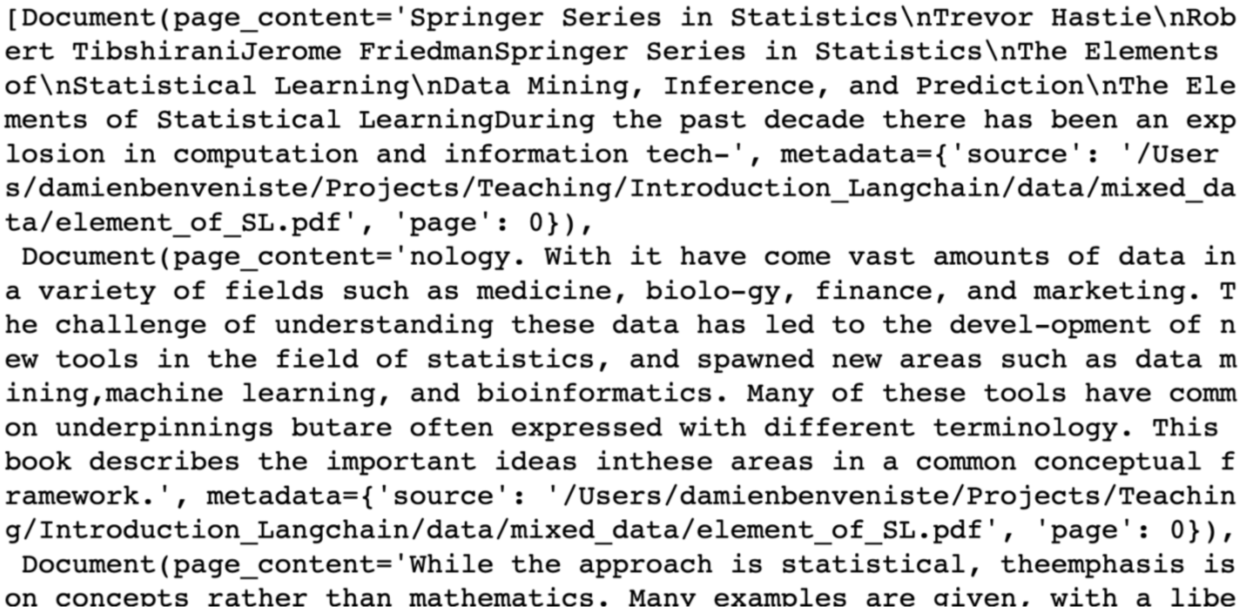
Retrieval Augmented Generation

Indexing data

Let’s load and split the pdf file of Element of Statistical Learning:

```
from langchain.document_loaders import PyPDFLoader
from langchain.text_splitter import RecursiveCharacterTextSplitter

file_path = '...'

loader = PyPDFLoader(file_path=file_path)

text_splitter = RecursiveCharacterTextSplitter(
    chunk_size=500,
    chunk_overlap=0
)

data = loader.load_and_split(text_splitter=text_splitter)
data
```



```
[Document(page_content='Springer Series in Statistics\nTrevor Hastie\nRob
ert TibshiraniJerome FriedmanSpringer Series in Statistics\nThe Elements
of\nStatistical Learning\nData Mining, Inference, and Prediction\nThe Ele
ments of Statistical LearningDuring the past decade there has been an exp
losion in computation and information tech-', metadata={'source': '/User
s/damienbenveniste/Projects/Teaching/Introduction_Langchain/data/mixed_da
ta/element_of_SL.pdf', 'page': 0}),
 Document(page_content='nology. With it have come vast amounts of data in
a variety of fields such as medicine, biolo-gy, finance, and marketing. T
he challenge of understanding these data has led to the devel-opment of n
ew tools in the field of statistics, and spawned new areas such as data m
ining,machine learning, and bioinformatics. Many of these tools have comm
on underpinnings butare often expressed with different terminology. This
book describes the important ideas inthese areas in a common conceptual f
ramework.', metadata={'source': '/Users/damienbenveniste/Projects/Teachin
g/Introduction_Langchain/data/mixed_data/element_of_SL.pdf', 'page': 0}),
 Document(page_content='While the approach is statistical, theemphasis is
on concepts rather than mathematics. Many examples are given, with a libe
```

