Querying documents with LLMs

April 23, 2024 · 203 min · Ismaël Rousseau, Géraldine Damnati

Table of Contents

Introduction

Documents are at the core of many business processes, containing valuable information that drives decision-making. With the rise of generative AI, we're now turning to this technology for document intelligence, seeking to make use of their impressive capabilities to handle and interpret text. This is especially the case right now for legal and financial documents, where we're experimenting with LLMs to verify a document's compliance, compare it with a collection of other documents or facilitate information gathering on a single document or a group of documents.

Despite the advancements in generative AI, applying current LLMs to real-world applications presents certain challenges, primarily due to the discrepancy between the rather short raw text segments they are designed to work with and the length and complexity of actual documents.

This post offers an introduction to various existing methods one can use to query documents, followed by a performance comparison on a case-study: **questioning the AI Act.**. The objective is to illustrate that RAG configuration can have a significant impact on the results, and, beyond RAG, that different types of questions should be handled with different types of approaches.

Each section includes illustrations, examples and code snippers for better understand on how you could implement those systems yourself.

Note: This post mostly focuses on techniques related to querying a single long document or a set of few documents, but not really on large collections with thousands of documents. While most of wh... writing should also apply for those cases, you might face issues related to scale, that I'm not covering here.

Context augmentation

In this section, we'll explore various methods designed to augment the LLM with additional context when analyzing documents. These strategies incorporate additional information in the model's prompt, either directly extracted from or inferred based on the document in question. We'll examine four distinct methods, each with increasing levels of complexity, to understand their respective strengths and applications. Hopefully, by the end of this article, you'll have a better understanding of which approach works best for your own use cases.

In-context learning

In-context learning (ICL) is the most straightforward method to process documents with LLMs. Essentially, this approach involves including the entire document within the prompt. Despite its simplicity, this method can be remarkably effective and is often enough for most tasks. By integrating the document's content into the prompt, the LLM is now provided with a foundation for its responses, leveraging both its internal knowledge and the information within the document. Since the model itself cannot access the company's internal data or updates beyond a specific point, this integration significantly improves the LLM's understanding and response accuracy in our use-cases at Orange.

▸ **Click here to view an example**

While powerful for short documents, ICL is not suitable for long documents or for large collections of documents. This is because LLMs have a limit to how much text can be included in the prompt. This limit varies depending on the model, but is usually around 8k tokens, which represents about 15 pages of text. To address this issue, some alternative approaches have emerged, such as the popular **Retrieval Augmented Generation** (RAG) and the use of **Agents**, which we will now look into.

Retrieval Augmented Generation

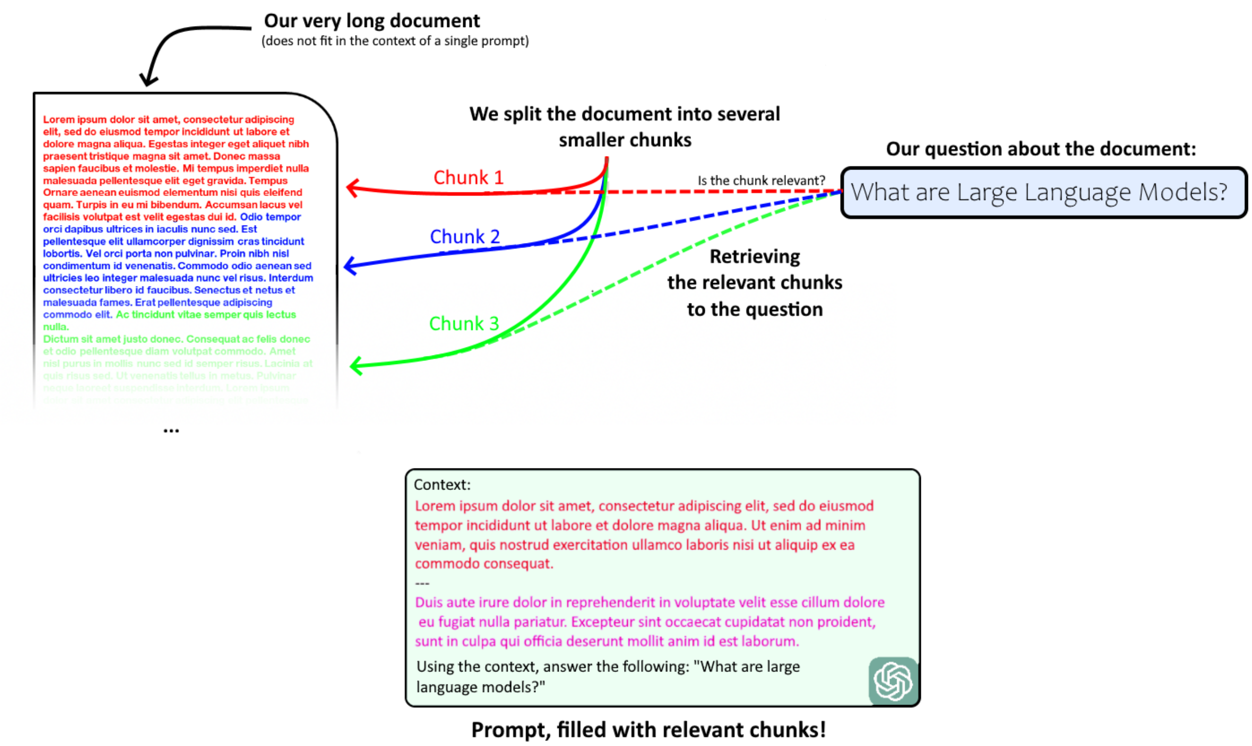
As we just said, it is sometimes impossible to include the full document's content in the prompt of a LLM. **Retrieval Augmented Generation** is an approach that solves this problem by incorporating only the relevant sections of a document within the prompt, rather than the complete document.

It is made of two components: the **retriever** and the **generator** .

- The **retriever** is the component responsible for identifying and selecting the most relevant parts of the document to include in the prompt. The retrieval process can be as simple as searching for keywords in the document, or can be more sophisticated, like, for instance, using dense vector searches to more accurately grasp the semantics of the question & the documents.

- The **generator** is simply the part that will *write* the answer in natural language, using the retrieved passages. Nowadays, the generator is almost always a LLM.

As of April 2024, this is the method used in Dinootoo Search to search in documents.



Here's how you could implement RAG using DocLLM, a library I created in order to work with documents.

```
from doc_llm.documents.document import Document
from doc_llm.orange import AzureForOrangeEngine

ai_act = Document("src/my_data/ai_act.pdf") # Without additional arguments, the document will be proces
engine = AzureForOrangeEngine("dai-semafor-nlp-gpt-4-model-fr")
answer = ai_act.query("What does the AI act say regarding facial regognition?", engine, max_tokens=512,
```

Note: This library was mainly created to experiment freely and learn the concepts described here. For production purposes, you might want to check some frameworks like LangChain or LlamaIndex.

▼ **Click here to view the full prompt generated by the code**

We asked the question *"What does the AI act say regarding facial regognition?"* on the AI Act. The

