IN5550: Question Answering and Retrieval Augmented Language Models

Lecture 12, part 1

Sondre Wold



Asking questions to machines and hoping for a reply

Question Answering (QA) is one of the more traditional tasks within NLP.

Asking questions to machines and hoping for a reply

Question Answering (QA) is one of the more traditional tasks within NLP. In this lecture we are going to look at:

Asking questions to machines and hoping for a reply

Question Answering (QA) is one of the more traditional tasks within NLP. In this lecture we are going to look at:

1: Question answering as a task;

Asking questions to machines and hoping for a reply

Question Answering (QA) is one of the more traditional tasks within NLP. In this lecture we are going to look at:

1: Question answering as a task;

2: How to combine language models with techniques from information retrieval in order to answer questions with language models.

Question answering

In the context of NLP, a question answering system is a computer program that can answer questions formulated in natural language.

Question answering

In the context of NLP, a question answering system is a computer program that can answer questions formulated in natural language. The specific task formulation has historically varied alot, as almost anything can be reformulated into a question-answer pair.

Question answering

In the context of NLP, a question answering system is a computer program that can answer questions formulated in natural language. The specific task formulation has historically varied alot, as almost anything can be reformulated into a question-answer pair. However, following Rogers et al., 2023, we can seperate QA into two main categories:

Question answering

In the context of NLP, a question answering system is a computer program that can answer questions formulated in natural language. The specific task formulation has historically varied alot, as almost anything can be reformulated into a question-answer pair. However, following Rogers et al., 2023, we can seperate QA into two main categories:

- Information-seeking:
  - You want to know something you do not already know
  - Google searches, StackOverflow questions, etc
  - Typically assumes there is no additional context
  - Open-ended answers (can be running text, lists, images, whatever)

Question answering

In the context of NLP, a question answering system is a computer program that can answer questions formulated in natural language. The specific task formulation has historically varied alot, as almost anything can be reformulated into a question-answer pair. However, following Rogers et al., 2023, we can seperate QA into two main categories:

- Information-seeking:
  - You want to know something you do not already know
  - Google searches, StackOverflow questions, etc
  - Typically assumes there is no additional context
  - Open-ended answers (can be running text, lists, images, whatever)
- Probing questions:
  - You want to test a machine on something you already know
  - Used to evaluate computer programs
  - Typically assumes some context, e.g a provided passage
  - Also referred to as "machine reading comprehension"
  - Often formulated in a multiple-choice format

Blurry lines

There are no clear-cut taxonomies, and the lines are blurry, e.g:

Which program at Notre Dame offers a Master of Education degree?

Blurry lines

There are no clear-cut taxonomies, and the lines are blurry, e.g:

Which program at Notre Dame offers a Master of Education degree?

...could appear both in information-seeking datasets like Natural Questions (Kwiatkowski et al., 2019) and typical LM evaluation datasets like SQuAD (Rajpurkar et al., 2016)

That being said...

There are ways of distinguishing between datasets based on the format as opposed to the "motivation" behind the questions.

Answer format

Given the context "Einstein was born in 1879", as per Rogers et al., 2023:

Format	**Question**	**Answer**
Extractive	When was Einstein born?	1879/Token 5
Multiple-choice	When was Einstein born?	a) 1899, **b)** 1879, c) 1900
Categorical	Was Einstein born in 1879?	Yes
Freeform	When was Einstein born?	1879 (generated)

Domains

It is also common to distinguish based on domain/scope:

