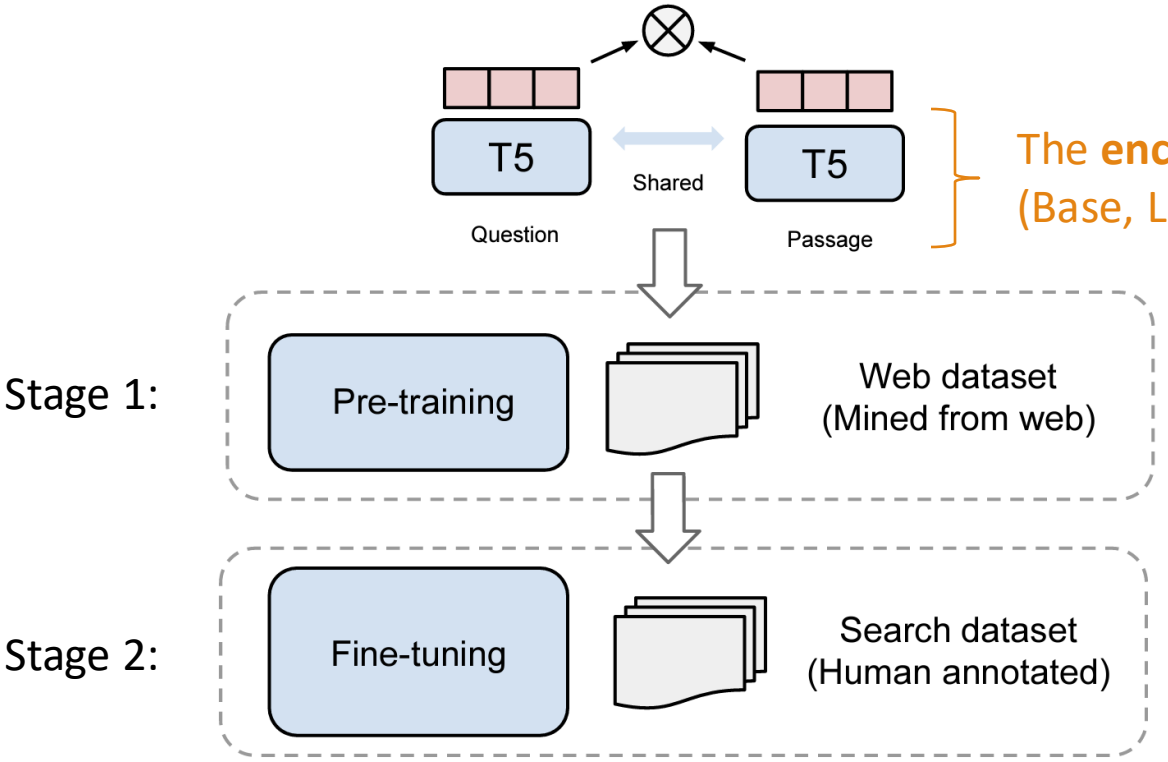
Interactions of Query and Document



ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT, SIGIR 2020

Generalizable T5-based Retriever (GTR)

Multi-stage training for GTR



The **encoder** part from pretrained T5 models (Base, Large, XL, XXL)

2 billion **QA pairs** from:
- Reddit
- Stack-Overflow

Dataset:
- MS Marco (~10%, reach strong out-domain performance)
- NaturalQuestions

https://arxiv.org/pdf/1611.09268

MS MARCO: A Human Generated MAchine Reading COmprehension Dataset

- Machine Reading Comprehension (MRC) dataset
  - by Microsoft 2016, which was adapted in 2018 for retrieval.
- 8.8 M passages from the Web to 1M real-world queries.
- Each query is associated with sparse relevance judgments of one (or very few) documents marked as relevant and no documents explicitly indicated as irrelevant.
- three different tasks
  - (i) predict if a question is answerable given a set of context passages, and extract and synthesize the answer as a human would
  - (ii) generate a well-formed answer (if possible) based on the context passages that can be understood with the question and passage context
  - (iii) rank a set of retrieved passages given a question.

Generalizable T5-based Retriever (GTR)

- **Data efficiency**: Only needs **10%** of MS Marco *supervised data* to achieve the best **out-of-domain** performance!

*GTR **w/o** Pre-training *GTR w/ Pre-training + Fine-tuning

	GTR-FT		GTR		
Ratio of data	Large	XL	Large	XL	XXL
	NDCG@10 on MS Marco *in-domain				
10%	0.402	0.397	0.428	0.426	-
100%	0.415	0.418	0.430	0.439	0.430
	Zero-shot average NDCG@10 w/o MS Marco *out-of-domain				
10%	**0.413**	0.418	**0.452**	**0.462**	**0.465**
100%	0.412	**0.433**	0.445	0.453	0.458

Summary Parameter Sharing

	BERT (Devlin et al., NAACL 2019)	Sentence-BERT (Reimers et al., EMNLP 2019)	SimCSE (Gao et al., EMNLP 2021)	DPR (Karpukhin et al., EMNLP 2020)	GTR (Ni et al., EMNLP 2022)
Encoder Type	Cross	Dual	Dual	Dual	Dual
Weight Sharing	Nan	Yes	Yes	No (Separate Query Encoder and Passage Encoder)	Yes

https://arxiv.org/pdf/2104.08663

BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models

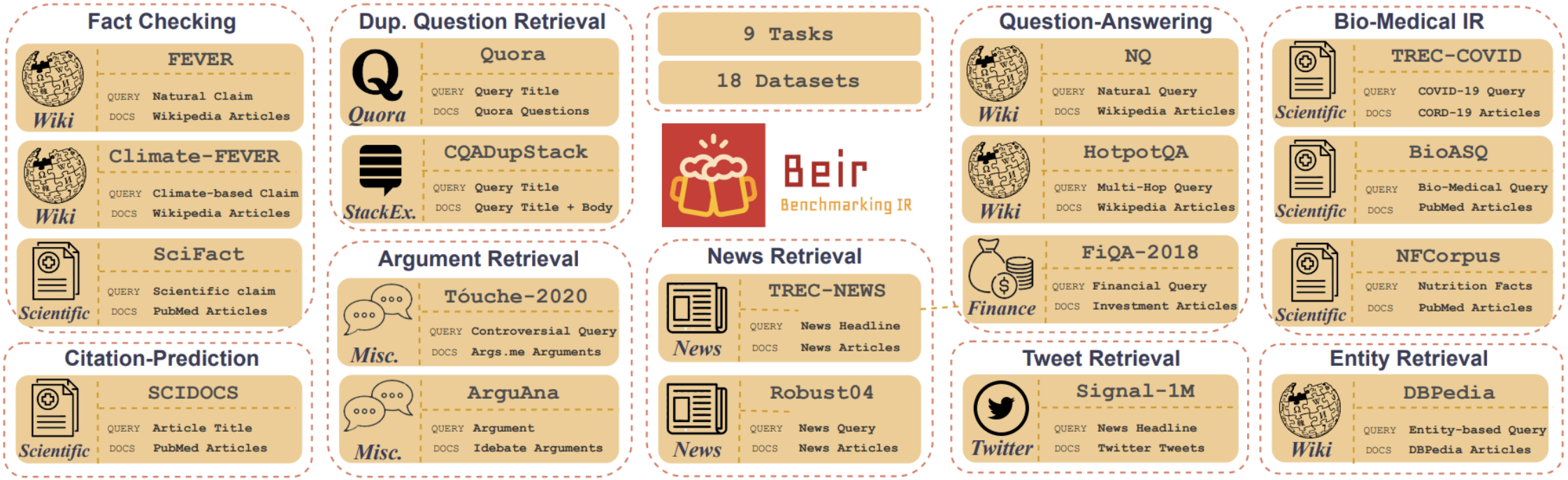


Figure 1: An overview of the diverse tasks and datasets in BEIR benchmark.

https://arxiv.org/pdf/2104.08663

BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models

Model (→)	Lexical	Sparse			Dense				Late-Interaction	Re-ranking
Dataset (↓)	**BM25**	**DeepCT**	**SPARTA**	**docT5query**	**DPR**	**ANCE**	**TAS-B**	**GenQ**	**ColBERT**	**BM25+CE**
MS MARCO	0.228	0.296‡	0.351‡	0.338‡	0.177	0.388‡	0.408‡	0.408‡	0.401‡	**0.413**‡
TREC-COVID	0.656	0.406	0.538	0.713	0.332	0.654	0.481	0.619	0.677	**0.757**
BioASQ	0.465	0.407	0.351	0.431	0.127	0.306	0.383	0.398	0.474	**0.523**
NFCorpus	0.325	0.283	0.301	0.328	0.189	0.237	0.319	0.319	0.305	**0.350**
NQ	0.329	0.188	0.398	0.399	0.474‡	0.446	0.463	0.358	0.524	**0.533**
HotpotQA	0.603	0.503	0.492	0.580	0.391	0.456	0.584	0.534	0.593	**0.707**
FiQA-2018	0.236	0.191	0.198	0.291	0.112	0.295	0.300	0.308	0.317	**0.347**
Signal-1M (RT)	0.330	0.269	0.252	0.307	0.155	0.249	0.289	0.281	0.274	**0.338**
TREC-NEWS	0.398	0.220	0.258	0.420	0.161	0.382	0.377	0.396	0.393	**0.431**
Robust04	0.408	0.287	0.276	0.437	0.252	0.392	0.427	0.362	0.391	**0.475**
ArguAna	0.315	0.309	0.279	0.349	0.175	0.415	0.429	**0.493**	0.233	0.311
Touché-2020	**0.367**	0.156	0.175	0.347	0.131	0.240	0.162	0.182	0.202	0.271
CQADupStack	0.299	0.268	0.257	0.325	0.153	0.296	0.314	0.347	0.350	**0.370**
Quora	0.789	0.691	0.630	0.802	0.248	0.852	0.835	0.830	**0.854**	0.825
DBPedia	0.313	0.177	0.314	0.331	0.263	0.281	0.384	0.328	0.392	**0.409**
SCIDOCS	0.158	0.124	0.126	0.162	0.077	0.122	0.149	0.143	0.145	**0.166**
FEVER	0.753	0.353	0.596	0.714	0.562	0.669	0.700	0.669	0.771	**0.819**
Climate-FEVER	0.213	0.066	0.082	0.201	0.148	0.198	0.228	0.175	0.184	**0.253**
SciFact	0.665	0.630	0.582	0.675	0.318	0.507	0.643	0.644	0.671	**0.688**
Avg. Performance vs. BM25		**- 27.9%**	**- 20.3%**	**+ 1.6%**	**- 47.7%**	**- 7.4%**	**- 2.8%**	**- 3.6%**	**+ 2.5%**	**+ 11%**

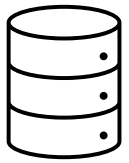
From Retrievers to QA

From now we focus on the full QA pipeline.

Input Query

Which year was Oppenheimer born?

Retriever
(e.g., BERT)



Knowledge Base
(e.g., Wikipedia)

Robert Oppenheimer (born April 22, 1904, New York, New York, U.S.—died February 18, 1967, Princeton, New Jersey) was an American theoretical physicist and science administrator ...

Retrieved Documents

Generator
(e.g., LLM)

LLM Output

1967 (Without RAG)
1904 (With RAG)

(Also called "Reader", because of QA is a reading comprehension task)

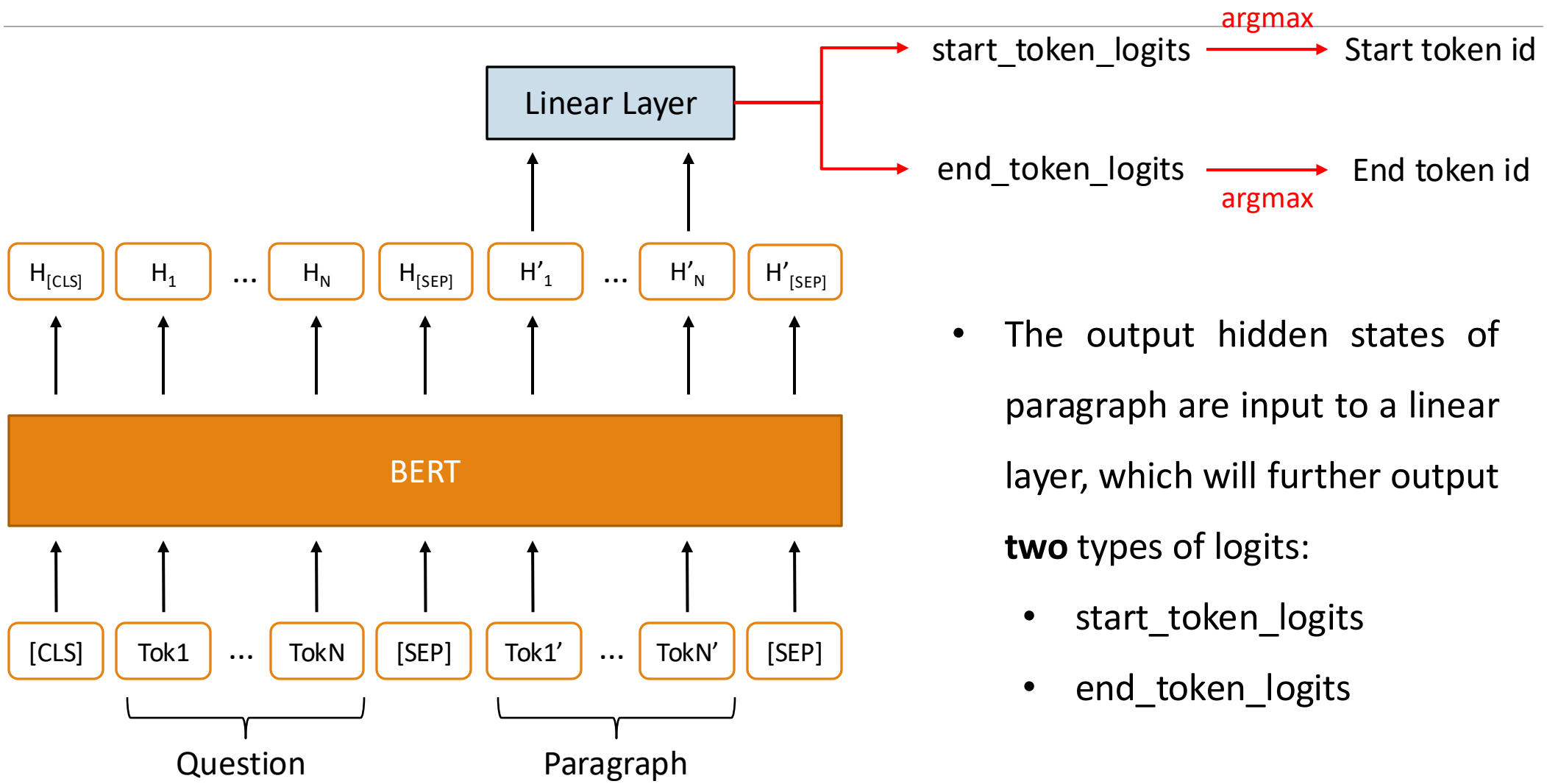
Gao, Yunfan, et al. "Retrieval-augmented generation for large language models: A survey." *arXiv preprint arXiv:2312.10997* (2023).

Open-domain QA with retrievers

- The QA model is an encoder (e.g., BERT).
  - ORQA
  - REALM
- The QA model is a generator (e.g., BART).

