

Question and Answering

The different types of models in question and answers are Open domain question and answering and Reading comprehension question answering.

In the context of question answering:

1. Open Domain Question Answering:

- Open domain question answering involves developing systems that can answer questions across a broad range of topics, without specific limitations on the subject matter.
- In open-domain question answering, the system needs to understand the question posed by the user and retrieve or generate an appropriate answer from a vast pool of knowledge.
- These systems often rely on large-scale language models and knowledge bases to provide accurate and informative responses to user queries.
- Example: A user might ask an open-domain question answering system, "What are the effects of climate change?" The system would need to understand the question and generate a coherent response based on its understanding of climate change and its impacts.

2. Reading Comprehension Question Answering:

- Reading comprehension question answering focuses on answering questions based on a specific passage or document provided to the system.
- In reading comprehension tasks, the system is given a passage of text (the context) and a question related to the content of that passage.
- The system's task is to read and comprehend the passage, extract relevant information, and generate an accurate answer to the question based solely on the information contained within the passage.
- Example: Given a passage about the life of Abraham Lincoln, a reading comprehension system might be asked, "When was Abraham Lincoln

born?" The system would need to locate the relevant information in the passage and provide the correct date of Lincoln's birth.

While both open-domain and reading comprehension question answering involve providing answers to questions, they differ in terms of the scope of topics and the source of information used to generate the answers.

Open-domain question answering is more general, requiring systems to draw from a wide range of knowledge sources, while reading comprehension question answering focuses on understanding and extracting information from specific passages or documents.

Key components for Q/A:

In the context of natural language processing (NLP), retrievers, readers, and generators are components or modules within a larger system, typically used in question answering or text generation tasks:

1. Retrievers:

- Retriever modules are responsible for efficiently retrieving relevant documents or passages of text from a large corpus or knowledge base.
- These modules use various retrieval techniques, such as keyword matching, semantic search, or vector similarity algorithms, to identify documents or passages likely to contain relevant information.
- Retriever modules are often used as the first step in question answering systems to narrow down the search space and focus on the most relevant sources of information.

2. Readers:

- Reader modules, also known as comprehension models, are designed to understand and extract information from the retrieved documents or passages.

- These modules typically use machine learning models, such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), or transformer-based architectures like BERT (Bidirectional Encoder Representations from Transformers), to analyze the text and identify relevant information.
- Reader modules aim to accurately comprehend the meaning of the text, identify key entities or concepts, and extract answers to specific questions posed by the user or generated by the system.

3. Generators:

- Generator modules are responsible for generating new text, such as responses, summaries, or completions, based on input from the retriever and reader modules.
- These modules can use various techniques, including rule-based approaches, template-based generation, or neural language models such as GPT (Generative Pre-trained Transformer) or GPT-3, to generate text that is coherent, contextually relevant, and grammatically correct.
- Generator modules are often used to produce final answers or responses in question answering systems, summarize information from multiple sources, or generate text in natural language generation tasks.

In summary, retrievers retrieve relevant documents or passages, readers comprehend and extract information from the retrieved text, and generators produce new text based on the input from the retriever and reader modules. Together, these components form the building blocks of complex NLP systems for tasks such as question answering, summarization, and text generation.

These different models can be used in different combinations

The diagram illustrates the architecture of a Retrieval-Augmented Generation (RAG) system, divided into two main components: the Information Retrieval System and the Reading Comprehension module.

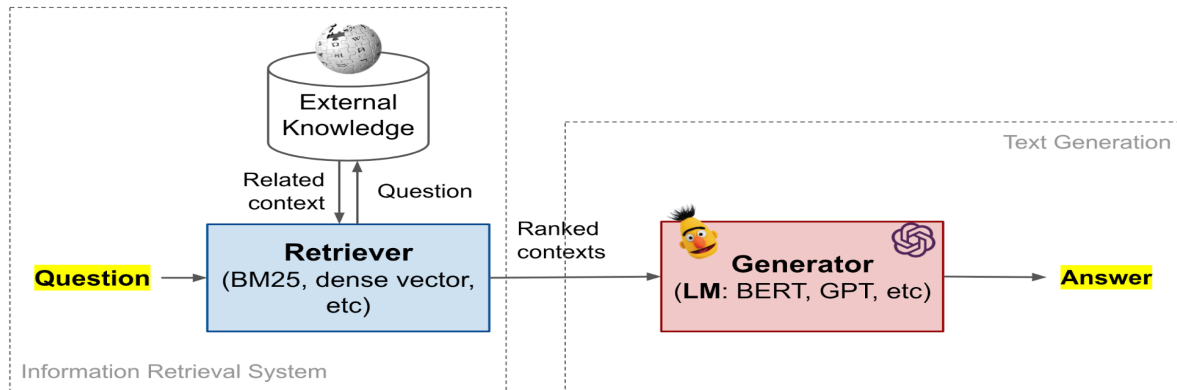
Information Retrieval System:

- A **Question** (highlighted in yellow) is input into the **Retriever** (blue box).
- The **Retriever** uses methods like BM25 or dense vectors to search for information.
- It interacts with **External Knowledge** (represented by a globe icon in a cylinder), which provides **Related context** to the Retriever and receives **Questions** from it.
- The Retriever outputs **Ranked contexts** to the Reader.

Reading Comprehension:

- The **Reader** (orange box) processes the ranked contexts.
- It identifies the **start pos** (start position) and **end pos** (end position) within the context.
- The **selected context** (the segment between start and end positions) is then used to generate the **Answer** (highlighted in yellow).
- The selected context is visually represented as a sequence of tokens: a red box with 'S', a grey box, and a green box with 'E'.

- ## 2)Retriever-Generator model:



- A retriever-generator model is a combination of retriever and generator components, often used in text generation or summarization tasks.
- The retriever component retrieves relevant documents or passages of text from a corpus or knowledge base.
- The generator component then processes the retrieved text and generates new text based on the input.
- This model is useful for tasks such as document summarization, where the retriever identifies relevant information, and the generator condenses this information into a concise summary.