PROJET KG-Enabled DevOps RAG chatbot

Maziz Yassine
Yaici Walid

Table of contents

01

Etat de l’art

Chatbot integration in few patterns

- Ce document traite de l’intégration des chatbots dans divers systèmes logiciels

- Les chatbots sont des agents logiciels qui peuvent interagir avec les humains en utilisant le langage naturel.

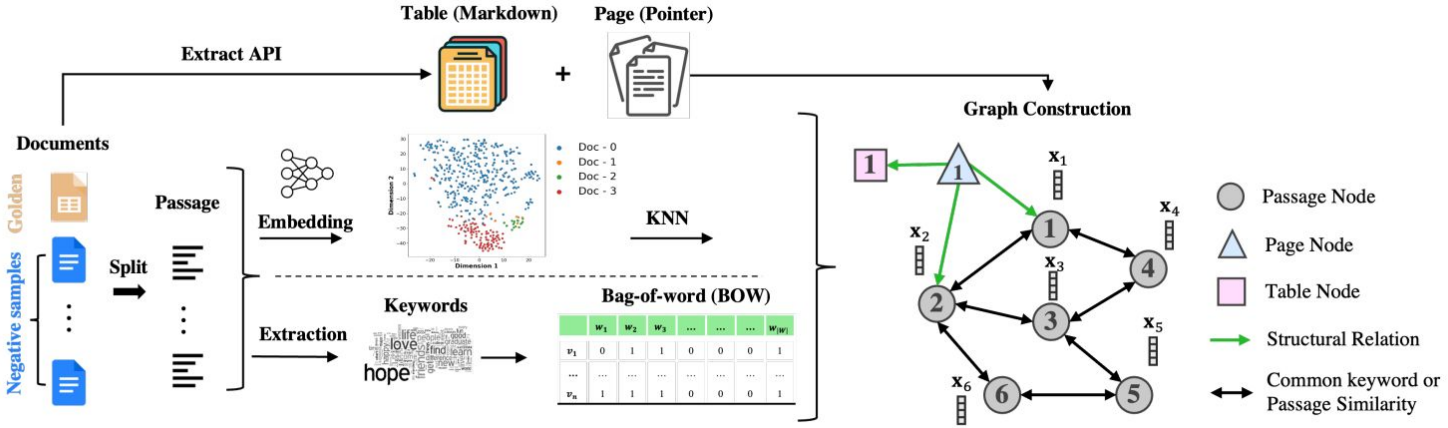
Chatbot integration in few patterns

- IN-APP Assistant: chatbot qui réside dans une application existante, telle qu'un site web ou une application de bureau
- Agent GUI: Intégration dans les interfaces utilisateur graphiques (GUI) en rendant des widgets indépendants ou des composants UI qui ont leur propre logique d'application
- API caller: Interagit avec d'autres systèmes logiciels par l'intermédiaire de leurs API. Il peut faire appel à ces API pour effectuer des tâches spécifiques ou récupérer des informations.

Knowledge Graph Prompting for Multi-Document Question Answering

- Ce document traite d'une méthode appelée Knowledge Graph Prompting (KGP) pour la réponse aux questions multi-documents (MD-QA).
- Dans le module de construction de graphe, les nœuds du graphe représentent des passages ou des structures de documents, et les arêtes représentent la similitude sémantique/lexicale.
- Le graphe sert de règle globale qui régule l'espace de transition entre les passages et réduit la latence de la recherche.