QA model is an encoder (e.g., BERT)

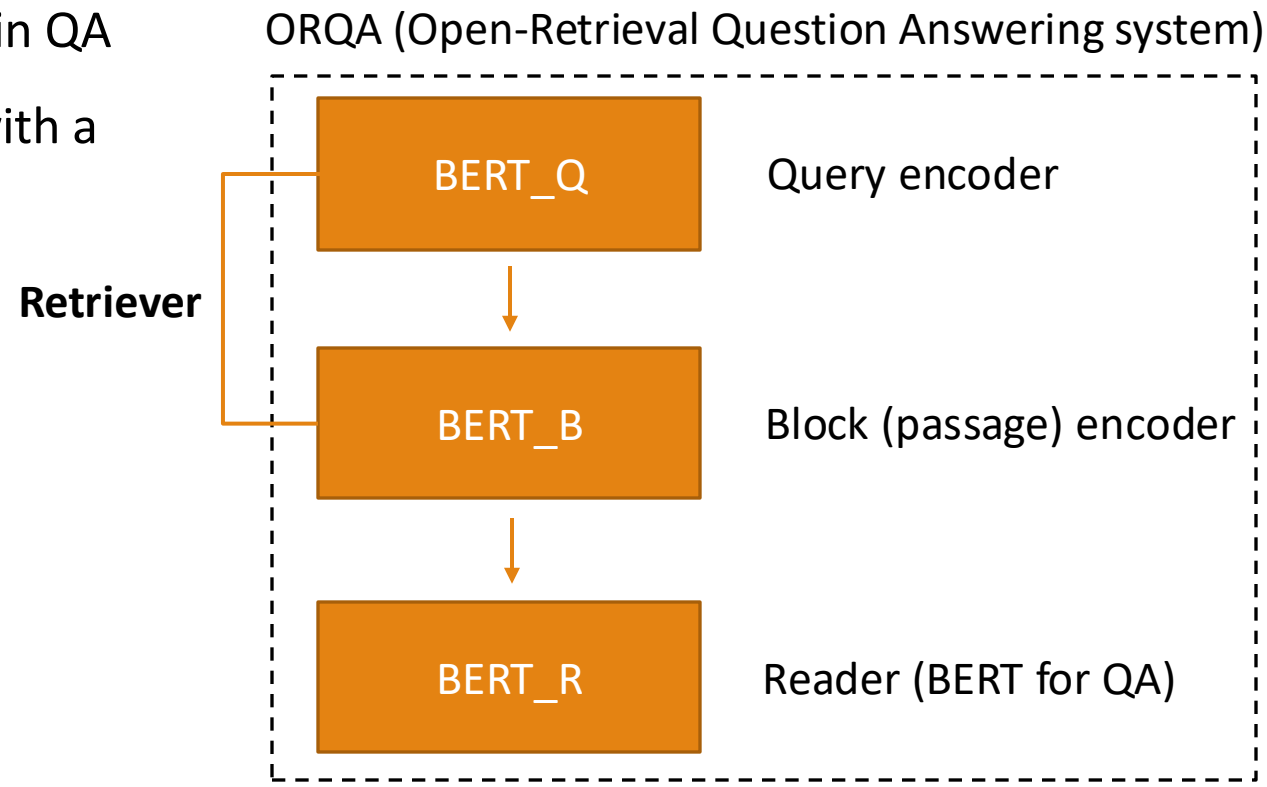
BERT for QA (Reading Comprehension)



- The output hidden states of paragraph are input to a linear layer, which will further output **two** types of logits:
  - start_token_logits
  - end_token_logits

ORQA (Open-Retrieval Question Answering system)

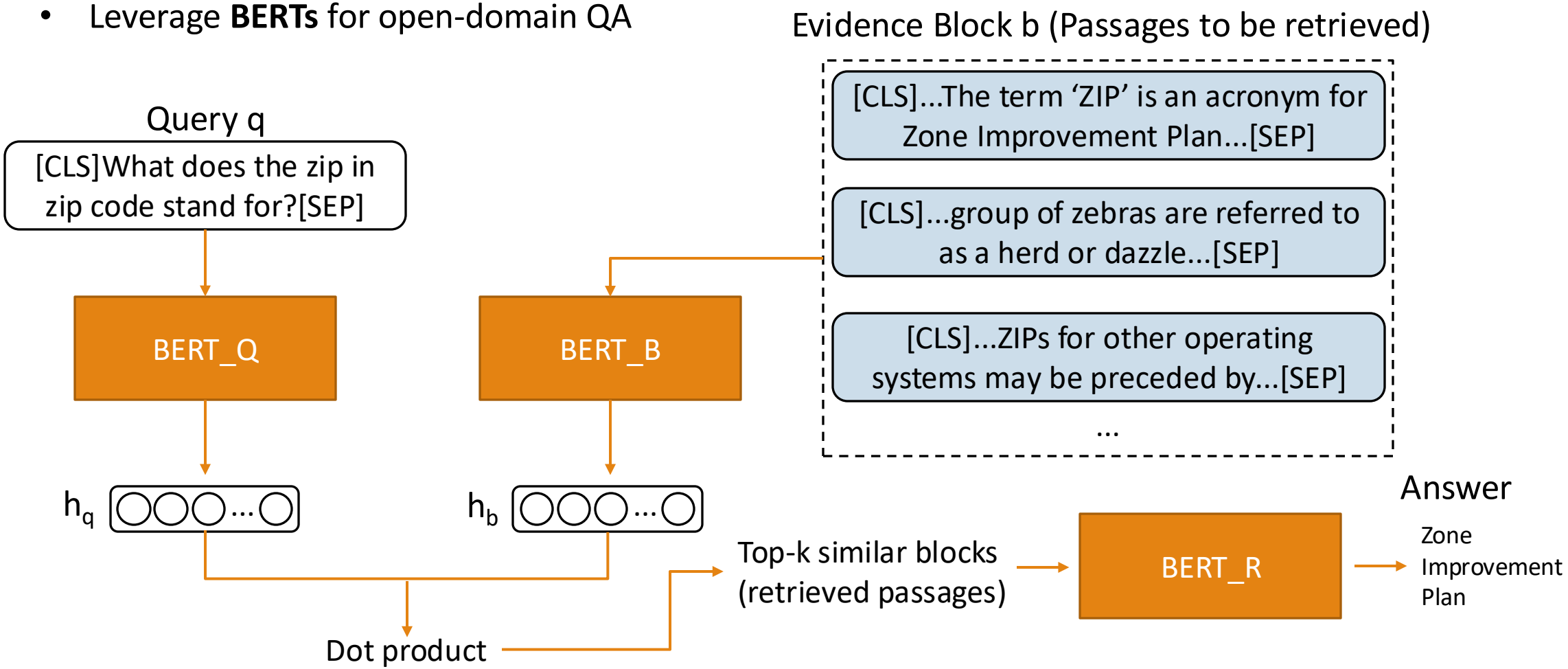
- Leverage **BERTs** for open-domain QA
- Extend BERT for QA to **ODQA** with a retriever model



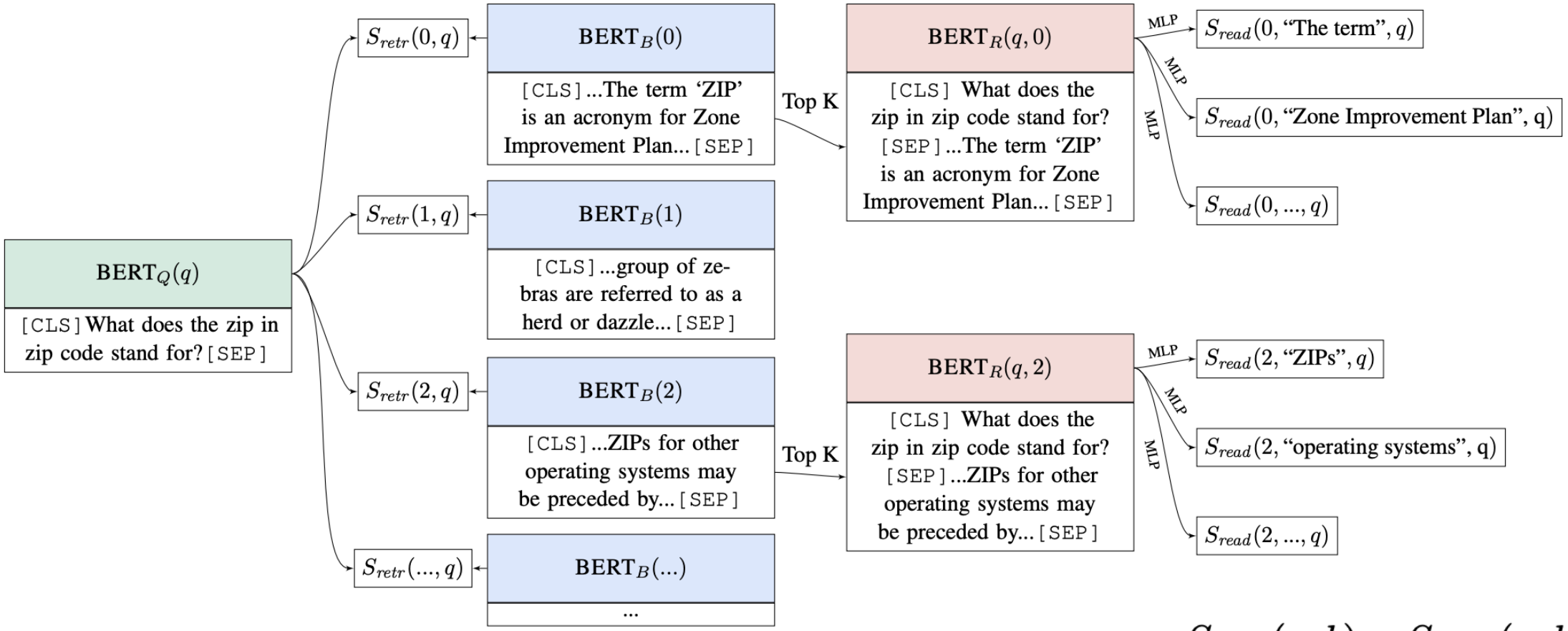
Lee, Kenton, Ming-Wei Chang, and Kristina Toutanova. "Latent Retrieval for Weakly Supervised Open Domain Question Answering." ACL 2019.

ORQA (Open-Retrieval Question Answering system)

- Leverage **BERTs** for open-domain QA



ORQA (Open-Retrieval Question Answering system)



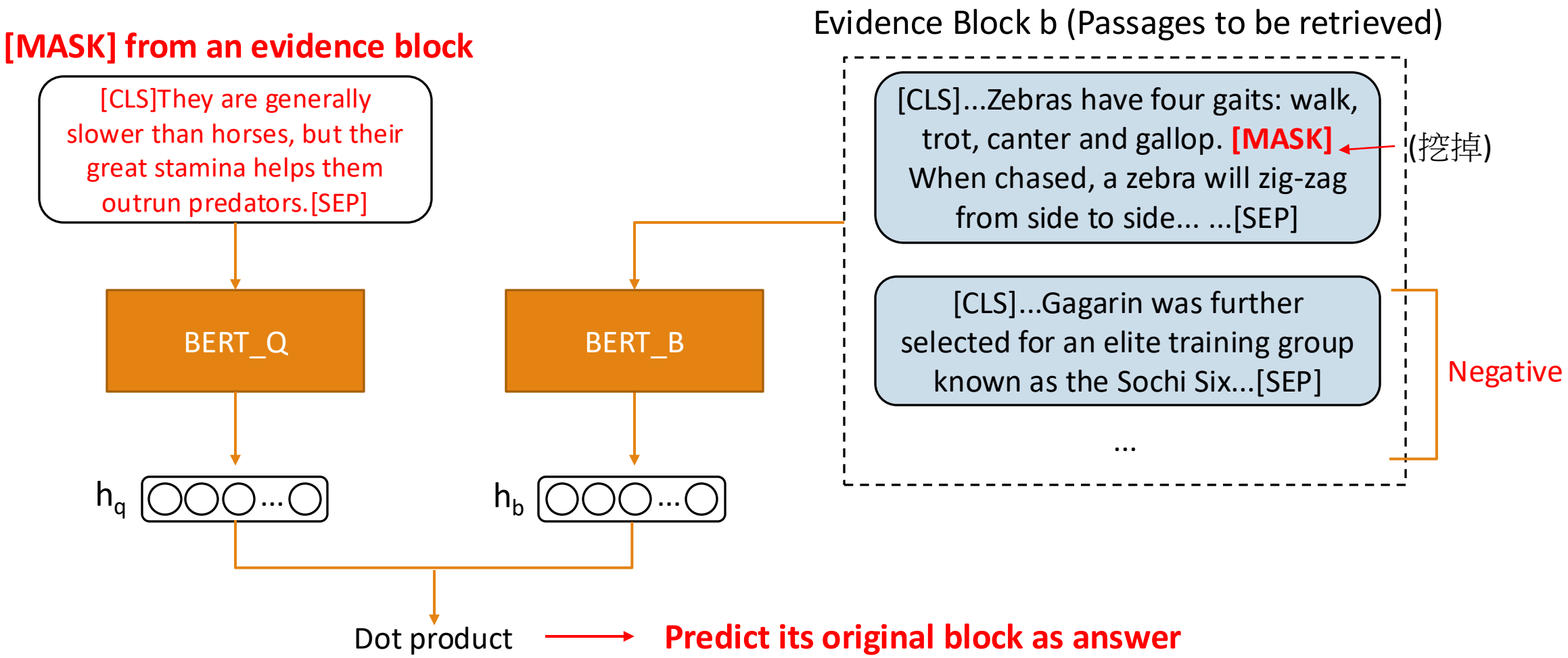
Final score: $S_{retr}(q, b) + S_{read}(q, b, s)$

Inverse Cloze Task for Pre-training a Retriever

- Cloze task: predict masked token from a corrupted context
- **Inverse cloze task (ICT): predict a corrupted context from a randomly sampled sentence (unsupervised)**
  - This is a **continual** pre-training (CP) task starting from the pre-trained BERT model in **ORQA**.

Lee, Kenton, Ming-Wei Chang, and Kristina Toutanova. "Latent Retrieval for Weakly Supervised Open Domain Question Answering." ACL 2019.

ICT for Pre-training a Retriever

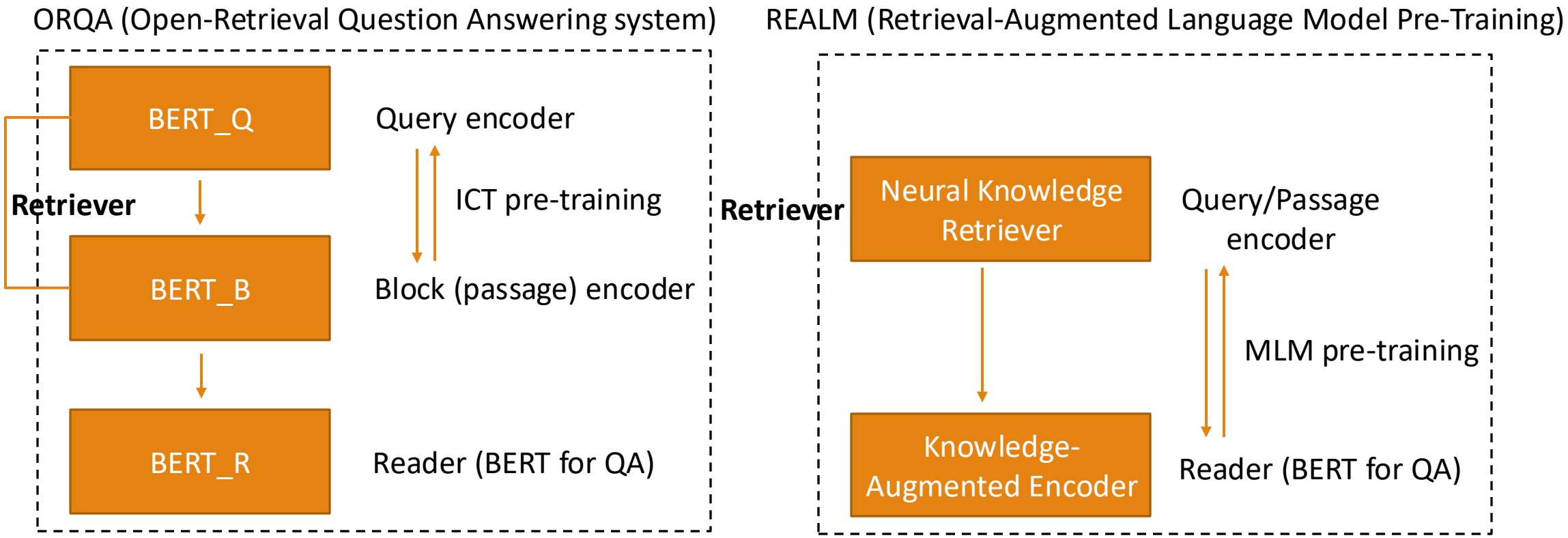


ICT for Pre-training a Retriever

- Cloze task: predict masked token from a corrupted context
- **Inverse cloze task (ICT): predict a corrupted context from a randomly sampled sentence**
  - This is a **continual** pre-training (CP) task starting from the pre-trained BERT model in **ORQA**.
- Model to train:
  - Pre-training stage: BERT_Q and BERT_B
  - Fine-tuning stage: BERT_Q and BERT_R

Lee, Kenton, Ming-Wei Chang, and Kristina Toutanova. "Latent Retrieval for Weakly Supervised Open Domain Question Answering." ACL 2019.