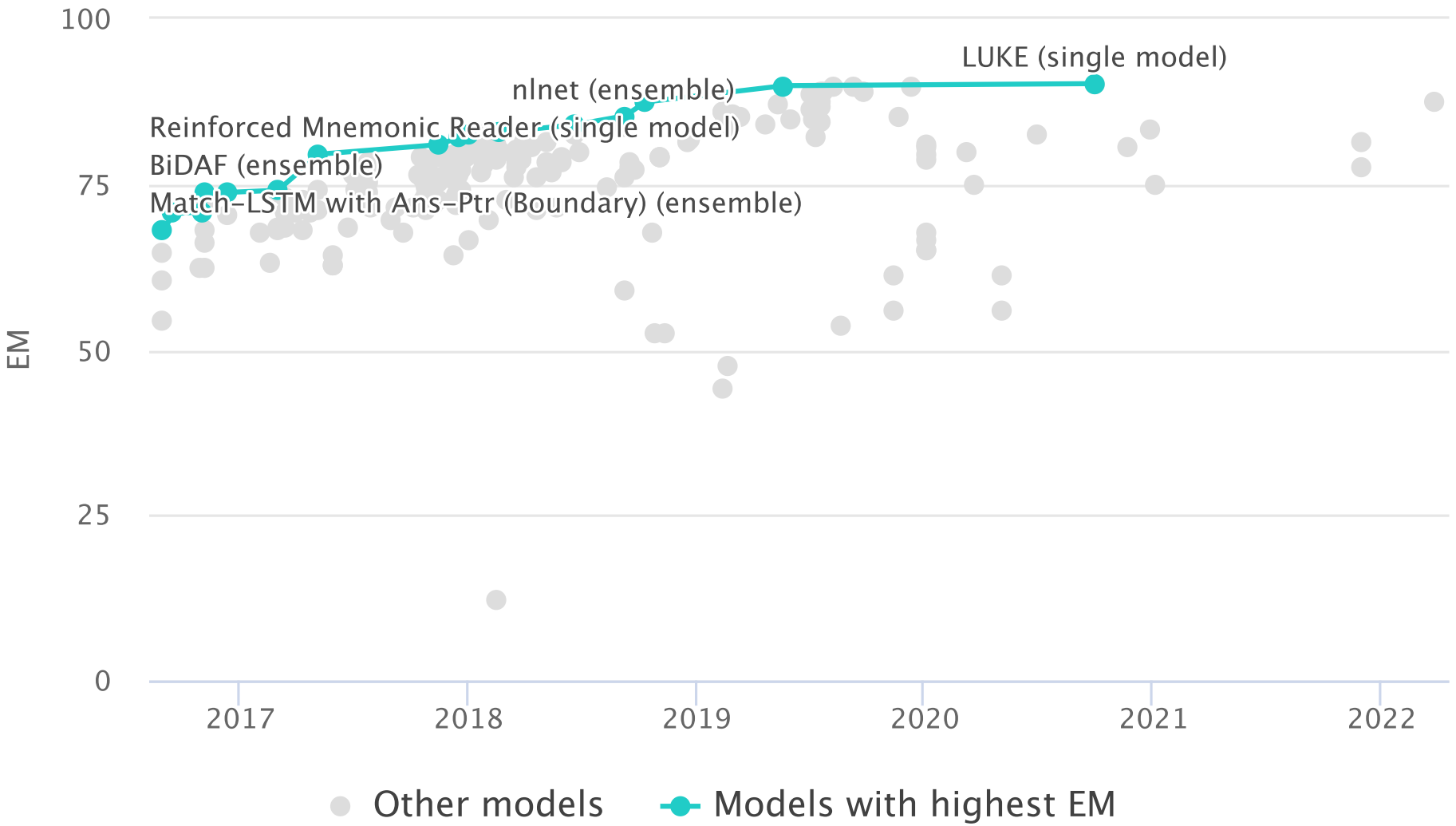
Scope	Example
Open-domain	The questions target a wide range of sources, and are not limited to one specific domain.
Closed-doman	The questions targets only one domain, e.g medicine.
Open-book	The model has access to information external to the question and any provided context, e.g the whole of wikipedia.
Closed-book	The model does not have access to anything outside the quesiton and has to rely solely on the provided information and any knowledge already stored in the system (e.g weights).

Example: SQuAD

One of the most widely used QA datasets is the Stanford Question Answering set (SQuAD) (Rajpurkar et al., 2016):

- 100 000+ questions about Wikipedia articles, written by crowdworkers.
- Probing, Extractive, open-domain, closed-book.
- Evaluated with Exact Match (EM) and F1
- Human performance is at 77% EM and 86.8% F1.