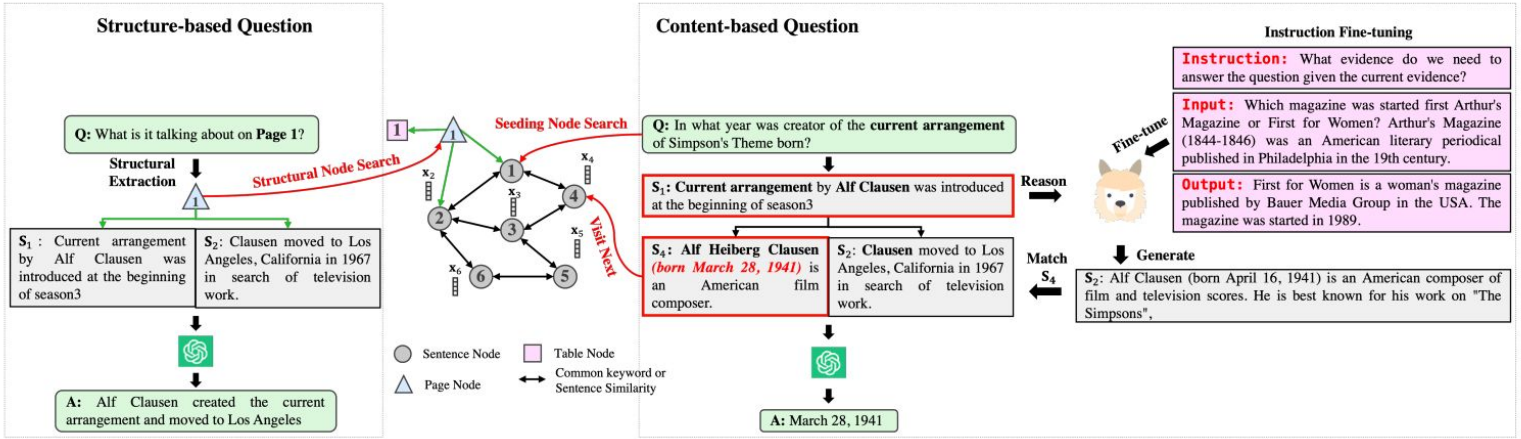
Knowledge Graph Prompting for Multi-Document Question Answering



Knowledge Graph Prompting for Multi-Document Question Answering

- Dans le module de parcours de graphe, Un LLM est utilisé pour le parcours afin de rassembler l'ensemble des noeuds nécessaires pour répondre à la question.
- Un LLM est entraîné à générer la réponse suivante étant donnée le noeud courant.
- Le LLM est aussi responsable de générer la réponse finale basée sur l'ensemble des noeuds récoltés.

Knowledge Graph Prompting for Multi-Document Question Answering



Exploring Large Language Models for Knowledge Graph Completion

- Le document explore l'utilisation de grands modèles de langage (LLM) pour compléter les graphes de connaissance.
- Différentes tâches pour compléter un graph:
  - La classification de triplets
  - La prediction de relations
  - La prediction de Entity link
- Un LLM est fine tune sur ces tâches la et ensuite utilisé pour compléter des graphs.

02
Construction graph contenu

Documents Utilisés

- Utilisation de texte extrait à partir de Wikipedia (l'API WIKIPEDIA) sur le thème des films
- Création d'une liste de films, et extraction des pages Wikipedia pour chaque films

Construction graph contenu

- Afin de construire un graph sur le contenu des documents extrait de wikipedia nous avons suivie la même approche que l'article [Knowledge Graph Prompting for Multi-Document Question Answering].
- Dans un premier temps on commence par diviser les document en chunk.

Construction graph contenu

Etape 1: Extraction d’entités

- On parcours l’ensemble des chunks extraits
  - Pour chaque chunk on extraits les entités en utilisant spacy
  - On forme un dictionnaire qui contient pour chaque mot clé, les chunks qui le contiennent.
  - On forme un deuxième dictionnaire avec pour chaque titre de document, les chunks de ce dernier.