Let’s install the FAISS package:

```
%pip install faiss-cpu
```

And let’s embed text with the OpenAI embeddings

```
from langchain.embeddings.openai import OpenAIEmbeddings

embeddings = OpenAIEmbeddings(show_progress_bar=True)
vector1 = embeddings.embed_query('How are you?')
len(vector1)

> 1536
```

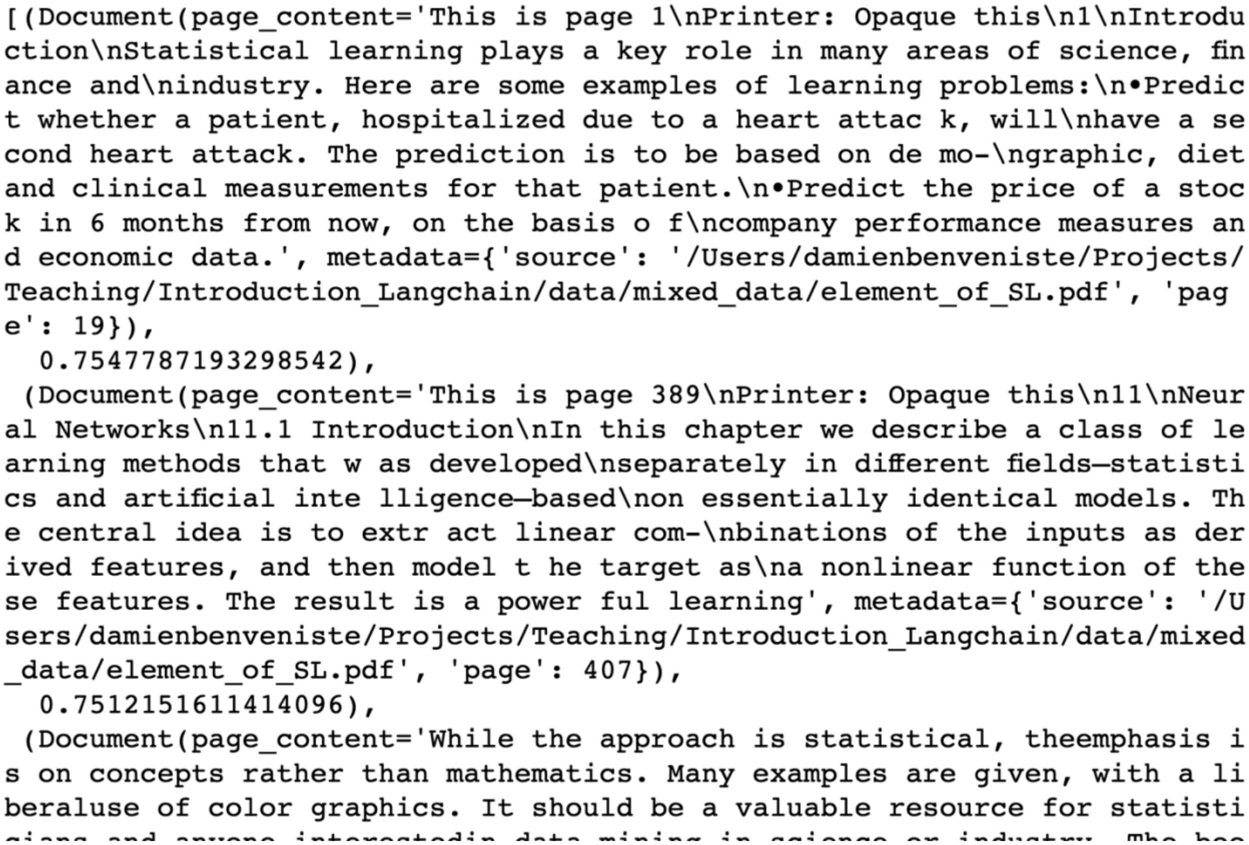
Let’s embed the book data

Retrieval Augmented Generation

```
from langchain.vectorstores import FAISS

index = FAISS.from_documents(data, embeddings)
```