REALM (Retrieval-Augmented Language Model Pre-Training)

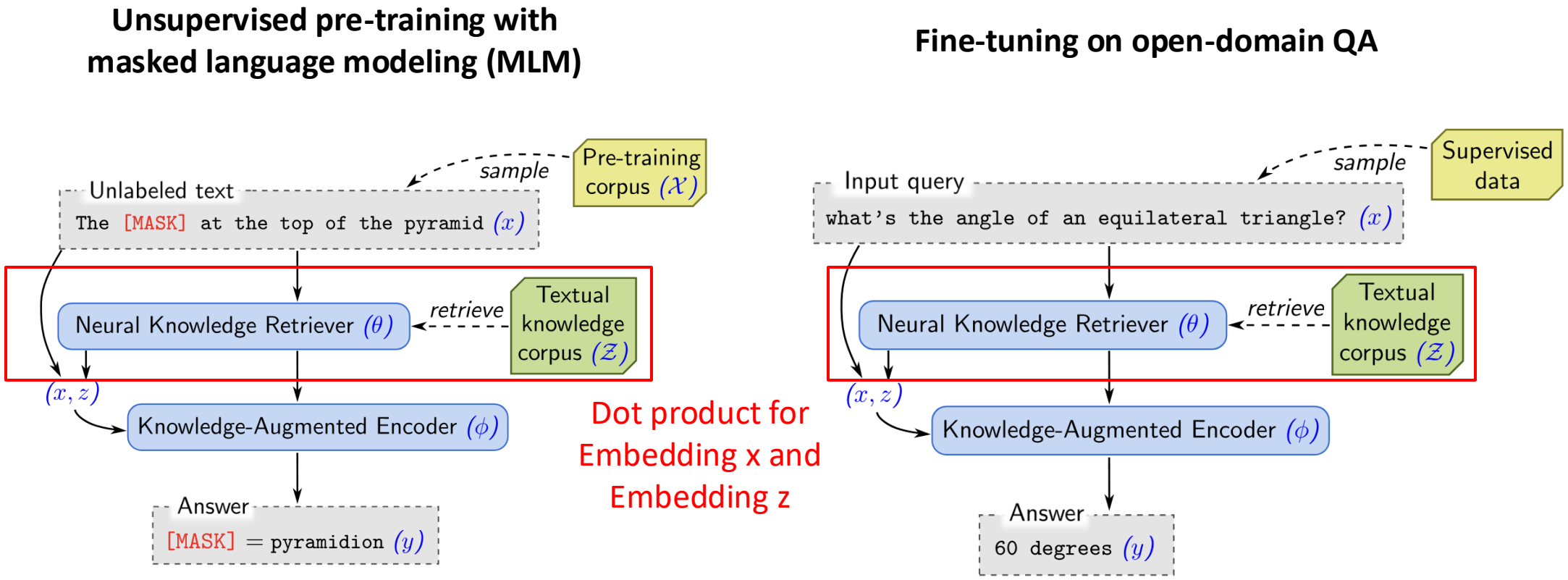


Lee, Kenton, Ming-Wei Chang, and Kristina Toutanova. "Latent Retrieval for Weakly Supervised Open Domain Question Answering." ACL 2019.

Guu, Kelvin, et al. "Retrieval augmented language model pre-training." ICML 2020.

https://arxiv.org/abs/2002.08909

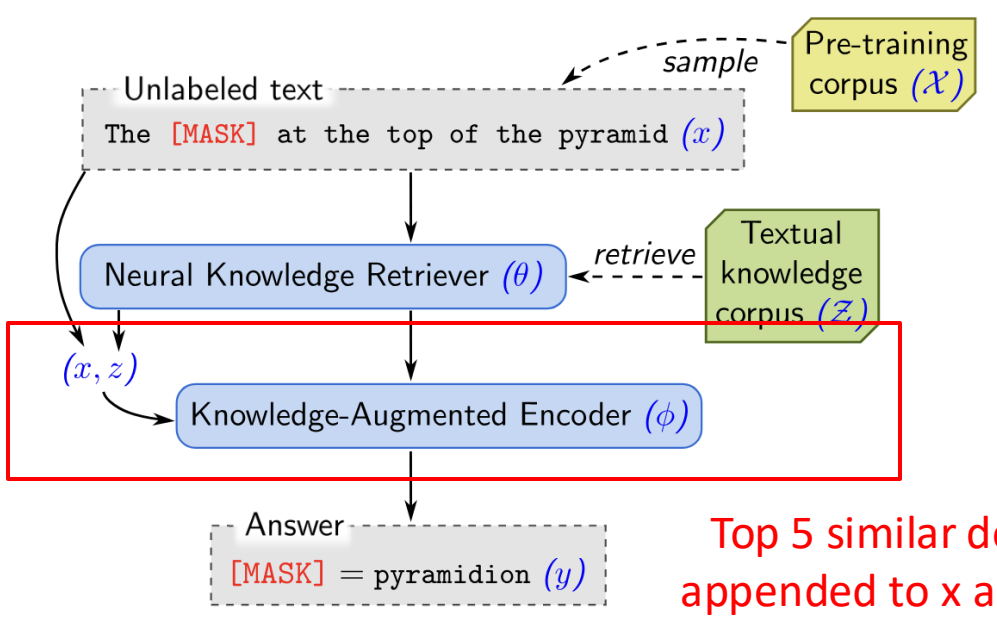
REALM (Retrieval-Augmented Language Model Pre-Training)



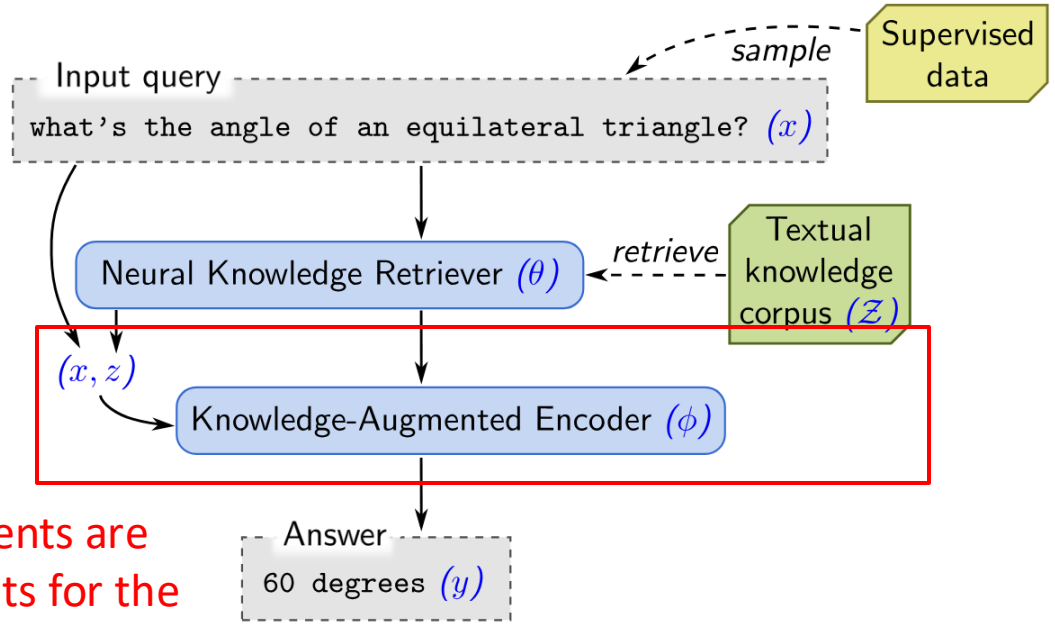
Guu, Kelvin, et al. "Retrieval augmented language model pre-training." ICML 2020.

REALM (Retrieval-Augmented Language Model Pre-Training)

Unsupervised pre-training with masked language modeling (MLM)



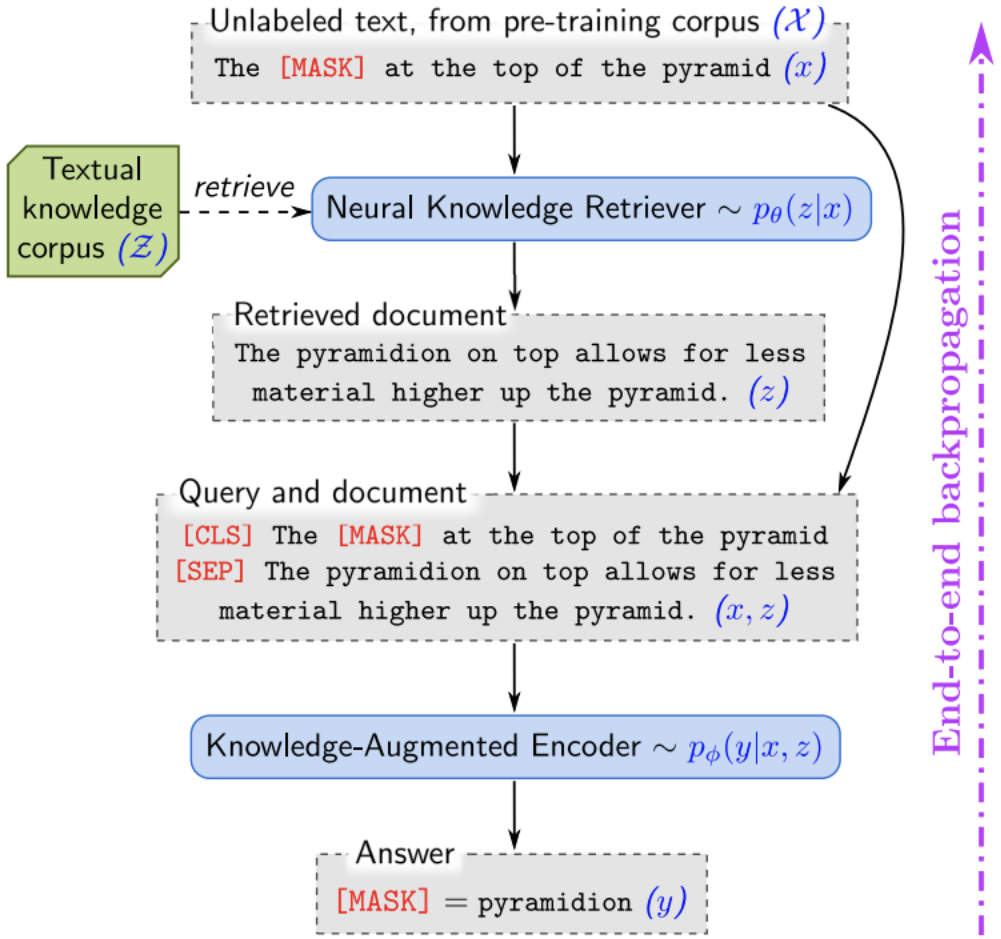
Fine-tuning on open-domain QA



Top 5 similar documents are appended to x as inputs for the Knowledge-Augmented Encoder

Guu, Kelvin, et al. "Retrieval augmented language model pre-training." ICML 2020.

REALM (Retrieval-Augmented Language Model Pre-Training)

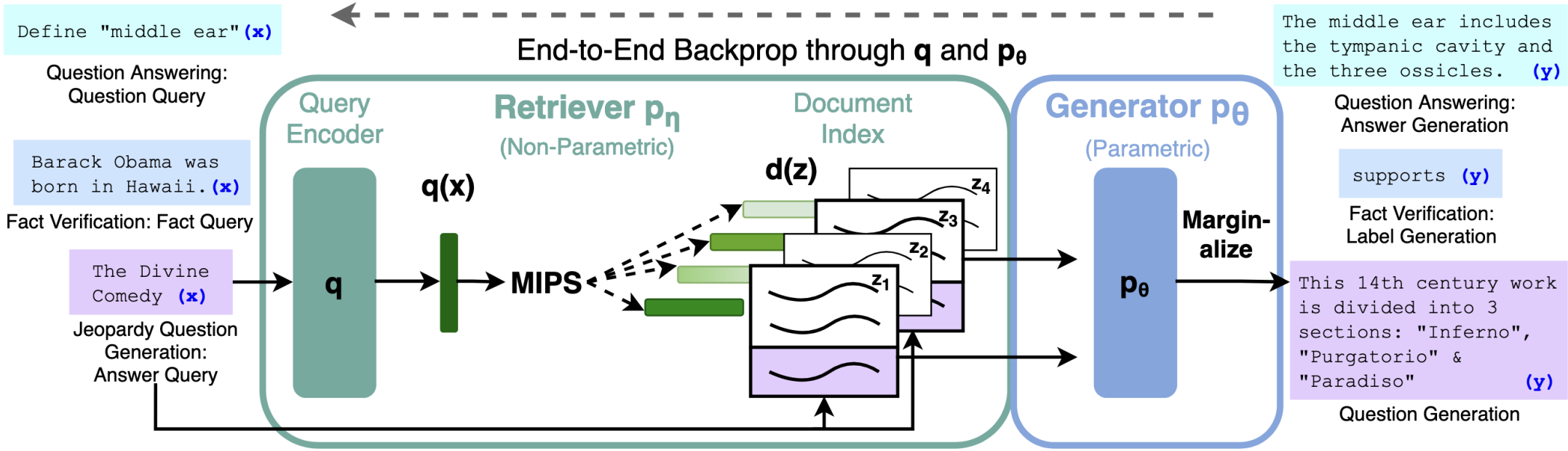


Some issues

- Corpus is big. (all of Wikipedia)
  - Index in embedding level
- $\mathcal{X}$ contain [MASK]
- End-to-end training

QA model is a generator (e.g., BART)

Retrieval-Augmented Generation (RAG)



- This approach is the first RAG paper.
- No pre-training is involved. Only fine-tuning on open-domain QA.
- MIPS: Maximum Inner-Product Search (speed-up approach, supported by indexing packages like FAISS.)