retriever (here, a dense-vector search) found that the top three segments of text most similar/relevant to the question were the chunks number **4**, **52** and **2**. As such, we've included them in the prompt.

```
Context:
---
<Chunk: ai_act.pdf-4>
This article is part of 'Title II: Prohibited Artificial Intelligence Practices'.

Article 5: Prohibited Artificial Intelligence Practices content:

1. The following artificial intelligence practices shall be prohibited:
(a) the placing on the market, putting into service or use of an AI system that deploys subliminal tech
(b) the placing on the market, putting into service or use of an AI system that exploits any of the vul
(ba) the placing on the market or putting into service for this specific purpose, or use of biometric c
(c) the placing on the market, putting into service or use of AI systems for the evaluation or classifi
(i) detrimental or unfavourable treatment of certain natural persons or whole groups thereof in social
(ii) detrimental or unfavourable treatment of certain natural persons or groups thereof that is unjusti
(d) the use of ‘real-time’ remote biometric identification systems in publicly accessible spaces for th
(i) the targeted search for specific victims of abduction, trafficking in human beings and sexual explo
(ii) the prevention of a specific, substantial and imminent threat to the life or physical safety of na
(iii) the localisation or identification of a person suspected of having committed a criminal offence,
(da) the placing on the market, putting into service for this specific purpose, or use of an AI system
(db) the placing on the market, putting into service for this specific purpose, or use of AI systems th
(iiid) deleted.
1a. This Article shall not affect the prohibitions that apply where an artificial intelligence practice
1. The use of ‘real-time’ remote biometric identification systems in publicly accessible spaces for the
(a) the nature of the situation giving rise to the possible use, in particular the seriousness, probabi
(b) the consequences of the use of the system for the rights and freedoms of all persons concerned, in
1. As regards paragraphs 1, point (d) and 2, each use for the purpose of law enforcement of a ‘real-tim
3a. Without prejudice to paragraph 3, each use of a ‘real-time’ remote biometric identification system
1. A Member State may decide to provide for the possibility to fully or partially authorise the use of
2 and 3. Member States concerned shall lay down in their national law the necessary detailed rules for
1. National market surveillance authorities and the national data protection authorities of Member Stat
2. The Commission shall publish annual reports on the use of ‘real-time’ remote biometric identificatio
</Chunk: ai_act.pdf-4>

<Chunk: ai_act.pdf-52>
This article is part of 'Title IV: Transparency Obligations for Providers and Deployers of Certain AI S

Article 52: Transparency Obligations for Providers and Users of Certain AI Systems and GPAI Models cont

1. Providers shall ensure that AI systems intended to directly interact with natural persons are design
1a. Providers of AI systems, including GPAI systems, generating synthetic audio, image, video or text c
2. Deployers of an emotion recognition system or a biometric categorisation system shall inform of the
3. Deployers of an AI system that generates or manipulates image, audio or video content constituting a
3a. The information referred to in paragraphs 1 to 3 shall be provided to the concerned natural persons
4. Paragraphs 1,
2 and 3 shall not affect the requirements and obligations set out in Title III of this Regulatio    s
4a. The AI Office shall encourage and facilitate the drawing up of codes of practice at Union level to
</Chunk: ai_act.pdf-52>
```

```
<Chunk: ai_act.pdf-2>
This article is part of 'Title I: General Provisions'.

Article 3: Definitions content:

For the purpose of this Regulation, the following definitions apply:
(1) ‘AI system’ means a machine-based system designed to operate with varying levels of autonomy and th
(1a) ‘risk’ means the combination of the probability of an occurrence of harm and the severity of that
(2) ‘provider’ means a natural or legal person, public authority, agency or other body that develops an
(4) ‘deployer means any natural or legal person, public authority, agency or other body using an AI sys
(5) ‘authorised representative’ means any natural or legal person located or established in the Union w
(6) ‘importer’ means any natural or legal person located or established in the Union that places on the
(7) ‘distributor’ means any natural or legal person in the supply chain, other than the provider or the
(8) ‘operator’ means the provider, the product manufacturer, the deployer, the authorised representativ
(9) ‘placing on the market’ means the first making available of an AI system or a general purpose AI mo
(10) ‘making available on the market’ means any supply of an AI system or a general purpose AI model fo
(11) ‘putting into service’ means the supply of an AI system for first use directly to the deployer or
(12) ‘intended purpose’ means the use for which an AI system is intended by the provider, including the
(13) ‘reasonably foreseeable misuse’ means the use of an AI system in a way that is not in accordance w
(14) ‘safety component of a product or system’ means a component of a product or of a system which fulf
(15) ‘instructions for use’ means the information provided by the provider to inform the user of in par
(16) ‘recall of an AI system’ means any measure aimed at achieving the return to the provider or taking
(17) ‘withdrawal of an AI system’ means any measure aimed at preventing an AI system in the supply chai
(18) ‘performance of an AI system’ means the ability of an AI system to achieve its intended purpose;
(19) ‘notifying authority’ means the national authority responsible for setting up and carrying out the
(20) ‘conformity assessment’ means the process of demonstrating whether the requirements set out in Tit
(21) ‘conformity assessment body’ means a body that performs third-party conformity assessment activiti
(22) ‘notified body’ means a conformity assessment body notified in accordance with this Regulation and
(23) ‘substantial modification’ means a change to the AI system after its placing on the market or putt
(24) ‘CE marking of conformity’ (CE marking) means a marking by which a provider indicates that an AI s
(25) ‘post-market monitoring system’ means all activities carried out by providers of AI systems to col
(26) ‘market surveillance authority’ means the national authority carrying out the activities and takin
(27) ‘harmonised standard’ means a European standard as defined in Article 2(1)(c) of Regulation (EU) N
(28) ‘common specification’ means a set of technical specifications, as defined in point 4 of Article 2
(29) ‘training data’ means data used for training an AI system through fitting its learnable parameters
(30) ‘validation data’ means data used for providing an evaluation of the trained AI system and for tun
(31) ‘testing data’ means data used for providing an independent evaluation of the AI system in order t
(32) ‘input data’ means data provided to or directly acquired by an AI system on the basis of which the
(33) ‘biometric data’ means personal data resulting from specific technical processing relating to the
(33a) ‘biometric identification’ means the automated recognition of physical, physiological, behavioura
(33c) ‘biometric verification’ means the automated verification of the identity of natural persons by c
(33d) ‘special categories of personal data’ means the categories of personal data referred to in Articl
(33e) ‘sensitive operational data’ means operational data related to activities of prevention, detectio
(34) ‘emotion recognition system’ means an AI system for the purpose of identifying or inferring emotio
(35) ‘biometric categorisation system’ means an AI system for the purpose of assigning natural persons
(36) ‘remote biometric identification system’ means an AI system for the purpose of identifying natural
(37) ‘‘real-time’ remote biometric identification system’ means a remote biometric identification syste
(38) ‘‘post’ remote biometric identification system’ means a remote biometric identification system oth
(39) ‘publicly accessible space’ means any publicly or privately owned physical place accessible to an
(40) ‘law enforcement authority’ means:
(a) any public authority competent for the prevention, investigation, detection or prosecution o
(b) any other body or entity entrusted by Member State law to exercise public authority and public powe
(41) ‘law enforcement’ means activities carried out by law enforcement authorities or on their behalf f
(42) ‘Artificial Intelligence Office’ means the Commission’s function of contributing to the implementa
```

```
(43) ‘national competent authority’ means any of the following: the notifying authority and the market
(44) ‘serious incident’ means any incident or malfunctioning of an AI system that directly or indirectl
(a) the death of a person or serious damage to a person’s health;
(b) a serious and irreversible disruption of the management and operation of critical infrastructure.
(ba) breach of obligations under Union law intended to protect fundamental rights;
(bb) serious damage to property or the environment.
(44a) ‘personal data’ means personal data as defined in Article 4, point (1) of Regulation (EU) 2016/67
(44c) ‘non-personal data’ means data other than personal data as defined in point (1) of Article 4 of R
(be) ‘profiling’ means any form of automated processing of personal data as defined in point (4) of Art
(bf) ‘real world testing plan’ means a document that describes the objectives, methodology, geographica
(44 eb) ‘sandbox plan’ means a document agreed between the participating provider and the competent aut
(bg) ‘AI regulatory sandbox’ means a concrete and controlled framework set up by a competent authority
(bh) ‘AI literacy’ refers to skills, knowledge and understanding that allows providers, users and affec
(bi) ‘testing in real world conditions’ means the temporary testing of an AI system for its intended pu
(bj) ‘subject’ for the purpose of real world testing means a natural person who participates in testing
(bk) ‘informed consent’ means a subject’s freely given, specific, unambiguous and voluntary expression
(bl) “deep fake” means AI generated or manipulated image, audio or video content that resembles existin
(44e) ‘widespread infringement’ means any act or omission contrary to Union law that protects the inter
(a) which has harmed or is likely to harm the collective interests of individuals residing in at least
(i) the act or omission originated or took place;
(ii) the provider concerned, or, where applicable, its authorised representative is established; or,
(iii) the deployer is established, when the infringement is committed by the deployer;
(b) which protects the interests of individuals, that have caused, cause or are likely to cause harm to
(44h) ‘critical infrastructure’ means an asset, a facility, equipment, a network or a system, or a part
(44b) ‘general purpose AI model’ means an AI model, including when trained with a large amount of data
(44c) ‘high-impact capabilities’ in general purpose AI models means capabilities that match or exceed t
(44d) ‘systemic risk at Union level’ means a risk that is specific to the high-impact capabilities of g
(44e) ‘general purpose AI system’ means an AI system which is based on a general purpose AI model , tha
(44f) ’floating-point operation’ means any mathematical operation or assignment involving floating-poin
(44g) ‘downstream provider’ means a provider of an AI system, including a general purpose AI system, wh
</Chunk: ai_act.pdf-2>
---
Using the provided context, answer the user query in the same language as his: 'What does the AI act sa
```

▾ **Click here to view GPT’s answer**

```
The AI Act addresses facial recognition in several articles.

In Article 5, it prohibits the use of AI systems that categorize individuals based on their biometric d

In Article 52, it states that deployers of a biometric categorisation system should inform the natural

In Article 3, it defines 'remote biometric identification system' as an AI system for the purpose of id

These are just a few examples, and the AI Act addresses facial recognition and biometric data in other
```

As you can see, RAG is a rather accessible method to query long documents. However, it should be noted that it works better for some questions than others and that it might not work at all for some questions.

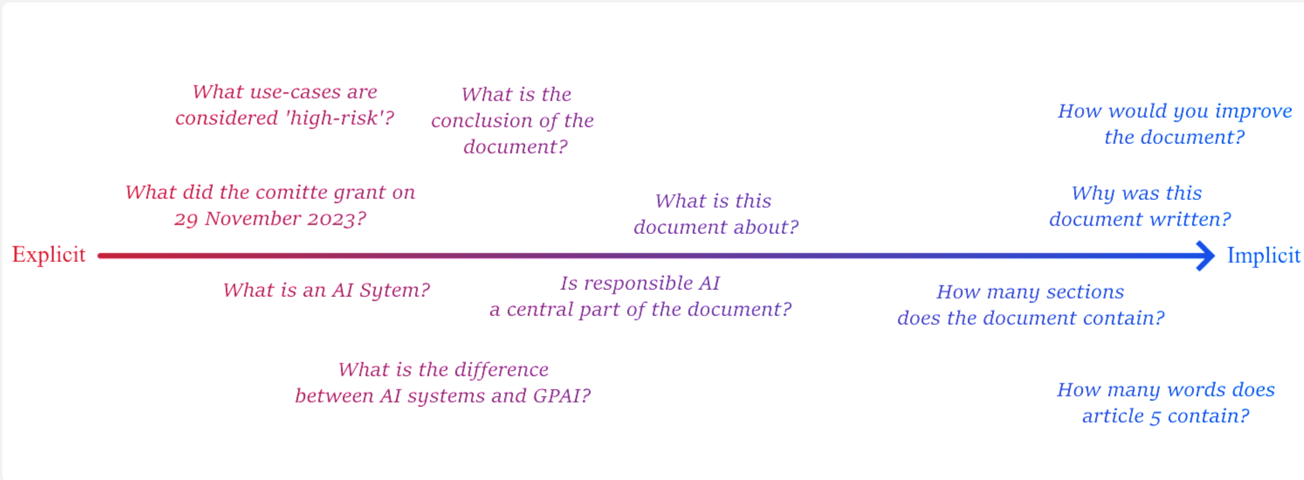
Intrisic limitations of RAG

Indeed, while it works most of the time for precise, factual questions, it will struggle answering questions that require information that is not explicitly stated in the text. This includes questions that are about the structure of the document rather than its content directly or questions that are very broad and require to piece together multiple parts of the document to get an answer from the overall context.

Here are a few examples of questions for which RAG may not be the best fit:

- *Questions about the document structure*
  - How many sections of the document talk about X?
  - Is section X longer than section Y?
  - Is topic X covered in section Y or Z?
  - What is the last item listed in the document?
- *Broad questions*
  - What is this document about?
  - Can you summarize the document?
  - What is the author's main point?
  - What are the main questions tackled in the document?
  - Who is the intended audience?

As a general guideline, in order to know if RAG is appropriate for your use-case, you should determine if the answers to the questions you want to ask are explicitly stated in your documents, or if they're rather deduced implicitely from the context and the structure. Of course, this is a spectrum, and some questions may fall somewhere in between, requiring a combination of direct information retrieval and contextual understanding to provide the most accurate and helpful responses.



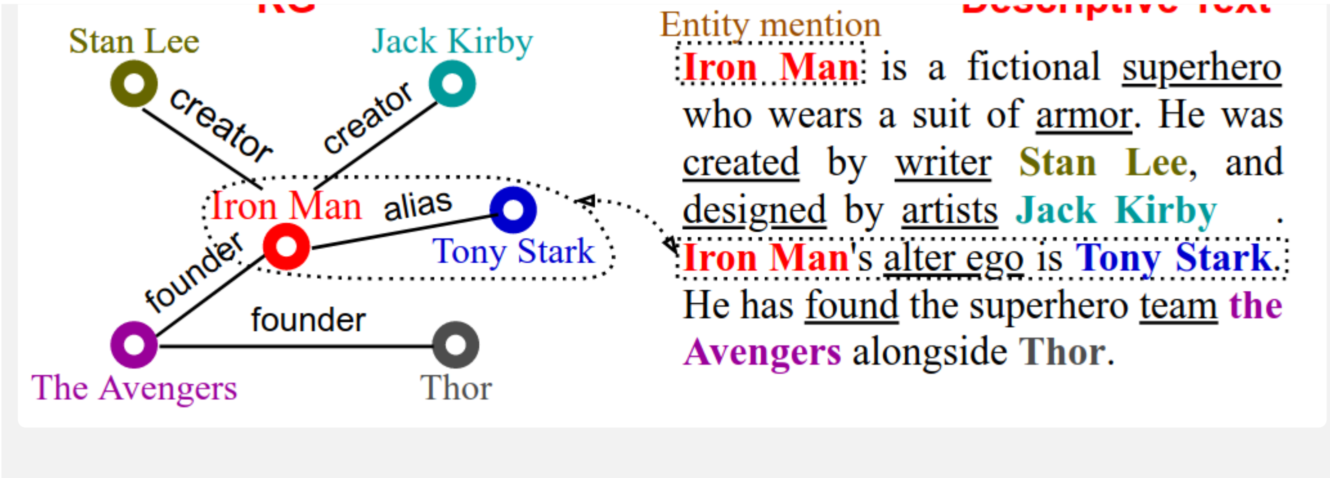
Questions spectrum, RAG is very effective for questions on the left. For questions on the right, GraphRAG and Agents might work better

The purpose here is to highlight that quens come in various forms, and that, consequently, there isn't a single system capable of accurately answering all of them. You should instead try to identify what kind of answers you want to retrieve from your document to choose the appropriate approach.

