PES
UNIVERSITY

ALGORITHMS FOR INFORMATION RETRIEVAL AND INTELLIGENT WEB

Question Answering, Neural Models for IR

Question Answering

- Question Answering is a computer science discipline within the fields of information retrieval and natural language processing, which focuses on building systems that automatically answer questions posed by humans in a natural language.

- Question answering systems are designed to meet human information needs in contexts like virtual assistants, chatbots, search engines, or database queries. They primarily target factoid questions, seeking concise factual answers.
  Eg: Where is the Louvre Museum located?

Question Answering

PES UNIVERSITY

- By the early 1960s, systems used the two major paradigms of question answering—information retrieval-based and knowledge-based—to answer questions about baseball statistics or scientific facts.

- We will learn the two major paradigms for factoid question answering. Information-retrieval (IR) based QA, sometimes called open domain QA, relies on the vast amount of text on the web or in collections of scientific papers like PubMed.

Question Answering

PES UNIVERSITY

- QA systems can be based on various techniques, including information retrieval, knowledge-based, generative, and rule-based approaches. Each method has its strengths and weaknesses, and the choice of method depends on the project's specific needs.

- QA systems can be used in many places, like customer service, search engines, healthcare, education, finance, e-commerce, voice assistants, chatbots, and virtual assistants.

Question Answering - Typical flow

A natural language question-answering (QA) system is a computer program that automatically answers questions using NLP. The basic process of a natural language QA system includes the following steps:

1. **Text pre-processing**: The question is pre-processed to remove irrelevant information and standardise the text's format. This step includes tokenization, lemmatization, and stop-word removal, among others.

2. **Question understanding**: The pre-processed question is analysed to extract the relevant entities and concepts and to identify the type of question being asked. This step can be done using natural language processing (NLP) techniques such as named entity recognition, dependency parsing, and part-of-speech tagging.

Question Answering

3. Information retrieval: The question is used to search a database or corpus of text to retrieve the most relevant information. This can be done using information retrieval techniques such as keyword search or semantic search.

4. Answer generation: The retrieved information is analysed to extract the specific answer to the question. This can be done using various techniques, such as machine learning algorithms, rule-based systems, or a combination.

5. Ranking: The extracted answers are ranked based on relevance and confidence score.

Question Answering

PES UNIVERSITY

- Given a user question, information retrieval is used to find relevant passages. Then neural reading comprehension algorithms read these retrieved passages and draw an answer directly from spans of text.

- In the second paradigm, knowledge-based question answering, a system instead builds a semantic representation of the query, such as mapping
  What states border Texas? to the logical representation: λx.state(x) ∧borders(x,texas), or
  When was Ada Lovelace born? to the gapped relation: birth-year (Ada Lovelace, ?x).

  These meaning representations are then used to query databases of facts.

Question Answering - Types

PES UNIVERSITY

Information retrieval-based QA

This approach uses information retrieval techniques, such as keyword or semantic search, to identify the documents or passages most likely to hold the answer to a given question.

Knowledge-based QA

Knowledge-based question answering (QA) automatically answers questions using a knowledge base, such as a database or ontology, to retrieve the relevant information.

Question Answering – Types

Generative QA

Generative question answering (QA) automatically answers questions using a generative model, such as a neural network, to generate a natural language answer to a given question.

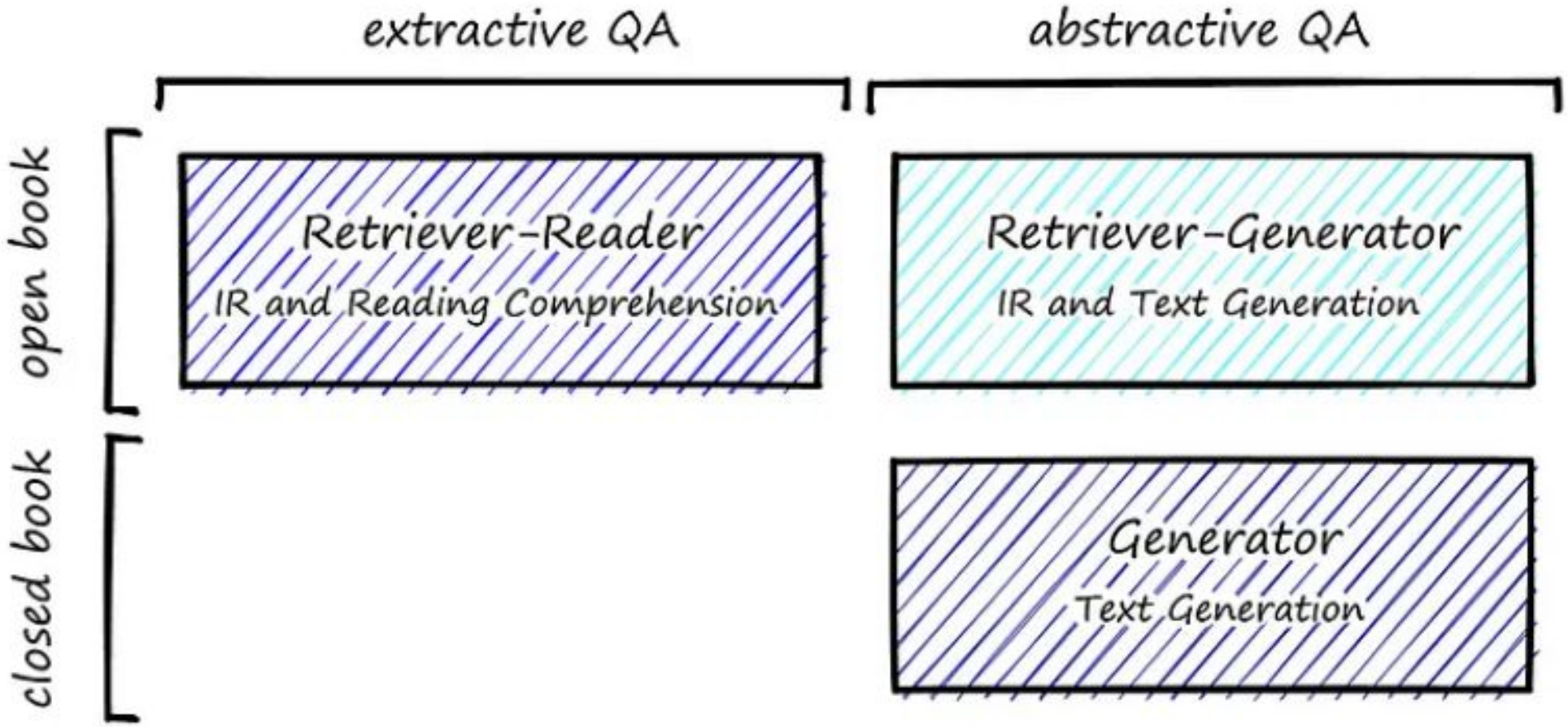
Hybrid QA

Hybrid question answering (QA) automatically answers questions by combining multiple QA approaches, such as information retrieval-based, knowledge-based, and generative QA.

