Timbr LLM Docs

# Timbr LLM Connector

TimbrLlmConnector is timbr’s model handles llm inputs then generate timbr’s sql & push them to the DB to get query results.

## Variables

|  |  |  |
| --- | --- | --- |
| Key | Type (default) | Required |
| url | String | Yes |
| token | String | Yes |
| llm | langchain.llms.base.LLM | Yes |
| ontology | String (‘system\_db’) | No |

## Methods

### Initiating

llm\_wrapper = LlmWrapper(

llm\_type=<LlmTypes.ChatGPT | LlmTypes.Claude | LlmTypes.Timbr>,

api\_key=<llm\_api\_key>,

)

llm\_connector = TimbrLlmConnector(

url=<timbr\_url>,

token=<timbr\_token>,

llm=<llm\_wrapper or another LLM instance>,

ontology=<ontology to use for queries>

)

### get\_ontologies

Get a list of available knowledge graphs in timbr environment.

Usage:

ontologies\_list = llm\_connector.get\_ontologies()

Return value:

A list of string elements representing the ontologies names.

### set\_ontology

Wether you initiate the TimbrLlmConnector with ontology or without, you can change the ontology to use though this method.

Note, if you did not initiate the TimbrLlmConnector with an ontology, you wouldn’t be able to use any other methods (besides get\_ontologies) unless you set an ontology.

Usage:

llm\_connector.set\_ontology(‘<ontology name>’)

### determine\_table

Given user input, use LLM to figure which table to query

Usage:

llm\_connector.determine\_table(user\_input)

Return value:

String representing the table name

### generate\_sql

Given user input, use LLM to generate timbr sql query.

Usage:

llm\_connector.generate\_sql(user\_input, table\_name)

\*table\_name is not mandatory, if table name provided – will use this table in the query, else determine table by the user\_input.

Return value:

String representing timbr’s sql query

### run\_timbr\_query

Given timbr sql query, query the selected ontology and return the query results.

Usage:

llm\_connector.run\_timbr\_query(sql\_query)

Return value:

A list of lists representing a table data set

### run\_llm\_query

Given user input, use llm to generate timbr sql query, then query the ontology and return the query results.

Usage:

llm\_connector.run\_llm\_query(user\_input)

Return value:

Dictionary by the following schema:

{

“table”: <string - queried table name>,

“sql”: <string - generated timbr sql query>,

“rows”: <list of lists - query results from the ontology>

}

# ExecutePromptAgent

Langchain agent handles user input, generates timbr sql query using LLM, query the ontology and return the query results.

## Variables

|  |  |  |
| --- | --- | --- |
| Key | Type (default) | Required |
| timbr\_url | String | Yes |
| timbr\_token | String | Yes |
| llm | langchain.llms.base.LLM | Yes |
| ontology | String (‘system\_db’) | No |
| table | String | No |

## Usage

agent = ExecutePromptAgent(

llm=<llm>,

timbr\_url=<timbr\_url>,

timbr\_token=<timbr\_token>,

ontology=<ontology>,

table=<table>

)

OR

agent = create\_execute\_prompt\_agent\_executor(

   llm=<llm>,

   timbr\_url=<timbr\_url>,

   timbr\_token=<timbr\_token>,

   ontology=<ontology>,

   table=<table>

)

result = agent.run("What are the total sales for last month?")

Output:

Dictionary by the following schema:

{

“table”: <string - queried table name>,

“sql”: <string - generated timbr sql query>,

“rows”: <list of lists - query results from the ontology>

}

# GenerateTimbrSqlChain

Langchain chain handles user input and generates timbr sql query using LLM.

## Variables

|  |  |  |
| --- | --- | --- |
| Key | Type (default) | Required |
| url | String | Yes |
| token | String | Yes |
| llm | langchain.llms.base.LLM | Yes |
| ontology | String (‘system\_db’) | No |
| table | String | No |

## Usage

generate\_timbr\_llm\_chain = GenerateTimbrSqlChain(

url=<timbr\_url>,

token=<timbr\_user\_token>,

llm=<llm or timbr\_llm\_wrapper instance>,

ontology=<ontology\_name>,

table=<table\_name>,

)

results = generate\_timbr\_llm\_chain({ "prompt": question })

Output:

Dictionary by the following schema:

{

“table”: <string - queried table name>,

“sql”: <string - generated timbr sql query>,

}

# ExecuteTimbrLlmChain

Langchain chain handles timbr sql query and query the ontology to get results.

## Variables

|  |  |  |
| --- | --- | --- |
| Key | Type (default) | Required |
| url | String | Yes |
| token | String | Yes |
| llm | langchain.llms.base.LLM | Yes |
| ontology | String (‘system\_db’) | No |
| table | String | No |

## Usage

execute\_timbr\_llm\_chain = ExecuteTimbrLlmChain(

url=<timbr\_url>,

token=<timbr\_user\_token>,

llm=<llm or timbr\_llm\_wrapper instance>,

ontology=<ontology\_name>,

table=<table\_name>,

)

results = execute\_timbr\_llm\_chain({ "prompt": question })

Output:

Dictionary by the following schema:

{

“table”: <string - queried table name>,

“sql”: <string - generated timbr sql query>,

“rows”: <list of lists>,

}