

Бараненков Сергей
Демочкина Полина
Тихобаев Илья
Яхтин Леонид

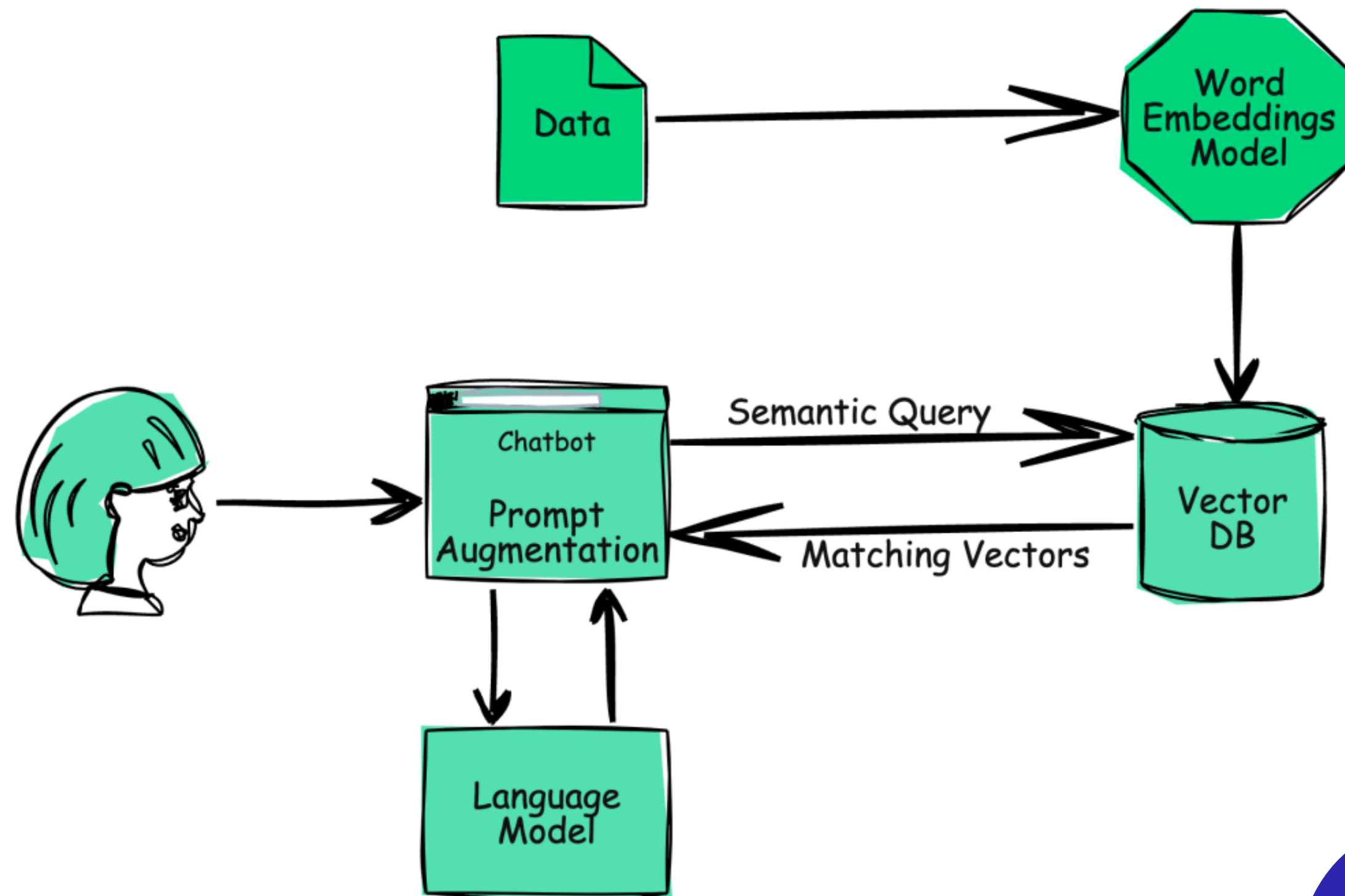
ОРКЕСТРАЦИЯ ВЕКТОРНОЙ БД ДЛЯ ПРИМЕНЕНИЯ В LLM МОДЕЛЯХ.

ЦЕЛЬ:

Разработать чат бота, способного отвечать на вопросы из специфичной, незнакомой ему области с помощью дополнительной векторной базы данных.