We can search in that index:

```
index.similarity_search_with_relevance_scores(
    "What is machine learning?"
)
```



```
[(Document(page_content='This is page 1\nPrinter: Opaque this\n1\nIntrodu
ction\nStatistical learning plays a key role in many areas of science, fin
ance and\nindustry. Here are some examples of learning problems:\n•Predic
t whether a patient, hospitalized due to a heart attac k, will\nhave a se
cond heart attack. The prediction is to be based on de mo-\ngraphic, diet
and clinical measurements for that patient.\n•Predict the price of a stoc
k in 6 months from now, on the basis o f\ncompany performance measures an
d economic data.', metadata={'source': '/Users/damienbenveniste/Projects/
Teaching/Introduction_Langchain/data/mixed_data/element_of_SL.pdf', 'pag
e': 19}),
  0.7547787193298542),
 (Document(page_content='This is page 389\nPrinter: Opaque this\n11\nNeur
al Networks\n11.1 Introduction\nIn this chapter we describe a class of le
arning methods that w as developed\nseparately in different fields—statisti
cs and artificial inte lligence—based\non essentially identical models. Th
e central idea is to extr act linear com-\nbinations of the inputs as der
ived features, and then model t he target as\na nonlinear function of the
se features. The result is a power ful learning', metadata={'source': '/U
sers/damienbenveniste/Projects/Teaching/Introduction_Langchain/data/mixed
_data/element_of_SL.pdf', 'page': 407}),
  0.7512151611414096),
 (Document(page_content='While the approach is statistical, theemphasis i
s on concepts rather than mathematics. Many examples are given, with a li
beraluse of color graphics. It should be a valuable resource for statisti
cians and anyone interestedin data mining in science or industry. The boo
```

We can use that index in a chain

```
from langchain.chains import RetrievalQA
from langchain.chat_models import ChatOpenAI
from langchain.callbacks import StdOutCallbackHandler

retriever = index.as_retriever()
retriever.search_kwargs['fetch_k'] = 20
retriever.search_kwargs['maximal_marginal_relevance'] = True
retriever.search_kwargs['k'] = 10

llm = ChatOpenAI()

chain = RetrievalQA.from_chain_type(
    llm=llm,
    retriever=retriever,
    verbose=True
)

handler = StdOutCallbackHandler()

chain.run(
    'What is machine learning?',
    callbacks=[handler]
)
```

```
Machine learning is a field of study that involves the development of algorithms and models that can
learn from data and make predictions or decisions without being explicitly programmed. It focuses on
creating computer systems that can automatically learn and improve from experience, rather than
being explicitly programmed for specific tasks. Machine learning algorithms analyze large amounts of
data to identify patterns, make predictions, or learn from examples and feedback. It is widely used in
various fields such as science, finance, and industry for tasks like predicting stock prices, medical
diagnoses, and customer behavior analysis.
```

Loading data into a Vector Database

We are going to load the data in Pinecone. Let’s install the Python package

```
%pip install pinecone-client
```

And let’s load the data into the database

```
import pinecone
from langchain.vectorstores import Pinecone

pinecone.init(
    api_key=PINECONE_API_KEY,  # find at app.pinecone.io
    environment=PINECONE_ENV  # next to api key in console
)

index_name = "langchain-demo"
db = Pinecone.from_documents(
    data,
    embeddings,
    index_name=index_name
)
```