GraphRAG

Note: *An article focused on GraphRAG will be published by HUYNH Viet Phi soon, I will update this post with the link once it's available. However, note that the use-cases are a bit different. I will be focusing on creating small Knowledge Graphs on the fly to query documents, whereas Phi will write about using GraphRAG to query a pre-existing big Knowledge Graph (ie. TelcoGraph, Wikidata). This has different implications in terms of scale, use-cases, approaches etc.*

In order to tackle the issues previously mentioned, an approach called GraphRAG has been proposed. The idea is to first create a **Knowledge Graph** of the document(s) prior to answering the questions. A **Knowledge Graph** is a network of entities, concepts, and relationships that provides a structured representation of knowledge present in your document(s). It organizes data in an interconnected way, allowing for more complex queries and a better understanding of the content within the documents.



Representation of a knowledge graph: *Li, Junyi, et al. "Few-shot knowledge graph-to-text generation with pretrained language models."*

There are multitude of ways to create such Knowledge Graphs. In our case, following Microsoft reasearch blog post on the subject, they were created by a LLM, iterating through the whole document(s) and updating the graph when encountering new relationships. Now, in order to use the graph to answer a question, we'll have to retrieve the relevant information from the graph.

Overall, GraphRAG offers two main advantages:

1. The retrieved context doesn't only come from a few segments of the text, but can represent knowledge that is scattered in various parts of the document or collection of documents.
2. You not only retrieve knowledge of the entities mentioned in the user's query, but also other entities that are related (and you know the type of relationship between them). Which often results in more complete answers.

Some companies like WikiHow.ai are also experimenting with Graph Reasoning in order to constraint our search following certain rules. I have not experimented with those techniques myself, but it shows there's a lot of potential at the intersection of Knowledge Graphs and RAG. However, GraphRAG still is inherently limited by some of the characteristics of LLMs, such as its inhability to count, to sort or do any kind of advanced reasoning which might be required to answer some questions.

LlamaIndex implementation of GraphRAG

LlamaIndex proposes an implementation that uses NebulaGraph (an open-source distributed graph database) and retrieves information about relevant entities using keywords search. More specif the retriever performs the following steps:

1. Search related Entities to the question/task

2. Get SubGraph of those Entities (default 2-depth) from the KnowledgeGraph
3. Build context based on the SubGraph

However, not much else is described about the specifics of this implementation in the documentation.

▸ **Here is an example of the context passed in the prompt from the LlamaIndex implementation**

DocLLM implementation of GraphRAG

On my side, in order to experiment with this approach, I have personally reimplemented a version of GraphRAG in my DocLLM framework. One significant difference between my implementation and LlamaIndex lies in the graph creation process. In my case, I also request GPT to provide a natural language description of the relationship when creating the graph (and I store the source chunk from which the relationship was created).

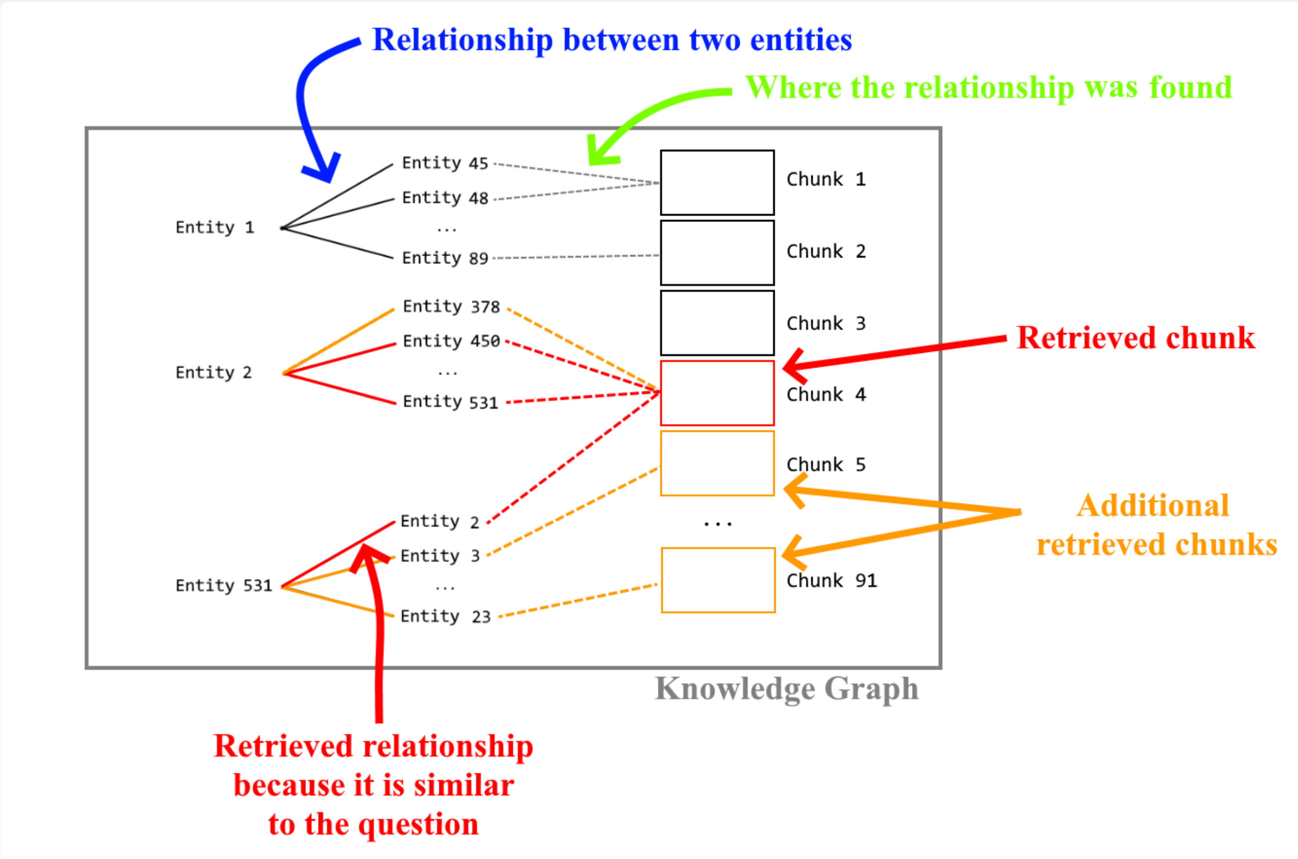
Relationship for the 'AI systems' entity:

```
- Relationship: 'Intended to be used in'
- Target entity: 'Areas listed in Annex III'
- Description: 'The AI systems are intended to be used in any of the areas listed in points 1 to 8 of A

- Relationship: 'Risk of harm to'
- Target entity: 'Health and safety'.
- Description: 'The AI systems pose a risk of harm to health and safety.'

- Relationship: 'Developed for'
- Target entity: 'Safeguarding substantial public interest'.
- Description: 'AI systems shall be developed for safeguarding substantial public interest.'

- Relationship: 'Components'
- Target entity: 'Large-scale IT systems'.
- Description: 'AI systems are components of the large-scale IT systems'

[...]
```

From there, I propose the following retrieval method:

1. We retrieve the top 20 descriptions that are the most similar to the question (cosine similarity on dense vectors).
2. We list all the entities which contain at least one relevant description
3. **We include in the prompt the concatenation of all the chunks sources that are related to the relationships of the top entities.**

In a way, what we do is that we use the descriptions from the relationships as a proxy to then retrieve the relevant chunks, as illustrated by the following:



Using the generated Knowledge Graph as a proxy to retrieve relevant chunks

Doing this offers two key main benefits. Firstly, it compares the question with descriptions of a similar length, which makes semantic search more effective. Secondly, it not only retrieves information directly relevant to the question but also includes related entities, thereby offering a wider yet focused context.

Agents and tools

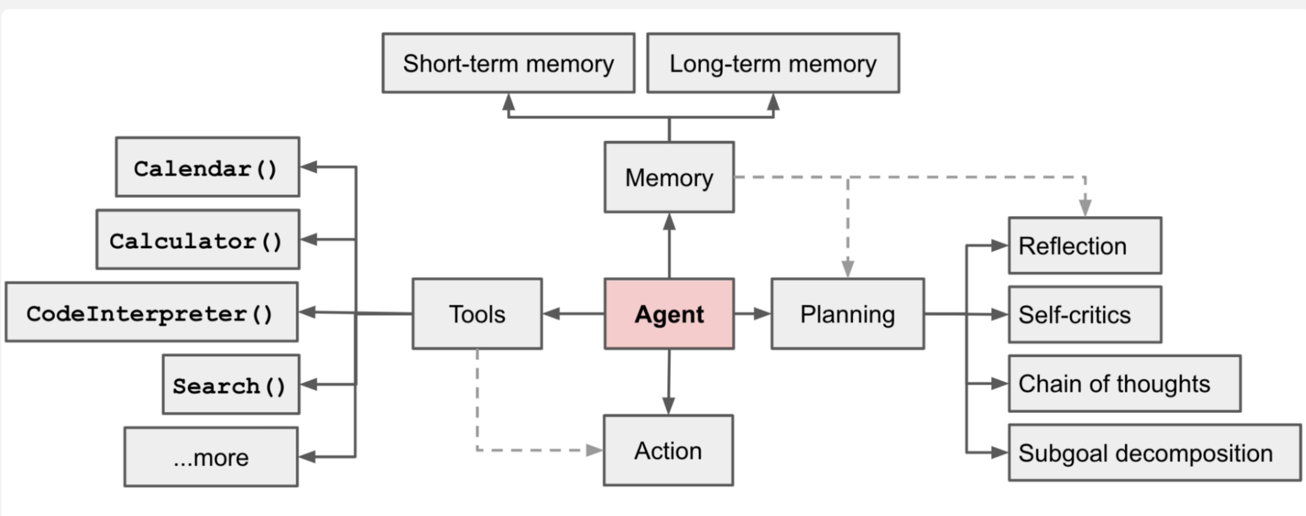
In order to answer to very complex questions, that might require multi-step, multi-documents reasoning, as well as a good document structure comprehension or even statistical analysis, age show great promise as an alternative to the approaches presented so far.

What are agents?

An agent is an autonomous system able to perceive its environnement and execute actions in order to reach a given objective. While the concept of agents is not new and has been at the center of a lot of research in the field of reinforcement learning, it has certainly gotten a fresh look with the arrival of LLMs. We'll focus today on the latter.

Currently, the field of Natural Language Processing (NLP) is investigating how to harness the capabilities of LLMs for complex reasoning and action within dynamic environments. These models are being enhanced with sophisticated memory systems that handle both immediate and long-term information. Moreover, LLMs are being equipped with a suite of 'tools' they can employ to effectively address and resolve the tasks at hand.

Characteristics of LLM-based agentic approaches



Agent overview, from https://lilianweng.github.io/posts/2023-06-23-agent/

Most of the systems that incorporate LLMs as agents have adopted methods that ask the model to articulate its reasoning before actually giving a final answer. Asking the LLM to explain its reasoning gives it "space to think", where space is measured in terms of tokens. Rather than prompting the model for an immediate answer, requesting an explanation has been demonstrated to significantly enhance the model's performance (this is also the principle behind *chain-of-thought*).