- Необходимо реализовать бота в Телеграмме
- В качестве базовой LLM необходимо использовать GigaChat
- Необходимо собрать собственную БД с дополнительными знаниями, используя библиотеку LangChain

PIPELINE



1. DATASET

Что такое инструменты в контексте LangChain?	Инструменты - это функции, которые агенты могут использовать для взаимодействия с миром.
Как загрузить инструменты в LangChain?	<pre>from langchain.agents import load_tools tool_names = [...] tools = load_tools(tool_names)</pre>
Какие импорты необходимы для создания пользовательских инструментов в LangChain?	<pre>from langchain.agents import AgentType, initialize_agent from langchain.chains import LLMMathChain from langchain.chat_models import ChatOpenAI from langchain.tools import BaseTool, StructuredTool, Tool, tool from langchain.utilities import SerpAPIWrapper</pre>
Как определить простой инструмент в LangChain, принимающий одну строку и возвращающий строку?	<pre>search = SerpAPIWrapper() llm_math_chain = LLMMathChain(llm=llm, verbose=True) tools = [Tool.from_function(func=search.run, name="Search", description="useful for when you need to answer questions about current events"),]</pre>

2. VECTOR DB SETUP

Embeddings



Vector search

OpenAI Ada

HNSW index

```
query = "Что такое Langchain?"  
db.similarity_search(query, k=5)
```

```
[Document(page_content='Что такое LangChain и для чего он используется? LangChain – это фреймворк для разработки приложений, работающих на основе языковых моделей. Он позволяет создавать приложения, которые осведомлены о контексте и могут осуществлять рассуждения на основе предоставленного контекста.', metadata={'question': 'Что такое LangChain и для чего он используется?', 'answer': 'LangChain – это фреймворк для разработки приложений, работающих на основе языковых моделей. Он позволяет создавать приложения, которые осведомлены о контексте и могут осуществлять рассуждения на основе предоставленного контекста.', 'id': 0}),  
Document(page_content='Какие ключевые преимущества предлагают пакеты LangChain? Основные преимущества пакетов LangChain включают в себя компоненты для работы с языковыми моделями и готовые цепочки, которые облегчают начало работы и позволяют настраивать существующие цепочки или создавать новые.', metadata={'question': 'Какие ключевые преимущества предлагают пакеты LangChain?', 'answer': 'Основные преимущества пакетов LangChain включают в себя компоненты для работы с языковыми моделями и готовые цепочки, которые облегчают начало работы и позволяют настраивать существующие цепочки или создавать новые.', 'id': 2}),  
Document(page_content='Какова роль LangChain в экосистеме инструментов? LangChain является частью богатой экосистемы инструментов, которые интегрируются с этим фреймворком и строятся на его основе, включая различные интеграции и руководства по лучшим практикам разработки.', metadata={'question': 'Какова роль LangChain в экосистеме инструментов?', 'answer': 'LangChain является частью богатой экосистемы инструментов, которые интегрируются с этим фреймворком и строятся на его основе, включая различные интеграции и руководства по лучшим практикам разработки.', 'id': 6}),  
Document(page_content='Какие интеграции предлагает LangChain и как они способствуют созданию приложений? LangChain предлагает обширную экосистему интеграций с различными внешними ресурсами, такими как локальные и удаленные файловые системы, API и базы данных. Эти интеграции позволяют разработчикам создавать гибкие приложения, сочетающие возможности языковых моделей (LLM) с доступом, взаимодействием и манипулированием внешними ресурсами.', metadata={'question': 'Какие интеграции предлагает LangChain и как они способствуют созданию приложений?', 'answer': 'LangChain предлагает обширную экосистему интеграций с различными внешними ресурсами, такими как локальные и удаленные файловые системы, API и базы данных. Эти интеграции позволяют разработчикам создавать гибкие приложения, сочетающие возможности языковых моделей (LLM) с доступом, взаимодействием и манипулированием внешними ресурсами.', 'id': 13}),  
Document(page_content='Как установить LangChain CLI и в чем его предназначение? LangChain CLI полезен для работы с шаблонами LangChain и другими проектами LangServe. Установить его можно с помощью команды 'pip install langchain-cli'.', metadata={'question': 'Как установить LangChain CLI и в чем его предназначение?', 'answer': 'LangChain CLI полезен для работы с шаблонами LangChain и другими проектами LangServe. Установить его можно с помощью команды 'pip install langchain-cli'.', 'id': 11})]
```


3. VECTOR SEARCH TESTING

	questions	rephrased_questions
0	Что такое LangChain и для чего он используется?	Для чего предназначен LangChain и что это за технология?
1	Какие основные части включает в себя фреймворк LangChain?	Какие компоненты входят в состав фреймворка LangChain?
2	Какие ключевые преимущества предлагают пакеты LangChain?	Какие основные преимущества предоставляют пакеты LangChain?
3	Что такое LCEL в контексте LangChain?	В контексте LangChain, что означает LCEL?
4	Какие модули предоставляет LangChain?	Какие функциональные возможности предлагает LangChain?
5	Какие примеры использования LangChain приведены в документации?	В документации приведены какие примеры использования LangChain?
6	Какова роль LangChain в экосистеме инструментов?	Какую роль играет LangChain в экосистеме инструментов?
7	Как установить LangChain с использованием Pip и Conda?	Как можно установить пакет LangChain с помощью инструментов Pip и Conda?
8	Как установить LangChain из исходного кода?	Как можно установить LangChain из исходного кода?
9	Что представляет собой пакет 'langchain-experimental' и как его установить?	Какой пакет называется 'langchain-experimental' и как его можно установить?
10	Как установить LangServe и для чего он используется?	Что такое LangServe и как его использовать?
11	Как установить LangChain CLI и в чем его предназначение?	Что такое LangChain CLI и как его использовать?
12	Что такое LangSmith SDK и как его установить?	Как можно установить LangSmith SDK?
13	Какие интеграции предлагает LangChain и как они способствуют созданию приложений?	Какие возможности для интеграции предоставляет LangChain и как они помогают в создании приложений?

```
print(f'Точность на переформулированных вопросах {relevant_in_top_n / doc_length}')
```