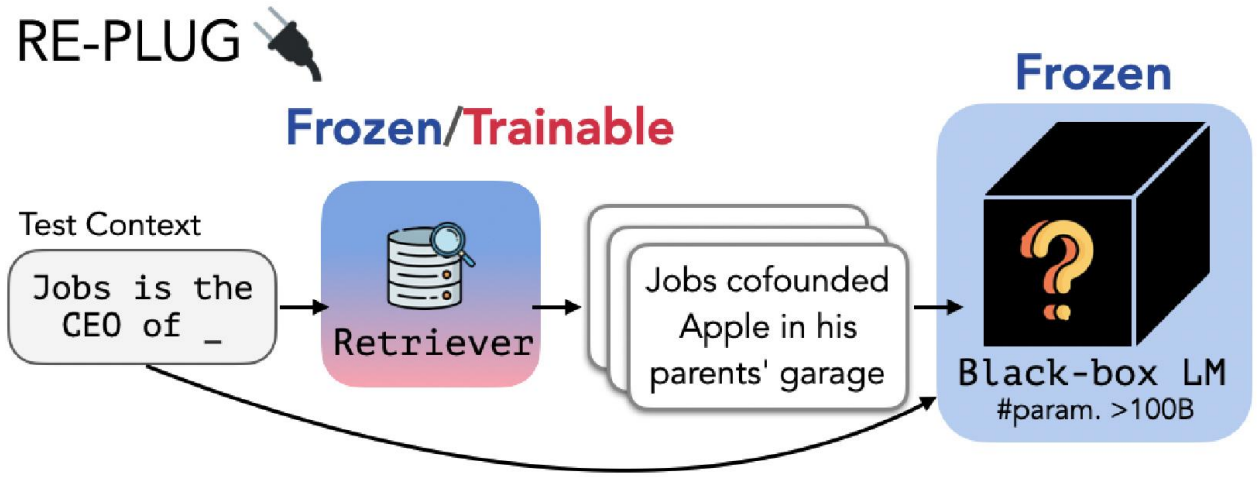
Lewis, Patrick, et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks." NeurIPS 2020.

https://aclanthology.org/2024.naacl-long.463/

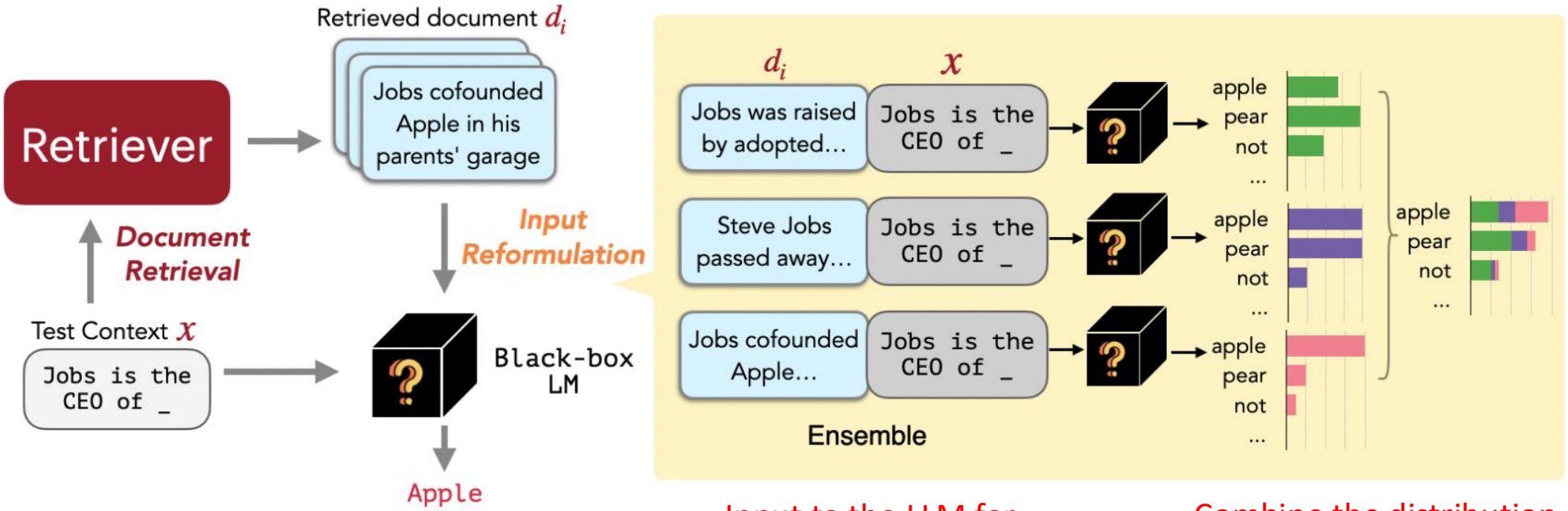
RAG with an LLM as the generator

Shi, Weijia, et al. "REPLUG: Retrieval-Augmented Black-Box Language Models." NAACL 2024.

- Due to a very large number of parameters, LLM is hard to train.
- **REPLUG (Retrieve and Plug)** proposes to train an LLM in the RAG framework.



REPLUG (at Inference Time)



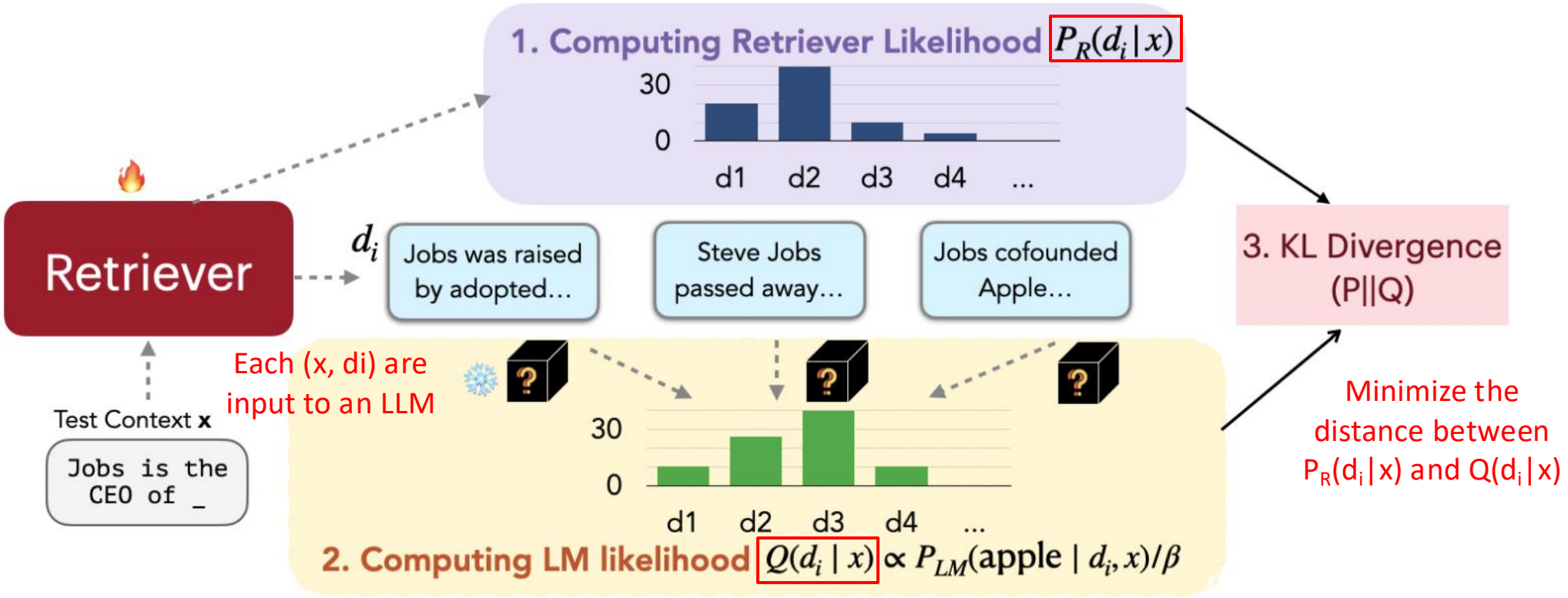
Input to the LLM for each retrieved doc

Combine the distribution for each retrieved doc

Shi, Weijia, et al. "REPLUG: Retrieval-Augmented Black-Box Language Models." NAACL 2024.

Training REPLUG with a Frozen LLM

REPLUG LSR (LM-Supervised Retrieval)



Shi, Weijia, et al. "REPLUG: Retrieval-Augmented Black-Box Language Models." NAACL 2024.

Comparison between REPLUG and REPLUG LSR

Model		# Parameters	Original	+ REPLUG	Gain %	+ REPLUG LSR	Gain %
GPT-2	Small	117M	1.33	1.26	5.3	1.21	9.0
	Medium	345M	1.20	1.14	5.0	1.11	7.5
	Large	774M	1.19	1.15	3.4	1.09	8.4
	XL	1.5B	1.16	1.09	6.0	1.07	7.8
GPT-3 (black-box)	Ada	350M	1.05	0.98	6.7	0.96	8.6
	Babbage	1.3B	0.95	0.90	5.3	0.88	7.4
	Curie	6.7B	0.88	0.85	3.4	0.82	6.8
	Davinci	175B	0.80	0.77	3.8	0.75	6.3

Table 1: **Both REPLUG and REPLUG LSR consistently enhanced the performance of different language models.** Bits per byte (BPB) of the Pile using GPT-3 and GPT-2 family models (Original) and their retrieval-augmented versions (+REPLUG and +REPLUG LSR. The gain % shows the relative improvement of our models compared to the original language model.

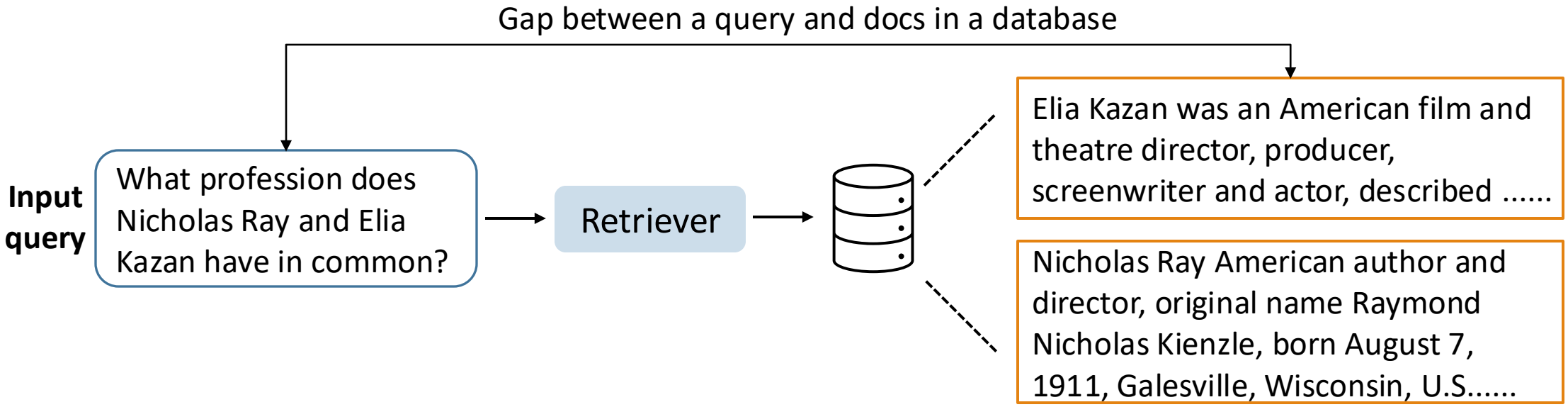
Shi, Weijia, et al. "REPLUG: Retrieval-Augmented Black-Box Language Models." NAACL 2024.

Recent RAG Developments

Type	Method	Paper	Venue Year
Enhancing Retrieval	Query Rewriting	Query Rewriting in Retrieval-Augmented Large Language Models	EMNLP 2023
	HyDE	Precise Zero-Shot Dense Retrieval without Relevance Labels	ACL 2023
Enhancing RAG	RetRobust	Making Retrieval-Augmented Language Models Robust to Irrelevant Context	ICLR 2024
	RAFT	RAFT: Adapting Language Model to Domain Specific RAG	COLM 2024
	Self-RAG	Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection	ICLR 2024
	RAAT	Enhancing Noise Robustness of Retrieval-Augmented Language Models with Adaptive Adversarial Training	ACL 2024
Enhancing RAG with Continual Retrieval	FLARE	Active Retrieval Augmented Generation	EMNLP 2023

Query Rewriting

- Motivation:
  - There is inevitably a gap between the input text and the knowledge that is really needed to query.

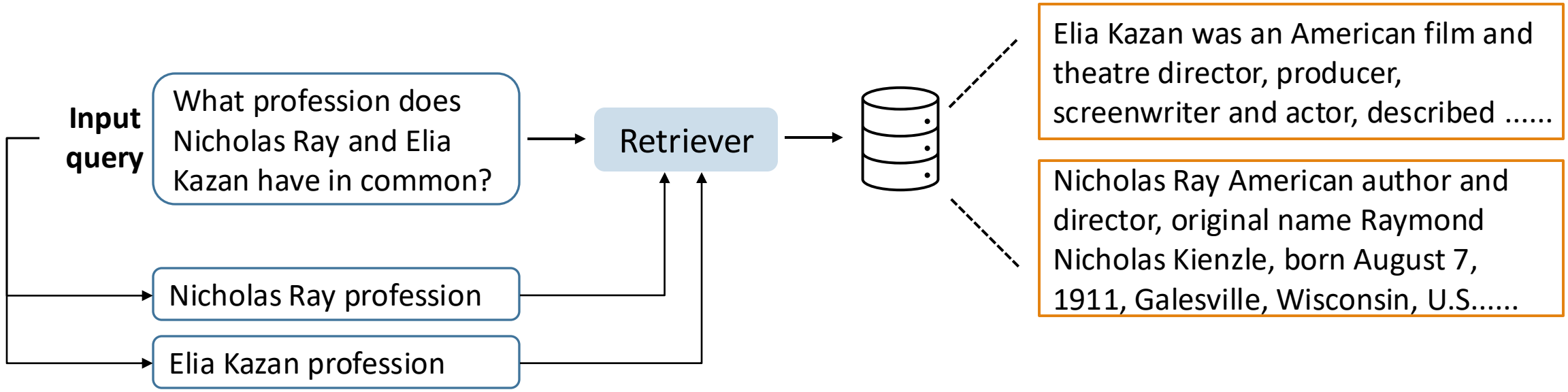


Ma, Xinbei, et al. "Query Rewriting in Retrieval-Augmented Large Language Models." EMNLP 2023.