Performance over time



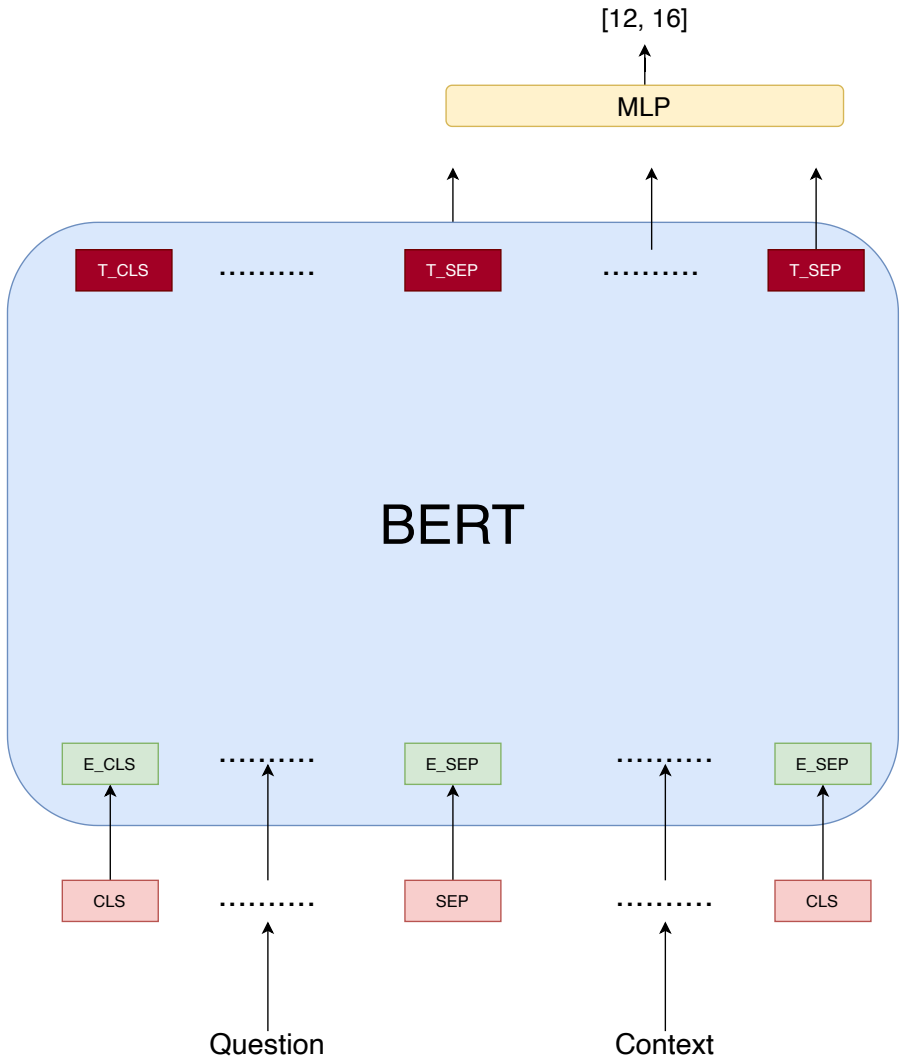
Example item

The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in **France**. *They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.*

Q: In what country is Normandy located? **A:** France (pos: 159)

How to solve this using an encoder-only model?

E.g using BERT (Devlin et al., 2018):



Other types of QA

QA is also a task where NLP meets other fields within ML.

Visual Question Answering, e.g Shin et al., 2016



Q: What is the color of the cat?
VQA: gray
→FSVQA: The color of the cat is gray.



Q: Does the man have a mustache?
VQA: no
→FSVQA: No, the man does not have a mustache.



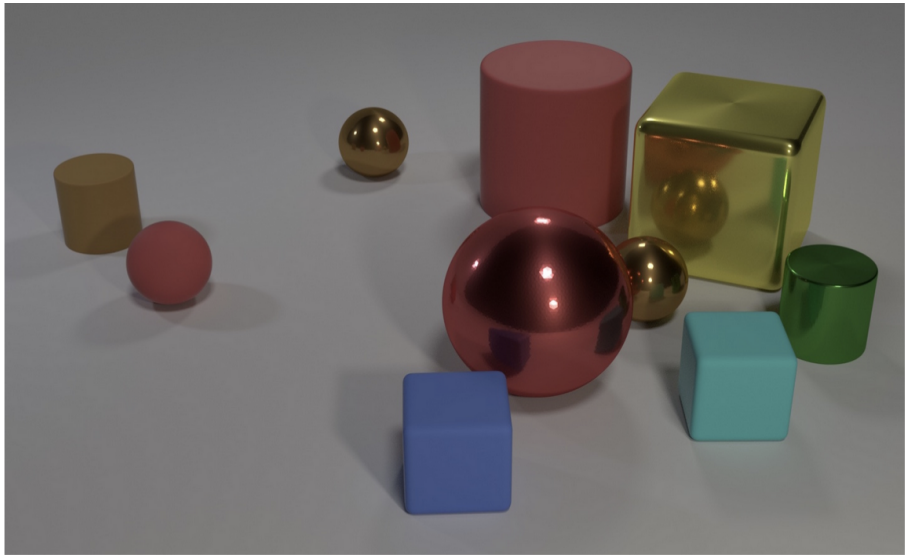
COCO: A hand is holding a group of scissors.
→ Q: What is a hand holding?
FSVQA: A hand is holding a group of scissors.