Rule-based QA

Rule-based question answering (QA) automatically answers questions using a predefined set of rules based on keywords or patterns in the question.

Question Answering - Variants based on Answer creation methods



There are a few approaches to question answering (QA).

Question Answering

1. **Extractive QA**: In this approach, the model directly retrieves the answer from the provided context and presents it to the user. This type of QA is commonly solved using BERT-like models.
2. **Generative QA**: Contrasting with extractive QA, generative QA involves the model generating free-form text as the answer, directly based on the given context. This method relies on Text Generation models.

QA systems differ in the **source of their answers:**

1. **Open QA**: Here, the answer is extracted from a context, similar to extractive QA.
2. **Closed QA**: In contrast, closed QA does not receive any context for answering. The model generates the entire answer autonomously without external context.

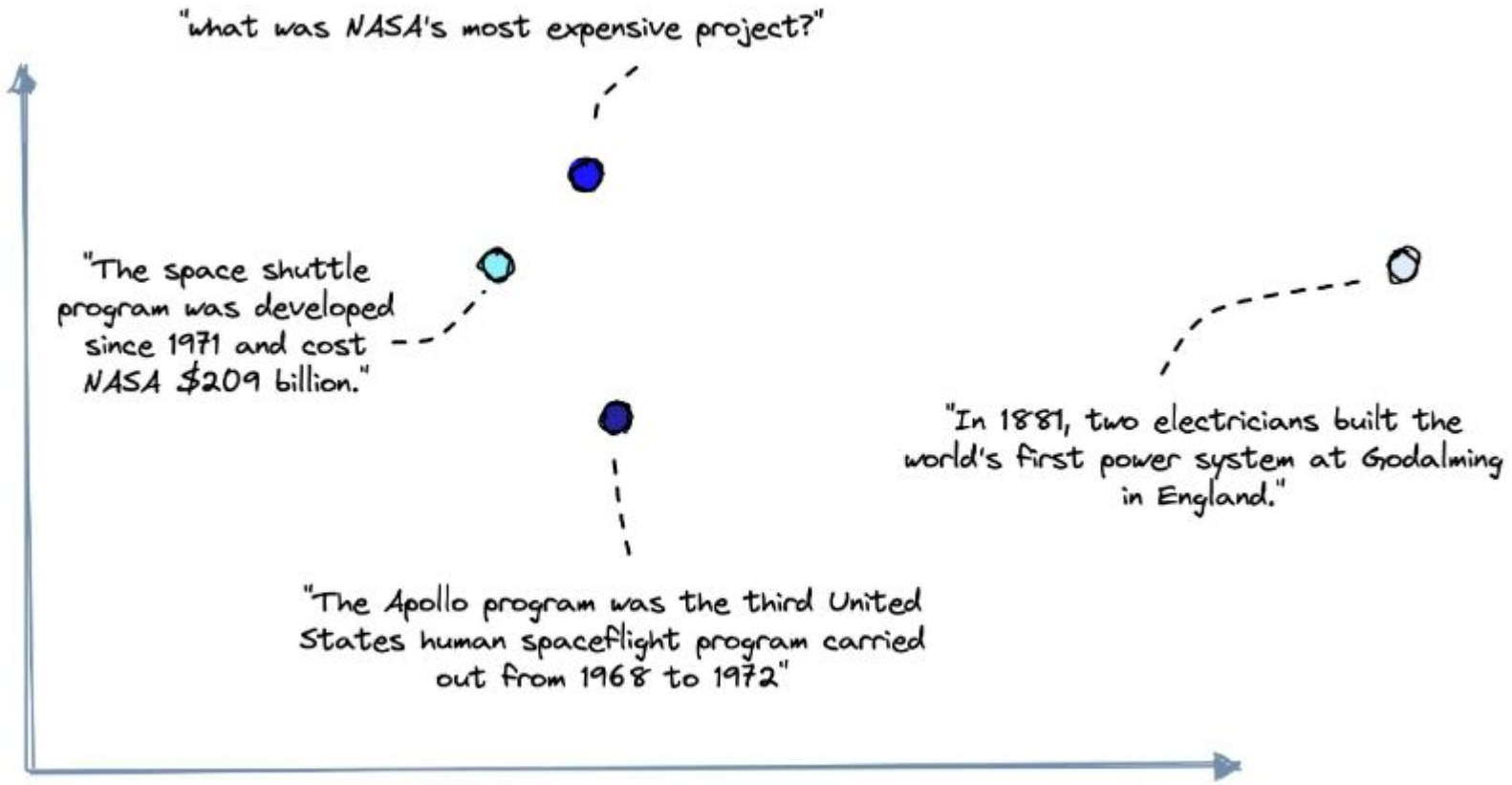
Question Answering - Components

Indexed data (document store/vector database) i.e. External Knowledge

Indexed data storage is a way to organize documents so that they can be quickly retrieved when a question is asked. This is done by creating indexes of the documents, which are essentially lists of the words and phrases that appear in the documents. When a question is asked, the model can use the indexes to quickly find documents that are likely to contain the answer to the question.Some popular indexed data storage systems include Elasticsearch, Solr, and MongoDB.

A **vector database** is a type of document store that stores documents as vectors. The vectors for different words and phrases are similar to each other if the words and phrases have similar meanings.Some popular vector databases include Pincone, Chrome, and Wiavate.

