

THE 23RD INTERNATIONAL SEMANTIC WEB CONFERENCE

November 11, 2024 – November 15, 2024

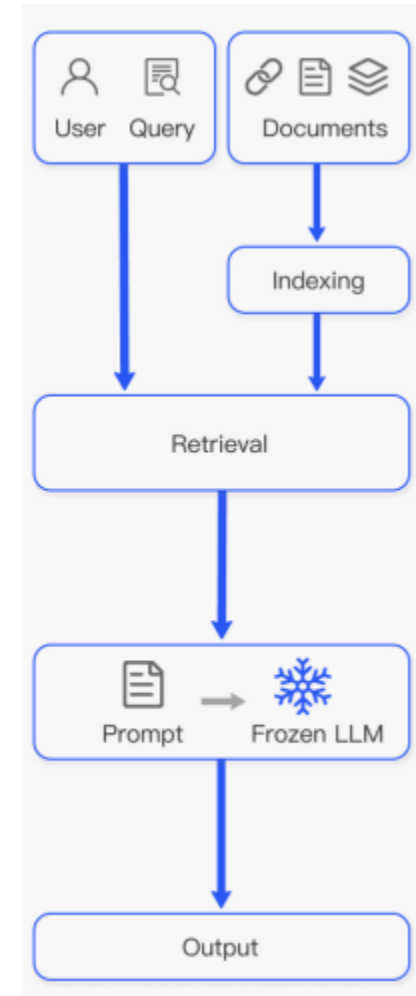
Live! Casino & Hotel Maryland

Slot 1: Basic RAG Techniques

ISWC 2024

Drawing Parallels: RAG and Graph RAG

- Any RAG Setup consists of the following approach:
 - *Indexing*: Chunking of document
 - *Retrieval*: For a given query retrieve the top K chunks
 - *Generation*: Given the query and K chunks are given to LLM to construct the answer
- Analogues of the RAG setup in Graph Domain
 - *Indexing*: knowledge graph
 - *Retrieval*: **What could be a analogues of this in KG domain?**
 - *Generation*: Generate appropriate response from the retrieved data.



Basic RAG Approach [1]

Create a sparql query in wikidata for the following query: who is prince george of cambridge parents ?

Here's a SPARQL query to retrieve the parents of Prince George of Cambridge from Wikidata:

sparql

Copy code

```
SELECT ?parent ?parentLabel WHERE {  
  wd:Q13590459 wdt:P40 ?parent .  
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }  
}
```

Explanation:

- `wd:Q13590459` is the Wikidata entity ID for Prince George of Cambridge.
- `wdt:P40` is the property ID for "child's parent".
- The `SERVICE wikibase:label` clause ensures that the labels are displayed in the desired language (e.g., English).

You can run this query in the [Wikidata Query Service](#).

Retrieval in KG

- Should satisfy the following properties
 - Should be able to fetch the required details to answer the question
 - Should be able to give consistent results to minimize hallucinations
- As we are working with KG, to retrieve knowledge from KG it should be able to construct the right query.
- As LLMs are ubiquitous can we just not use LLMs to construct the query ?
 - What about hallucinations ?
 - How will LLM know what entities are present in the question ?
- Q12590459: Dictyosquilla tuberculata (aquatic animal)
- P40: child

Chain of Thought

- Recent research in LLMs have shown that, to solve a complex problem like maths problems, LLMs perform better when they solve the task by breaking down into smaller task. Also known as Chain Of Thought (COT)
- Applying the same COT principle here, to generate the right query, the LLMs should be able to
 - Identify the entities involved in the question.
 - Identify the relation involved in the question.
 - Construct the query based on the entities and relations.
 - Does this sound familiar ?

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✅

Entity Linking and Relation Linking

- The task of matching a mention in a textual context to a KG context (unique identifier).
- The retrieval allows the LLM to be grounded to KG.
- While the community has worked hard to implement different methods, in this tutorial we focus on readily available tools for entity linking and relation linking.
- In this tutorial for joint entity and relation extraction we use Falcon[2]

We know 'Sebastian Thrun' is a person
but do we know which person exactly?