COCO: Two dogs are playing frisbee.
→ Q: How many dogs are playing frisbee?
FSVQA: Two dogs are playing frisbee.

VQA can be quite hard, e.g CLEVR Johnson et al., 2017

Questions in CLEVR test various aspects of visual reasoning including **attribute identification**, **counting**, **comparison**, **spatial relationships**, and **logical operations**.



Q: Are there an **equal number** of **large things** and **metal spheres**?

Q: **What size** is the **cylinder** **that is left of** the **brown metal** thing **that is left of** the **big sphere**?

Q: There is a **sphere** with the **same size as** the **metal cube**; is it **made of the same material as** the **small red sphere**?

Q: **How many** objects are **either** **small cylinders** or **red** things?

Back to NLP

Two core problems related to LLMs and QA:

Back to NLP

Two core problems related to LLMs and QA:

- Models store knowledge implicitly in a huge parameter space

Back to NLP

Two core problems related to LLMs and QA:

- Models store knowledge implicitly in a huge parameter space
- Models are expensive to update

Retrieval

One way to mitigate these problems is through *retrieval*.

Retrieval

One way to mitigate these problems is through *retrieval*.

- Models can *learn* to leverage information from external knowledge sources:

$$P(y|x,\phi) \rightarrow P(y|x,\phi,\theta,Z),$$

where θ is a retriever model and Z is some non-parametric information store.

Retrieval

One way to mitigate these problems is through *retrieval*.

- Models can *learn* to leverage information from external knowledge sources:

$$P(y|x, \phi) \rightarrow P(y|x, \phi, \theta, Z),$$

  where θ is a retriever model and Z is some non-parametric information store.

- We can leverage techniques from the field of *information retrieval* and combine them with NLP

Retrieval

One way to mitigate these problems is through *retrieval*.

- Models can *learn* to leverage information from external knowledge sources:

$$P(y|x, \phi) \rightarrow P(y|x, \phi, \theta, Z),$$

  where θ is a retriever model and Z is some non-parametric information store.
- We can leverage techniques from the field of *information retrieval* and combine them with NLP
- Started out as a way of improving QA, but is currently extremly popular due to the increased usage of LLMs in industry.

