DIPARTIMENTO
MATEMATICA

Introduction to RAG Models

Retrieval Augmented Generation (RAG) models are a powerful class of language models that combine the strengths of retrieval-based and generation-based approaches to tackle complex natural language tasks. These models leverage external knowledge sources to enhance their understanding and generation capabilities.

What are RAG Models?

Hybrid Approach

RAG models combine retrieval and generation techniques, leveraging the strengths of both to tackle complex natural language tasks.

Knowledge Augmentation

These models enhance language understanding by incorporating information from external knowledge sources, such as knowledge bases or text corpora.

Flexible Architecture

RAG models can be adapted to a variety of applications, from question answering to text summarization, by adjusting the retrieval and generation components.

How RAG Models Work

1

Retrieval

The model retrieves relevant information from a large knowledge base.

2

Encoding

The retrieved information is encoded along with the input text.

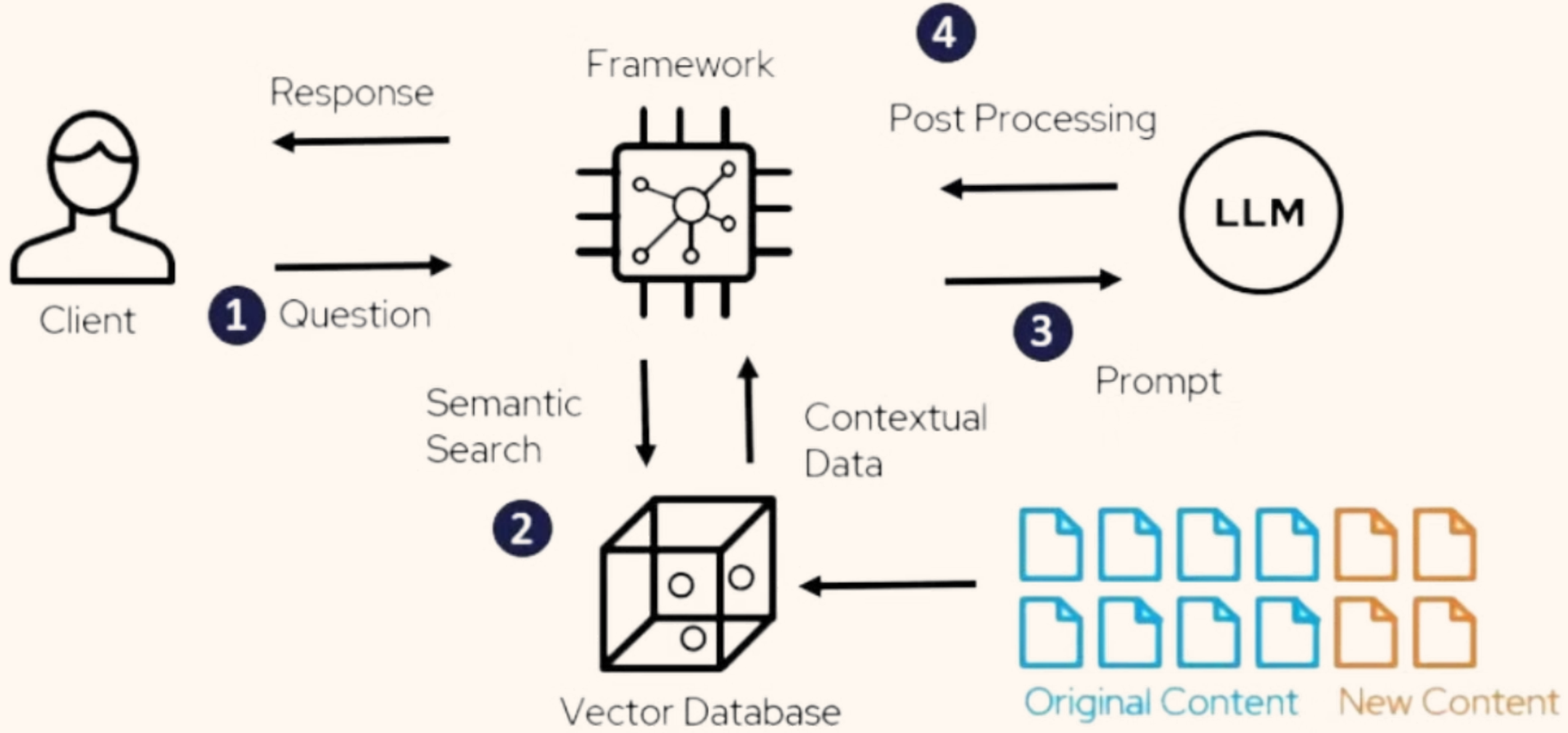
3

Generation

The encoded representation is used to generate the output sequence.

RAG models work by first retrieving relevant information from a large knowledge base, such as Wikipedia or a custom corpus. This retrieved information is then encoded alongside the input text, allowing the model to leverage both the external knowledge and the context of the task at hand. Finally, the encoded representation is used to generate the output sequence, producing a result that is informed by the retrieved knowledge.

RAG Architecture Model



Advantages of RAG Models

Enhanced Knowledge Utilization

RAG models can effectively leverage external knowledge sources to improve their understanding and generation capabilities, leading to more informed and contextual outputs.

Improved Flexibility

The hybrid architecture of RAG models allows for adaptability to a wide range of natural language processing tasks, from question answering to summarization.

Enhanced Performance