Question Answering - Components



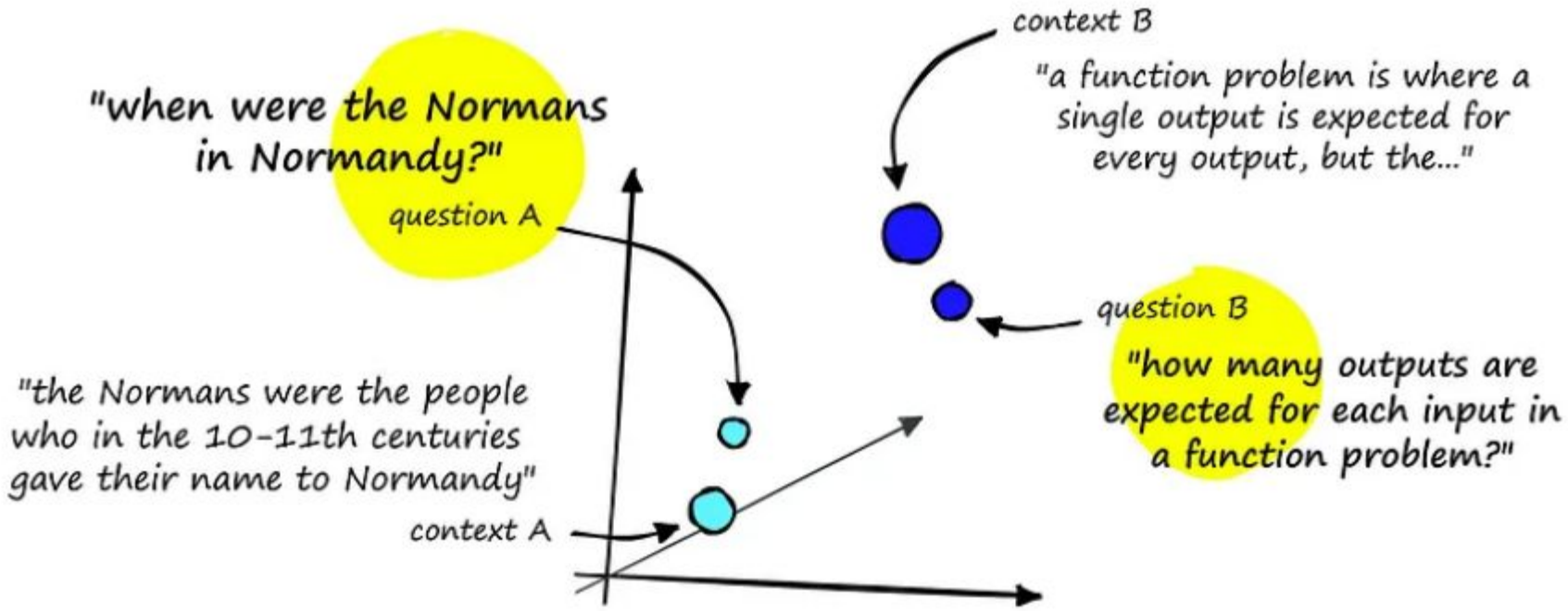
Queries and contexts with similar meaning are embedding into a similar vector space.

Question Answering - Components

PES UNIVERSITY

Retriever

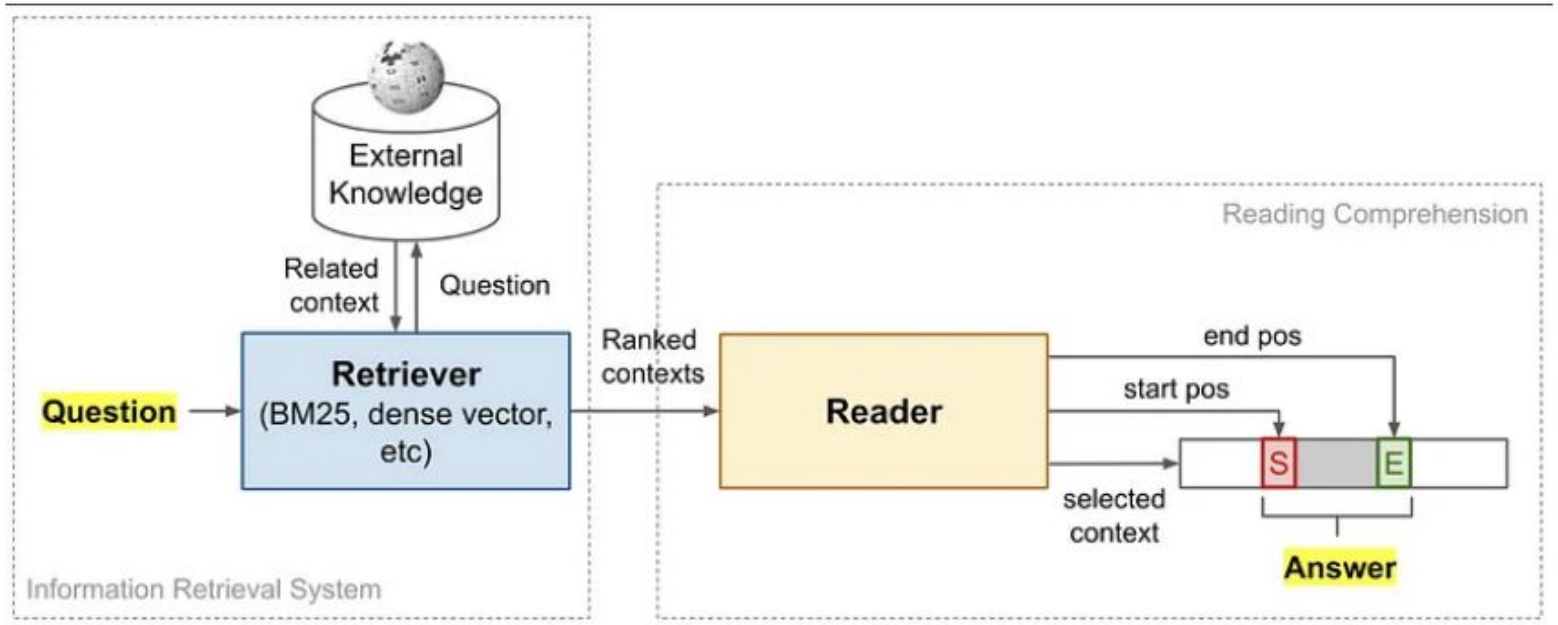
A retriever is responsible for retrieving the most relevant documents from a knowledge base that are likely to contain the answer to a given question i.e. it retrieves a set of contexts for our QA model to extract answers.



Question Answering

Reader

A reader is responsible for extracting the answer from the retrieved documents/set of Contexts. Using the Reader's model alone solves the reading comprehension (RC) QA task, but if it is used with the Retriever model, we have the Open-Domain QA model.

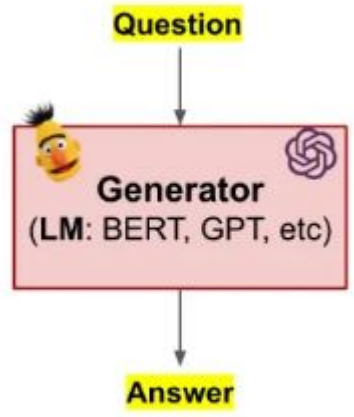


Question Answering - Components

PES UNIVERSITY

Generator

Generator does not extract the answer to a question from a context. Instead, it generates text directly to answer the question. This is done by using a language model, such as GPT-3. The language model is trained on a large dataset of text, and it can be used to generate text that is relevant to the question.The context can be optionally provided to the model. If the context is provided, the model will use the context to generate a more informative and accurate answer. However, the model can also generate an answer without the context.