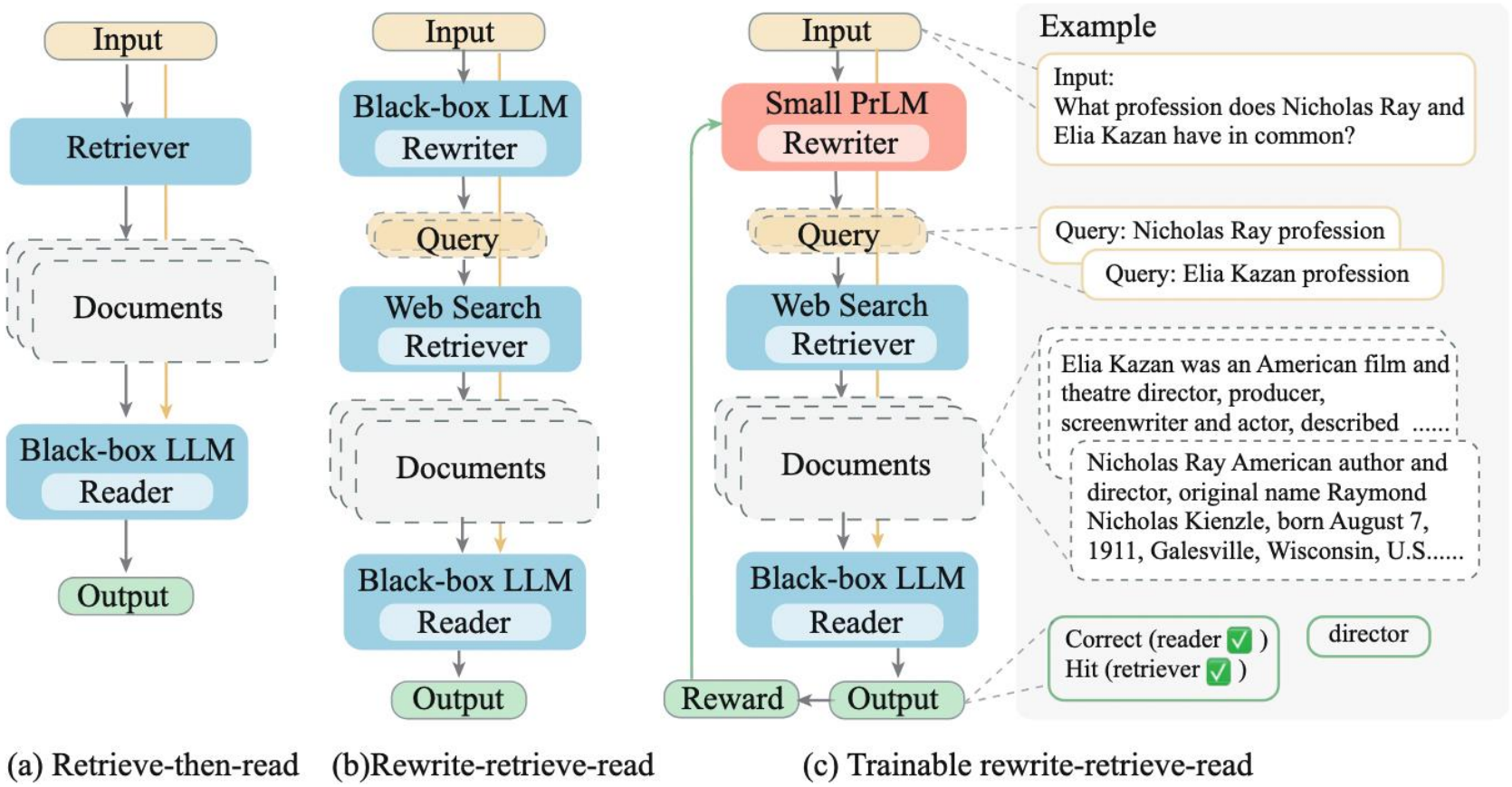
Query Rewriting

Ma, Xinbei, et al. "Query Rewriting in Retrieval-Augmented Large Language Models." EMNLP 2023.

- Motivation:
  - There is inevitably a gap between the input text and the knowledge that is really needed to query.



Query Rewriting – Approach



(a) Retrieve-then-read (b)Rewrite-retrieve-read (c) Trainable rewrite-retrieve-read

Ma, Xinbei, et al. "Query Rewriting in Retrieval-Augmented Large Language Models." EMNLP 2023.

Query Rewriting – Prompt for the LLMs

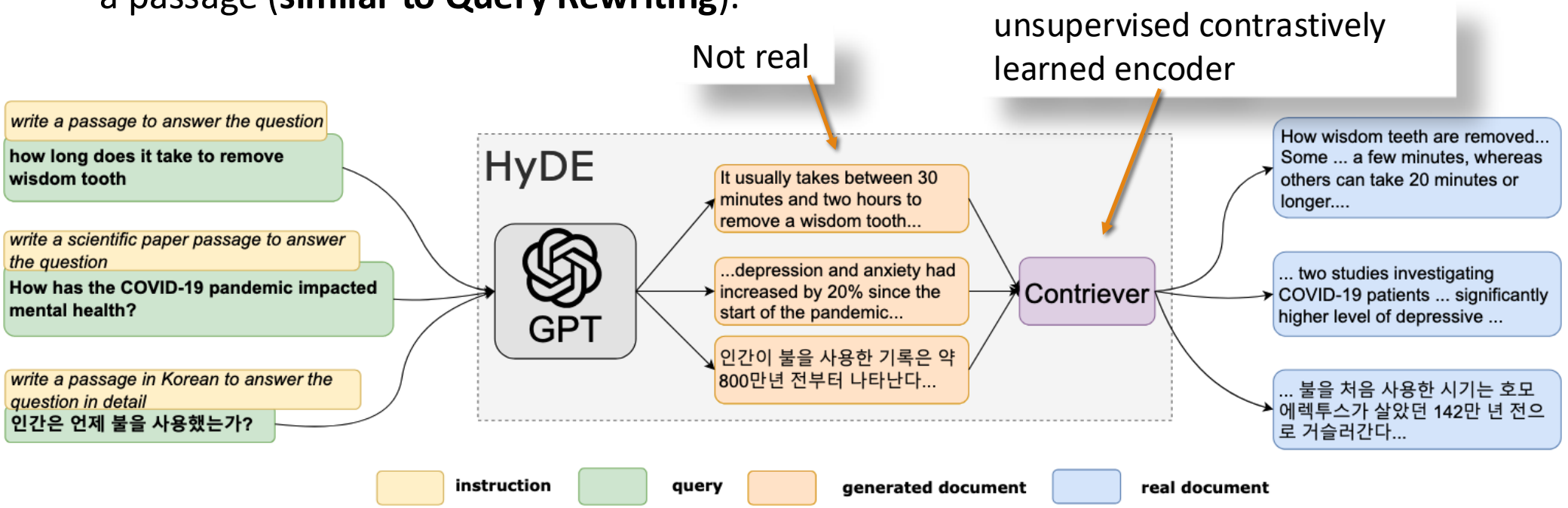
Direct prompt
Answer the question in the following format, end the answer with '**'. {demonstration} Question: {x} Answer:
Reader prompt in retrieval-augment pipelines
Answer the question in the following format, end the answer with '**'. {demonstration} Question: {doc} {x} Answer:
Prompts for LLM as a frozen rewriter
Open-domain QA: Think step by step to answer this question, and provide search engine queries for knowledge that you need. Split the queries with ';' and end the queries with '**'. {demonstration} Question: {x} Answer: *Multiple choice QA:* Provide a better search query for web search engine to answer the given question, end the queries with '**'. {demonstration} Question: {x} Answer:

Table 1: Prompt lines used for the LLMs.

Ma, Xinbei, et al. "Query Rewriting in Retrieval-Augmented Large Language Models." EMNLP 2023.

Hypothetical Document Embeddings (HyDE)

- Use an LLM (InstructGPT) to perform query transformations by asking the LLM to write a passage (**similar to Query Rewriting**).



Gao, Luyu, et al. "Precise Zero-Shot Dense Retrieval without Relevance Labels." ACL 2023.

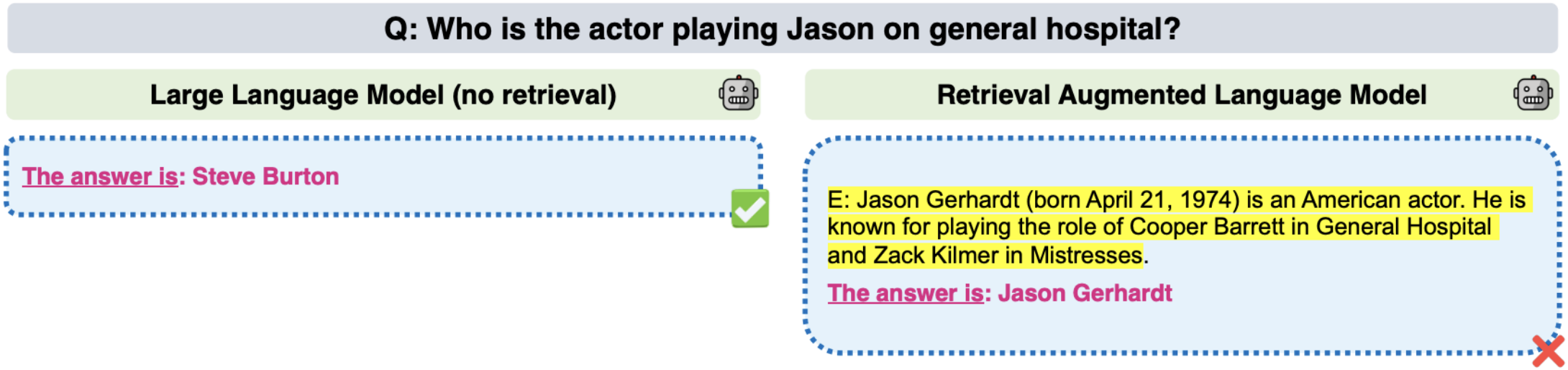
HyDE can be better than Query Rewriting for retrieval tasks

Method	TREC DL19					TREC DL20				
	mAP	nDCG@10	R@50	R@1k	Latency	mAP	nDCG@10	R@50	R@1k	Latency
unsupervised										
BM25	30.13	50.58	38.32	75.01	**0.07**	28.56	47.96	46.18	78.63	**0.29**
Contriever	23.99	44.54	37.54	74.59	3.06	23.98	42.13	43.81	75.39	0.98
supervised										
LLM-Embedder	44.66	70.20	49.06	84.48	2.61	45.60	68.76	61.36	84.41	0.71
+ Query Rewriting	44.56	67.89	51.45	85.35	7.80	45.16	65.62	59.63	83.45	2.06
+ Query Decomposition	41.93	66.10	48.66	82.62	14.98	43.30	64.95	57.74	84.18	2.01
+ HyDE	50.87	**75.44**	54.93	88.76	7.21	50.94	**73.94**	63.80	88.03	2.14
+ Hybrid Search	47.14	72.50	51.13	89.08	3.20	47.72	69.80	64.32	88.04	0.77
+ HyDE + Hybrid Search	**52.13**	73.34	**55.38**	**90.42**	11.16	**53.13**	72.72	**66.14**	**90.67**	2.95

Table 7: Results for different retrieval methods on TREC DL19/20. The best result for each method is made bold and the second is underlined.

Wang, Xiaohua, et al. "Searching for best practices in retrieval-augmented generation." EMNLP 2024.

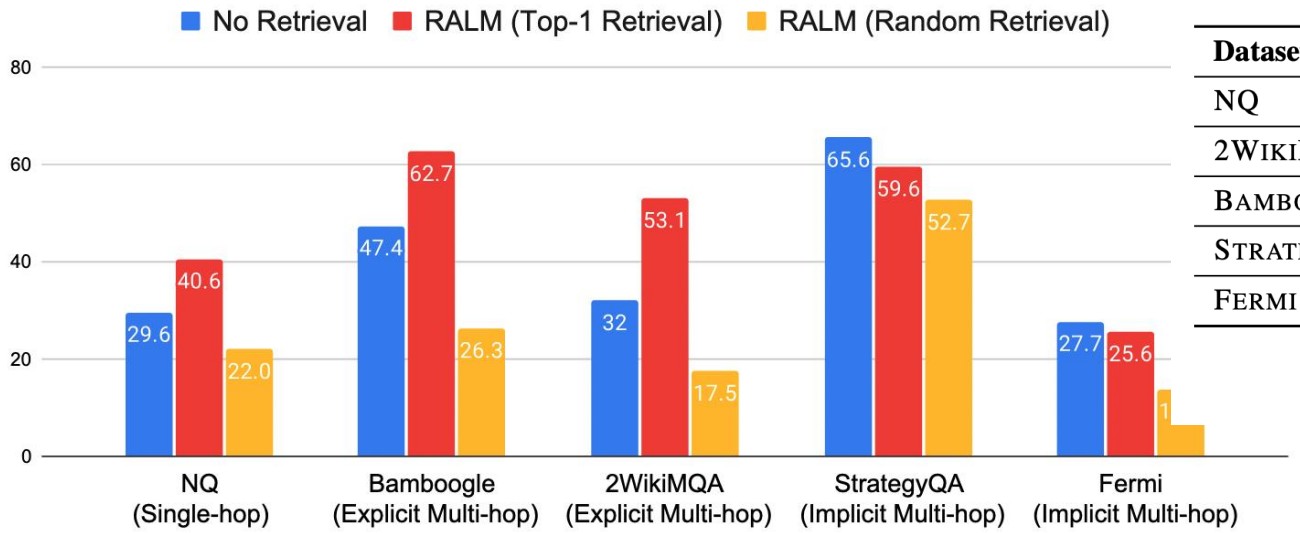
RetRobust: Making Retrieval-Augmented Language Models Robust to Irrelevant Context



RALM: Retrieval Augmented Language Model

Yoran, Ori, et al. "Making Retrieval-Augmented Language Models Robust to Irrelevant Context." ICLR 2024.

RetRobust: Making Retrieval-Augmented Language Models Robust to Irrelevant Context



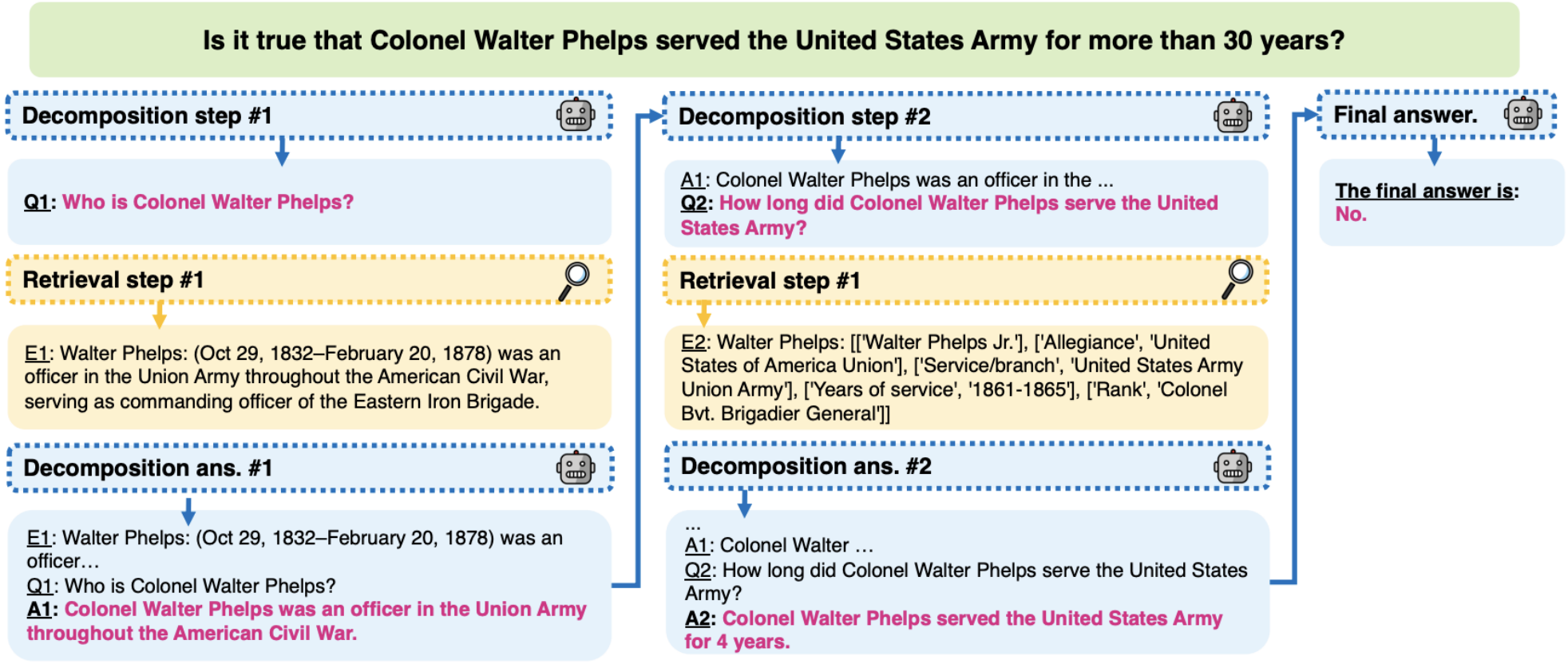
Dataset	Type	Example
NQ	Single-hop	What episode of law and order svu is mike tyson in?
2WIKIMQA	Explicit	Where was the place of death of Isabella Of Bourbon's father?
BAMBOOGLE	Explicit	What is the maximum airspeed (in km/h) of the third fastest bird?
STRATEGYQA	Implicit	Can Arnold Schwarzenegger deadlift an adult Black rhinoceros?
FERMI	Implicit	How many high fives has Lebron James given/received?