By combining the strengths of retrieval and generation, RAG models often demonstrate superior performance compared to standalone retrieval or generation-based approaches.

Limitations of RAG Models

Dependency on Knowledge Bases

RAG models rely heavily on the quality and coverage of the underlying knowledge bases, which can limit their performance on tasks requiring specialized or niche information.

Retrieval Accuracy

The retrieval component of RAG models is crucial, and inaccuracies in this step can negatively impact the overall performance of the model.

Computational Complexity

The dual processes of retrieval and generation can make RAG models computationally intensive, especially for large-scale applications or real-time deployments.

Potential Bias

The knowledge sources used by RAG models may contain biases, which could then be reflected in the model's outputs, requiring careful consideration and mitigation.

Applications of RAG Models

Question Answering

RAG models excel at retrieving relevant information from knowledge bases to provide accurate and informative answers to complex questions.

Dialogue and Chatbots

By leveraging external knowledge, RAG-based chatbots and dialogue systems can engage in more natural and contextual conversations.

Summarization

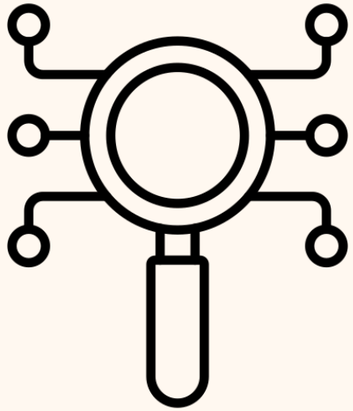
RAG models can generate concise and informative summaries by selectively retrieving and incorporating key details from large amounts of source material.

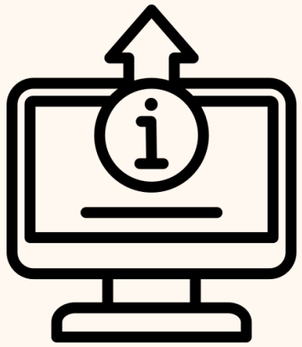
Content Generation

RAG models can be used to generate high-quality, fact-based content such as news articles, reports, and creative writing by drawing upon a broad knowledge base.

Retrieval Mechanisms in RAG Models

The retrieval component of RAG models is crucial, as it determines the quality and relevance of the external information that is used to augment the generation process.





By leveraging powerful indexing and retrieval algorithms, RAG models can efficiently sift through vast amounts of information and identify the most salient pieces of knowledge to include in the generation phase.