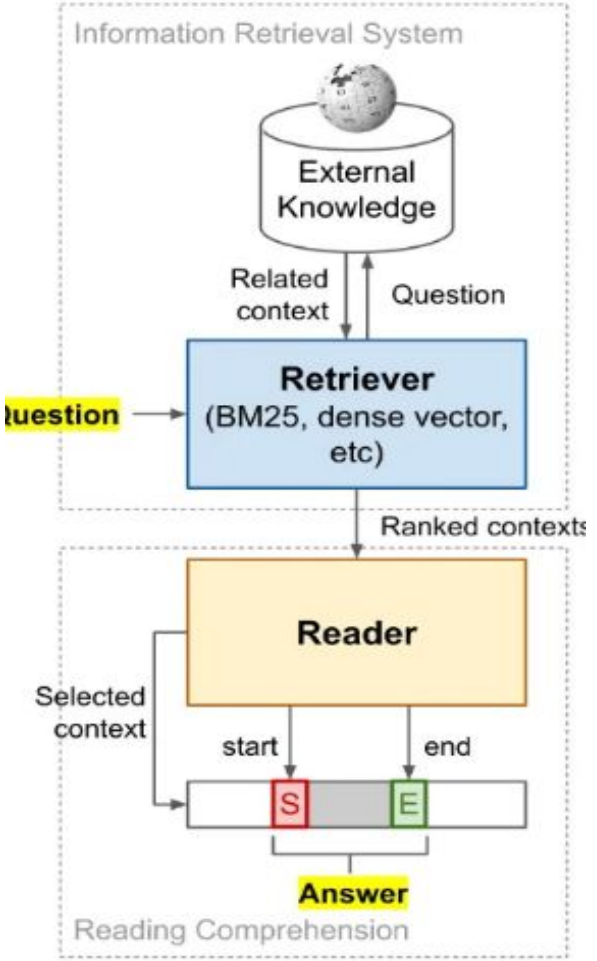


Generator

Question Answering - Combining Components to form variants

Retriever — Reader (Open-book Extractive QA)

The retriever takes the question and context, which is indexed in the database. It creates a query vector(vector representation) that is compared against all indexed context vectors in the database and returns the (n) most similar contexts. The reader model then takes in the most similar contexts and the question to provide a span selection of the answer.

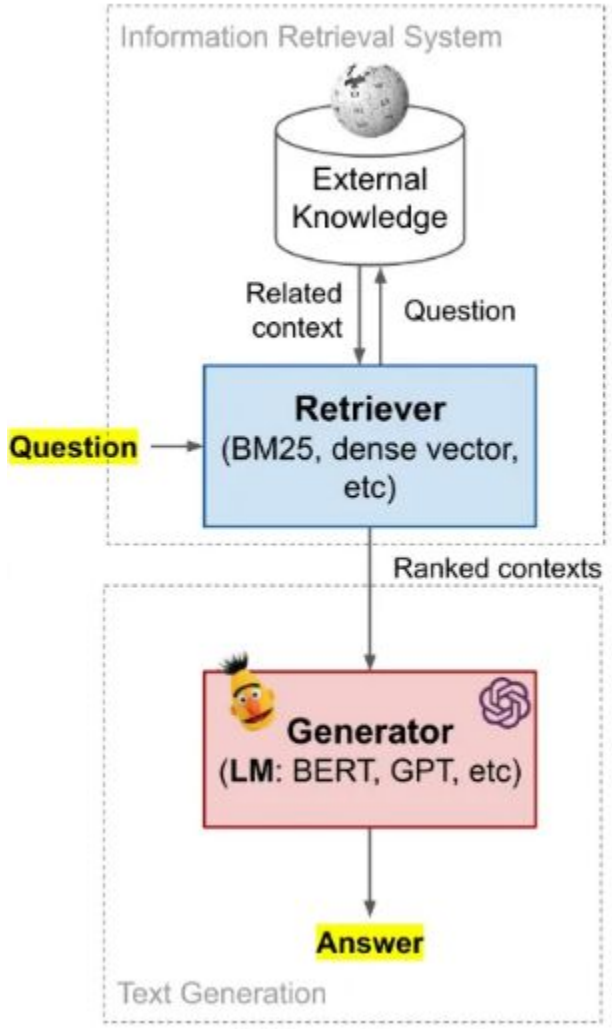


Retriever-Reader

Question Answering

Retriever — Generator (Open-book Abstractive QA)

In this system, reader model is replaced with a generator model. The generator model utilizes the question, along with the provided contexts and knowledge, to produce answers. This approach is referred to as abstractive question-answering and is termed “open-book” because the data source is external.The system is employed to address more conceptual questions that do not demand precise responses. Consequently, the answers it provides are more in the form of suggestions or opinions rather than direct, verbatim answers.

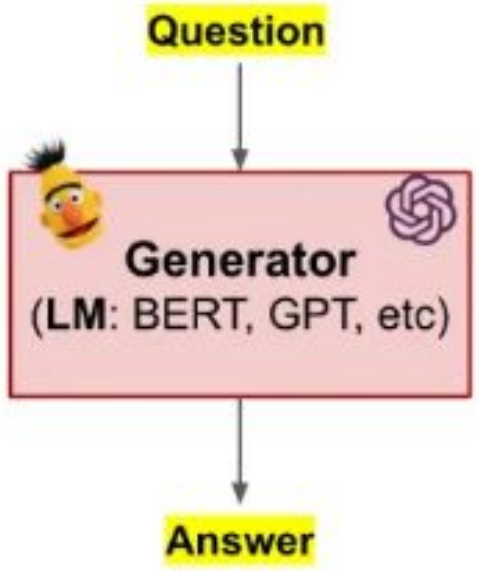


Retriever-Generator

Question Answering

Generator (Closed-book Abstractive QA)

This architecture is known as closed-book abstractive QA since it relies solely on the model's internal knowledge to generate answers. The retriever and reader models are removed, and the generative model handles the answers.



Generator

References

"Speech and Language Processing", Third Edition, Daniel Jurafsky, James H.Martin, Chapter 14: Question Answering and Information Retrieval, 2023.

https://iprathore71.medium.com/q-a-model-76957da40e07

https://spotintelligence.com/2023/01/20/question-answering-qa-system-nlp/

PES UNIVERSITY

ALGORITHMS FOR INFORMATION RETRIEVAL AND INTELLIGENT WEB

PES UNIVERSITY

Neural Models for IR, QA as an Information Retrieval task - Factoid QA - 1

Neural models for IR

The idea is that huge pretrained language models have read a lot of facts in their pre-training data, presumably including the location of the Louvre, and have encoded this information in their parameters.

For some general factoid questions this can be a useful approach and is used in practice. But prompting a large language model is not yet a solution for question answering.

One way to employ large pretrained language models, utilizing their exposure to extensive training data. The approach involves prompting these models with questions and relying on their ability to generate responses based on their encoded knowledge.

Neural models for IR

PES UNIVERSITY

However, a significant challenge arises as large language models often produce incorrect or "hallucinated" answers. Hallucinations refer to responses that deviate from factual information available in the real world. This limitation highlights that while this approach can be useful for certain factoid questions, it is not a foolproof solution.