Additionally, these systems often utilize a record of past actions to guide their current reasoning. By referencing earlier actions, agents powered by LLMs can refine their responses based on accumulated experiences. This historical insight serves as a form of memory, enriching the model's reference pool for drawing connections, recognizing trends, and enhancing the accuracy of its predictions or choices. Through this iterative process, where the model evaluates its prior outputs and rationales, it can achieve a deeper level of comprehension and more adeptly navigate intricate tasks.

Such techniques include *ReAct, Reflexion, LearnAct etc*. For more details, please refer to our blog post on this matter: Takeaways of : A Path Towards Autonomous Machine Intelligence and The Rise and Potential of Large Language Model Based Agents: A Survey

Use of external tools

With the advent of GPT-4, a novel approach has been developed which involves the concept of "tools" or "function calling." This method entails instructing GPT to utilize a specific "tool", which is then executed externally, and the outcome is subsequently input back into GPT.

For example, we can enhance GPT's capabilities by integrating a calculator tool, thus enabling it to carry out precise mathematical computations.

Consider the following scenario where we request GPT to calculate the product of 7894 and 1144:

1. Answer without tool augmentation

```
from doc_llm.orange import AzureForOrangeEngine
from doc_llm.agents import Agent
from doc_llm.agent_functions.document_functions import tool

engine = AzureForOrangeEngine("dai-semafor-nlp-gpt-4-model-fr")

a = 7894
b = 1144
gpt_answer = engine.query(f"{a} * {b}=?")
#-------------
# True answer: 9030736
# GPT answer: 9024256
```

GPT gets the answer wrong! Which is to be expected, since it is a language model and is not meant to handle calculations. Now, let's give GPT a tool to help it: **a calculator**.

```
from doc_llm.orange import AzureForOrangeEngine
from doc_llm.agents import Agent
from doc_llm.agent_functions.document_functions import tool
```

```
engine = AzureForOrangeEngine("dai-semafor-nlp-gpt-4-model-fr")

a = 7894
b = 1144

# This is our 'calculator' tool. Because we describe what it does, and provide typing hints, GPT will k
@tool(name="multiply", description="multiplies two numbers")
def multiply(a: int, b: int):
    return a*b

# We include the calculator into GPT.
agent = Agent(engine, [multiply])
answer = agent.run(f"{a}*{b}=?", max_actions=2, verbose=True)
print(answer.content)

#-------------
# (Turn 1) The user has provided two numbers, 7894 and 1144, and has requested their product. To find t
# (Turn 2) The multiplication of the numbers 7894 and 1144 has been calculated and the result is 903073
# GPT Answer: 9030736
```

I understand this might feel a bit mysterious. What is actually happening behind the scenes? Let's dive in the different steps that were executed!

Prompt (Turn 1)

The following prompt was created by my framework.

```
Complete the user request: '7894*1144=?'

---
To achieve the user goal you will make good use of the following functions:
- final_answer(your_final_answer: str) -> Gives your final answer to the user
- multiply(a: int, b: int) -> multiplies two numbers
---
Note: You will respect the required types and you will not compose functions.
You have 2 actions left.

You will now answer with an action (using a function) by precisely following this template :

- Explanation: explain the reasoning behind your next action.
- Action: function(argument1, ...)
```

GPT Answer (Turn 1)

```
- Explanation: The user has provided two numbers, 7894 and 1144, and has requested their product
- Action: multiply(7894, 1144)
```

We then parsed the answer, to retrieve the function name as well as the arguments, **and we execute**

the multiply() function on our end, with the provided arguments. This gives us a result; *9030736* that we feed back into GPT (see next prompt).

Prompt (Turn 2)

See how we now include the result of the previous action in the prompt.

```
Complete the user request: '7894*1144=?'


Previous actions:
(1) Explanation: The user has provided two numbers, 7894 and 1144, and has requested their product. To
-> multiply(7894, 1144): 9030736

---
To achieve the user goal you will make good use of the following functions:
- final_answer(your_final_answer: str) -> Gives your final answer to the user
- multiply(a: int, b: int) -> multiplies two numbers
---

Note: You will respect the required types and you will not compose functions.
You have 1 action left.

You will now answer with an action (using a function) by precisely following this template :

- Explanation: explain the reasoning behind your next action.
- Action: function(argument1, ...)
```

GPT Answer (Turn 2)

```
- Explanation: The multiplication of the numbers 7894 and 1144 has been calculated and the result is 90
- Action: final_answer(9030736)
```

The final_answer() function is special. It will stop the loop and output to the user what's inside the parenthesis. The answer provided to the user here is: **9030736**, which is the correct answer.

Note: *All this process of injecting the query, available tools, history, number of actions left etc. is automatically handled by the code behind our Agent object.*

> As we can see, we can define tools to do things that the LLM itself couldn't do. But why are tools so useful for answering questions on documents? Well, it is possible to define some document exploration tools that the agent will then be able to execute and **reason on the output of these**.

Defining tools for document exploration

To answer questions on long document using agents, we define a set of tools that might be useful to retrieve the relevant context and iteratively reason on the data it has seen so far.

For instance, we could define this set of tools:

- **Document reader**: When this tool is called, it will return the content of the full document if it fits in the prompt, if not it will return *'Document is too long for analysis, try read_chunk()'*.
- **Table of contents**: Returns the table of contents of the document
- **Chunk reader**: Returns a chunk content by its ID
- **Page reader**: Returns a page content by its number
- **Words counter**: Returns the number of words between page X & Y
- **Search Engine**: Searches for an expression in a text with for each suggestion the corresponding chunk ID

By giving the LLM access to those tools, we can let it explore the document at its will and let it iteratively reason according to the retrieved information. Here is how one could implement this with code:

```
@tool(name="read_chunk", description="Reads the first chunk of the document", context=my_collection)
def read_chunk(document: str, chunk_number: int = 0):
    return document.chunks[chunk_number].content

@tool(name="read_document", description="Reads the document in full.", context=my_collection)
def read_document(document: str):
    if document.n_tokens > 8000:
        return "Document is too long for analysis, try read_chunk()"
    return document.content

@tool(name="read_article", description="Reads an article by its ID (example: '2' or '8b')", context=my_
def read_article(document: str, article_id: str):
    for chunk in document.chunks:
        article_info = chunk.content.split("\n\n")[1]
        if f"Article {article_id}" in article_info:
            target_chunk = chunk
            break
    return target_chunk

@tool(name="count_words_in_article", description="Counts on many words are present in an article.", con
def count_words_in_article(document: str, article_id: str):
    for chunk in document.chunks:
        article_info = chunk.content.split("\n\n")[1]
        if f"Article {article_id}" in article_info:
            target_chunk = chunk
            break
    return target_chunk.content.count(" ")
```

```python
@tool(name="table_of_contents", description="Returns the table of contents of the document", context=my
def read_table_of_contents(document: str):
    if document.table_of_contents:
        return document.table_of_contents
    return "The document does not contain a table of content."

@tool(name="count_words", description="Counts on many words are present between chunk X and Y in a docu
def count_words(document: str, starting_chunk: int, ending_chunk: int):
    words = 0
    for chunk_number in range(starting_chunk, ending_chunk + 1, 1):
        chunk = document.chunks[chunk_number].content
        words += chunk.count(" ")
    return words
```

Now, when trying to answer a question, the LLM will call several tools iteratively until it gets all the information it needs to provide a final answer. Here is an example of the kind of reasoning the LLM will do and the overall process it follows when using tools. For clarity, I only included the explanation given by the LLM and not the output of the functions, as they can be very long.

```
Question: 'What is Simbow?'

(1) Explanation: First, I will read the document to understand what information it contains.
(2) Explanation: Since the document is too long to be read at once, I will use the read_chunk() functio
(3) Explanation: The document mentions SimBow on page 11. I will use the read_page function to read the
(4) Explanation: The document mentions SimBow on page 11, but it doesn't provide a clear definition. I
(5) Explanation: The search function has returned multiple instances of 'SimBow' being mentioned in the
(6) Explanation: The chunk 10 provides a good explanation of what SimBow is. However, it seems to be cu
(7) Explanation: The previous chunks provided some information about SimBow, but it seems that there is

Final answer: SimBow is a tool based on the word2vec technology that allows semantic similarity calcula
```

Document structure

While the retrieval method used to contextualize the LLM answer is important, another important yet often overlooked aspect is *how* you split your document into multiple chunks. If your chunking method is not adapted, you're fighting an uphill battle and will probably not get very good results.

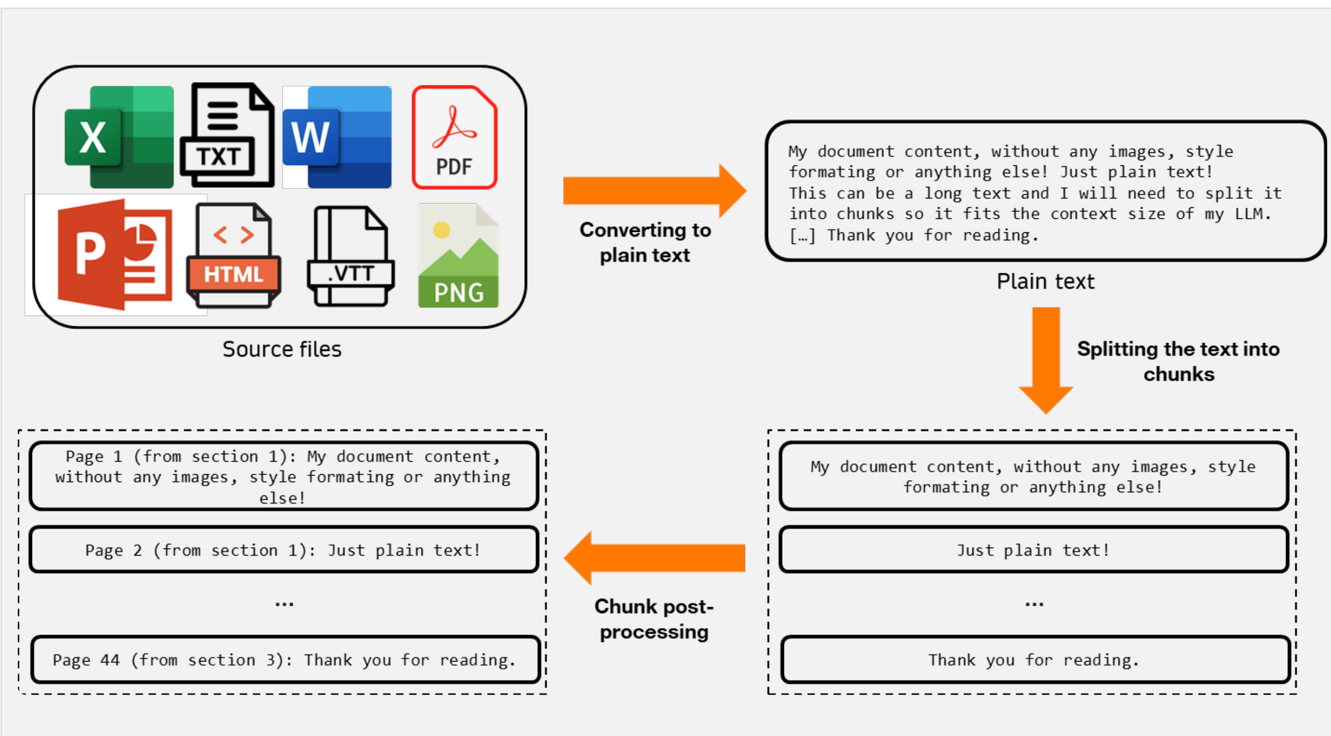
Here are my recommendations for your chunking

- Find the most natural way to split your text into chunks
  - Ideally, one chunk should provide enough context to be understood by itself
  - It should respect the overall structure of the document (titles / sections / annexes etc.)