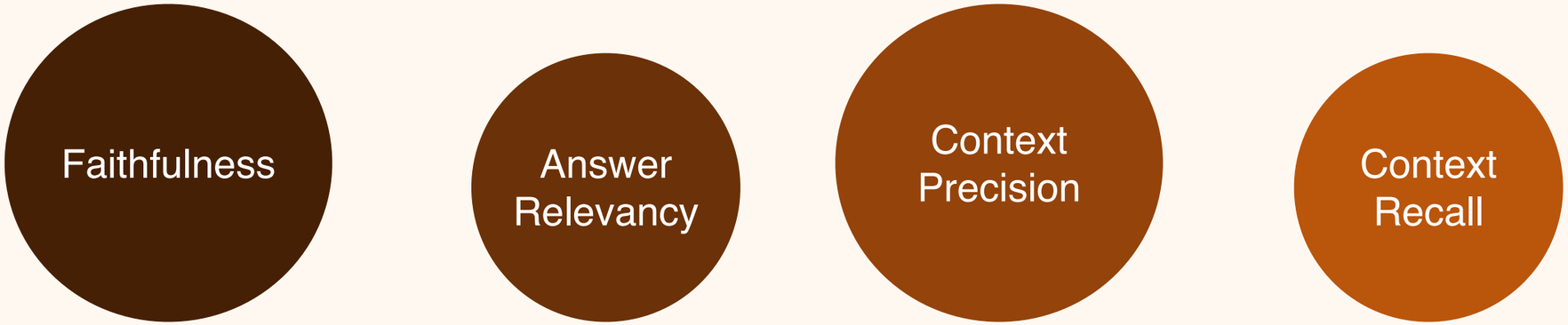
Generation Mechanisms in RAG Models

The generation component in RAG models uses retrieved knowledge to produce coherent and informative output through advanced language modeling techniques, like **transformer-based architectures.**

By conditioning the generation on the encoded representation of the input and the retrieved knowledge, RAG models can generate responses that are not only grammatically correct but also semantically relevant and factually accurate.

Evaluation Metrics for RAG Models

Evaluating the performance of Retrieval Augmented Generation (RAG) models is a crucial step in understanding their strengths and limitations. RAG models are typically assessed using a combination of metrics that capture their retrieval accuracy, generation quality, and overall task-specific effectiveness.



The context precision and context recall metrics evaluate the quality of the information retrieved from the knowledge base, while the faithfulness and relevancy scores assess the performance and relevance of the generated text. The task-specific accuracy metric measures the overall effectiveness of the RAG model in completing the target natural language processing task, such as question answering or summarization.

Future Directions and Research Opportunities

Exploring Novel Architectures

Researchers are investigating innovative RAG model architectures to enhance retrieval, improve knowledge integration, and enable more sophisticated reasoning.

Advancing Retrieval Techniques

Ongoing research aims to develop more accurate and efficient retrieval mechanisms, including few-shot and zero-shot learning approaches.

Enhancing Reasoning Capabilities

Researchers are exploring ways to equip RAG models with stronger commonsense and logical reasoning skills to handle complex tasks.

Leveraging Diverse Data Sources

Incorporating a wider range of knowledge bases, including dynamic and domain-specific sources, could further improve the versatility of RAG models.

Sources for our work

Code of Justinian

The Code of Justinian, compiled in 529-534 CE, streamlined Roman laws into a coherent legal system, profoundly influencing European legal traditions and modern civil law.

Magna Carta

The Magna Carta, signed in 1215, established foundational principles of legal rights and limitations on royal power, influencing the development of constitutional law and individual freedoms.

The Universal Declaration of Human Rights

The Universal Declaration of Human Rights, adopted in 1948 by the UN General Assembly, sets out fundamental rights and freedoms for all people worldwide.