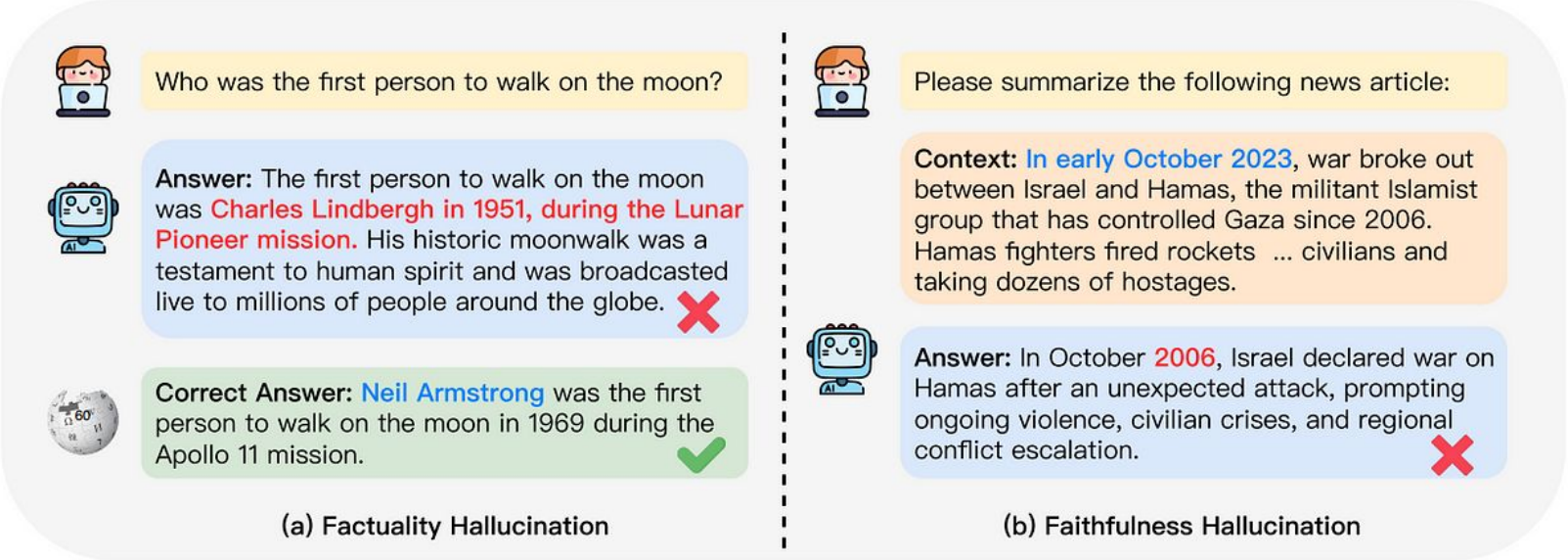


Figure 1: An intuitive example of LLM hallucination.

Neural models for IR

- The recognition of inaccuracies in generated responses calls for further refinement in the training and fine-tuning processes of question answering systems.

- One issue with language models is the lack of well-calibrated confidence estimates in their answers. In a calibrated system, the confidence of the system in the correctness of its answer is closely linked to the probability of that answer being correct. Ideally, if the system is unsure or incorrect, it should express hesitation or suggest verifying information from another source.

Neural models for IR

- However, language models often lack this calibration, leading to situations where they provide very wrong answers with a high degree of certainty.

- A second problem is that simply prompting a large language model doesn't allow us to ask questions about proprietary data, ie, data that is owned and controlled by an individual, an organization or a group. A common use of question-answering is to query private data, like asking an assistant about our email or private documents, or asking a question about our own medical records.

Neural models for IR

- internal datasets, or even the web itself, can be especially useful for rapidly changing or dynamic information; by contrast, large language models are often only released at long increments of many months and so may not have up-to-date information.
- Due to these downfalls, the retriever/reader model is the current dominated solution.
- In a retriever/reader model, information retrieval techniques are employed initially to retrieve documents that are likely to contain relevant information for answering a given question. Subsequently, the model may either extract an answer from specific spans of text within these documents or utilize large language models to generate an answer, a process sometimes referred to as retrieval-augmented generation.

Neural models for IR

- This approach addresses some of the challenges associated with using simple prompting methods for question answering. By basing answers on retrieved documents, it ensures that the response is grounded in facts sourced from a curated dataset. Additionally, the answer is provided in conjunction with the context of the passage or document from which it was extracted.

- Utilizing retrieval techniques offers several advantages. It enhances user confidence in the accuracy of the answer by providing context, allowing users to assess the reliability of the information. Moreover, this methodology can be applied to proprietary datasets, enabling the system to retrieve information from specialized domains such as legal or medical data for specific applications.

IR - with Dense Vectors

- The classic tf-idf or BM25 algorithms for IR have long been known to have a conceptual flaw: they work only if there is exact overlap of words between the query and document. In other words, the user posing a query (or asking a question) needs to guess exactly what words the writer of the answer might have used, an issue called the vocabulary mismatch problem.

- The solution to this problem is to use an approach that can handle synonymy: instead of (sparse) word-count vectors, using (dense) embeddings. This idea is implemented in modern times via encoders like BERT.