Construction graph contenu

Etape 2:

Script de requetes

```
def kw_graph_construct(file, d):

    with open(file, "w") as f:
        chunk2id = {}
        for i, chunk in enumerate(d['title_chunks']):
            _, chunk = chunk
            chunk_sans_guillemet = chunk
            if '"' in chunk:
                chunk_sans_guillemet = chunk.replace('"', '')
            f.write('MERGE (:Content {contenu:"' + chunk_sans_guillemet +
                '", id:'
                    + str(i) + '});\n')
            chunk2id[chunk] = i

        for kw, chunks in d['kw2chunk'].items():
            for i in range(len(chunks)):
                for j in range(i+1, len(chunks)):
                    f.write("MATCH (node1:Content), (node2:Content)\n")
                    f.write("WHERE node1.id = "+str(chunk2id[chunks[i]])+"
                        AND node2.id = "+str(chunk2id[chunks[j]])+"\n")
                    f.write('MERGE (node1)-[:has_commun_entity{entity:"'+kw
                        +'"}]-(node2);\n')

        for kw, chunks in d['title2chunk'].items():
            for i in range(len(chunks)):
                for j in range(i+1, len(chunks)):
                    f.write("MATCH (node1:Content), (node2:Content)\n")
                    f.write("WHERE node1.id = "+str(chunk2id[chunks[i]])+"
                        AND node2.id = "+str(chunk2id[chunks[j]])+"\n")
                    f.write('MERGE (node1)-[:has_commun_title{title:"'+kw+'
                        "}]-(node2);\n')
```

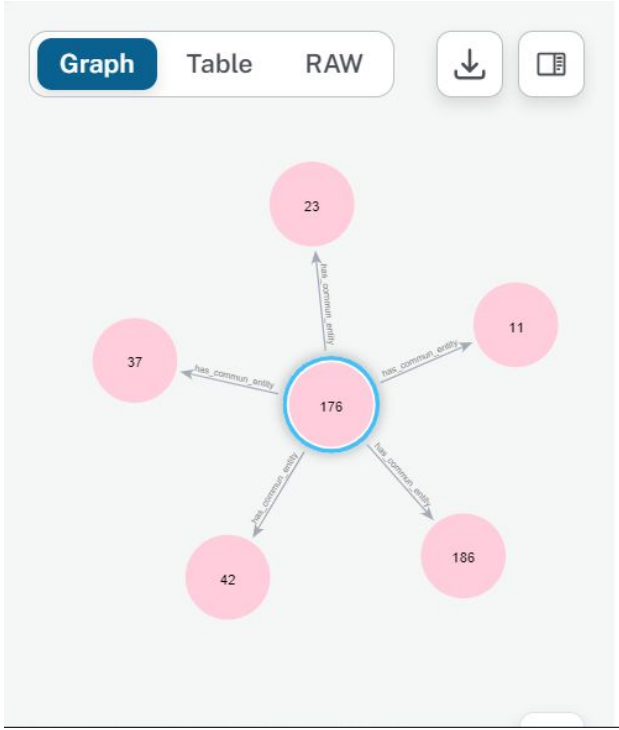
Construction graph contenu

Etape 3: Execution du script

```
path = './script.txt'
uri = ""
user = ""
password = ""
driver = GraphDatabase.driver(uri, auth=(user, password))
with driver.session() as session:
  with open(path, "r") as file:
      contenu = file.read()
      requetes = contenu.split('\n')
      for i in tqdm(range(len(requetes))):
        session.run(requetes[i])
```

Construction graph contenu

Exemple:



embedding

[0.010828135535120964, -0.01811053417623043, -0.012021412141621113, -0.040585052222013474, 0.008346118964254856, 0.023020016029477712, -0.0042207906953... Show all

contenu

"Sound design Gregg Landaker and Gary Rizzo were the films audio engineers tasked with audio mixing while sound editor Richard King supervised the process Christopher Nolan sought to mix the sound to take maximum advantage of theater equipment and paid close attention to designing the sound mix like f

03
Construction graph structure

Construction graph structure

- Extraction des entitées/relations

- Construction du graphe

Extraction des entitées/relations

- (Movie title; Directed_by; Director)
- (Movie title; Written_by; Writer1)
- (Movie title; Written_by; Writer2)
- (Movie title; Written_by; WriterN)
- (Movie title; Produced_by; Producer person)
- (Movie title; Edited_by; Editor)
- (Movie title; Production_company; company)
- (Movie title; Distributed_by ; distributer)
- (Movie title; Release_date; date)
- (Movie title; Cast; Actor)