- Add metadata in the header of your chunk
    - The document name or section name
    - Possibly, a summary of the last few pages to provide context

Of course, extracting the structure from a document is a very hard task. Sometimes, like with the AI Act, some people have already done the job of splitting the text into coherent parts. If your documents always follow the same formating, I highly recommend you take some time to find the best way to split your documents into sound chunks. Most of the time however, we don't have any hints on how to chunk a document and might be tempted to simply split the document by page or block of texts. I would suggest instead to look into tools like Grobid or Marker which might help determining the overall structure of the document automatically.

The type of document you want to work with will strongly make this structuring task more or less difficult. For instance, documents with mostly text, like Word files, are usually easier to turn into plain text for the model to analyze. However, Powerpoint presentations are more challenging because they often use pictures, diagrams, and complex designs that aren't easy to convert into text. When we try to change these presentations into text, we might lose the meaning that was shown through images, animations, and how things are placed on the slides. This is where multimodal models could come handy *(refer to the "Multimodality" section for more information)*



An high-level overview of one way to process documents for RAG

In order to keep this blog post at a reasonable length, I decided to not spend too much time on this part, however, this section could probably be a post of its own since it's such an important (and hard) topic, with much more additional experimentation needed.

A case-study on the AI Act

Now that we've seen a bunch of different methods to answer questions on documents, let's see how each one compares to the other. To answer this question, I prepared a set of 30 questions about the AI Act, of different styles and difficulties, and asked the following systems to answer them:

- **RAG #1**: My own implementation of RAG using DocLLM. The AI Act was split by pages. The retrieval method used is cosine similarity using the 'all-MiniLM-L6-v2' SentenceTransformers model. We retrieve the 5 most similar segments.
- **RAG #2**: The basic RAG implementation proposed by LlamaIndex. The OpenAI embeddings were replaced by the "BAAI/bge-small-en-v1.5" embeddings. The chunks are blocks of 512 tokens.
- **RAG #3**: The basic RAG implementation proposed by LangChain. The OpenAI embeddings were replaced by the "BAAI/bge-small-en-v1.5" embeddings. The chunks are blocks of 512 tokens.
- **RAG #4**: The tool provided in Dinootoo search (https://find.dinootoo.ai.orange). The document is split by pages, the retrieval method used is BM-25. Two variants are proposed: only the first retrieved chunk is included in the LLM context (RAG #4a) or the 5 best retrieved chunks are passed to the context (RAG #4b).
- **GraphRAG #1**: My implementation of GraphRAG in DocLLM, using the Knowledge graph as a proxy to retrieve the most relevant chunks. The knowledge graph is generated with GPT-4. The final answer is generated with GPT-3.5.
- **GraphRAG #2**: LlamaIndex's implementation of GraphRAG using NebulaGraph & a keywords retriever. I couldn't find much more details about the exact implementation
- **Agent (augmented with a structure toolkit)**: My own implementation of agents. The agent is augmented with a tool to read the table of content (list of titles & articles) as well as tools related to structure : read_article(), count_words(), count_items() and compute_expression() which allows the LLM to do math computation. We limit the maximum number of iterations to 5. *We use GPT-4*.

Implementation ID	Framework used	LLM	Chunking Strategy	Prompting Strategy	Retriever	L
RAG #1	DocLLM	GPT-3.5	By page	Including top 5 retrieved chunks	sentence-transformers/all-MiniLM-L6-v2	

Implementation ID	Framework used	LLM	Chunking Strategy	Prompting Strategy	Retriever	L
RAG #2	LlamaIndex	GPT-3.5	By blocks of 512 tokens	Starter example from the LlamaIndex documentation	BAAI/bge-small-en-v1.5	
RAG #3	Langchain	GPT-3.5	By blocks of 512 tokens	Quickstart from the Langchain documentation	BAAI/bge-small-en-v1.5	
RAG #4a	Dinootoo Search	GPT-3.5	By page	Including top 1 retrieved chunk	BM-25	
RAG #4b	Dinootoo Search	GPT-3.5	By page	Including top 5 retrieved chunks	BM-25	
GraphRAG #1	DocLLM	GPT-4 for graph creation / GPT-3.5 for answer generation	By page for Knowledge Graph creation and for prompting	Retrieving the relevant chunks using the knowledge graph as a proxy	sentence-transformers/all-MiniLM-L6-v2	
GraphRAG #2	LlamaIndex	GPT-3.5	Unclear from the documentation	Including the relevant graph triplets in the prompt	Cypher query	
Agents (structure toolkit)	DocLLM agents	GPT-4	By articles	Iterative approach 'ReAct style' with tool use	Table of contents + read_article() tool	

I wrote the questions mostly based on this article and by exploring the AI Act itself. By doing this, I was write the answers I was expecting at the same time, and that I knew were true, which comes handy in order to evaluate the answers.

Questions asked & answers

- **Factual questions**

▶ What is an 'AI System'?

▶ What key elements in the definition of an AI system the document provides clearly differenciate an AI system from any other software?

▶ Can you give a list of the prohibited AI practices described in the document?

▶ What does the AI act say regarding social scoring systems?

▶ What does the AI act say regarding facial regognition?

▶ What is a 'high-risk' AI system?

▶ What requirements must AI systems providers meet?

▼ What obligation do the users have when using AI systems?

System	Answer
Reference answer	Deployers, formerly known as users of high-risk AI systems, have several obligations, including:- Using the high-risk AI system in accordance with the provider's instructions.- Implementing human oversight where possible.- Monitoring the input data and the operation of the system.- Retaining automated logs for at least six months.
RAG #2 (LlamaIndex)	The context information does not provide a clear answer to this query. However, according to the Guidelines of HLEG, AI systems should be developed and used in a way that can be appropriately controlled and overseen by humans, and that respects human dignity and personal autonomy. Additionally, AI systems should be developed and used in compliance with existing privacy and data protection rules, while avoiding discriminatory impacts and unfair biases that are prohibited by Union or national law. Therefore, it can be inferred that users of AI systems have an obligation to ensure that the systems they use are developed and used in a way that aligns with these principles.
RAG #3 (Langchain)	Users of AI systems have an obligation to ensure a sufficient level of AI literacy of their staff and individuals involved in the operation and use of the AI systems. Deployers of high-risk AI systems must take appropriate measu ensure compliance with instructions for use. Providers of AI systems must specify necessary information, capabilities, technical access, and other

System	Answer
	assistance to enable compliance with obligations set out in regulations.
RAG #1 (DocLLM)	The documents do not provide information on the specific obligations of users when using AI systems. However, they do outline the obligations of providers and deployers of AI systems, including the need for conformity assessments, risk assessments, and human oversight measures. Additionally, the documents encourage providers and deployers of all AI systems, high-risk or not, to apply on a voluntary basis additional requirements related to ethical guidelines, environmental sustainability, inclusive and diverse design, and stakeholder participation.
GraphRAG #1 (DocLLM)	According to the information provided in the document, there is no specific obligation mentioned for users of AI systems. However, Article 4b of Title I states that providers and deployers of AI systems should ensure that their staff and other persons dealing with the operation and use of AI systems on their behalf are sufficiently trained and informed about the AI systems they are using. This implies that users of AI systems should also be adequately trained and informed about the AI systems they are using. Therefore, it is recommended that users of AI systems should seek proper training and information about the AI systems they are using to ensure their safe and effective use. (Source: European Commission. "Proposal for a Regulation Laying Down Harmonized Rules on Artificial Intelligence (Artificial Intelligence Act)." Brussels, 21 April 2021, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021PC0201.)
GraphRAG #2 (LlamaIndex)	The context information does not provide a clear answer to this query as it is too broad. The Guidelines of HLEG mention several principles that should be followed when developing and using AI systems, including human agency and oversight, technical robustness and safety, privacy and data governance, transparency, diversity, non-discrimination and fairness, and social and environmental well-being. The Regulation also outlines various obligations for providers of AI systems, such as ensuring compliance with the Regulation, providing information and documentation upon request, and cooperating with the AI Office and national competent authorities. However, the context information does not specify any obligations for users of AI systems.
Agent (structure toolkit)	Users, referred to as 'deployers' in the context of high-risk AI systems, h several obligations. These include: 1. Ensuring the use of such systems in accordance with the instructions of use.2. Assigning human oversight to competent individuals.3. Ensuring input data is relevant and representative for

System	Answer
	the intended purpose of the AI system.4. Monitoring the operation of the AI system and informing providers, distributors, and relevant authorities if there's a risk or serious incident.5. Keeping logs generated by the AI system for at least six months.6. Informing workers and public authorities about the use of high-risk AI systems.7. Complying with registration obligations.8. Carrying out a data protection impact assessment.9. Requesting authorization for the use of AI systems for post remote biometric identification in the framework of a criminal investigation.10. Informing natural persons when they are subject to the use of the high-risk AI system.11. Cooperating with relevant national competent authorities.These obligations are outlined in Article 29: Obligations of Deployers of High-Risk AI Systems.
RAG #4a (Dinootoo Search, k=1)	Sorry, I am not able to write a relevant response based on this document.
RAG #4b (Dinootoo Search, k=5)	The users of AI systems have the obligation to comply with the requirements of the Regulation.

▸ What are public sector bodies required to do?

▸ If an entity puts their name or trademark on an AI system, what impact does that have on their responsibilities under the AI Act?

▸ Can trade secrets be used as a legitimate reason to refuse disclosure of an algorithm used in high-risk AI systems?

▸ Who is granted the right to lodge a complaint under the AI Act?

▸ What can be said about GPAI models?

▾ What is the difference between a GPAI model and an AI system?

System	Answer
Reference answer	The difference between a General Purpose AI (GPAI) model and an AI system as defined by the AI Act is that a GPAI model is a foundational component that can be used to build various AI systems, but in itself, it is not considered an AI system. The AI Act treats GPAI models and AI systems under different regulatory frameworks. While a GPAI model is a more general tool o framework that can be adapted for different uses, an AI system refers to a specific application of AI technology designed for a particular function or set of tasks. Consequently, a GPAI model will never be classified as a high-risk AI

System	Answer
	system, but a GPAI system—created using a GPAI model—could be considered high-risk depending on its application and impact. Providers of GPAI models have obligations such as maintaining technical documentation and respecting copyright law, with additional responsibilities for those models that pose systemic risks, including risk assessments and incident reporting.
RAG #2 (LlamaIndex)	The context information does not provide a direct comparison between a GPAI model and an AI system. It only defines the terms "AI system" and "general purpose AI model" separately. An AI system is defined as a machine-based system designed to operate with varying levels of autonomy and that may exhibit adaptiveness after deployment, while a general purpose AI model is not explicitly defined in the given context.
RAG #3 (Langchain)	The compromise agreement defines GPAI systems as AI systems based on GPAI models. The AI Office will evaluate codes of practice for these systems and may provide common rules for implementation if necessary. GPAI models may present systemic risks and have horizontal obligations for compliance, including providing technical documentation and information to downstream providers.
RAG #1 (DocLLM)	According to the documents provided, a general purpose AI system is an AI system which is based on a general purpose AI model, that has the capability to serve a variety of purposes, both for direct use as well as for integration in other AI systems. On the other hand, a GPAI model refers to a specific type of AI model that is considered high-risk and subject to additional regulations and oversight. Therefore, the main difference between a GPAI model and an AI system is that the former is a specific type of AI model while the latter refers to a broader category of AI systems that can be based on various types of AI models, including GPAI models.