Table 1: The QA datasets in our experiments.

Retrieval augmentation can boost performance, but it also hurts performance on StrategyQA and Fermi, and random contexts reduce performance dramatically.

Yoran, Ori, et al. "Making Retrieval-Augmented Language Models Robust to Irrelevant Context." ICLR 2024.

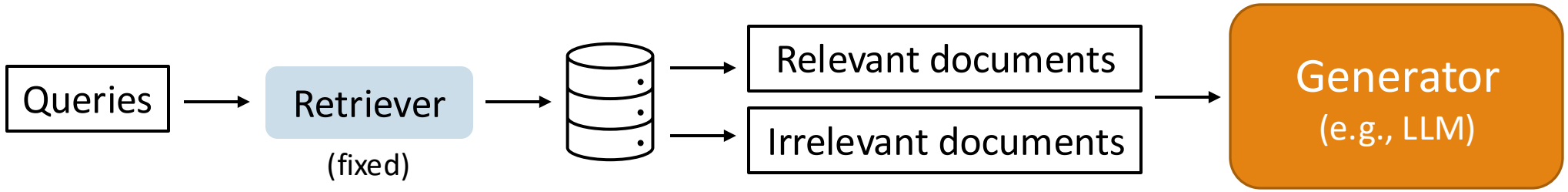
RetRobust: Making Retrieval-Augmented Language Models Robust to Irrelevant Context

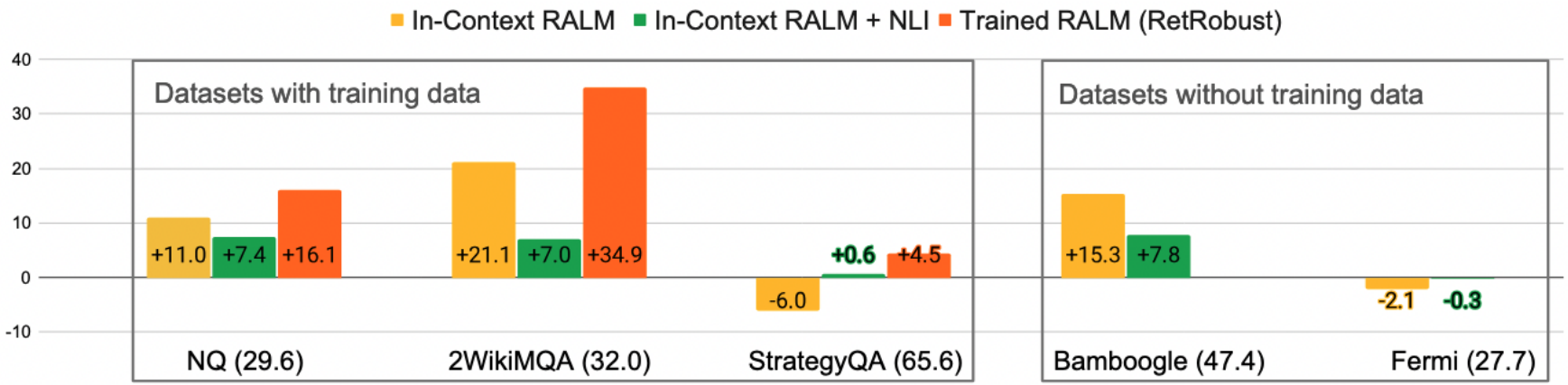


Yoran, Ori, et al. "Making Retrieval-Augmented Language Models Robust to Irrelevant Context." ICLR 2024.

RetRobust: Making Retrieval-Augmented Language Models Robust to Irrelevant Context

Train the generator on QA tasks with both relevant and irrelevant retrieved documents.

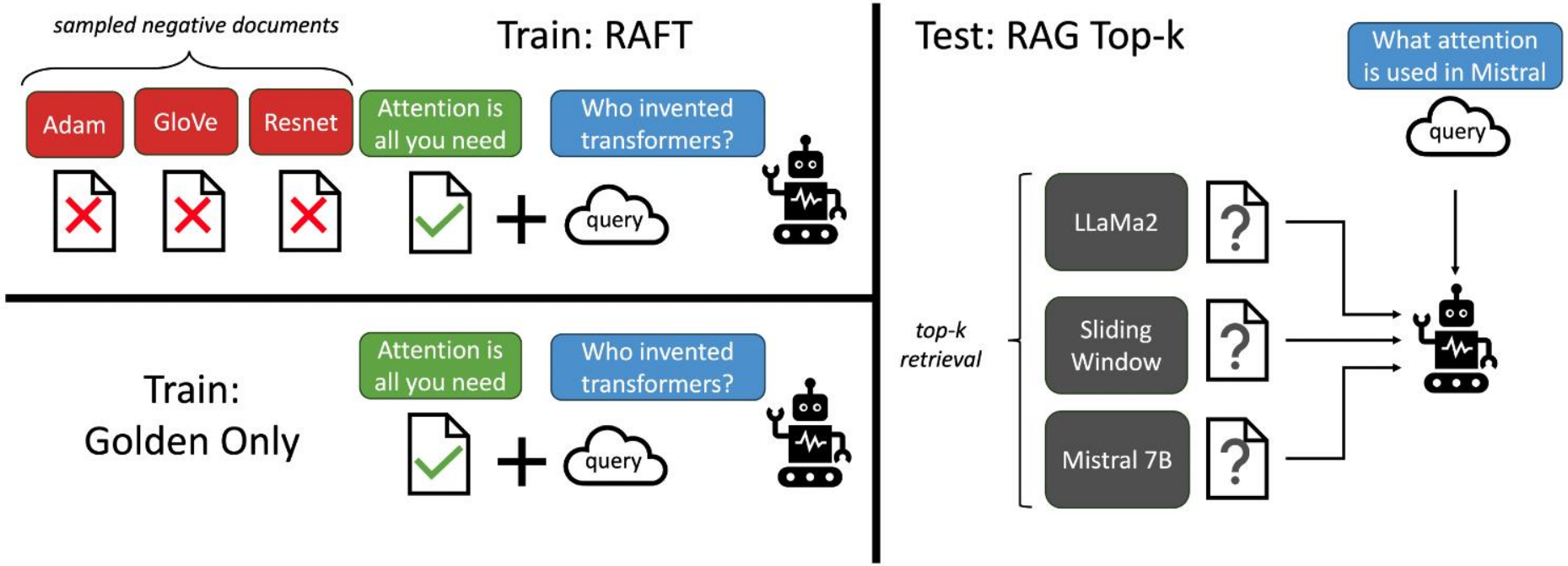




Yoran, Ori, et al. "Making Retrieval-Augmented Language Models Robust to Irrelevant Context." ICLR 2024.

RAFT: Adapting LLM to RAG

- In this work, the retriever is fixed. An LLM is trainable.
- Both correct and incorrect documents are included during training.



MateiZaharia, TianjunZhang ShishirG Patil NamanJain ShengShen, and IonStoica JosephE Gonzalez. "RAFT: Adapting Language Model to Domain Specific RAG." COLM 2024.

Difference between RetRobust and RAFT

RAFT uses **CoT Answer (answer with rationales).**

```
Question: The Oberoi family is part of a hotel company that has a head office
in what city?

context: [The Oberoi family is an Indian family that is famous for its
involvement in hotels, namely through The Oberoi Group]...[It is located in
city center of Jakarta, near Mega Kuningan, adjacent to the sister JW Marriott
Hotel. It is operated by The Ritz-Carlton Hotel Company. The complex has two
towers that comprises a hotel and the Airlangga Apartment respectively]...[The
Oberoi Group is a hotel company with its head office in Delhi.]

Instruction: Given the question, context and answer above, provide a logical
reasoning for that answer. Please use the format of: ##Reason: {reason}
##Answer: {answer}.

--------------------------------------------------------------------------------

CoT Answer: ##Reason: The document ##begin_quote## The Oberoi family is an
Indian family that is famous for its involvement in hotels, namely through The
Oberoi Group. ##end_quote## establishes that the Oberoi family is involved in
the Oberoi group, and the document ##begin_quote## The Oberoi Group is a hotel
company with its head office in Delhi. ##end_quote## establishes the head
office of The Oberoi Group. Therefore, the Oberoi family is part of a hotel
company whose head office is in Delhi. ##Answer: Delhi
```

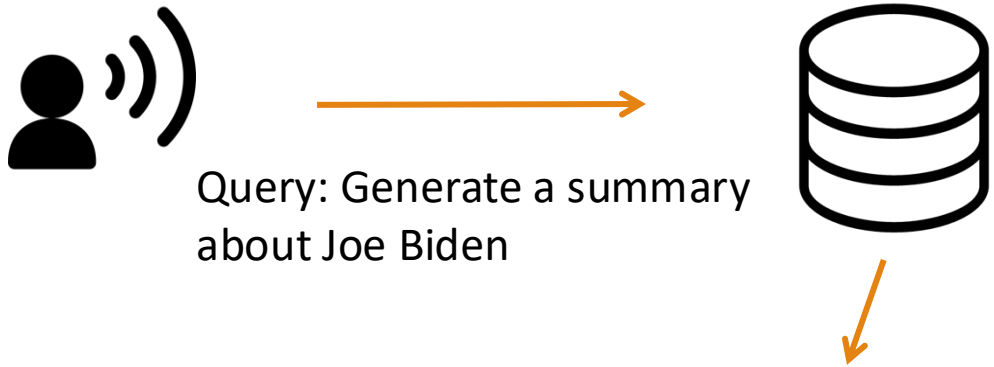
MateiZaharia, TianjunZhang ShishirG Patil NamanJain ShengShen, and IonStoica JosephE Gonzalez. "RAFT: Adapting Language Model to Domain Specific RAG." COLM 2024.

EMNLP 2023
Carnegie Mellon University

Active Retrieval Augmented Generation

Problem Formation

- For a naïve retrieval augmented generation:



Redundant information (Noise) may confuse the language model and make the generation wrong.

1. Joseph Robinette Biden Jr. was born on November 20, 1942
2. Neilia Hunter Biden, the first wife of Joe Biden, was born on July 28, 1942.
3. I started thinking as I was coming over here, why is it that Joe Biden is the first in his family ever to go to a university?

Noise

Methodology

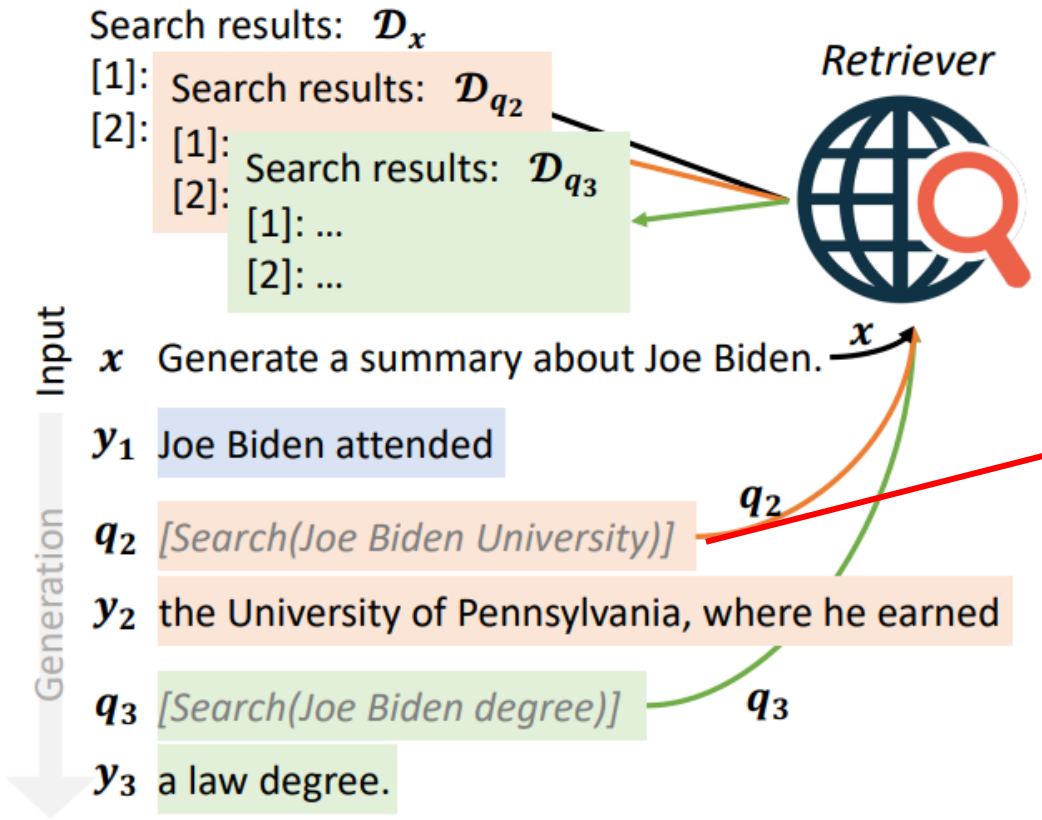
In retrieval-augmented generation, it is crucial to determine:

- **What to retrieve:** Selecting relevant and useful information.
- **When to retrieve:** Identifying the appropriate timing for retrieval.

The retrieved information should:

- Be essential for the model's generation process.
- Directly address the model's specific needs.

What to retrieve

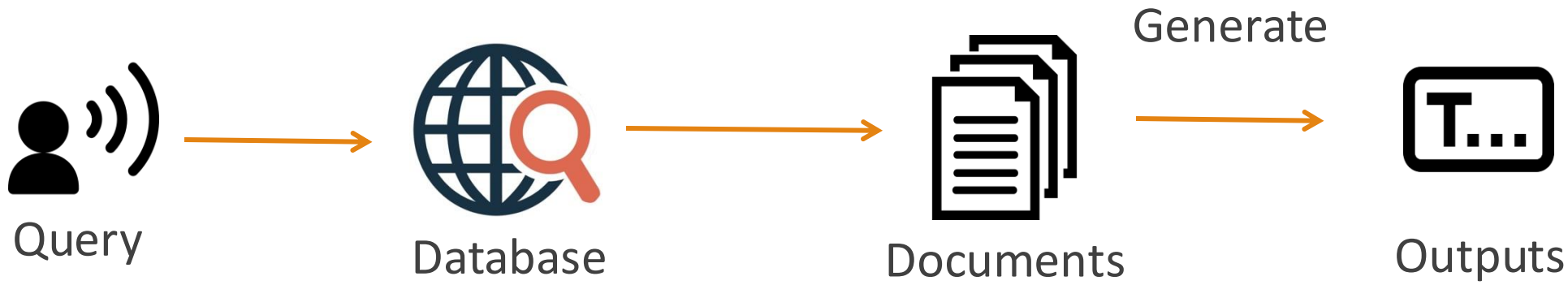


1. LMs should only retrieve information when they do not have the necessary knowledge to avoid unnecessary or inappropriate retrieval.
2. the retrieval queries should reflect the intents of future generations

Output = concat(y_1, y_2, y_3, ...,)

When to retrieve

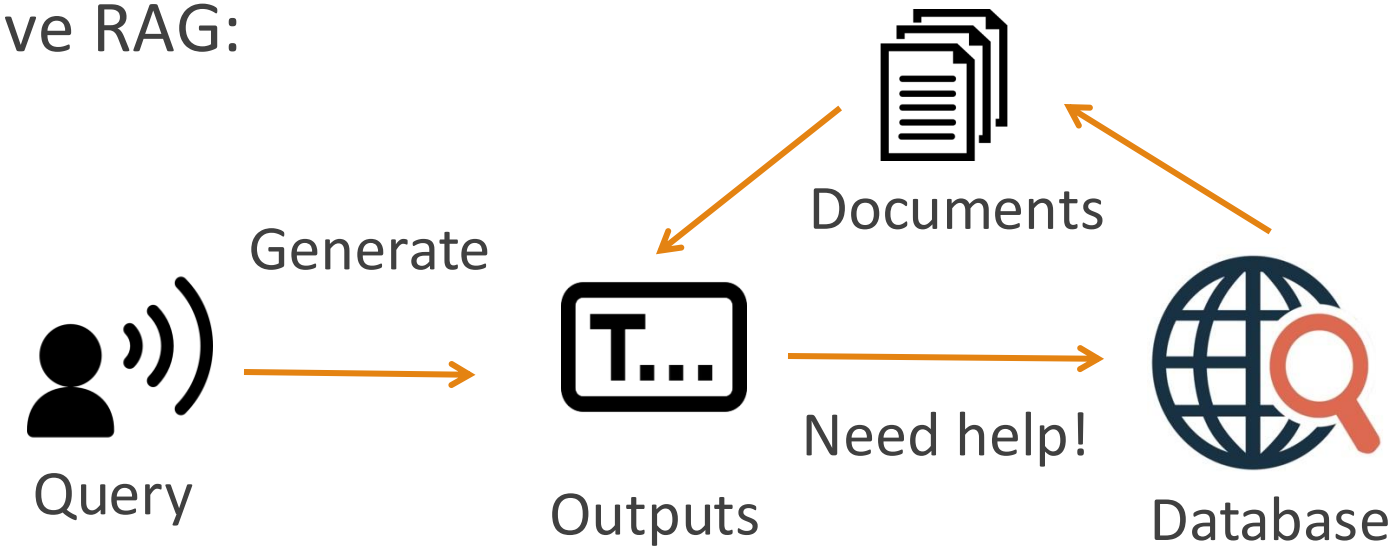
For traditional RAG:



The documents are retrieved based on the input query before generation.