IR - with Dense Vectors

PES UNIVERSITY

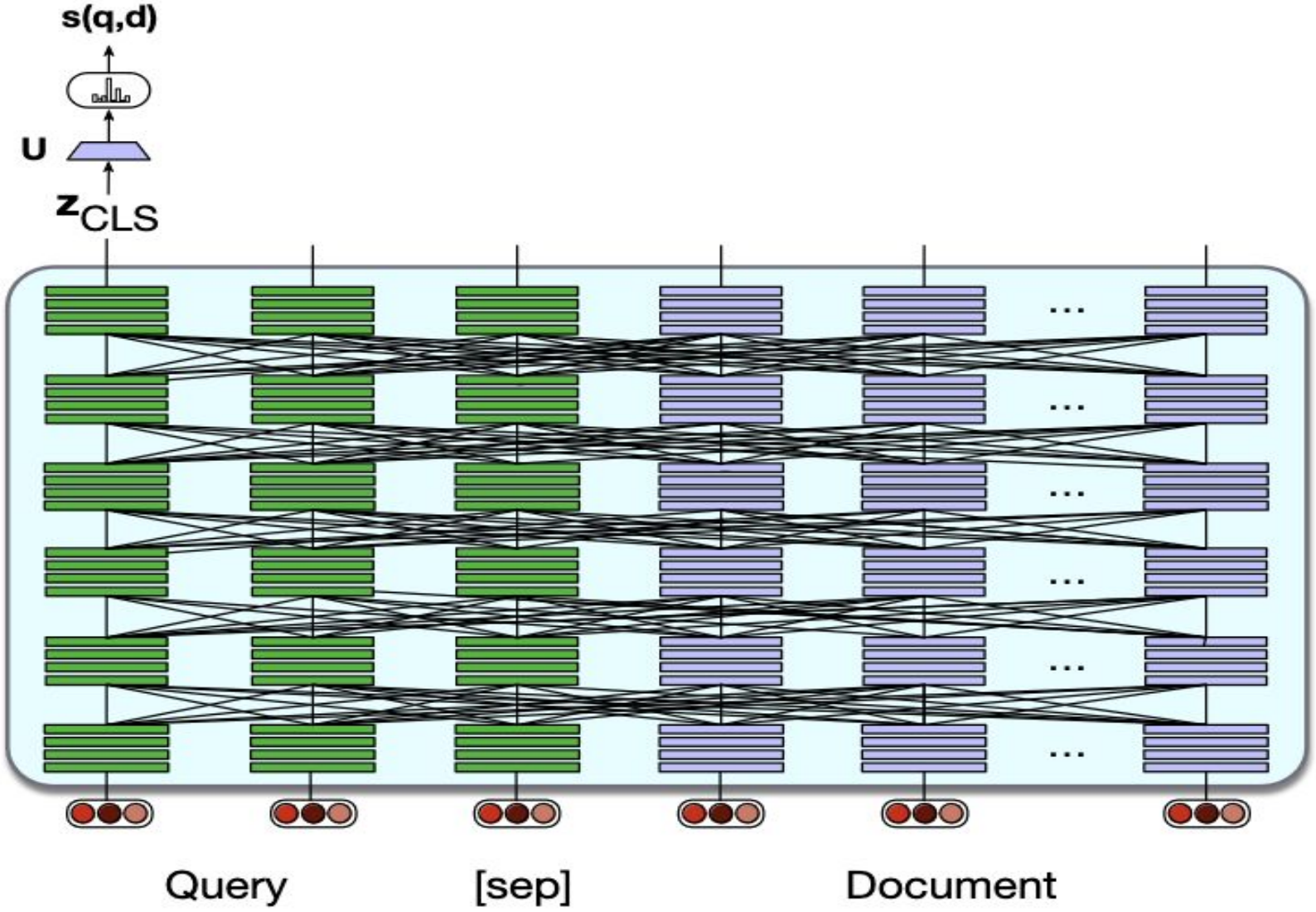
- The most powerful approach is to present both the query and the document to a single encoder. This enables the transformer's self-attention mechanism to see all the tokens of both the query and the document, and thus building a representation that is sensitive to the meanings of both query and document. Then a linear layer can be put on top of the [CLS](classification) token to predict a similarity score for the query/document tuple:

$$\mathbf{z} = \text{BERT}(\text{q}; [\text{SEP}]; \text{d})[\text{CLS}]$$

$$\text{score}(q, d) = \text{softmax}(\mathbf{U}(\mathbf{z}))$$

IR - with Dense Vectors

- This architecture is shown in the diagram.

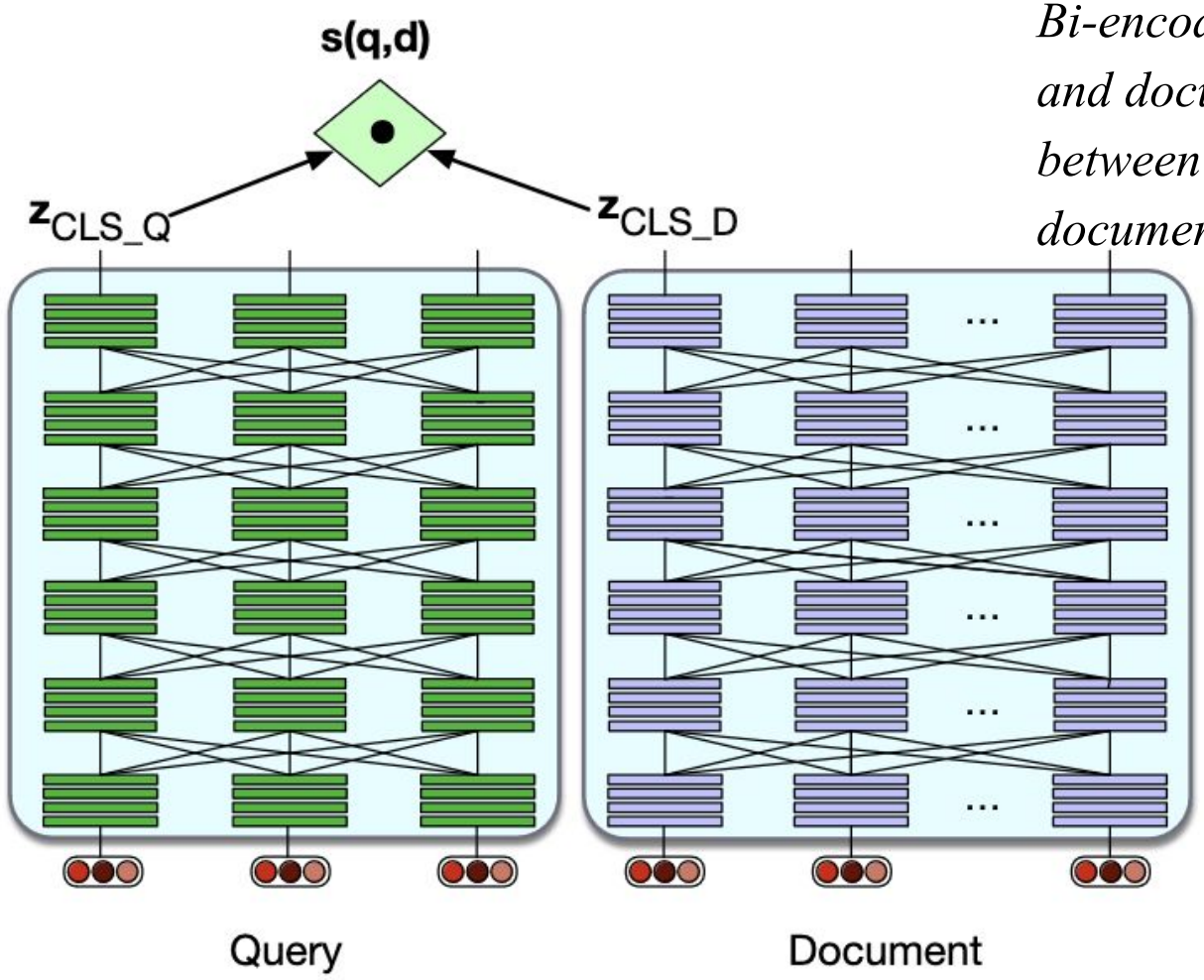


Query/ Document encoder: Use a single encoder to jointly encode query and document and finetune to produce a relevance score with a linear layer over the CLS token

IR - with Dense Vectors

- Here the retrieval step is not done on an entire document. Instead documents are broken up into smaller passages, such as non-overlapping fixed-length chunks of say 100 tokens, and the retriever encodes and retrieves these passages rather than entire documents
- The query and document have to be made to fit in the BERT 512-token window, for example by truncating the query to 64 tokens and truncating the document if necessary so that it, the query, [CLS], and [SEP](separator) fit in 512 tokens.
- The BERT system together with the linear layer U can then be fine-tuned for the relevance task by gathering a tuning dataset of relevant and non-relevant passages.
- The problem with this architecture is the expense in computation and time. With this architecture, every time we get a query, we have to pass every single document in our entire collection through a BERT encoder jointly with the new query. This enormous use of resources is impractical for real cases.