Extraction des entitées/relations

```
assistant = "Here is the text: \n" + page_text + "\n and Here is the
    triplets we want to extract for this texte: " + relations
question = "Extract triplets from the text using the provided ontology.
    Replace the movie title, by the movie name as an entity. \n As response,
    only give the list of the triplets"


messages = [ {"role": "system", "content": "You are an expert in Extracting
    Knowledge Graph relation"},
            {"role": "assistant", "content": assistant},
            {"role": "user", "content": question},
            ]
```

Extraction des entitées/relations

- Le résultat se présente en format texte.

```
(Dune: Part Two; Directed_by; Denis Villeneuve)
(Dune: Part Two; Written_by; Jon Spaihts)
(Dune: Part Two; Produced_by; Legendary Pictures)
(Dune: Part Two; Edited_by; Joe Walker)
(Dune: Part Two; Production_company; Warner Bros. Pictures)
(Dune: Part Two; Distributed_by; Warner Bros. Pictures)
(Dune: Part Two; Release_date; 2024)
(Dune: Part Two; Cast; Timothée Chalamet)
(Dune: Part Two; Cast; Rebecca Ferguson)
(Dune: Part Two; Cast; Josh Brolin)
(Dune: Part Two; Cast; Stellan Skarsgård)
(Dune: Part Two; Cast; Dave Bautista)
(Dune: Part Two; Cast; Zendaya)
(Dune: Part Two; Cast; Charlotte Rampling)
(Dune: Part Two; Cast; Javier Bardem)
(Dune: Part Two; Cast; Austin Butler)
(Dune: Part Two; Cast; Florence Pugh)
(Dune: Part Two; Cast; Christopher Walken)
(Dune: Part Two; Cast; Léa Seydoux)
(Dune: Part Two; Cast; Souheila Yacoub)
```

Construction du graphe

(Dune: Part Two; Directed_by; Denis Villeneuve)

Construction du graphe

Dune: Part Two Directed_by Denis Villeneuve

Construction du graphe

Dune: Part Two	Directed_by	Denis Villeneuve
Movie	Directed_by	?

Construction du graphe

Dune: Part Two

Movie

Directed_by

'Directed_by'

'Written_by'

'Produced_b'

'Edited_by'

'Cast'

Denis Villeneuve

Person

Construction du graphe

Dune: Part Two

Directed_by

Denis Villeneuve

'Edited_by'

Movie

'Production_company'

Company

'Distributed_by'

Construction du graphe

Dune: Part Two

Directed_by

Denis Villeneuve

Movie

Date

Date

Construction du graphe

```
session.run(
    "MERGE (m:Movie {title: $movie})"
    "MERGE (p:" + type_relation+ " {"+property_name+": $value})"
    "MERGE (m)<-[:%s]-(p)" % relation,
    movie=movie,
    value=value)
```