When to retrieve

For Active RAG:



The documents are retrieved only when the language model needs help.

The definition of "need help"

Language models are often well-calibrated, meaning that low probability or confidence typically reflects a lack of knowledge.

This paper define a threshold $\theta \in [0, 1]$, when the probability of next token prediction is lower than θ, a retrieval is triggered.

There is no need to perform retrieval.

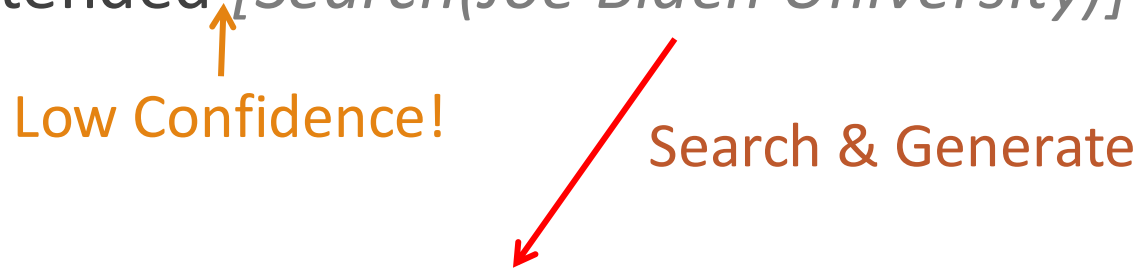
$$\boldsymbol{y}_t = \begin{cases} \hat{\boldsymbol{s}}_t & \text{if all tokens of } \hat{\boldsymbol{s}}_t \text{ have probs} \geq \theta \\ \boldsymbol{s}_t = \mathrm{LM}([\mathcal{D}_{\boldsymbol{q}_t}, \boldsymbol{x}, \boldsymbol{y}_{<t}]) & \text{otherwise} \end{cases}$$

Use $\widehat{S}_t$ to perform **a next** retrieval

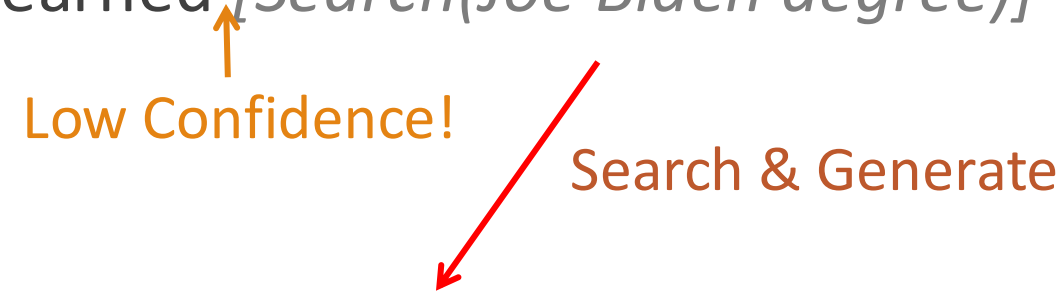
Generation

Query: Generate a summary about Joe Biden

Outputs: Joe Biden attended *[Search(Joe Biden University)]*



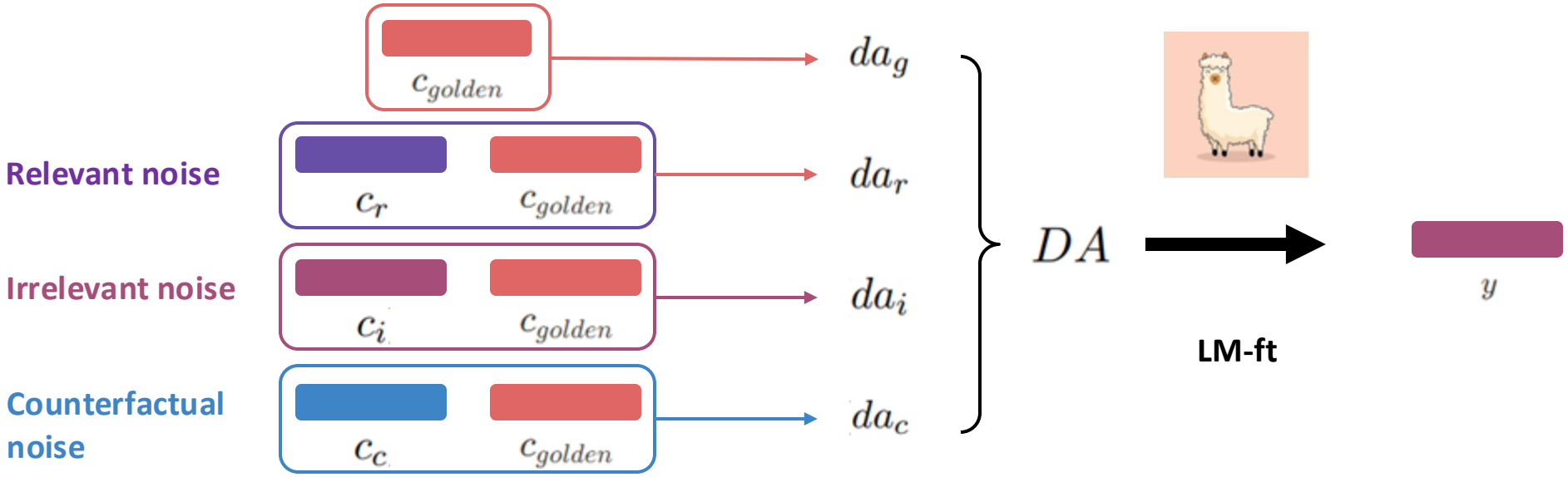
the University of Pennsylvania, where he earned *[Search(Joe Biden degree)]*



a law degree.

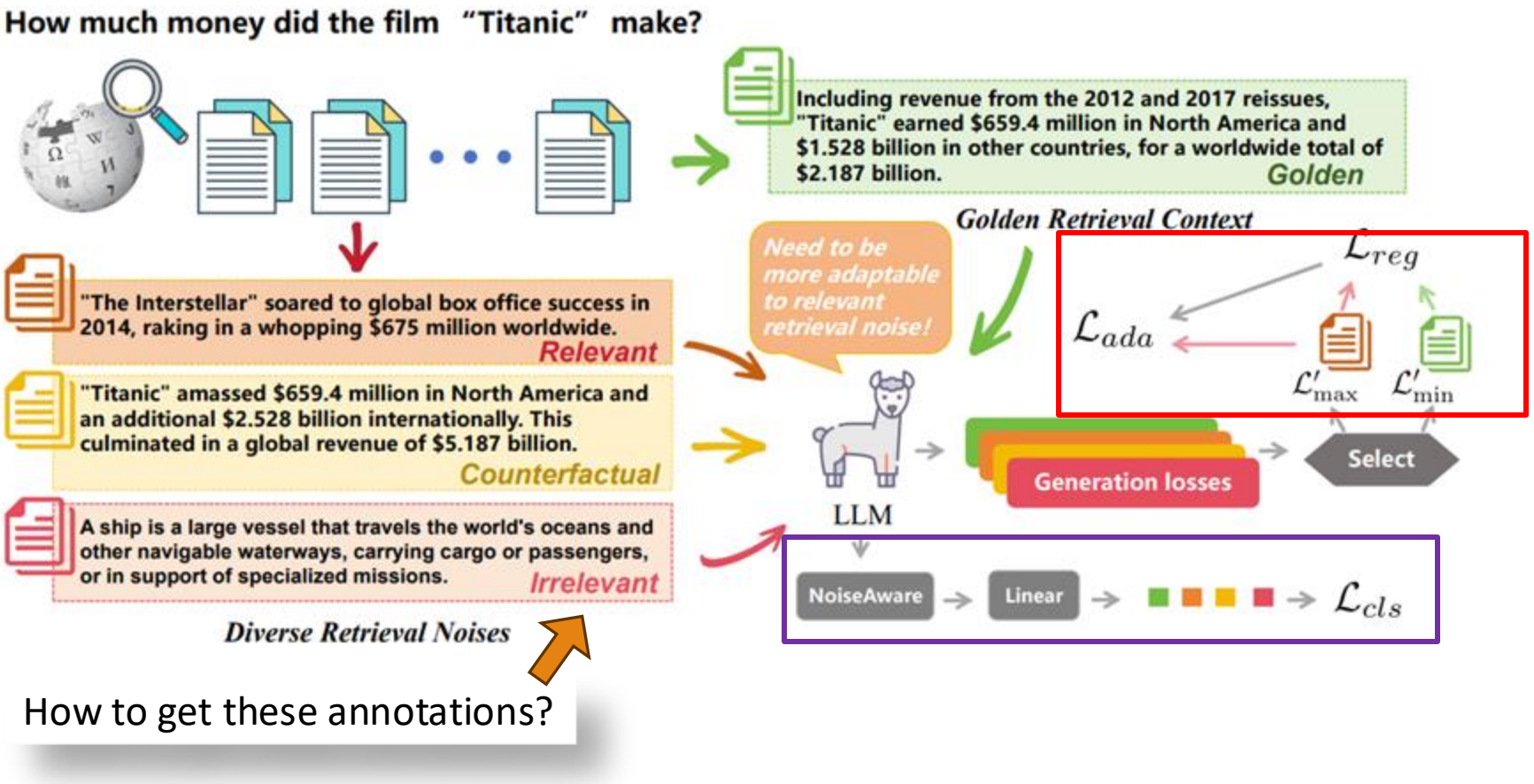
RAAT

- Recently, several studies have attempted to **enhance the noise robustness of RALMs through noisy training**, which involves incorporating retrieved noisy contexts into fine-tuning data.



Fang, Feiteng, et al. “Enhancing Noise Robustness of Retrieval-Augmented Language Models with Adaptive Adversarial Training.” ACL 2024.

RAAT



Adaptive learning maximizes input manipulation and minimizes by fine-tuning to enhance model robustness.

An auxiliary task enhances RALMs' robustness by detecting and addressing retrieval noise.

Fang, Feiteng, et al. "Enhancing Noise Robustness of Retrieval-Augmented Language Models with Adaptive Adversarial Training." ACL 2024.

RAAT Conclusion

Method	Golden Only		Golden & c_i		Golden & c_r		Golden & c_c		Avg	
	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
LLaMA2_{7B}	65.56	51.80	56.14	42.87	53.10	39.73	51.81	38.37	56.68	43.19
Qwen_{7B}	62.57	47.07	61.48	46.06	55.50	40.50	53.26	36.90	58.20	42.63
LLaMA2_{13B}	69.27	55.00	63.25	49.47	62.27	47.97	62.07	47.17	64.22	49.90
Qwen_{14B}	67.45	51.43	66.71	51.20	61.88	46.16	58.65	41.30	63.67	47.52
LLaMA2_{70B}	71.43	56.56	70.05	55.13	65.97	51.33	63.91	48.27	67.84	52.82
$\text{ChatGPT}_{3.5}$	73.98	60.50	72.24	60.30	70.65	56.89	69.00	54.64	71.47	58.10
RALM_{golden}	80.31	74.03	79.33	72.73	73.26	66.33	73.08	65.40	76.50	69.62
RetRobust	80.10	73.80	79.25	72.97	74.81	68.30	75.46	68.43	77.41	70.88
$\text{RALM}_{retrieved}$	80.04	73.40	81.09	74.80	75.99	69.10	73.10	65.67	77.55	70.74
$\text{RALM}_{multiple}$	85.47	80.17	85.27	81.20	83.07	78.33	83.25	79.23	84.27	79.73
RAAT	**87.15**	**83.07**	**86.80**	**82.73**	**85.14**	**81.00**	**86.29**	**82.10**	**86.35**	**82.23**

Table 2: Experimental results on our RAG-Bench benchmark. "Golden Only" denotes a scenario where LLMs only consult the golden retrieval context. In "Golden & $c_i/c_r/c_c$", LLMs consider both the golden retrieval context and *irrelevant retrieval noise/relevant retrieval noise/counterfactual retrieval noise*.

Fang, Feiteng, et al. "Enhancing Noise Robustness of Retrieval-Augmented Language Models with Adaptive Adversarial Training." ACL 2024.

RAAT Conclusion

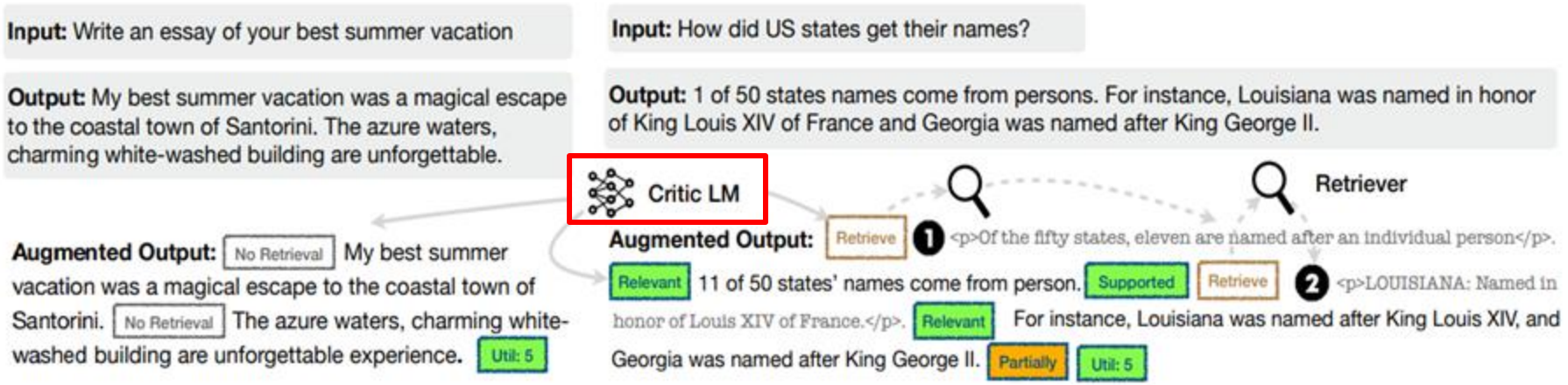
1. Investigated **retrieval noises** in RALMs and categorized them into **three distinct types**, reflecting real-world environments.
2. Introduced **RAAT** as a solution to address the noise robustness challenges faced by RALMs, which leveraged **adaptive adversarial learning** and **multi-task learning** to enhance the model's capability.
3. **Established a benchmark** to verify the effectiveness of RAAT based on three open-domain QA datasets.