Reformulated QA pipeline

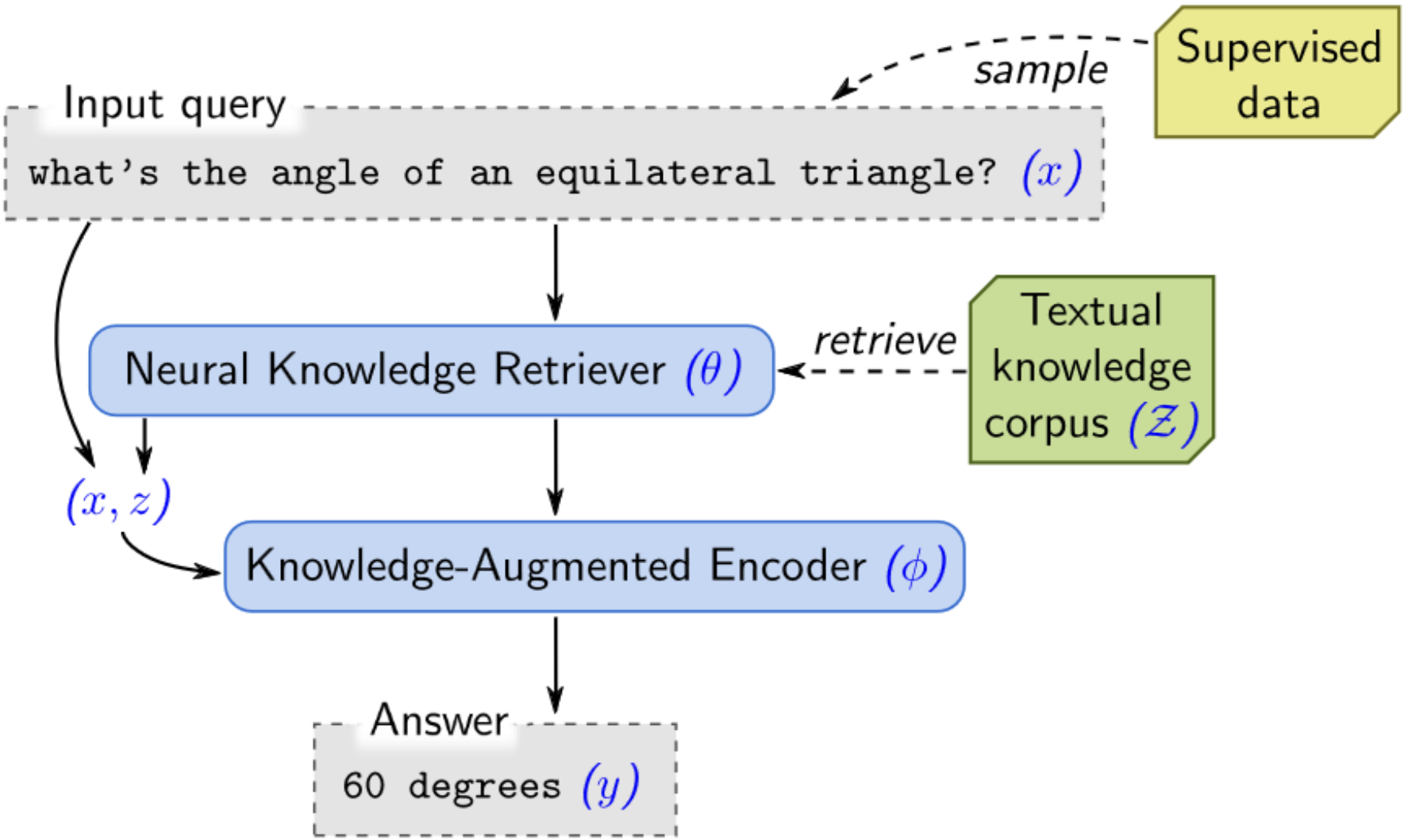


Figure from Guu et al., 2020

REALM (Guu et al., 2020)

You can also add a retriever to the language modeling task!