When **Sebastian Thrun** PERSON started at **Google** ORG in **2007** DATE, few people outside of the company took him seriously. "I can tell you very senior CEOs of major **American** NORP car companies would shake my hand and turn away because I wasn't worth talking to," said **Thrun** PERSON, now the co-founder and CEO of online higher education startup Udacity, in an interview with **Recode** ORG **earlier this week** DATE.

A little **less than a decade later** DATE, dozens of self-driving startups have cropped up while automakers around the world clamor, wallet in hand, to secure their place in the fast-moving world of fully automated transportation.

About: [Sebastian Thrun](#)

An Entity of Type: *scientist*, from Named Graph: *http://dbpedia.org*, within Data Space: *dbpedia.org*

Sebastian Thrun (born May 14, 1967) is an innovator, entrepreneur educator, and computer scientist from Germany. He was CEO and cofounder of Udacity. Before that, he was a Google VP and Fellow, and a Professor of Computer Science at Stanford University. At Google, he founded Google X. He is currently also an Adjunct Professor at Stanford University and at Georgia Tech.

Property	Value
<i>db:abstract</i>	<ul style="list-style-type: none">Sebastian Thrun (born May 14, 1967) is an innovator, entrepreneur educator, and computer scientist from Germany. He was CEO and cofounder of Udacity. Before that, he was a Google VP and Fellow, and a Professor of Computer Science at Stanford University. At Google, he founded Google X. He is currently also an Adjunct Professor at Stanford University and at Georgia Tech. Thrun led development of the robotic vehicle Stanley which won the 2005 DARPA Grand Challenge, and which has since been placed on exhibit in the Smithsonian Institution's National Museum of American History. His team also developed a vehicle called Junior, which placed second at the DARPA Grand Challenge (2007). Thrun led the development of the Google self-driving car. Thrun is also known for his work on probabilistic algorithms for robotics with applications including robotic mapping. In recognition of his contributions, and at age 39, Thrun was elected into the National Academy of Engineering and also into the Academy of Sciences Leopoldina in 2007. In 2011, Thrun received the Max Planck Research Award, and the inaugural AAAI EA Fagenbaum Prize. Fast Company selected Thrun as the 6th most creative person in the business world. The Guardian recognized Thrun as one of 20 "fighters for internet freedom".

http://dbpedia.org/page/Sebastian_Thrun

Query Construction

- While **LLMs** excel at understanding natural language **instructions**, generating SPARQL queries remains a **challenging task**.
- Techniques such as **Chain of Thought (CoT)** enable LLMs to break down query construction into smaller subtasks, facilitating a step-by-step solution
- Let us understand with help of example:
 - **Question:** What countries are within the European Union?
 - **Entities:** Q39731, Q104724460, Q458 (Entities retrieved by Falcon), we need to select only one
 - Entities cannot be fully understood by simply examining their IDs, making it difficult for LLMs to interpret them effectively.
 - Therefore, we add corresponding label and description, for example
 - Q39731: Baltic states - 3 countries east of the Baltic Sea (Estonia, Latvia, Lithuania)
 - Q104724460: Russian peripheral countries : Borrowing of foreign countries
 - Q458 : European Union - political and economic union of 27 European states
 - The description makes it easy for the LLM to select the right entity id
 - **Relations:** The same steps are also followed for relations

Query Construction

- Example Continued
 - The question, the candidate entities and the relation extracted are then passed through a **prompt**.
 - **Prompt:** The prompt consists of a question, the SPARQL query and the reason of how the query is constructed. This helps the LLM to understand the reason for the query.

Answer Construction

- The output of the SPARQL query and the question is then passed through the LLM to construct the final answer.

References

- Retrieval-Augmented Generation for Large Language Models: A Survey
- Falcon 2.0: An Entity and Relation Linking Tool over Wikidata

Hands-On

Building a Basic RAG Approach

Drawbacks of the approach

- Even when grounded with entities and relationships from a knowledge graph, the LLM generates different queries with each run. Its probabilistic nature makes it inconsistent and unreliable.
- In the current setup, the generated SPARQL query are for mostly easy questions, but for hard questions it still does not perform well.
- The accuracy of the current of the setup depends on the entity and relation linker.
- The setup does not work for complex question, which requires multi-hop reasoning.

Thankyou

See you after the break.