System	Answer
GraphRAG #1 (DocLLM)	According to the EU's AI regulation, a GPAI model is a type of AI model that is capable of performing a wide range of tasks and can be integrated into various systems, while an AI system is a machine-based system that generates outputs based on the input it receives (Answer 1). The regulation distinguishes between general purpose AI (GPAI) models and AI systems, with the AI office having powers to monitor and supervise compliance of AI systems based on a general purpose AI model (Article 63a). Additionally, criteria for classifying a general-purpose AI model as having systemic risk are defined in Article 52a. Therefore, a GPAI model is a type of AI system, but not all AI systems are based on a general purpose AI model (Answer 2).
GraphRAG #2 (LlamaIndex)	According to the provided context information, a general purpose AI model is a component of an AI system and does not constitute an AI system on its own. AI models require the addition of further components, such as a user interface, to become AI systems. The definition of a general purpose AI model should be based on the key functional characteristics of the model, in particular, the generality and the capability to competently perform a wide range of distinct tasks. On the other hand, an AI system is a machine-based system designed to operate with varying levels of autonomy and that may exhibit adaptiveness after deployment. It infers, from the input it receives, how to generate outputs such as predictions, content, recommendations, or decisions that can influence physical or virtual environments.
Agent (structure toolkit)	The document reviewed did not provide specific information about the difference between a GPAI model and an AI system. Further reading or resources may be needed to fully understand the difference between these two concepts.
RAG #4a (Dinootoo Search, k=1)	According to the context, a GPAI model is a specific type of AI model that is used as a basis for GPAI systems. GPAI systems, on the other hand, are AI systems that are built using GPAI models and can serve a variety of purposes. In other words, a GPAI model is a component of a GPAI system, which is a broader term encompassing the entire AI system.
RAG #4b (Dinootoo Search, k=5)	The difference between a GPAI model and an AI system is that GPAI models are a type of AI system that can serve a variety of purposes. GPAI mod have specific obligations, such as keeping up-to-date and making availa technical documentation, providing information to downstream providers for compliance, and performing model evaluation and risk assessments for

System	Answer
	models with systemic risks. On the other hand, AI systems refer to a broader category of artificial intelligence systems that can include various types of models and applications.

- ▸ What are the regulations around models finetuning?
- ▸ What use-cases would be considered 'high-risk'?
- ▸ What are the penalties for non-compliance to the AI Act?
- ▸ When will the AI Act be enforced?
- ▸ Must US-based companies comply with the AI Act?
- ▸ What kind of information should I provide about my AI system?
- ▸ Am I a provider if I use a foundation model but use RAG?
- ▾ What must I keep for 10 years?

System	Answer
Reference answer	As a provider, you must keep the following documentation for a period of 10 years after the AI system has been placed on the market or put into service:(a) The technical documentation as specified in Article 11 of the AI Act. (b) Documentation concerning the quality management system as referred to in Article 17. (c) Documentation concerning any changes approved by notified bodies, if applicable. (d) Decisions and other documents issued by notified bodies, if applicable. (e) The EU declaration of conformity as mentioned in Article 48.Additionally, Member States will set conditions to ensure that this documentation remains available to national competent authorities for the full 10-year period, even if the provider goes bankrupt or ceases activity before the end of that period.For providers that are financial institutions and are subject to Union financial services legislation, the maintenance of the technical documentation should be integrated into the documentation kept under the relevant financial services legislation.
RAG #2 (LlamaIndex)	The context information does not provide any clear indication of what must be kept for 10 years.
RAG #3 (Langchain)	Importers must keep a copy of the certificate issued by the notified body, instructions for use, and the EU declaration of conformity for 10 years after the AI system has been placed on the market or put into service. This is i accordance with the requirements set out in Chapter 2 of the Title.

System	Answer
RAG #1 (DocLLM)	There is no specific document mentioned in the provided documents that must be kept for 10 years. However, according to Article 13 of the AI Act, providers of high-risk AI systems must keep the technical documentation for a period of 10 years after the AI system has been placed on the market.
GraphRAG #1 (DocLLM)	If you are a provider of a high-risk AI system, you must keep a written machine-readable, physical or electronically signed EU declaration of conformity for each high-risk AI system and keep it at the disposal of the national competent authorities for 10 years after the AI high-risk system has been placed on the market or put into service. This is stated in Article 48(1) of 'Title III: High-Risk AI System'. However, if there is no information provided in the given context, it is unclear what must be kept for 10 years.
GraphRAG #2 (LlamaIndex)	The provider of an AI system must keep the technical documentation and the documentation concerning the quality management system for a period ending 10 years after the AI system has been placed on the market or put into service, as stated in Article 18 of the AI Act.
Agent (structure toolkit)	According to Article 18: Documentation Keeping of the 'ai_act_data/ai_act_by_pages.json' document, the provider of an AI system must keep the following for a period of 10 years after the AI system has been placed on the market or put into service: technical documentation, quality management system documentation, documentation concerning changes approved by notified bodies, decisions and other documents issued by the notified bodies, and the EU declaration of conformity.
RAG #4a (Dinootoo Search, k=1)	According to the context, you must keep the technical documentation referred to in Article 11 and the documentation concerning the quality management system referred to in Article 17 for a period of 10 years after the AI system has been placed on the market or put into service.
RAG #4b (Dinootoo Search, k=5)	You must keep the technical documentation referred to in Article 11 and the documentation concerning the quality management system referred to in Article 17 for a period of 10 years.

▸ What are my obligations when it comes to the transparency of my AI system?

- **Broad questions**

▾ What is this document about?

System	Answer
Reference answer	This document is about the AI Act, a proposed European Union regulation on artificial intelligence in the European Union. Proposed by the European Commission on 21 April 2021 and not yet enacted, it would introduce a common regulatory and legal framework for artificial intelligence. Its scope would encompass all types of artificial intelligence in a broad range of sectors (exceptions include AI systems used solely for military, national security, research, and non-professional purpose). As a piece of product regulation, it would not confer rights on individuals, but would regulate the providers of AI systems, and entities using AI in a professional context.The proposed EU Artificial Intelligence Act aims to classify and regulate artificial intelligence applications based on their risk to cause harm.
RAG #2 (LlamaIndex)	The document is about a proposal for a Regulation laying down harmonised rules on artificial intelligence (the AI Act) and its main elements, including subject matter and scope, technical documentation, and processing of personal data.
RAG #3 (Langchain)	The document is about technical documentation referred to in Annex IV, which includes documentation concerning changes approved by notified bodies, decisions and other documents issued by notified bodies, and the EU declaration of conformity referred to in Article 48. The Member States determine the conditions under which the documentation remains at the disposal of national competent authorities. The document also outlines systems and procedures for record-keeping and resource management.
RAG #1 (DocLLM)	There are multiple documents provided. The first document (AI_Act.pdf-129) contains information on the technical documentation that providers of high-risk AI systems must maintain, as well as the automatically generated logs that must be kept. The second document (AI_Act.pdf-10) is a proposal for a regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence. The third document (AI_Act.pdf-1) provides information on the negotiation process and main elements of the compromise for the proposed regulation. The fourth document (AI_Act.pdf-172) contains information on codes of practice for the proper application of the regulation.
GraphRAG #1 (DocLLM)	This document is about the regulation of artificial intelligence (AI) systen the European Union. It includes provisions for high-risk AI systems, technical documentation, conformity assessment, post-market monitoring, and

System	Answer
	evaluation and review. The document also addresses the obligations of providers of high-risk and general-purpose AI models, as well as the powers of authorities protecting fundamental rights. The document includes several articles that amend existing EU directives and regulations to take into account the regulation of AI systems. (Various chunks from multiple articles in the document provide this information.)
GraphRAG #2 (LlamaIndex)	This document is about a proposal for a Regulation laying down harmonised rules on artificial intelligence (the AI Act) and its main elements, including subject matter and scope, technical documentation, alerts of systemic risks by the scientific panel, power to request documentation and information, and power to conduct evaluations. The purpose of the AI Act is to ensure a high level of protection of health, safety, and fundamental rights enshrined in the Charter, which includes democracy, rule of law, and environmental protection. The scope of the AI Act excludes national security.
Agent (structure toolkit)	The document is a regulation providing definitions and guidelines for the use of AI systems. It applies to various entities involved with AI systems and outlines exceptions for when it does not apply. It also emphasizes the importance of AI literacy among staff and other persons dealing with the operation and use of AI systems. The document also outlines prohibited AI practices, such as the use of AI systems that deploy manipulative techniques, exploit vulnerabilities of certain groups, or use biometric data to infer sensitive information. It also provides guidelines for the use of ‘real-time’ remote biometric identification systems in publicly accessible spaces for the purpose of law enforcement, including the need for prior authorization and notification to relevant authorities.
RAG #4a (Dinootoo Search, k=1)	This document is about the design specifications and architecture of an AI system, including the logic, algorithms, and key design choices. It also covers the system’s intended use, classification choices, optimization goals, output description, and trade-offs made to comply with requirements. Additionally, it mentions the data requirements, including training methodologies, data sets, and labeling procedures.
RAG #4b (Dinootoo Search, k=5)	This document is about the AI Act.

▸ Who is the target audience for this document?

▾ Why is this document important?

System	Answer
Reference answer	This document is important as it tries to regulate an emerging technology: Artificial Intelligence, which could have a tremendous impact for the citizens of the EU. If unregulated, AI could pose risks for people, companies or even institutions and democracy as a whole. It's a major breakthrough when it comes to AI regulation in Europe.
RAG #2 (LlamaIndex)	The document is important because it outlines regulations and requirements for the development and use of high-risk AI systems, including the need for technical documentation and safeguards to protect personal data. It also addresses the need for providers of general purpose AI models to respect Union copyright law and make publicly available a summary of the content used for training.
RAG #3 (Langchain)	The document is important because it includes technical documentation referred to in Annex IV, which is necessary to demonstrate the conformity of a high-risk AI system with the requirements set out in Chapter 2 of the Title. The document also outlines the obligation of the manufacturer of products containing high-risk AI systems to provide documentation and information upon request from a national competent authority. Additionally, the document specifies that the scientific panel may consider any other relevant information gathered on its own initiative.
RAG #1 (DocLLM)	I'm sorry, but I cannot answer this question without more specific information about which document you are referring to. Please provide the document number or title.