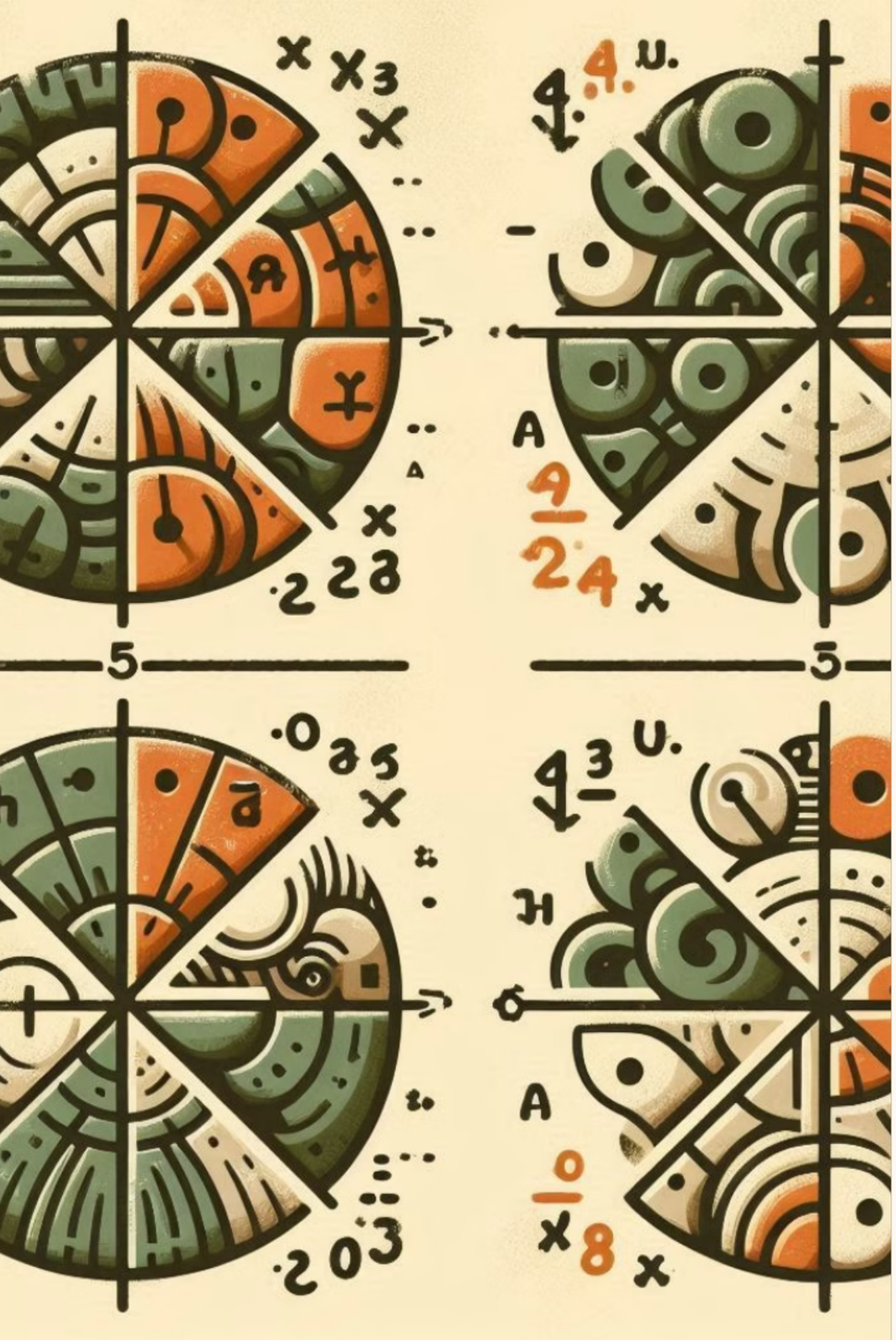
What did we do?