And we can now augment an LLM with the database

```
chain = RetrievalQA.from_chain_type(
    llm=llm,
    retriever=db.as_retriever(),
    verbose=True
)

chain.run(
    'What is machine learning?',
    callbacks=[handler]
)
```

Providing sources

I will show to provide the sources as we answer questions. Let’s install the NewsAPI Python package:

```
%pip install newsapi-python
```

Let’s get the news about “Artificial Intelligence“ from the past week:

```
from datetime import date, timedelta
from newsapi import NewsApiClient

newsapi = NewsApiClient(api_key=NEWS_API_KEY)

today = date.today()
last_week = today - timedelta(days=7)

latest_news = newsapi.get_everything(
    q='artificial intelligence',
    from_param=last_week.strftime("%Y-%m-%d"),
    to=today.strftime("%Y-%m-%d"),
    sort_by='relevancy',
    language='en'
)
```

and let’s create documents:

```
from langchain.docstore.document import Document
docs = [
    Document(
    page_content=article['title'] + '\n\n' + article['description'],
    metadata={
        'source': article['url'],
    }
    ) for article in latest_news['articles']
]
```

Let’s create a chain that provides the sources with the answers

```
from langchain.chains import create_qa_with_sources_chain
from langchain.chains.combine_documents.stuff import StuffDocumentsChain
from langchain.prompts import PromptTemplate

qa_chain = create_qa_with_sources_chain(llm)

doc_prompt = PromptTemplate(
    template="Content: {page_content}\nSource: {source}",
    input_variables=["page_content", "source"],
)

final_qa_chain = StuffDocumentsChain(
    llm_chain=qa_chain,
    document_variable_name="context",
    document_prompt=doc_prompt,
)

index = FAISS.from_documents(docs, embedding=embeddings)


chain = RetrievalQA(
    retriever=index.as_retriever(),
    combine_documents_chain=final_qa_chain
)
```

Let’s ask a question:

```
question = """
What is the most important news about artificial intelligence from last week?
"""

answer = chain.run(question)
answer
```

```
{
  "answer": "The most important news about artificial intelligence from last week is the use of AI to train on the works of authors Stephen King and Margaret Atwood. These authors responded to the revelation that their work is being used to train AI. Additionally, AI took the stage at the Edinburgh Fringe festival, raising the question of whether AI can deliver a satisfying punchline. Furthermore, a tech expert from the University of Oxford highlighted the potential workplace threats of AI, including the possibility of AI becoming a monitoring boss. Finally, AI is being seen as a tool that can help companies connect with customers in a more personalized and efficient way.",
  "sources": [
    "https://www.theatlantic.com/newsletters/archive/2023/09/books-briefing-ai-stephen-king-margaret-atwood/675213/?utm_source=feed",
    "https://www.cnet.com/tech/ai-took-the-stage-at-the-worlds-largest-arts-festival-heres-what-happened/",
    "https://www.foxnews.com/tech/tech-expert-existential-fears-ai-are-overblown-sees-very-disturbing-workplace-threats",
    "https://www.techradar.com/pro/ai-could-help-companies-connect-with-customers-like-never-before"
  ]
}
```

Retrieval Augmented Generation

Indexing a website

We are going to use Apify to crawl a website. Let’s download the Python package

```
%pip install apify-client chromadb
```

and let’s create a loader that will crawl the AiEdge Newsletter website:

```
from langchain.utilities import ApifyWrapper
from langchain.document_loaders.base import Document

apify = ApifyWrapper()

loader = apify.call_actor(
    actor_id="apify/website-content-crawler",
    run_input={
        "startUrls": [{"url": "https://newsletter.theaiedge.io/"}],
        "aggressivePrune": True,
    },
    dataset_mapping_function=lambda item: Document(
        page_content=item["text"] or "", metadata={"source": item["url"]}
    ),
)
```

Let’s index the website data:

```
from langchain.indexes import VectorstoreIndexCreator

text_splitter = RecursiveCharacterTextSplitter(
    chunk_size=500,
    chunk_overlap=0
)

index = VectorstoreIndexCreator(
    text_splitter=text_splitter
).from_loaders([loader])
```

