HARNESSING POWER OF MULTI-PROMPT MULTI-RETRIEVERS

HTTPS://GITHUB.COM/INSIGHTBUILDER

CONTROL USER EXPERIENCE SEEMLESSLY WITH LESS EFFORT

NEW PRIMITIVES : CHALLENGE SOLVED

- WHEN THE USER CAN CHAT REGARDING DIFFERENT TOPICS, THE LLM HAS TO BE PROMPTED DIFFERENT. MULTI-PROMPT SOLVES THIS CHALLENGE
- USER MIGHT REQUIRE DATA THAT IS IN DIFFERENT VECTOR STORES. LLMS NEED TO QUERY ACCORDINGLY. MULTI-RETRIEVAL WORKS HERE

USE CASES OF MULTI-PROMPT:

- DIFFERENT BOT FOR DIFFERENT USERS
- CONDITIONAL PROMPTING
- PROVIDING USERS NEW CHOICES

USE CASES OF MULTI-RETRIVER:

- SAME CHAT BOT SERVING DIFFERENT TEAMS/ DOMAIN
- KEEPING INFORMATION SECURE
- ADDING DIFFERENT TYPES OF RETRIEVERS

IMPLEMENTING MULTI-PROMPT

```
FROM LANGCHAIN.CHAINS.ROUTER IMPORT MULTIPROMPTCHAIN
FROM LANGCHAIN.LLMS IMPORT OPENAI

SPARK_TEMPLATE = """YOU ARE A VERY BIG DATA ENGINEER. \YOU ARE GREAT AT ANSWERING QUESTIONS ABOUT DATA
ENGINEERING IN A CONCISE AND EASY TO UNDERSTAND MANNER. \WHEN YOU DON'T KNOW THE ANSWER TO A QUESTION YOU
ADMIT THAT YOU DON'T KNOW.HERE IS A QUESTION:{INPUT}"""

MACHINELEARN_TEMPLATE = """YOU ARE EXPERT MACHINE LEARNING DEVELOPER. YOU ARE GREAT AT ANSWERING MACHINE
LEARNING QUESTIONS. \YOU ARE CAN BREAK DOWN HARD PROBLEMS INTO THEIR COMPONENT PARTS, \ANSWER THE
COMPONENT PARTS, AND THEN PUT THEM TOGETHER TO ANSWER THE BROADER QUESTION.HERE IS A QUESTION:{INPUT}"""

PROMPT_INFOS = [{"NAME": "SPARK","DESCRIPTION": "GOOD FOR ANSWERING QUESTIONS ABOUT
SPARK","PROMPT_TEMPLATE": SPARK_TEMPLATE},
{"NAME": "MACHINELEARN","DESCRIPTION": "GOOD FOR ANSWERING MACHINE LEARNING QUESTIONS","PROMPT_TEMPLATE":
MACHINELEARN_TEMPLATE}]

CHAIN = MULTIPROMPTCHAIN.FROM_PROMPTS(OPENAI(), PROMPT_INFOS, VERBOSE=TRUE)
PRINT(CHAIN.RUN("HOW TO CREATE SCHEMA FOR SPARK TABLE?"))
```

IMPLEMENTING MULTI-RETRIEVER

```
FROM LANGCHAIN.LLMS IMPORT OPENAI
FROM LANGCHAIN.CHAINS.ROUTER IMPORT MULTIRETRIEVALQACHAIN

APP1_RETRIEVER = FAISS.FROM_DOCUMENTS(APP1_DOCS, OPENAIEMBEDDINGS()).AS_RETRIEVER()
APP2_RETRIEVER = FAISS.FROM_DOCUMENTS(APP2_DOCS, OPENAIEMBEDDINGS()).AS_RETRIEVER()
APP3_RETRIEVER = FAISS.FROM_TEXTS(APP3_DOCS, OPENAIEMBEDDINGS()).AS_RETRIEVER()

RETRIEVER_INFOS = [{
"NAME": "APP_1","DESCRIPTION": "USED FOR ANSWERING QUESTION ABOUT APP 1","RETRIEVER": APP1_RETRIEVER},
{"NAME": "APP_2","DESCRIPTION": "GOOD FOR ANSWER QUESITONS ABOUT APP 2","RETRIEVER": APP2_RETRIEVER},
{"NAME": "APP_3","DESCRIPTION": "GOOD FOR ANSWERING QUESTIONS ABOUT APP 3","RETRIEVER": APP3_RETRIEVER}]

CHAIN = MULTIRETRIEVALQACHAIN.FROM_RETRIEVERS(OPENAI(), RETRIEVER_INFOS, VERBOSE=TRUE)

PRINT(CHAIN.RUN("WHEN WAS APP 1 RELEASED TO THE MARKET?")
```

THANKS FOR WATCHING

REMEMBER TO PRACTICE WITH EXAMPLES

♥ LIKE

SHARE

SUBSCRIBE

HTTPS://GITHUB.COM/INSIGHTBUILDER