System	Answer
GraphRAG #1 (DocLLM)	This document, as described in Answer 1, is important because it establishes harmonized rules for the use of artificial intelligence systems in the European Union, with a focus on promoting human-centric and trustworthy AI while ensuring protection of fundamental rights and safety. It also includes specific requirements for high-risk AI systems and establishes a scientific panel of independent experts to support enforcement activities. Additionally, as mentioned in Answer 2, the document contains amendments to various regulations and directives related to AI systems, outlining the requirements for delegated and implementing acts. This information was found in the official press release from the European Commission on the adoption of the AI regulation.
GraphRAG #2 (LlamaIndex)	The document is important because it contains the compromise text of the AI Act, which aims to ensure a high level of protection of health, safety, and fundamental rights enshrined in the Charter, including democracy, rule of law, and environmental protection. It sets out requirements and obligations for the development and use of high-risk AI systems, including the processing of special categories of personal data, technical documentation, and transparency obligations for providers of general purpose AI models. The document also establishes a scientific panel to provide alerts of systemic risks posed by general purpose AI models and empowers the Commission to adopt implementing acts on the application of the provisions on the labeling and detection of artificially generated or manipulated content.
Agent (structure toolkit)	The document is important because it provides detailed and comprehensive guidelines and regulations for the use of AI systems, aiming to ensure their safe and ethical use.
RAG #4a (Dinootoo Search, k=1)	This document is important because it outlines the necessary safeguards and controls for the access and processing of personal data in high-risk AI systems. It emphasizes the need for strict documentation and confidentiality obligations to prevent misuse of personal data. Additionally, it specifies that special categories of personal data should not be transmitted or accessed by other parties and should be deleted once the bias has been corrected or the data has reached the end of its retention period. The document also highlights the requirement to justify the processing of special categories of personal data for the purpose of detecting and correcting biases. Furthermore, it states technical documentation for high-risk AI systems must be prepared and kept up-to-date before the system is placed on the market or put into service.

System	Answer
RAG #4b (Dinootoo Search, k=5)	This document is important because it provides regulations and guidelines for the development and use of AI systems, particularly those that are considered high-risk. It covers various aspects such as data protection, technical documentation, alerts of systemic risks, and post-market monitoring. The document aims to ensure the responsible and accountable use of AI systems, promote transparency and fairness, and protect individuals' rights and freedoms.

▸ How does the AI Act contribute to the development of trustworthy AI?
▸ What impact does the AI Act have on innovation and competitiveness in the AI sector?
▸ What role does the AI Act play in setting global standards for AI systems?
▸ What are the core values that the AI Act seeks to protect?

- **Structure-related questions**

▸ How many articles is the AI act made of?
▸ How many words in the chapter dedicated to Governance?
▾ Is chapter VI longer than chapter IX?

System	Answer
Reference answer	Yes, chapter VI is longer than chapter IX: chapter VI is made of around 2300 words, whereas chapter IX is made of 340 words.
RAG #2 (LlamaIndex)	I cannot answer the query as there is no information provided about the length of chapter VI or chapter IX. The context information only includes articles and sections of a regulation related to AI models and the delegation of power to the Commission.
RAG #3 (Langchain)	I don't know.
RAG #1 (DocLLM)	I'm sorry, but I cannot answer that question as there is no Chapter VI or Chapter IX mentioned in the provided documents. The documents only contain references to specific articles and annexes within the regulation.
GraphRAG #1 (DocLLM)	Based on the information provided in the document, it is not possible [illegible] determine whether chapter VI is longer than chapter IX. As stated in Answer 1 and Answer 2, the document does not contain information on the length of these chapters. Therefore, it is not possible to make a comparison between

System	Answer
	the two. (Source: None provided)
GraphRAG #2 (LlamaIndex)	I cannot determine whether chapter VI is longer than chapter IX based on the provided context information. The information provided includes various articles and sections within different chapters of the AI Act, but does not provide a comprehensive overview of the length of each chapter.
Agent (structure toolkit)	Yes, chapter VI is longer than chapter IX.
RAG #4a (Dinootoo Search, k=1)	Sorry, I am not able to write a relevant response based on this document.
RAG #4b (Dinootoo Search, k=5)	Sorry, I am not able to write a relevant response based on theses firsts documents.

- ▸ What does the very last article talk about?
- ▸ Can you summarize the article that talks about AI Regulatory Sandboxes?
- ▸ An article is dedicated to definitions, how many definitions are there?
- ▸ In which articles are there mentions of biometry?

Evaluation

Evaluating the results is a complex task firstly due to the large number of answers to analyze (240 answers here). Moreover, judging the quality of each answer depends on the criteria you prioritize, such as its relevance to the question, its conciseness, its completeness etc. Some answers may appear similar but differ on which aspect they focus on or the amount of detail they provide. This evaluation process is highly subjective, as two different people might have different expectations or value different aspects of the answer differently. Please keep that in mind when reading the following.

Qualitative evaluation

Firstly, we can try to give a qualitative evaluation of the different answers given by the different systems. For that, I've read all the answers carefully and tried to identify answer trends for each system. Remember that those are just my first impressions, and that I'm not an expert on the AI Act, so judging the quality of the answers can sometimes be a bit hard, even with a **Reference answer**.

RAG system

- **RAG #1 (DocLLM)** Works rather well, the answers from this system are pretty good when the relevant chunks have been correctly retrieved, which is not always the case. However, even if there is an answer, it can sometimes be lacking some details or context.
- **RAG #2 (LlamaIndex)** Same remarks as for RAG #1
- **RAG #3 (Langchain) ** Pretty similar to RAG #1 & RAG #2.
- **RAG #4 (BM-25)** While it’s the cheapest & fastest system of all, the answers from RAG #4 are often incomplete when the questions are too broad or simply too difficult. It often provides only limited information about the subject at hand and we’re left wanting to know more. Moreover, the retriever sometimes fail to retrieve the relevant(s) chunk(s) when the answer is provided in the document.

GraphRAG Systems

- **GraphRAG #1 (DocLLM)** Works very well except for very open-ended questions & strucural questions. Most of the time, it offers complete and nuanced answers that can span accross all the document.
- **GraphRAG #2 (LlamaIndex)** Also works really well, with complete answers.

Agent sytem

- **Agent with structure toolkit (DocLLM):** When it’s able to use the right tools and to follow a correct logic flow, it answers with very complete yet concise answers. However, they are very finicky with the provided tools & how they’re described. The version I used here was not provided tools that would help boost its performances, like an advanced search engine for instance.

Overall, it’s hard to say that one system is best for everything. There has always been a case where one system failed where other succeeded and inversely. My ‘vibe-check’ is that the GraphRAG systems often offer more complete & nuanced answers compared to baseline RAG. And while I may be biased, I feel like my DocLLM GraphRAG approach is very robust to a lot of questions. However, neither RAG or GraphRAG are able to answer questions that relate to the structure of the document. This is were the agentic approaches truly shine, as they can be provided tools to overcome some of the drawbacks of LLMs (counting, rigorous reasoning etc.). While the potential of agents may not have come accross very well in this test because most questions were very fact-based, I think they’re a very promising approach to document exploration. They are also not opposed to RAG or GraphRAG methods! Actually, we *can* include RAG systems as a tool. You can imagine the agent decomposing a hard question into easier subquestions and then use these RAG/GraphRAG systems to gather information and then synthetize them.

Quantitative evaluation

In order to evaluate the results in a more quantitative way, I mostly chose to ask questions whose answers could be found in the content of this blog article: The EU Artificial Intelligence Act: our 16 key takeaways. I created *gold* answers by extracting and slightly editing passages from the article. For questions that did not find a direct answer in the blog, I wrote the answers myself by researching information in the AI Act itself or from other various sources.

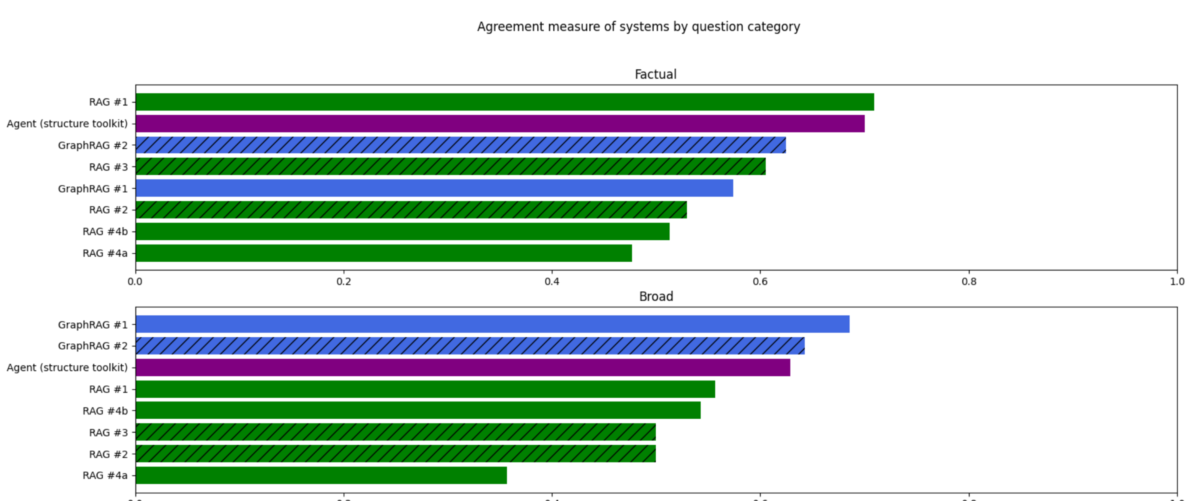
Response rate

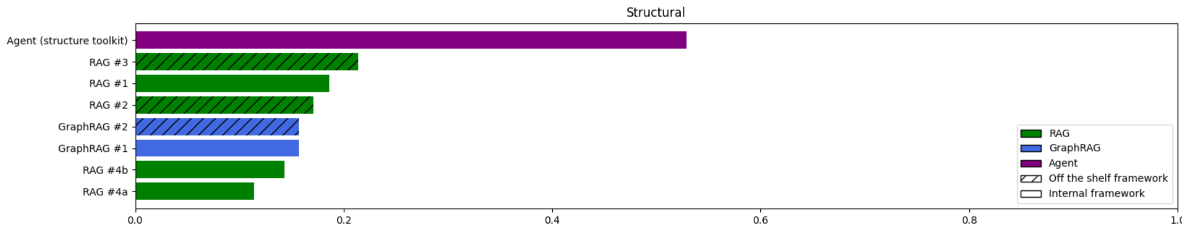
An interesting characteristic to evaluate is the response rate of our systems. It indicates how often the system provides a response to a question (not necessarly a correct response). If the system replies with a message like "I'm sorry, I don't have enough context to answer that question," we count that as a 'non-answer'. However, if it gives any information related to the question, that's considered an answer. To calculate the response rate, we divide the number of questions that received answers by the total number of questions asked. You can check the results in the result table.

Agreement measure

I decided to use the TruLens framework in order to automatically score the answers. The metric used is the "agreement" between the generated answer and the gold answer, the annotator is GPT-4. While evaluating generated answers by a generative model may sound like a weird idea, it's actually getting very popular for open-ended generation tasks, because classical metrics like ROUGE or METEOR don't work well at all. I looked into the scores given by GPT-4, and I was mostly agreeing with the results.

While not perfect (notably because it will favor answers that are the same length as the reference answer), this is at least a first way to try and rank the quality of the answers provided by each system.