Let’s make a search on that index:

```
query = "What is the main subject of the aiedge newsletter?"

index.query_with_sources(query)
```

```
{
    'question': 'What is the main subject of the aiedge newsletter?',
    'answer': ' The main subject of the AiEdge newsletter is Machine Learning applications, Machine Learning System Design, MLOps, and the latest techniques and news about the field.\n',
    'sources': ''
}
```

Let’s now pass that index to a chain

```
retriever = index.vectorstore.as_retriever()

qa = RetrievalQA.from_chain_type(
    llm=llm,
    retriever=retriever,
)

query = "What is the most recent article of the aiedge newsletter?"

qa.run(
    query,
    callbacks=[handler]
)
```

"I'm sorry, but I don't have access to the specific articles or the most recent content of the AiEdge Newsletter. As an AI language model, I don't have real-time access to current articles or newsletters. It would be best to subscribe to the newsletter and check the latest edition for the most recent article."

Indexing a GitHub repo

Let’s install the Python package:

```
%pip install GitPython
```

Let’s load a repo

```
from langchain.document_loaders import GitLoader

loader = GitLoader(
    clone_url="https://github.com/langchain-ai/langchain",
    repo_path="./data/repo/",
    file_filter=lambda file_path: file_path.endswith(".py"),
    branch="master",
)

documents = loader.load()
```

Let’s resplit the documents for the Python language:

```
from langchain.text_splitter import Language

python_splitter = RecursiveCharacterTextSplitter.from_language(
    language=Language.PYTHON,
    chunk_size=1000,
    chunk_overlap=200
)

documents = python_splitter.split_documents(documents)
```

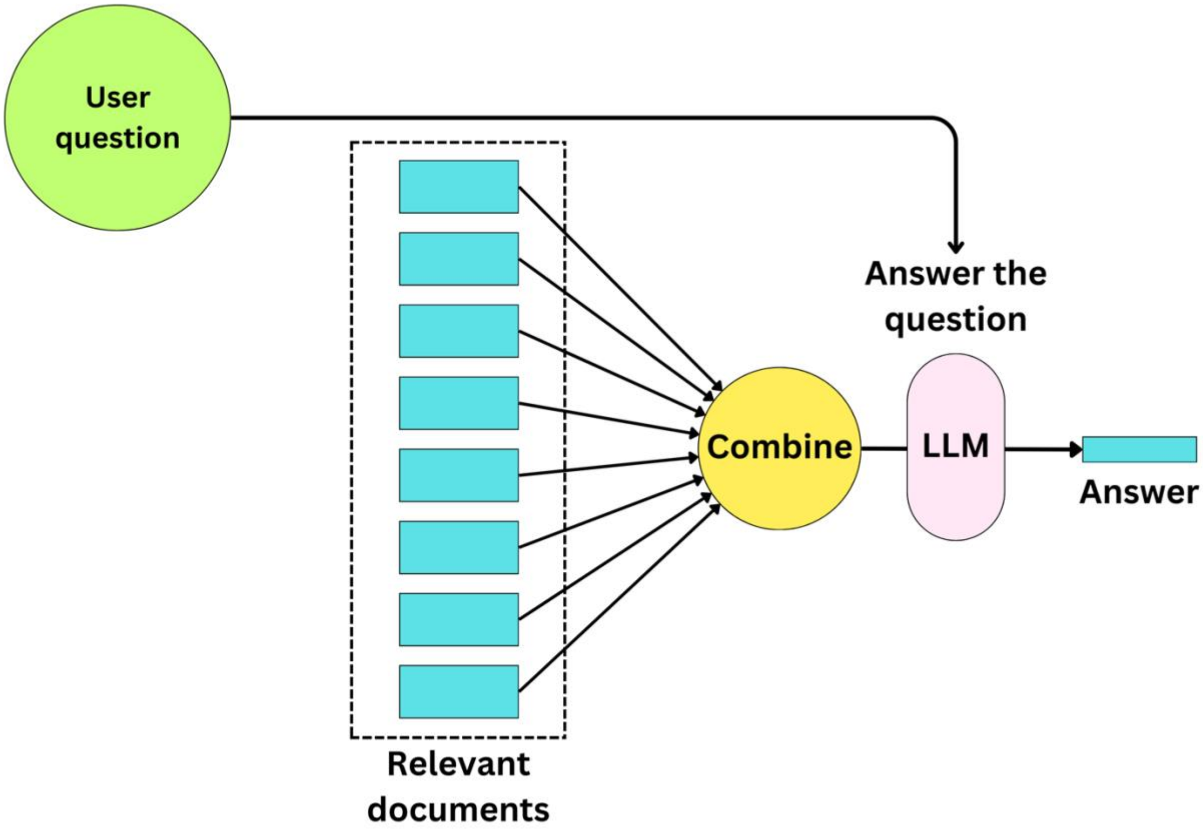
Let’s index the data and create a chain

```
index = FAISS.from_documents(documents, embeddings)
retriever = index.as_retriever()

qa = RetrievalQA.from_chain_type(
    llm=llm,
    retriever=retriever,
)

query = "What is a stuff chain?"

qa.run(query, callbacks=[handler])
```