IR - with Dense Vectors



Bi-encoder: Use separate encoders for query and document, and use the dot product between CLS token outputs for the query and document as the score.

- At the opposite end of the computational spectrum is a more efficient architecture known as the bi-encoder. In this architecture we can encode the documents in the collection only one time by using two separate encoder models, one to encode the query and one to encode the document.

IR - with Dense Vectors

- We encode each document, and store all the encoded document vectors in advance. When a query comes in, we encode just this query and then use the dot product between the query vector and the precomputed document vectors as the score for each candidate document.
- For example, if we used BERT, we would have two encoders BERT_Q and BERT_D and we could represent the query and document as the [CLS] token of the respective encoders:

$$\mathbf{z}_q = \text{BERT}_Q(\text{q})\,[\text{CLS}]$$
$$\mathbf{z}_d = \text{BERT}_D(\text{d})\,[\text{CLS}]$$
$$\text{score}(q,d) = \mathbf{z}_q \cdot \mathbf{z}_d$$

- The bi-encoder is much cheaper than a full query/document encoder, but is also less accurate, since its relevance decision can't take full advantage of all the possible meaning interactions between all the tokens in the query and the tokens in the document.

QA as an Information Retrieval task - Factoid QA

PES UNIVERSITY

- The goal of retrieval-based Question Answering (sometimes referred to as open-domain QA) is to answer a user's question by either identifying short text segments from the web or another extensive collection of documents or by generating an answer based on the information retrieved from these sources.
- Some examples of Factoid questions are:

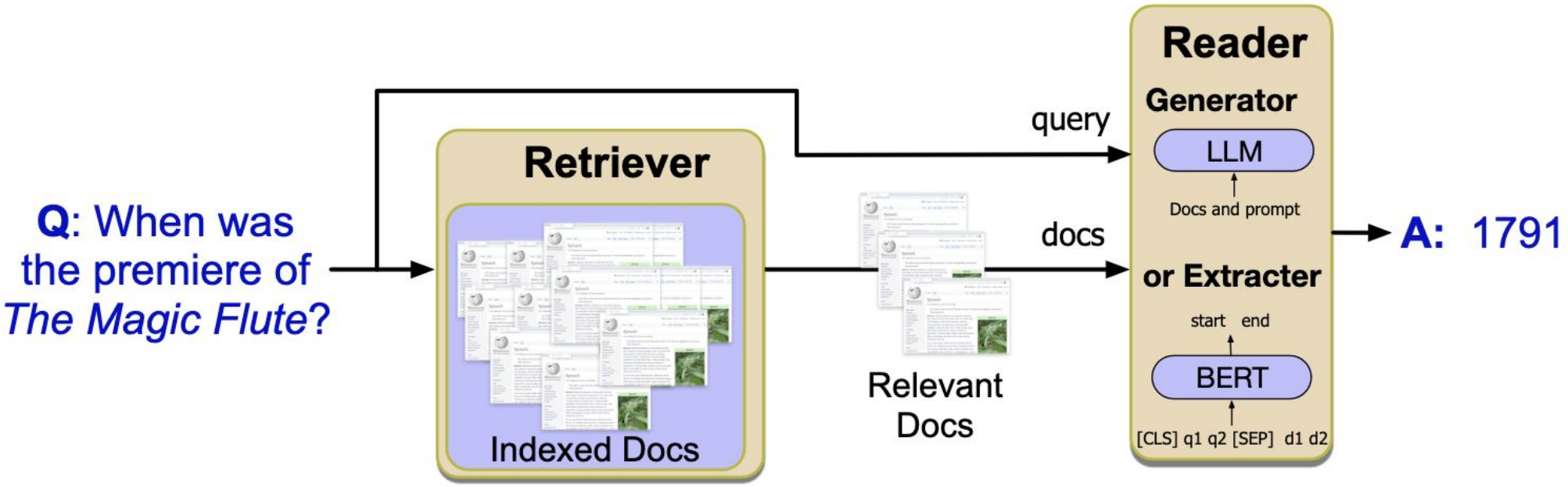
Question	Answer
Where is the Louvre Museum located?	in Paris, France
What are the names of Odin's ravens?	Huginn and Muninn
What kind of nuts are used in marzipan?	almonds
What instrument did Max Roach play?	drums
What's the official language of Algeria?	Arabic

QA as an Information Retrieval task - Factoid QA

PES UNIVERSITY

- The dominant paradigm for retrieval-based QA is sometimes called the retrieve and read model. In the first stage of this 2-stage model we retrieve relevant passages from a text collection, for example using the dense retrievers. The second stage, called the reader, is commonly implemented as either an extractor or a generator.
- The first method is span extraction, using a neural reading comprehension algorithm that passes over each passage and is trained to find spans of text that answer the question. The second method is also known as retrieval-augmented generation: we take a large pretrained language model, give it some set of retrieved passages and other text as its prompt, and autoregressively generate a new answer token by token.

QA as an Information Retrieval task - Factoid QA



QA as an Information Retrieval task - Factoid QA

Beyoncé Giselle Knowles-Carter (born September 4, 1981) is an American singer, songwriter, record producer and actress. Born and raised in Houston, Texas, she performed in various singing and dancing competitions as a child, and rose to fame in the late 1990s as lead singer of R&B girl-group Destiny's Child. Managed by her father, Mathew Knowles, the group became one of the world's best-selling girl groups of all time. Their hiatus saw the release of Beyoncé's debut album, Dangerously in Love (2003), which established her as a solo artist worldwide, earned five Grammy Awards and featured the Billboard Hot 100 number-one singles "Crazy in Love" and "Baby Boy".
Q: "In what city and state did Beyoncé grow up?" A: "Houston, Texas"
Q: "What areas did Beyoncé compete in when she was growing up?" A: "singing and dancing"
Q: "When did Beyoncé release Dangerously in Love?" A: "2003"

Fig: A passage from the SQuAD 2.0 dataset with 3 sample questions and the labeled answer spans.