Точность на переформулированных вопросах 1.0

4. PROMPT GENERATION

gigachat	gigachat + augmentation	prompt
<p>В LangChain можно объединять несколько шагов в цепочку, используя оператор <code>`</code> (запятая). Например, чтобы объединить два шага <code>step1`</code> и <code>step2`</code>, можно написать следующий код:</p> <pre><code>\nstep1;\nstep2;\n</code></pre> <p>Также можно использовать оператор <code>`</code> для объединения нескольких шагов в одну строку кода. Например:</p> <pre><code>\nstep1; step2; step3;\n</code></pre> <p>Обратите внимание, что оператор <code>`</code> не является обязательным элементом при объединении шагов в цепочку. Если шаги не объединены оператором <code>`</code>, они будут выполнены последовательно.</p>	<p>Для объединения нескольких шагов в цепочку в LangChain можно использовать пакеты 'ChatOpenAI' и 'ChatPromptTemplate'. Сначала необходимо импортировать эти пакеты, затем создать экземпляр модели и шаблона приглашения. Далее можно объединить их в цепочку, используя метод <code>combine_steps()</code>. При этом можно указать параметры, такие как <code>return_intermediate_steps=True</code>, чтобы получить доступ к промежуточным шагам агента.</p> <p>Пакеты LangChain предлагают ключевые преимущества, такие как компоненты для работы с языковыми моделями и готовые цепочки, которые облегчают начало работы и позволяют настраивать существующие цепочки или создавать новые. Они также предоставляют возможность выбора примеров по длине, что полезно при работе с длинными входными данными.</p>	<p>Ответь на вопрос: Как объединить несколько шагов в цепочку в LangChain?</p> <p>При ответе учитывай следующие имеющиеся данные:</p> <p>Каковы преимущества использования промежуточных шагов в LangChain? Использование промежуточных шагов в LangChain позволяет лучше понять, как агент приходит к окончательному ответу. Это обеспечивает дополнительную прозрачность и позволяет пользователям или разработчикам увидеть, какие инструменты использовались и какие наблюдения были сделаны на каждом шаге процесса</p> <p>Какие шаги необходимо выполнить для создания и использования простой цепочки 'PromptTemplate + ChatModel' в LangChain? Для создания и использования цепочки 'PromptTemplate + ChatModel' в LangChain, необходимо импортировать 'ChatOpenAI' и 'ChatPromptTemplate', создать экземпляр модели и шаблона приглашения, а затем объединить их в цепочку.</p> <p>Как можно получить доступ к промежуточным шагам агента в LangChain? В LangChain можно получить доступ к промежуточным шагам агента, используя параметр <code>return_intermediate_steps=True</code> при инициализации агента. Это позволяет отслеживать каждый шаг в цепочке действий агента, включая использование инструментов и наблюдения за результатами</p> <p>Какие ключевые преимущества предлагают пакеты LangChain? Основные преимущества пакетов LangChain включают в себя компоненты для работы с языковыми моделями и готовые цепочки, которые облегчают начало работы и позволяют настраивать существующие цепочки или создавать новые.</p> <p>В каких сценариях полезно использовать выбор примеров по длине в LangChain? Выбор примеров по длине в LangChain полезен, когда нужно учитывать ограничения по длине запроса, особенно для моделей языка с ограниченным размером контекстного окна. Это помогает убедиться, что включенные примеры не превышают максимально допустимую длину, что особенно важно при работе с длинными входными данными.</p> <p>Если вопрос тесно связан с предоставленными данными, используй их при формировании своего ответа.</p>

РЕЗУЛЬТАТ



ContextLLM

You can contact @ContextLLMBot right away.

Telegram



Как мне в CSVLoader в Langchain указать свой путь до файла если мой файл лежит в /home/user/data.csv?

12:57 PM ✓✓

Вопрос принят, скоро придет ответ, пожалуйста, подождите :)

12:57 PM

Для указания пути к файлу в CSVLoader в LangChain необходимо использовать аргумент `file_path` при создании экземпляра класса `CSVLoader`. В данном случае, если файл находится в директории `/home/user/data.csv`, можно создать экземпляр класса следующим образом:

```
python
from langchain.document_loaders.csv_loader import CSVLoader

loader = CSVLoader(file_path='/home/user/data.csv')
data = loader.load()
```

В этом примере мы указываем путь к файлу `/home/user/data.csv`. При использовании этого метода, CSVLoader автоматически загрузит данные из указанного файла и вернет их в виде списка документов.

12:57 PM