'A stuff chain is a sequence of operations performed on a language model (LLM) to generate or process text. It typically consists of a language model chain (LLMChain) and a document chain (StuffDocumentsChain). The LLMChain is responsible for generating text based on a prompt, while the StuffDocumentsChain is used to process and manipulate documents or summaries. The specific details and functionality of a stuff chain can vary depending on the context and configuration.'

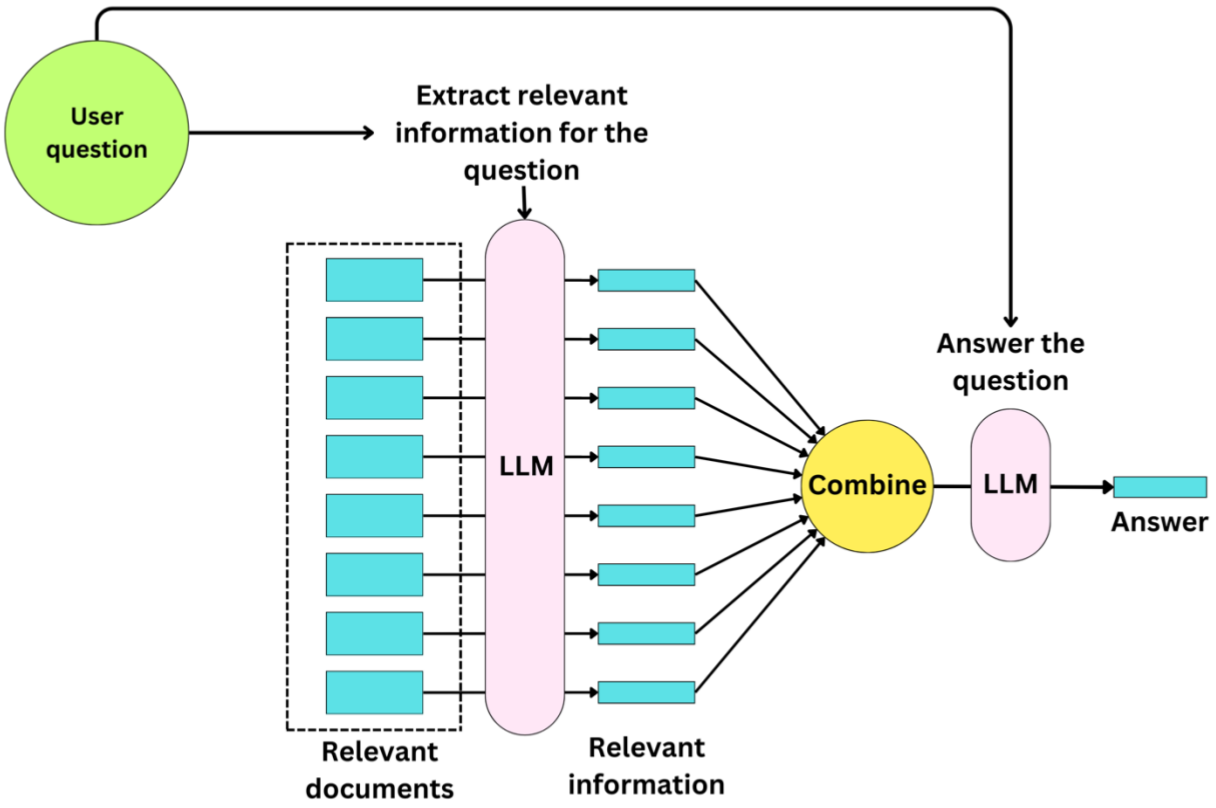
The Stuff strategy

Retrieval Augmented Generation

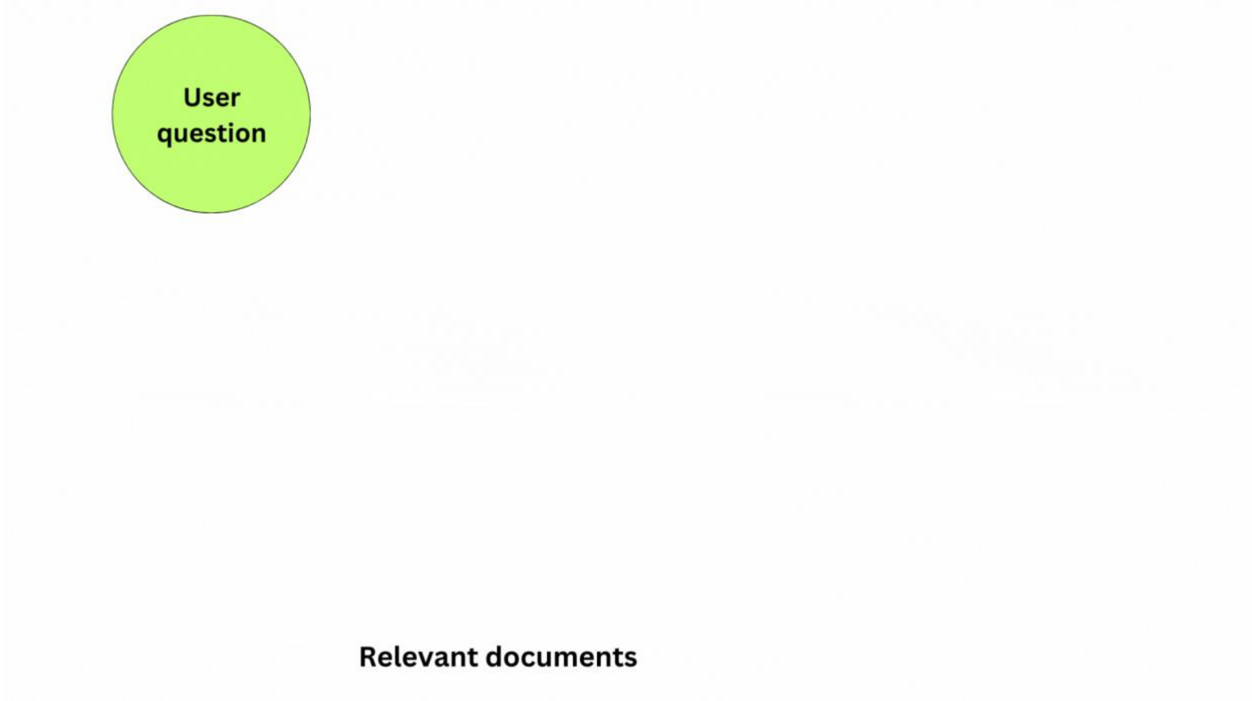


The Map-reduce strategy

Retrieval Augmented Generation