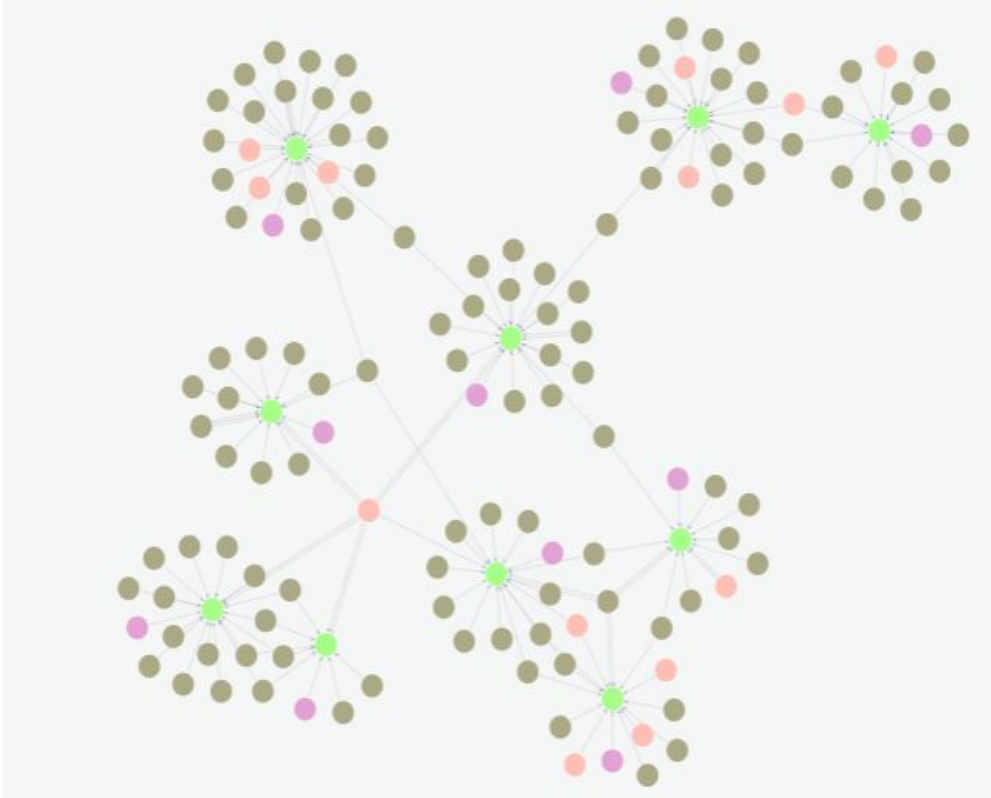
Construction du graphe

Nodes (146)

Company Date Movie Person

Relationships (173)

Cast Directed_by Distributed_by Edited_by Produced_by Production_company Release_date Written_by



04
Requetage graph contenu

Requetage graph contenu

- Nous avons imaginé une méthode qui se base sur la structure du graph pour récolter les informations nécessaire à une question et l'utilisation de LLM pour formuler une réponse à partir de ces informations
- Notre approche se divise en 3 étapes.

Requetage graph contenu

Etape 1: Creations des embeddings de noeuds

```
vector_index = Neo4jVector.from_existing_graph(
    OpenAIEmbeddings(),
    url=url,
    username=username,
    password=password,
    index_name='tasks',
    node_label="Content",
    text_node_properties=['contenu'],
    embedding_node_property='embedding',
)
```

Requetage graph contenu

Etape 2: Exploration du graph pour la r ́ecolte d'informations

```
def voisin_reponses(question,vector_index):
  response = vector_index.similarity_search(question)
  reponses = []
  for res in response:
      texte_recherche = res.page_content.split("contenu: ")[1]
      reponses.append(texte_recherche)
      cypher_query = f"MATCH (c:Content)-[:has_commun_entity]->(e:Content)
         WHERE c.contenu = '{texte_recherche}' RETURN DISTINCT e.contenu as
         contenu;"
      resultats = run_cypher_query(cypher_query)
      for record in resultats[:10]:
          reponses.append(record["contenu"])
  return list(set(reponses))
```

Requetage graph contenu

Etape 2: Utilisation du LLM pour générer la réponse

```
def reponse_question(question,vector_index, client):
  facts = voisin_reponses(question,vector_index)
  assistant = "here's a set of facts: " + str(facts)
  prompt_template = """
    give me an answer to this question based only on the set of facts: {}.
    Some fo these facts may not contain information necessery to the answer,
       you can ignore them.
    Dont add information from your knowledge, if you dont have the
       information in the set of facts dont add answer from your own.
    """.format(question)

  messages = [
             {"role": "user", "content": prompt_template},
             {"role": "assistant", "content": assistant},
             ]

  response = client.chat.completions.create(
      model="gpt-3.5-turbo",
      messages=messages,
      temperature=0
  )
  return response.choices[0].message.content
```

Requetage graph contenu

Exemple:

```
question = """who are the actors of hunger games. Also give some information about oppenheimer and interstellar
and dune and the greate gatsby and Harry Potter and the Deathly Hallows Part 1."""
print(reponse_question(question, vector_index, client))
```

Requetage graph contenu

The actors of Hunger Games are Jennifer Lawrence, Josh Hutcherson, and Liam Hemsworth.