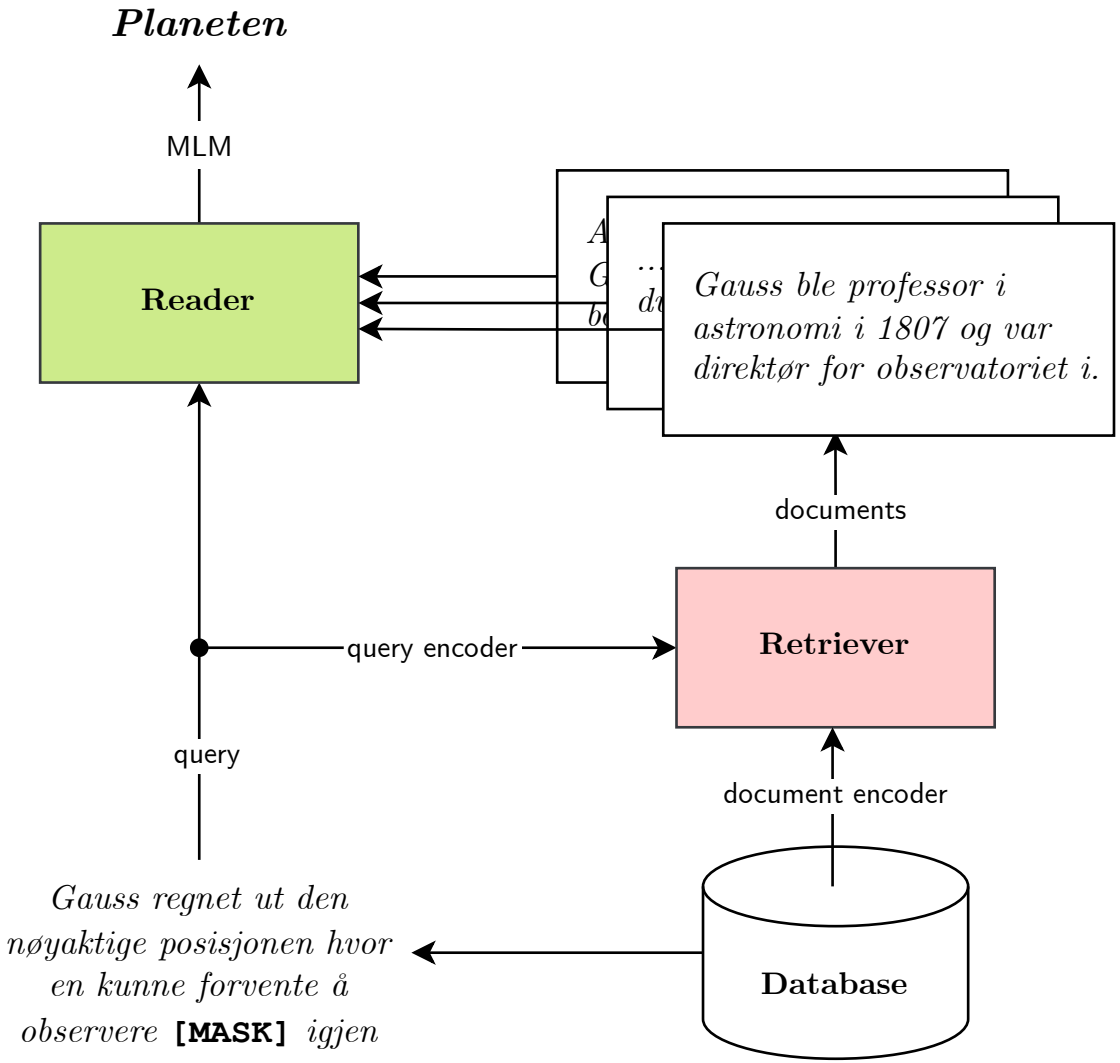


Figure from Charpentier et al., 2023

REALM contd.

Three components:

REALM contd.

Three components:

- A knowledge database, Z, such as Wikipedia.

REALM contd.

Three components:

- A knowledge database, Z, such as Wikipedia.
- A knowledge retriever:

$$f(x, z) = Embed_{input}(x)^\top Embed_{doc}(z),$$

where *Embed* is a text encoder, like BERT, x is a text query and $z \in Z$. This is calculated for all $z \in Z$ and the same x, and the highest product is selected as the retrieved document (MIPS).

REALM contd.

Three components:

- A knowledge database, Z, such as Wikipedia.
- A knowledge retriever:

$$f(x, z) = Embed_{input}(x)^\top Embed_{doc}(z),$$

  where *Embed* is a text encoder, like BERT, x is a text query and $z \in Z$. This is calculated for all $z \in Z$ and the same x, and the highest product is selected as the retrieved document (MIPS).
- A knowledge-augmented encoder:

$$P(y|z, x)$$

Retrieval Augmented Generation

The same principle can be applied to generative language models, nicknamed RAG (Lewis et al., 2020).

Retrieval Augmented Generation

The same principle can be applied to generative language models, nicknamed RAG (Lewis et al., 2020). The knowledge-augmented encoder from REALM is now a knowledge-augmented generator:

$$P_\theta(y_i|x, z, y_{1:i-1})$$

Retrieval Augmented Generation

The same principle can be applied to generative language models, nicknamed RAG (Lewis et al., 2020). The knowledge-augmented encoder from REALM is now a knowledge-augmented generator:

$$P_\theta(y_i|x, z, y_{1:i-1})$$

The retriever can be anything:

- Sparse
- Dense

RETRO

You can also pretrain a RAG model from scratch, as was done in Borgeaud et al., 2022:

> *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.*

RAG anno 2024