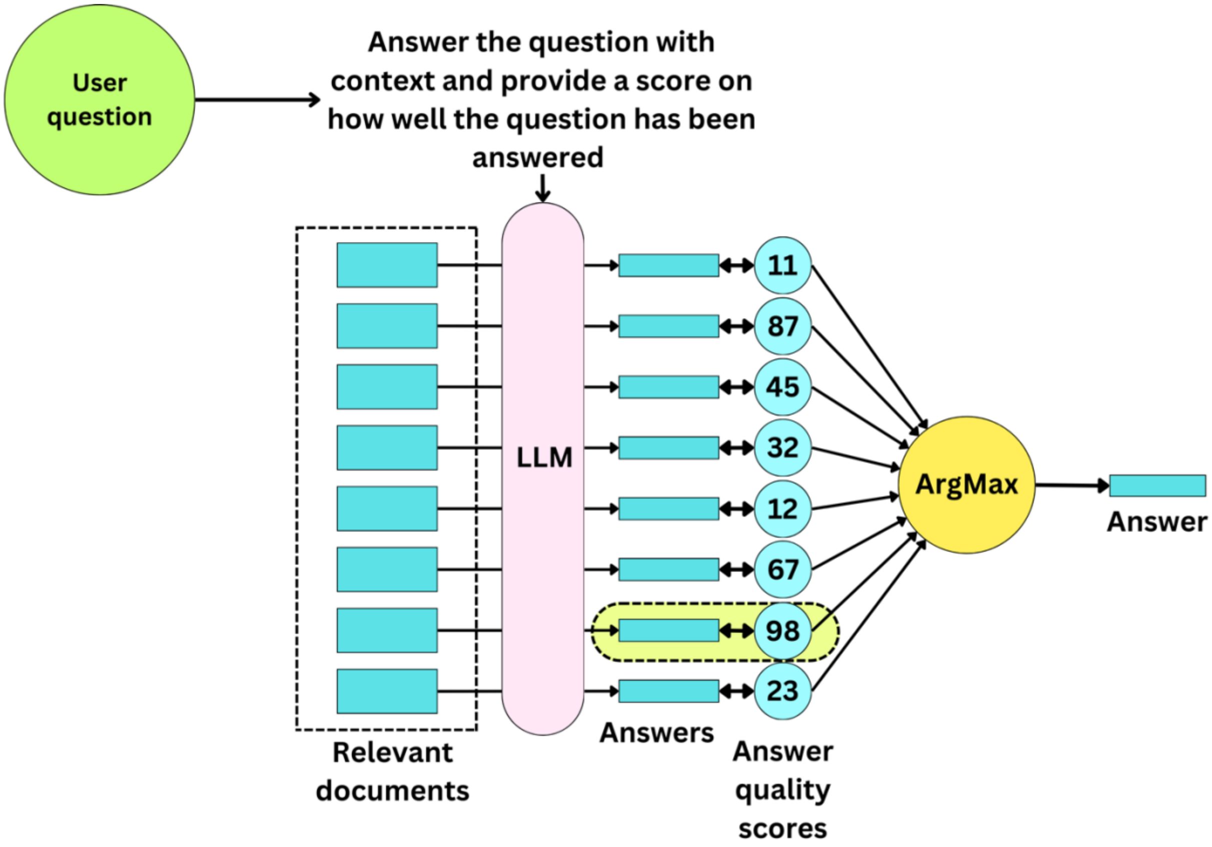


The Refine strategy



The Map-rerank strategy

Retrieval Augmented Generation


User question
Answer the question with context and provide a score on how well the question has been answered
LLM
11
87
45
32
12
67
98
23
ArgMax
Answer
Relevant documents
Answers
Answer quality scores