For more information about the implementation details of all these systems, please refer to the table in the blog post

System	Average answer length (number of words)	Response rate (%)	Agreement measure (%)
RAG #1	116 / 91 / 55	**95** / 86 / 42	**71** / 56 / 19
RAG #2	86 / 79 / 46	83 / 86 / 28	53 / 50 / 17
RAG #3	74 / 79 / 24	91 / **100** / 29	61 / 50 / 21
RAG #4a	54 / 78 / 27	70 / 71 / 14	48 / 36 / 11
RAG #4b	41 / 48 / 35	91 / **100** / 14	51 / 54 / 14
GraphRAG #1	121 / 102 / 78	82 / **100** / 57	57 / **69** / 16
GraphRAG #2	106 / 100 / 36	87 / 86 / 29	63 / 64 / 16
Agent (structure toolkit)	115 / 95 / 34	91 / **100** / **100**	70 / 63 / **53**

Note: The statistics follow this format: factual / broad / structural. Best values are **underlined**.

Cost & time

When evaluating these models, it's also important to consider their operational costs and execution times. Indeed, all the techniques described above strongly vary in terms of cost and time. With pay-as-you-go services like the GPT API, **both are proportional to the amount of tokens used in total to answer the question.**

RAG cost depends on the number of retrieved chunks as well as on the sizes of said chunks, bu usually pretty fast and cheap. It also depends on the retrieval method used & the way you store your vectors (if you work with vectors), though these factors typically have a negligible impact on the overall time compared to the duration the LLM takes to respond.

The same can be said about GraphRAG. When asked to answer a new question, it is not necessary more costly compared to baseline RAG. However, creating the Knowledge Graph with a LLM definitly has a cost, which depends on the LLM you use (GPT-4 is about 20x more costly than GPT-3.5 for instance), on the number of documents to analyze as well as their length.

Agents however, are usually a lot more costly than those two methods because it takes multiple turns to get an answer. If we provide the agent the list of all its previous actions & the associated outputs, the cost grows exponentially each turn, as the number of context tokens also grows each turn.

Here are a few graphs from my experiment on the AI Act (read next section for more details), to help you put in perspective the differences in cost:

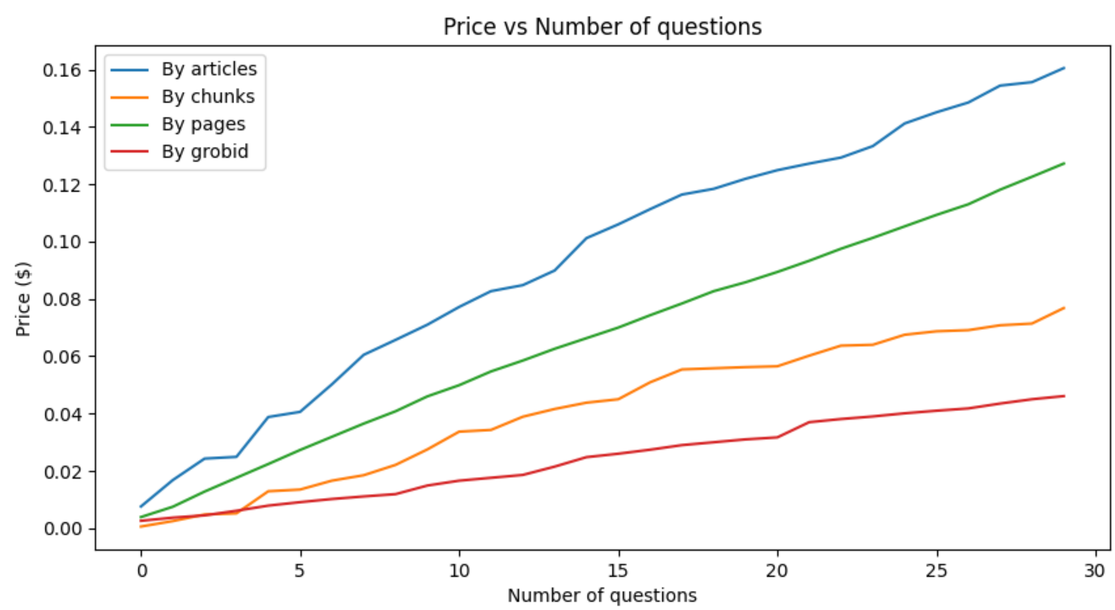
- **Approaches:**: The approach you choose to use has a big impact on the price. (GPT-4 was used as the LLM for the agent here, with GPT-3.5 the final price would have been around 0.5$, but it's not working well at all, so I decided not to include it.)

System	Cost at initialization ($)	Cost per 100 questions ($)
RAG (GPT-3.5)	0	0.42
GraphRAG (Graph initialization: GPT-4)	3.9	0.38
Agents (GPT-4)	0	20.81

Of course, we could explore several ways to help decrease the associated of these methods, for instance one could try to:

1. Summarize the context provided to the model with a lighter model (BART, T5)
2. Finetune a smaller model (like Llama or Mistral) to a specific approach. Some models work as well as GPT-4 with only 13b parameters when finetuned on the right data. See NexusRaven-V2 for example.
3. Run some tasks in parallel.

Chunking strategies: Different chunking strategies will split the document with different granularities. If you always retrieve the same amount of chunks, you'll use more tokens with big chunks. The following comes from experiments of different chunkings strategies using DocLLM (we did not evaluate the results from other frameworks yet, but they should be similar).



Evolution of cost for different chunking strategies as the number of questions asked grows

A few thoughts

So, what lessons did we learn here?

I think the first one is probably the fact that evaluating this kind of stuff is hard. Between all the different approaches, chunking strategies, embedding models, LLMs, prompts used, tools given to the agent, post-processing techniques and frameworks implementations that can be used in combination with each other, it gives us a *very* big parameters space to explore. To that, add the fact that the metrics we use to automatically evaluate performance don't necessarily always perfectly align with human judgment (which itself can vary from person to person) and you get a very complex problem to evaluate. This complexity makes it challenging to draw clear conclusions about the effectiveness of these systems. The list of systems tested in this study is not comprehensive, and there are many more that could be tested. However, including more experiments would make this post excessively long. The goal here is to provide a high-level overview of the current possibilities and to frame the problem in a way that promotes good practices for future work.

Still, even with all that said, I think we can pick up some interesting trends about the results:

1. **About the answers quality**: GraphRAG works great for most questions, RAG systems can work well on factual questions but are sensitive to the configuration & agents can work very well for

some questions, but are still finicky and need more work to gain in stability.

2. **About the cost of the systems**: All these systems have very different costs. As we saw in the "Cost & time" section , some systems (notably, agent-based systems) can be very costly, and definitly cannot be deployed at scale. The same might also be said about GraphRAG, which can *also* become expensive, especially if we generate the Graph for our documents with an expensive model like GPT-4. This means that there is a trade-off between performance and cost. For instance, the **BM-25 RAG** system may have lower performance, but it's **designed to be cost-effective and fast, with shorter answer lengths and a speed-optimized retrieval system.**
3. **About frameworks**: It seems like Langchain and LlamaIndex got pretty similar performances for RAG. I'm not sure I would strongly recommend one over the other. I do have a slight preference for LlamaIndex, as I feel like the documentation is a lot clearer and the whole framework might be a bit more stable. But if your goal is to implement a RAG system, you can definitly work with either of them (or even reimplement a RAG system from scratch).

Here are the answers to a few additional questions one could ask, that I did not really know where to include:

- **What about using GPT-3.5 for agents?** It could probably work, but I've had trouble getting GPT-3.5 to work consistently with function calling, it often outputs wrongly formated functions or arguments. A better prompt might improve the results, but I did not try it yet. Agents-based systems can be very finicky, and without a strong model like GPT-4, it might be hard to get anything of value.
- **What about using GPT-3.5 for knowledge graph creation?** It works! I tested also tested GraphRAG with on the AI Act knowledge graph being generated with GPT-3.5, and the answer were extremely similar.
- **Did you try using other LLMs (like Claude or Gemini)? Open-source models? Finetuning?** No I did not yet. It could be interesting to try though! I'm also interested in open-source models finetuning, but I have nothing to share *yet*.
- **Did you try using Langchain agents implementation?** I did not, but I know a few people in the NEPAL research group are experimenting with it.
- **Did you measure the impact of chunking & of the embedding model used?** Yes! Although it would be probably be nice to test more recent & bigger retrieval methods. Those I used pretty are rather old. In summary: Chunking by pages seem to work well (surprisingly, better than chunking by articles), chunking by blocks of tokens is clearly suboptimal. The chunking provided by Grobid here seemed not to work very well (it might be linked to the document format). Between MiniLM and BGE, MiniLM seems to be working better in our case.

Of course, if you have other questions, please send me a message!

What's next?

The end of RAG?

In the middle of writing this article, Google unveiled Gemini 1.5, a model with an impressive context length capability of several million tokens, equivalent to thousands of pages, while maintaining precise information retrieval.

In the meantime, we see emerging machine-learning architectures like Mamba, which also show promise in processing long documents.

Given that models like Gemini 1.5 can handle entire documents in a single prompt due to their large context window, the need for the techniques discussed in this article is reduced. Nonetheless, while RAG as a technique might disappear in the next years, I think it's important to keep working on it for now as :

- Large context models are not available for Orange yet.
- Processing millions of tokens per request can be time-intensive (taking several minutes for a response from Gemini 1.5) and expensive.
- The skills involved in splitting and indexing documents have inherent value.

As models get better and operate with increased autonomy, the concept of *agents* becomes more and more relevant. Thus, getting familiar with the state-of-the-art techniques involved with agents is important to understand where the technology is going and the potential applications that may emerge in the coming years.

Multimodality

One important aspect I did not mention in this article is multimodality. With multimodal models (able to process texts, images and audios), we would be able to not only retrieve information from the text of the documents, but also from illustrations and from the overall layout of the document. This would be very useful to process Powerpoint presentation or technical documents with many meaningful illustrations or graphs.

We already have large multimodal models with for instance Llava (open-source), GPT-4V or Ge[illegible] While they work rather well for some tasks, they're still showing limitations when presented with complex layouts or technical illustrations. But the trend is clear; models are getting multimodal and will continue improving in the next years.

There's also a field of smaller models that are trained specifically for document processing: see *UDOP*, *Nougat*, *Donut*, *DocLLM* etc.

In conclusion

In this post, we've explored various methods for answering questions based on documents. We've seen that it is important to select the appropriate approach for each type of question, as there is no one-size-fits-all solution. Instead, there is a range of strategies designed to meet diverse requirements and complexities in document analysis. The effectiveness of these methods is influenced by our understanding of the document structure and the performance of the retrieval models used.

We emphasized the unique advantages and constraints of each approach and highlighted the fact that a single method cannot address all scenarios. Instead, a spectrum of techniques is necessary to handle the varying demands of document analysis.

When trying to assess which system works best for what, it is very important that we can accurately evaluate the strength of each system. With this kind of open-generation tasks, and especially when working with long documents, it becomes very expensive to do this evaluation by hand. However, traditionnal automatic evaluation metrics fail us here to, which is why we must continue working on finding new ways to evaluate our systems at scale.

As generative AI continues to advance, our document processing methodologies are bound to evolve. We anticipate that in the coming years, we'll be able to effectively manage long, multimodal documents that currently pose challenges, potentially transforming the landscape of document analysis.

Future updates : I will probably post an update on this post with a more complete list of experiments, especially when it comes to agents. It would be interesting to add new powerful tools for document analysis, let the agent create its own tools or play more generally with the agent logic.

agents Documents llms Tools RAG GraphRAG Evaluation AI Act

© 2024 NEPAL LM Weekly Blog Powered by Hugo & PaperMod