Fang, Feiteng, et al. "Enhancing Noise Robustness of Retrieval-Augmented Language Models with Adaptive Adversarial Training." ACL 2024.

Self-RAG

Self-RAG (Training)

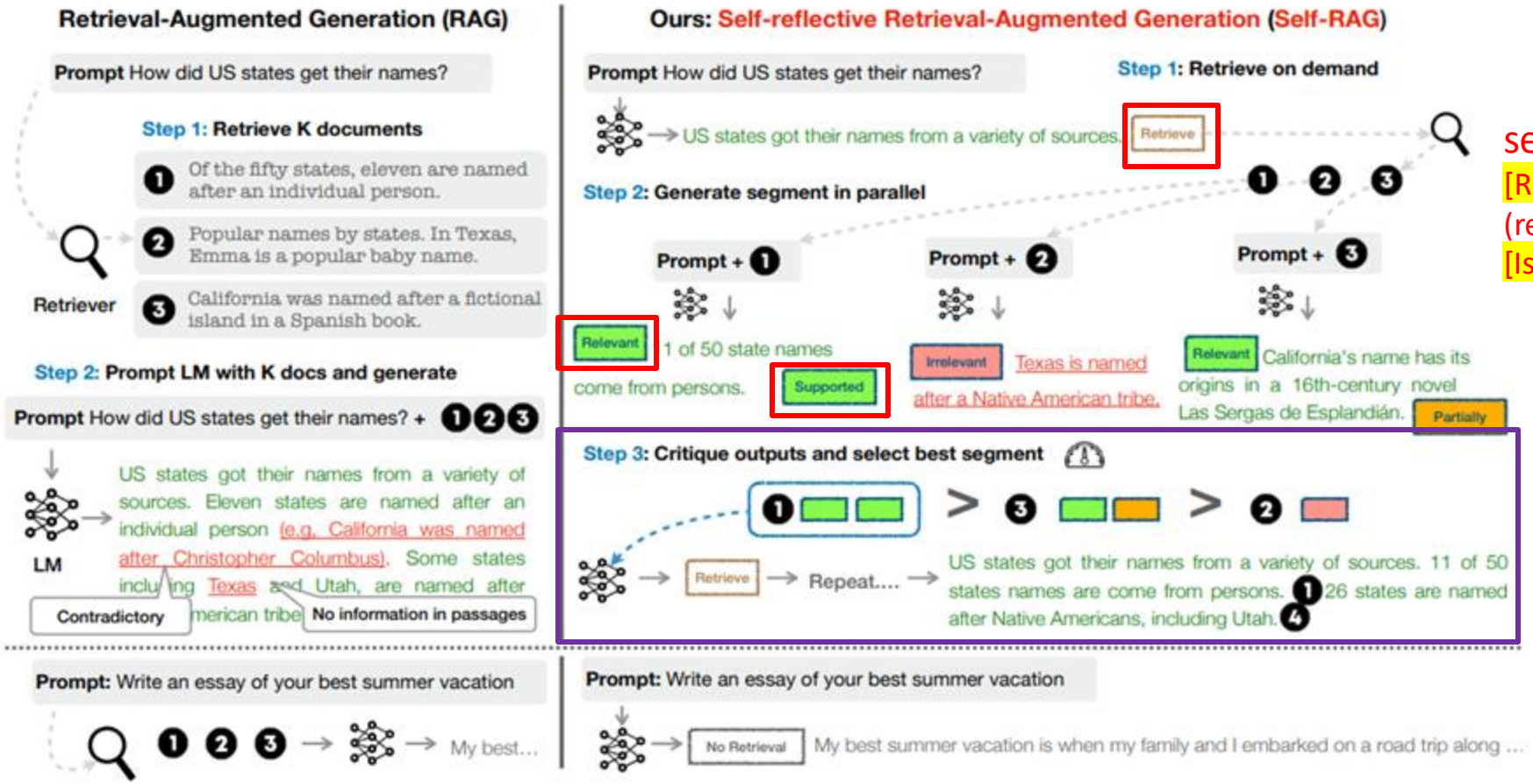
- Supervised data is created by prompting GPT-4 for reflection tokens, distilled into a critic model with token-specific definitions and inputs.
- Thus, they use critic model to collect training data for Generator.



Asai, Akari, et al. "Self-rag: Learning to retrieve, generate, and critique through self-reflection." ICLR 2024.

Self-RAG

Self-RAG (Inference)



self-reflection: [Retrieve], [IsRel], [IsSup] (response supported by d), [IsUse] (useful response)

obtain the top-B segment continuations at each timestamp

Asai, Akari, et al. "Self-rag: Learning to retrieve, generate, and critique through self-reflection." ICLR 2024.

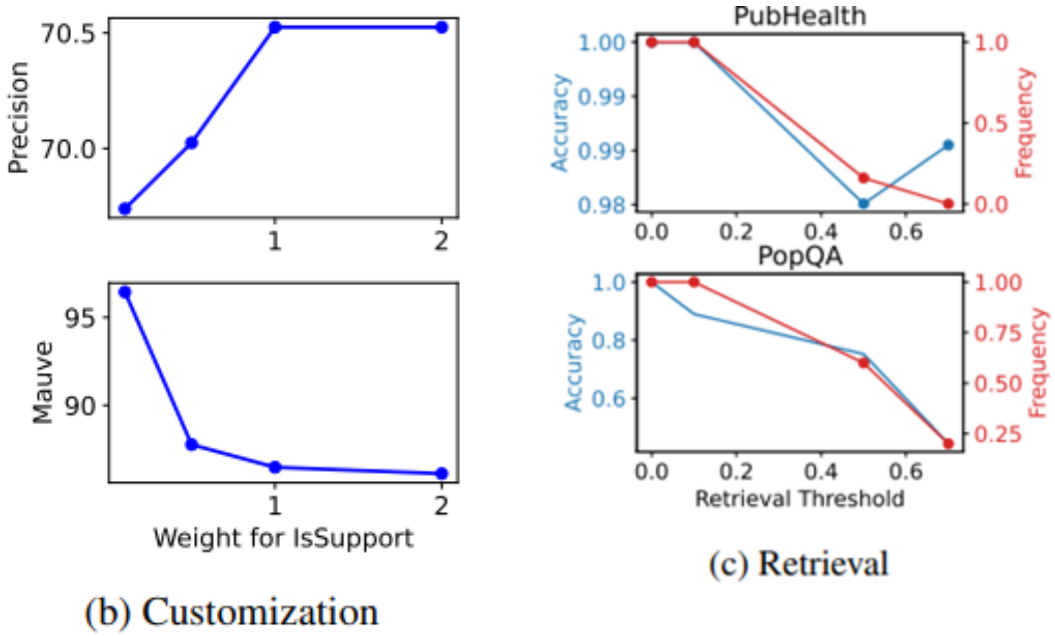
Self-RAG

LM	Short-form PopQA (acc)	Short-form TQA (acc)	Closed-set Pub (acc)	Closed-set ARC (acc)	Long-form generations (with citations) Bio (FS)	ASQA (em)	ASQA (rg)	ASQA (mau)	ASQA (pre)	ASQA (rec)
LMs with proprietary data										
Llama2-c$_{13B}$	20.0	59.3	49.4	38.4	55.9	22.4	29.6	28.6	–	–
Ret-Llama2-c$_{13B}$	51.8	59.8	52.1	37.9	79.9	32.8	34.8	43.8	19.8	36.1
ChatGPT	29.3	74.3	70.1	75.3	71.8	35.3	36.2	68.8	–	–
Ret-ChatGPT	50.8	65.7	54.7	75.3	–	40.7	39.9	79.7	65.1	76.6
Perplexity.ai	–	–	–	–	71.2	–	–	–	–	–
Baselines without retrieval										
Llama2$_{7B}$	14.7	30.5	34.2	21.8	44.5	7.9	15.3	19.0	–	–
Alpaca$_{7B}$	23.6	54.5	49.8	45.0	45.8	18.8	29.4	61.7	–	–
Llama2$_{13B}$	14.7	38.5	29.4	29.4	53.4	7.2	12.4	16.0	–	–
Alpaca$_{13B}$	24.4	61.3	55.5	54.9	50.2	22.9	32.0	70.6	–	–
CoVE$_{65B}$ *	–	–	–	–	71.2	–	–	–	–	–
Baselines with retrieval										
Toolformer*$_{6B}$	–	48.8	–	–	–	–	–	–	–	–
Llama2$_{7B}$	38.2	42.5	30.0	48.0	78.0	15.2	22.1	32.0	2.9	4.0
Alpaca$_{7B}$	46.7	64.1	40.2	48.0	76.6	30.9	33.3	57.9	5.5	7.2
Llama2-FT$_{7B}$	48.7	57.3	64.3	65.8	78.2	31.0	35.8	51.2	5.0	7.5
SAIL*$_{7B}$	–	–	69.2	48.4	–	–	–	–	–	–
Llama2$_{13B}$	45.7	47.0	30.2	26.0	77.5	16.3	20.5	24.7	2.3	3.6
Alpaca$_{13B}$	46.1	66.9	51.1	57.6	77.7	**34.8**	36.7	56.6	2.0	3.8
Our SELF-RAG $_{7B}$	54.9	66.4	72.4	67.3	**81.2**	30.0	35.7	**74.3**	66.9	67.8
Our SELF-RAG $_{13B}$	**55.8**	**69.3**	**74.5**	**73.1**	80.2	31.7	**37.0**	71.6	**70.3**	**71.3**

Asai, Akari, et al. "Self-rag: Learning to retrieve, generate, and critique through self-reflection." ICLR 2024.

Self-RAG

- Reflection token settings balance precision versus fluency and accuracy versus retrieval frequency.



Asai, Akari, et al. "Self-rag: Learning to retrieve, generate, and critique through self-reflection." ICLR 2024.

Self-RAG Conclusion

1. It introduces **SELF-RAG**, a new framework to enhance the quality and factuality of LLMs through **retrieval on demand** and **self-reflection**.
2. SELF-RAG trains an LM to learn to **retrieve**, **generate**, and **critique text passages and its own generation** by **predicting the next tokens** from its original vocabulary as well as newly added special tokens, called **reflection tokens**.
3. SELF-RAG further enables the **tailoring of LM behaviors** at test time by leveraging reflection tokens.

Asai, Akari, et al. "Self-rag: Learning to retrieve, generate, and critique through self-reflection." ICLR 2024.

Challenges in Modern RAG

- A significant amount of noise information even fake news in the content available on the Internet.
- Currently, there is a lack of comprehensive understanding of how each model can improve performance through information retrieval.

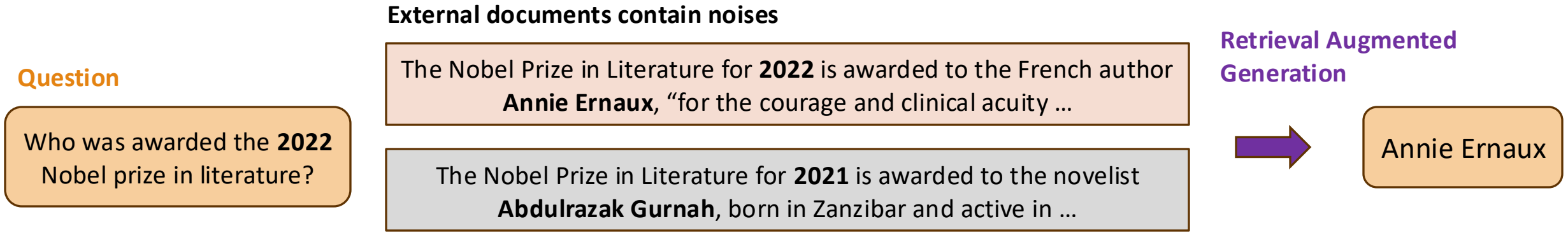
Type of Noises

- Relevant (semantically similar) but not contain the answer
- Counterfactual information
- Irrelevant information
- Black box digestion
- …

Capabilities that LLMs Should Have in RAG

Noise Robustness

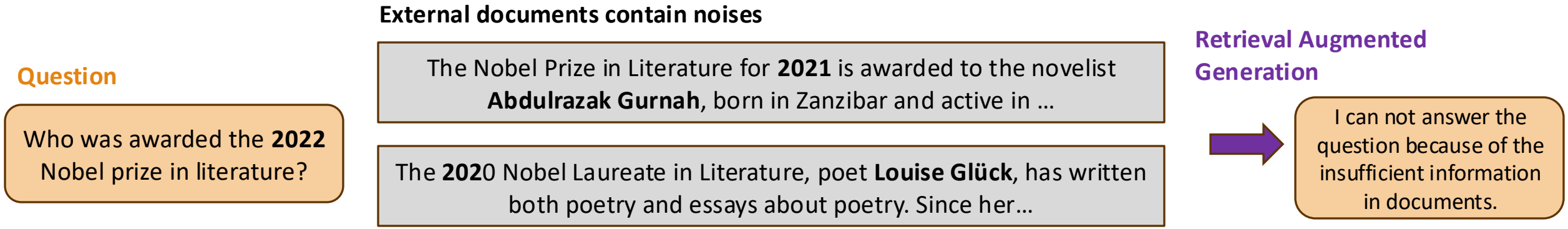
- LLMs must be able to extract the necessary information from documents despite there are noisy documents.



Capabilities that LLMs Should Have in RAG

Negative Rejection

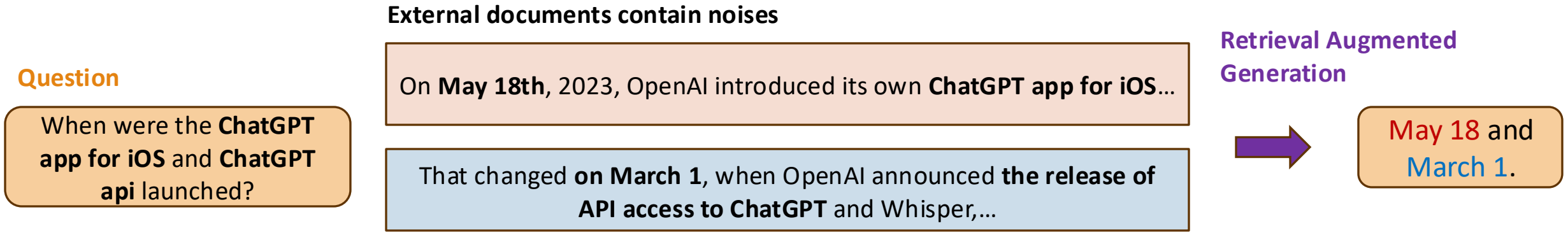
- In real-world situations, the search engine often fails to retrieve documents containing the answers.
- It is important for the model to have the capability to reject recognition and avoid generating misleading content.



Capabilities that LLMs Should Have in RAG

Information Integration

- In many cases, the answer to a question may be contained in multiple documents.
- To provide better answers to complex questions, it is necessary for LLMs to have the ability to integrate information.



Capabilities that LLMs Should Have in RAG

Counterfactual Robustness

- In the real world, there is an abundance of false information on the internet.
- LLMs should identify risks of known factual errors in the retrieved documents when the LLMs are given warnings about potential risks in the retrieved information through instruction.