Oppenheimer is a 2023 epic biographical thriller film written, directed, and co-produced by Christopher Nolan, starring Cillian Murphy as J. Robert Oppenheimer. The film also features Emily Blunt, Matt Damon, Robert Downey Jr., and Florence Pugh among others. It explores the life of Oppenheimer, his work on the Manhattan Project, and his eventual fall from grace.

Interstellar is a 2014 epic science fiction film directed by Christopher Nolan, starring Matthew McConaughey, Anne Hathaway, Jessica Chastain, and Michael Caine. The film follows a group of astronauts who travel through a wormhole near Saturn in search of a new home for humanity.

Dune is a 2021 epic science fiction film directed by Denis Villeneuve, based on the 1965 novel by Frank Herbert. The film stars Timothee Chalamet, Zendaya, Rebecca Ferguson, and Josh Brolin. It follows Paul Atreides as he unites with the Fremen people of the desert planet Arrakis to wage war against House Harkonnen.

The Great Gatsby is a novel by F. Scott Fitzgerald, first published in 1925. It follows the story of Jay Gatsby, a wealthy and mysterious man, and his love for Daisy Buchanan, a married woman. The novel explores themes of wealth, love, and the American Dream.

Harry Potter and the Deathly Hallows Part 1 is a 2010 fantasy film directed by David Yates, based on the novel by J.K. Rowling. It is the first part of the two-part film adaptation of the final Harry Potter book. The film stars Daniel Radcliffe, Emma Watson, and Rupert Grint.

05
Requetage graph structure

Requetage graph structure

- Cette partie consiste à tirer avantage du graphe de structure, pour extraire des information à fin que le llm puisse s'en servir pour répondre à la question
- Pour cela il faut passer par 3 étapes.
- Pour l'exemple prenons la question: "Tell me about Spiderman, and the cast of the film Far From Home"

Requetage graph structure

- Extraction de schéma

```
schema_visualisation = graph.query("call db.schema.visualization")
rel_properties = graph.query("call db.schema.relTypeProperties")
node_properties = graph.query("call db.schema.nodeTypeProperties")
```

Requetage graph structure

- Génération de la requête Cypher répondant à la question

```
assistant = "Here is the graph schema: " + str(schema_visualisation)
+ "\n Here is the properties of the relations of the graph: " + str(rel_properties)
+ "\n Here is the properties of the nodes of the graph:" + str(node_properties)
```

```
prompt_template = """
  give me a cypher Query to response to this:
  Question: {}
  """.format(question)
messages = [ {"role": "system", "content": "You are an expert in generating Cypher Query"},
          {"role": "user", "content": prompt_template},
          {"role": "assistant", "content": assistant}
          ]
```

Requetage graph structure

- Extraction de la requête Cypher répondant à la question

To respond to the question "tell me about Spiderman, and the cast of the film Far From Home" using Cypher Query, you can use the following query:

```cypher
MATCH (m:Movie)-[:Cast]->(p:Person)
WHERE m.title CONTAINS 'Spiderman' OR m.title CONTAINS 'Far From Home'
RETURN m.title AS Movie, COLLECT(p.name) AS Cast
```

This query will match all movies related to Spiderman or Far From Home and retrieve the movie title along with the cast members.

Requetage graph structure

- Extraction de la requête Cypher répondant à la question grâce à une expression régulière, en gardant que ce qui est entre triple backticks
- Remplacement des “->”, “<-” par des “-”

```
cypher
MATCH (m:Movie)-[:Cast]-(p:Person)
WHERE m.title CONTAINS 'Spiderman' OR m.title CONTAINS 'Far From Home'
RETURN m.title AS Movie, collect(p.name) AS Cast
```

Requetage graph structure

- Résultat de la requête

['Movie': 'Spider-Man: Far From Home', 'Cast': ['Zendaya', 'Tom Holland', 'Samuel L. Jackson', 'Cobie Smulders', 'Jon Favreau', 'J. B. Smoove', 'Jacob Batalon', 'Martin Starr', 'Tony Revolori', 'Marisa Tomei', 'Jake Gyllenhaal']]