ALGORITHMS FOR INFORMATION RETRIEVAL AND INTELLIGENT WEB

PES UNIVERSITY

Neural Models for IR, QA as an Information Retrieval task - Factoid QA - 2

QA as an Information Retrieval task - Factoid QA

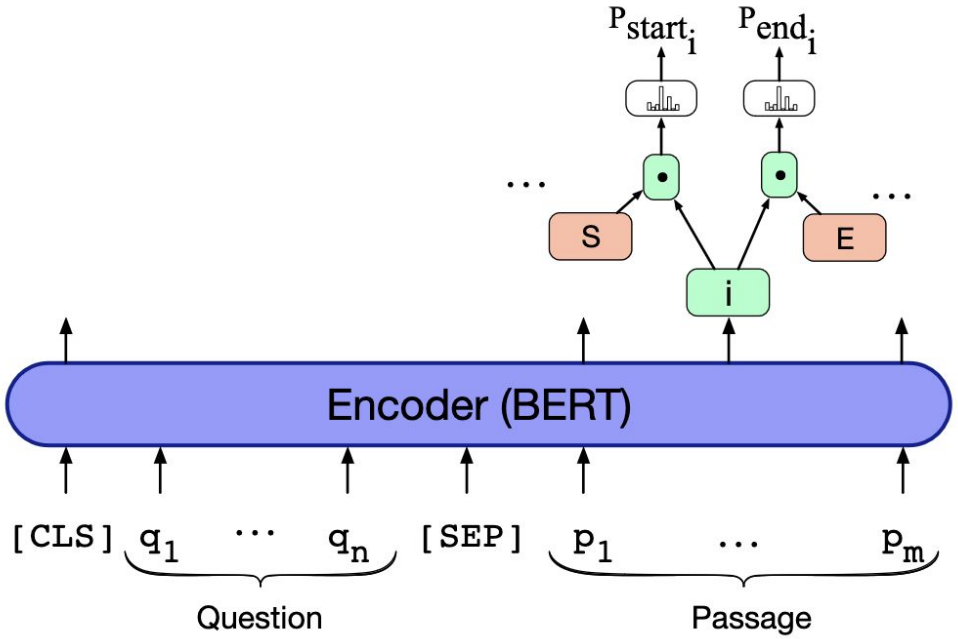
Reader Algorithms : 1. **Answer Span Extraction**

- The job of the reader is to take a passage as input and produce the answer. Here we introduce the span extraction style of reader, in which the answer is a span of text in the passage. For example given a question like “How tall is Mt. Everest?” and a passage that contains the clause Reaching 29,029 feet at its summit, a reader will output 29,029 feet.

- The answer extraction task is commonly modeled by span labeling: identifying in the passage a span (a continuous string of text) that constitutes an answer. Neuralspan algorithms for reading comprehension are given a question q of n tokens q1, ..., qn and a passage p of m tokens p1, ..., pm. Their goal is thus to compute the probability P(a|q, p) that each possible span a is the answer.

QA as an Information Retrieval task - Factoid QA

- If each span a starts at position as and ends at position a_e, we make the simplifying assumption that this probability can be estimated as $P(a|q, p) = P_{start}(a_s|q, p)P_{end}(a_e|q, p)$.
  Thus for for each token p_i in the passage we'll compute two probabilities: $p_{start}(i)$ that p_i is the start of the answer span, and $p_{end}(i)$ that pi is the end of the answer



QA as an Information Retrieval task - Factoid QA

- A standard baseline algorithm for reading comprehension is to pass the question and passage to any encoder like BERT, as strings separated with a [SEP] token, resulting in an encoding token embedding for every passage token p_i.

- For span-based question answering, we represent the question as the first sequence and the passage as the second sequence. We'll also need to add a linear layer that will be trained in the fine-tuning phase to predict the start and end position of the span.

QA as an Information Retrieval task - Factoid QA

- Many datasets (like SQuAD 2.0 and Natural Questions) also contain (question, passage) pairs in which the answer is not contained in the passage. We thus also need a way to estimate the probability that the answer to a question is not in the document. This is standardly done by treating questions with no answer as having the [CLS] token as the answer, and hence the answer span start and end index will point at [CLS]

- For datasets where annotated documents or passages exceed the maximum 512 input tokens allowed by BERT, such as Natural Questions with complete Wikipedia pages, a solution involves creating multiple pseudo-passage observations from the labeled Wikipedia page. Each observation is constructed by concatenating [CLS], the question, [SEP], and tokens from the document.

QA as an Information Retrieval task - Factoid QA

PES UNIVERSITY

- To manage longer documents, a sliding window of size 512 (or 512 minus the question length and special tokens) is utilized to create successive pseudo-passages. This process involves walking through the document, packing each window of tokens into the next pseudo-passage. The answer span for each observation is either marked as [CLS] (indicating no answer in that specific window) or the gold-labeled span.
- This approach can also be applied during inference by breaking up each retrieved document into separate observation passages and labeling each one. The answer is then chosen as the span with the highest probability (or nil if no span surpasses the probability of [CLS]).

QA as an Information Retrieval task - Factoid QA

PES UNIVERSITY

Retrieval-Augmented Generation

- The second standard reader algorithm is to generate from a large language model, conditioned on the retrieved passages. This method is known as retrieval-augmented generation, or RAG.
- In simple conditional generation, the task of question answering can be framed as word prediction. By providing a language model with a question and a token like "A:" to signal that an answer should follow, the model is prompted as follows:

```
Q: Who wrote the book 'The Origin of Species'? A:
```

QA as an Information Retrieval task - Factoid QA

- Subsequently, autoregressive generation is performed conditioned on this input text. The model generates responses based on the provided context, progressing sequentially to construct a coherent answer to the posed question.
- More formally the simple autoregressive language modeling computes the probability of a string from the previous tokens:

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | x_{<i})$$

QA as an Information Retrieval task - Factoid QA

- And simple conditional generation for question answering adds a prompt like Q: , followed by a query q , and A:, all concatenated:

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(\texttt{[Q:]}\ ;\ q\ ;\ \texttt{[A:]}\ ;\ x_{<i})$$

QA as an Information Retrieval task - Factoid QA

- The advantage of using a large language model is the enormous amount of knowledge encoded in its parameters from the text it was pretrained on. But, while this kind of simple prompted generation can work fine for many simple factoid questions, it is not a general solution for QA, because it leads to hallucination and is unable to answer questions from proprietary data.

- The idea of retrieval-augmented generation is to address these problems by conditioning on the retrieved passages as part of the prefix, perhaps with some prompt text like “Based on these texts, answer this question:”.

QA as an Information Retrieval task - Factoid QA

PES UNIVERSITY

- Datasets for IR-based QA are most commonly created by first developing reading comprehension datasets containing tuples of (passage, question, answer).
- Reading comprehension systems can use the datasets to train a reader that is given a passage and a question, and predicts a span in the passage as the answer.
- Including the passage from which the answer is to be extracted eliminates the need for reading comprehension systems to deal with IR.

QA as an Information Retrieval task - Factoid QA

PES UNIVERSITY

SQuAD Dataset : The Stanford Question Answering Dataset (SQuAD) consists of passages from Wikipedia and associated questions whose answers are spans from the passage.

HotpotQA Dataset: was created by showing crowd workers multiple context documents and asked to come up with questions that require reasoning about all of the documents.

The TriviaQA Dataset: contains 94K questions written by trivia enthusiasts, together with supporting documents from Wikipedia and the web resulting in 650K question-answer-evidence triples.

Natural Questions Dataset: incorporates real anonymized queries to the Google search engine

TyDi QA Dataset: contains 204K question-answer pairs from 11 typologically diverse languages, including Arabic, Bengali, Kiswahili, Russian, and Thai.