Chatbot Task

Introduction:

According to the meeting on 24 Aug with Zwinny AB Company, we decided to narrow the task down and focus first on the chatbot which could be discussed with the customer as a service provider assistant. For example, as a car rental company when customers need data like available cars for a specified period the Bot based on a database could have a natural conversation and give the proper information to customers.

OpenAI's GPT (generative pre-trained transformer) was the main thing we needed to explore, therefore after providing API's key by company OpenAi's docs on their webpage was the thing I used to explore the concepts and tools. At the same time, LangChain, a robust framework designed for building NLP applications with the use of large language models (LLMs) such as GPT-3, has been selected. LangChain offers standardized interfaces, efficient prompt handling, and memory functionalities, empowering language models to seamlessly interact with various data sources.

Experiments:

The basic idea behind this task is developing a chatbot designed for an imaginary car rental company. The user is able to ask different questions based on predefined template (which could be adaptable) and the Bot is tasked with generating proper response in natural language using the specified knowledge.

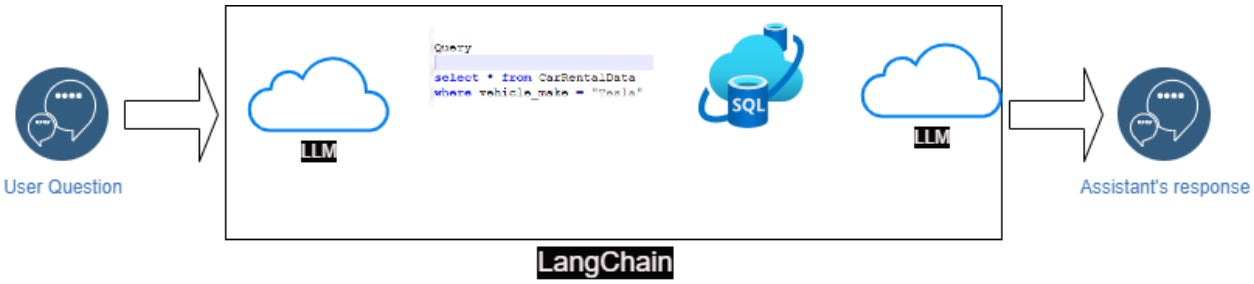

Figure1: Langchain provides tools to interact with SQL Databases

As shown in Figure 1, the user's question undergoes processing by the LLM (ChatGPT model) and is then transformed into an SQL query. To accomplish this task, the SQLite database engine is chosen, and relevant data from Kaggle is inserted into the database. The Langchain SQL Agent interacts with the SQL database in a versatile and robust manner. This entails translating a natural language question into SQL format, retrieving the appropriate response from the database, and subsequently converting it back into natural language so that the output is easy to understand to the user.

vehicle.year	vehicle.type	vehicle.model	vehicle.make	rate.daily	owner.id	location.longitude	location.latitude	location.country	location.city	reviewCount	renterTripsTaken	rating	fuelType
2019	suv	Model X	Tesla	135	12847615	-122.308841	47.449107	US	Seattle	12	13	5	ELECTRIC
2018	suv	Model X	Tesla	190	15621242	-106.276551	35.11106	US	Tijeras	1	2	5	ELECTRIC
2012	car	Prius	Toyota	35	10199256	-106.566681	35.127163	US	Albuquerque	24	28	4.92	HYBRID
2018	car	Mustang	Ford	75	9365496	-106.711425	35.149726	US	Albuquerque	20	21	5	GASOLINE
2010	car	Sebring	Chrysler	47	3553565	-106.601008	35.208659	US	Albuquerque	1	3	5	GASOLINE
2012	suv	GL-Class	Mercedes-Benz	58	7815747	-106.576111	35.068834	US	Albuquerque	12	13	5	GASOLINE
2005	suv	Yukon XL	GMC	42	3112016	-106.633815	35.045552	US	Albuquerque	12	13	4.42	GASOLINE
2018	suv	Expedition	Ford	117	9536762	-106.686481	35.186542	US	Albuquerque	10	12	4.9	GASOLINE
2016	car	Focus RS	Ford	102	14893743	-106.629406	35.061213	US	Albuquerque	1	1	5	GASOLINE
2018	suv	EcoSport	Ford	49	11389136	-106.556911	35.115363	US	Albuquerque	17	22	4.76	GASOLINE
2012	truck	F-150	Ford	75	9365496	-106.71102	35.148891	US	Albuquerque	22	23	4.95	GASOLINE
2012	car	Prius	Toyota	35	10199256	-106.598445	35.121564	US	Albuquerque	10	11	4.7	HYBRID
2018	suv	Tiguan	Volkswagen	88	9346980	-106.497368	35.078547	US	Albuquerque	7	11	5	GASOLINE
2013	car	Avalon Hybrid	Toyota	42	3070318	-106.727808	35.186381	US	Albuquerque	38	51	4.92	HYBRID
2011	car	3-Sep	Saab	42	3092621	-106.535379	35.161777	US	Albuquerque	56	61	4.88	GASOLINE
2014	minivan	CV Tradesman	Ram	65	14428787	-106.598589	35.108365	US	Albuquerque	2	2	4.5	GASOLINE
2018	suv	Cherokee	Jeep	42	1889102	-106.578805	35.115653	US	Albuquerque	0	0		GASOLINE
2007	car	Cayman	Porsche	116	3092621	-106.535441	35.161964	US	Albuquerque	20	23	5	GASOLINE
2018	suv	Wrangler JK	Jeep	119	4112693	-106.617357	35.049006	US	Albuquerque	3	5	4.67	GASOLINE

Figure 2: displays an example dataset containing information about available rental cars.

For instance, if a user wants to inquire about the number of electric SUVs availability, the resulting output might resemble the following:

```
←[1m> Entering new SQLDatabaseChain chain...←[0m
How many electric SUVs are currently available?
SQLQuery:←[32;1m←[1;3mSELECT COUNT(*) FROM "CarRentalData" WHERE "fuelType" = 'ELECTRIC' AND "vehicle_type" = 'suv';←[0m
SQLResult: ←[33;1m←[1;3m[(2,)]←[0m
Answer:←[32;1m←[1;3mThere are 2 electric SUVs currently available.←[0m
←[1m> Finished chain.←[0m
**********

User:  How many electric SUVs are currently available?

Assistant:  There are 2 electric SUVs currently available.
**********
```

For the upcoming question, the user is curious about the availability of Tesla cars manufactured after 2018.

```
Please verify if there are any Tesla vehicles with a manufacturing year beyond 2018.
SQLQuery:←[32;1m←[1;3mSELECT * FROM CarRentalData WHERE vehicle_make = 'Tesla' AND vehicle_year > 2018 LIMIT 300;←[0m
SQLResult: ←[33;1m←[1;3m[(None, 'ELECTRIC', 5, 13, 12, 'Seattle', 'US', 47.449107, -122.308841, 'WA', 12847615, 135, 'Tesla', 'Model X', 'suv',
19)]←[0m
Answer:←[32;1m←[1;3mYes, there is at least one Tesla vehicle with a manufacturing year beyond 2018.←[0m
**********

User:  Please verify if there are any Tesla vehicles with a manufacturing year beyond 2018.

Assistant:  Yes, there is at least one Tesla vehicle with a manufacturing year beyond 2018.
**********
```