1. **Chunking** - first step for creating a RAG model is to divide the document into different, semantically coherent parts or contexts. There are several chunking methods, but we used 3 automatic (comes from langchain) and 1 manual chunking. Automatic chunking methods we used are; recursive character text splitter, interquartile chunking, percentile chunking.
2. **Model definition** - after chunking our dataset, we proceed with the model definition. Model definition, in a nutshell, is creating a template for our RAG model and then combining the template with an LLM model. Our template consists of user-defined question and the relevant context retrieved from chunks.

3. **Model testing** - in this part, we create some questions from the documents to see how it answers basic, user-defined questions. As we'll see, in the evaluation part, we'll take this approach to a whole different level.

4. **Model evaluation** - isn't it boring to look at the document and come up with questions to test the model? Of course it is! That's why we're going to use an LLM model to generate **questions** from our synthetic dataset. Then we'll use the evaluation metrics we talked in the first part to assess the performance of our model.

Now let's analyze each of these topics in detail.



Chunking

- **Recursive Character Text Splitter** - splitter to recursively split text, aiming to keep semantically related pieces of text together. It is particularly recommended for initial text splitting efforts.

- **Interquartile Chunking** - in this method, the interquartile distance is used to split chunks.
- **Percentile Chunking** - in this method, all differences between sentences are calculated, and then any difference greater than the X percentile is split.
- **Manual Chunking** - chunking is done by the programmer's own conditions. For example, Magna Carta can be chunked based on the articles.

Model Definition

1. We create a template that will take a user-defined question, and augment it with the relevant context. We also instruct the template to say "I don't know" if it cannot find an answer to the question in the contexts.

2. Create a vector database that holds the contexts.

3. Then to access those contexts, we create a **retriever**.

4. Create the model by supplying the question and the context retriever, followed by our template which will combine these, followed by base LLM model which will take the question and the context and generate answers.

Model Testing

how do we know if they work?

We just create a question from the dataset and call the invoke method to generate an answer. If the chunking is done correctly (it is semantically coherent), then the answer will also be correct or at least it will have an answer.

but what if the chunking is problematic?

Then the model won't be able to find an answer and based on our template it generates an output "I don't know".

Model Evaluation and Metrics

- **Context Precision**

evaluates whether all of the ground-truth relevant items present in the contexts are ranked higher or not. Ideally all the relevant chunks must appear at the top ranks

- **Faithfulness**

this measures the factual consistency of the generated answer against the given context. It is calculated from answer and retrieved context. The generated answer is regarded as faithful if all the claims that are made in the answer can be inferred from the given context.

- **Answer Relevancy**

this evaluation metric focuses on assessing how pertinent the generated answer is to the given prompt. This metric is computed using the question, the context and the answer. The Answer Relevancy is defined as the mean cosine similarity of the original question to a number of artificial questions, which where generated (reverse engineered) based on the answer.

- **Context Recall**

measures the extent to which the retrieved context aligns with the annotated answer, treated as the ground truth. It is computed based on the ground truth answer and the retrieved context. In an ideal scenario, all sentences in the ground truth answer should be attributable to the retrieved context

Conclusion: The Power and Potential of RAG Models

RAG models represent a powerful and flexible approach to natural language processing, offering a range of advantages and potential applications. Looking ahead, continued research and innovation in this field holds great promise for advancing our understanding and capabilities in this critical domain.

Fun things are still ahead!

The breathtaking innovations happening in the AI field excites all of us, especially the researches done in the field of NLP.

RAG models could prove useful to many industries, businesses, technologies and etc. In the era of huge texts, policies, laws, medical texts, scientific researches and other areas that incorporate huge corpuses of texts may benefit from the innovations done in RAG.

Thank You for Keeping Up until the End

We hope this presentation has provided a valuable overview of RAG models and their powerful potential. We're excited to continue exploring the frontiers of natural language processing and AI-driven language generation.

Presented by Dinara Kurmangaliyeva, Emel Safarini, Ruslan Nuriev, Sezgi Cobanbas, Zeynep Sila Kaya