Requetage graph structure

- Formulation de la réponse à la question

```
question2 = "Generate a response to this question, using the given set of facts " + question
print(question2)
messages =  [{"role" : "system", "content" : "You are an expert in Answering generation using ONLY the given set of facts."},
             {"role": "user", "content": question2 },
             {"role": "assistant", "content": assistant}

             ]
```

Requetage graph structure

- Réponse de GPT

```
In the film "Spider-Man: Far From Home," the cast includes Zendaya, Tom Holland, Samuel L.
Jackson, Cobie Smulders, Jon Favreau, J. B. Smoove, Jacob Batalon, Martin Starr, Tony
Revolori, Marisa Tomei, and Jake Gyllenhaal.
```

Requetage graph structure

- Autre exemple: "Give me the set of persons who wrote and produced the same movie.

```cypher
MATCH (p:Person)-[:Produced_by]->(m:Movie)<-[:Written_by]-(p)
RETURN p.name, m.title
```

```
The set of persons who wrote and produced the same movie are Baz Luhrmann for "The Great Gatsby" and Adam McKay for "Don't Look Up".
```

06

Requetage final

Requetage final

- Le but de cette partie, est de rassembler, les deux approches de requêtage décrites au dessus, c'est à dire extraire le maximum d'information sur le contenu des documents, et sur la structure des documents, à fin de formuler une réponse final.

Requetage final

```
answer1 = questionAnswering(question, graph, client)
answer2 = reponse_question(question,vector_index, client, graph2)

question2 = "Generate a response to this question, using the given set of
    facts: \n" + answer1 + "\n" + answer2
print(question2)


messages = [{"role" : "system", "content" : "You are an expert in Answering
    generation using ONLY the given set of facts."},
            {"role": "user", "content": question2 },
            ]
response = client.chat.completions.create(
    model="gpt-3.5-turbo",
    messages=messages,
    temperature=0
)
```

Requetage final

- Question: "Tell me about Dune: Part Two, and the cast of the film Spider-Man: Far From Home"

Requetage final

```
MATCH (movie:Movie)
WHERE movie.title = 'Dune: Part Two' OR movie.title = 'Spider-Man: Far From Home'
OPTIONAL MATCH (movie)-[:Cast]-(actor:Person)
RETURN movie.title, collect(actor.name) as cast
```

Requetage final

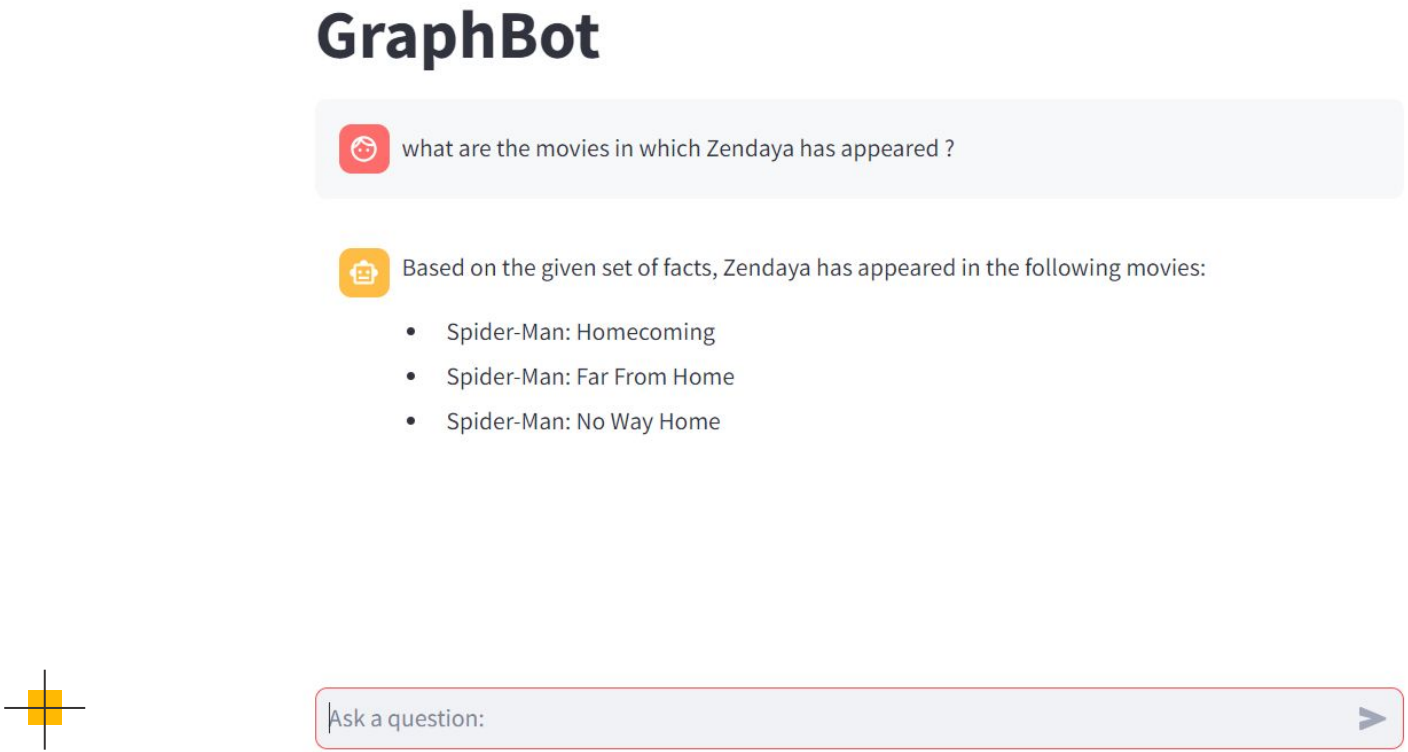
The cast of "Spider-Man: Far From Home" includes Zendaya, Tom Holland, Samuel L. Jackson, Cobie Smulders, Jon Favreau, J. B. Smoove, Jacob Batalon, Martin Starr, Tony Revolori, Marisa Tomei, and Jake Gyllenhaal.
Dune: Part Two is a 2024 American epic science fiction film directed and co-produced by Denis Villeneuve, serving as the sequel to Dune (2021). The film follows Paul Atreides as he unites with the Fremen people of the desert planet Arrakis to wage war against House Harkonnen. The cast includes Timothee Chalamet, Zendaya, Rebecca Ferguson, Josh Brolin, Stellan Skarsgard, Dave Bautista, Charlotte Rampling, and Javier Bardem. The film was greenlit in October 2021 and had its world premiere on February 15, 2024. On the other hand, the cast of the film Spider-Man: Far From Home includes Tom Holland, Jake Gyllenhaal, Zendaya, Marisa Tomei, and Michael Keaton. The film was released in July 2019 and received positive reviews from critics.
La réponse à la question est: The star-studded cast of "Dune: Part Two" features Timothée Chalamet, Zendaya, Rebecca Ferguson, Josh Brolin, Stellan Skarsgård, Dave Bautista, Charlotte Rampling, Javier Bardem, Austin Butler, Florence Pugh, Christopher Walken, Léa Seydoux, and Souheila Yacoub. Meanwhile, the cast of "Spider-Man: Far From Home" includes Zendaya, Tom Holland, Samuel L. Jackson, Cobie Smulders, Jon Favreau, J. B. Smoove, Jacob Batalon, Martin Starr, Tony Revolori, Marisa Tomei, and Jake Gyllenhaal.

07
Creation du chatbot

Creation du chatbot

- Nous avons conçu notre chatbot à partir de la bibliothèque streamlit qui est une bibliothèque open source en Python qui permet de créer rapidement des applications web interactives pour l'analyse de données et la visualisation.

Creation du chatbot



Passons maintenant aux tests !