



University of Stuttgart
Institute of Industrial Automation
and Software Engineering

**Transforming Vehicle
User Manuals into
Interactive AI Chatbot
Powered by Large
Language Model**

Speaker: Juntao Lin
Supervisor: Yuchen Xia
Data: 09.09.2024

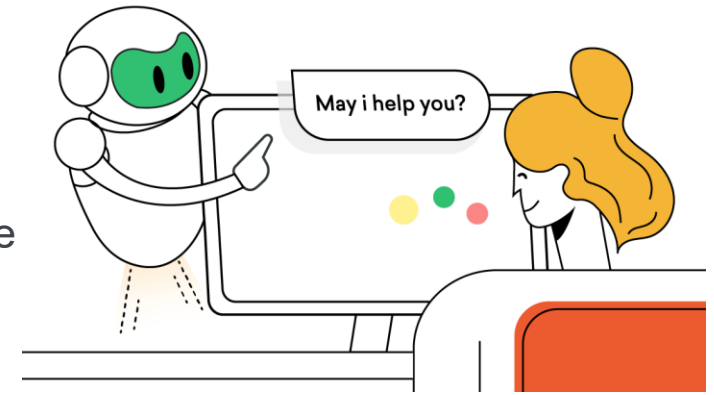


Contents

- Introduction
- Background
- Dataset Preparation
- RAG Chatbot
- Achievements
- Evaluation and Result Analysis
- Summary and Outlook

Introduction

- Motivation
 - machines understand and human language generate
 - Ability to handle a wide range of tasks
- Limitation
 - despite **generalization capabilities** models **lack specific domain knowledge**
- Objective
 - make general-purpose LLMs **specialized**
 - by transforming **vehicle user manuals into a knowledge base**
 - **RAG-based chat** assistant



Background

State of the art of LLM and RAG

Development Framework and Tools that support RAG

The dataset that support RAG

State of the art

LLM

- Large Language Models (LLMs)
 - Overview
 - versatile tools in AI applications

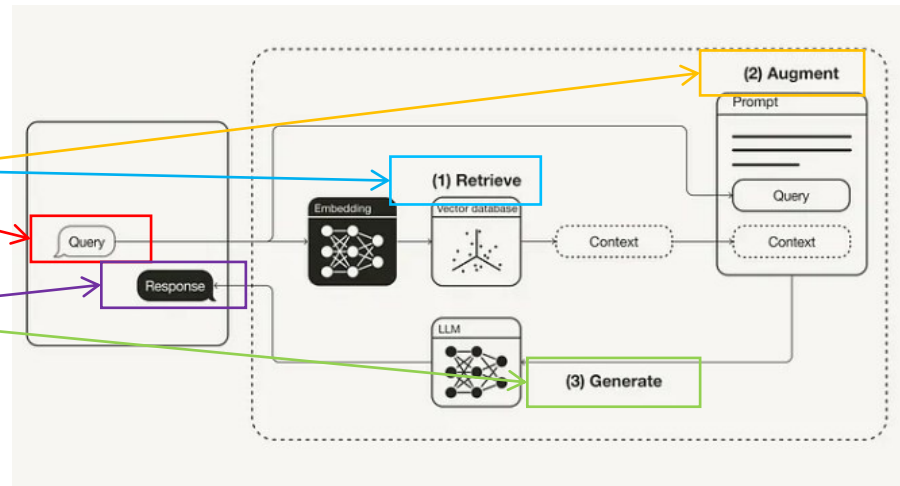
Model Name	Publishing Agency	Parameters	Token Limit	Model Structure	Open Source
GPT-3.5	OpenAI	175B	4096	GPT-3	no
GPT-4	OpenAI	1.76T	128k	GPT-4	no
T5	Google	13B	-	T5-style	yes
PaLM	Google	540B	8192	GPT-style	no
Chinchilla	DeepMind	70B	-	GPT-style	no
LLaMA	Meta	7B-65B	2048-16k	GPT-style	yes

State of the art

RAG

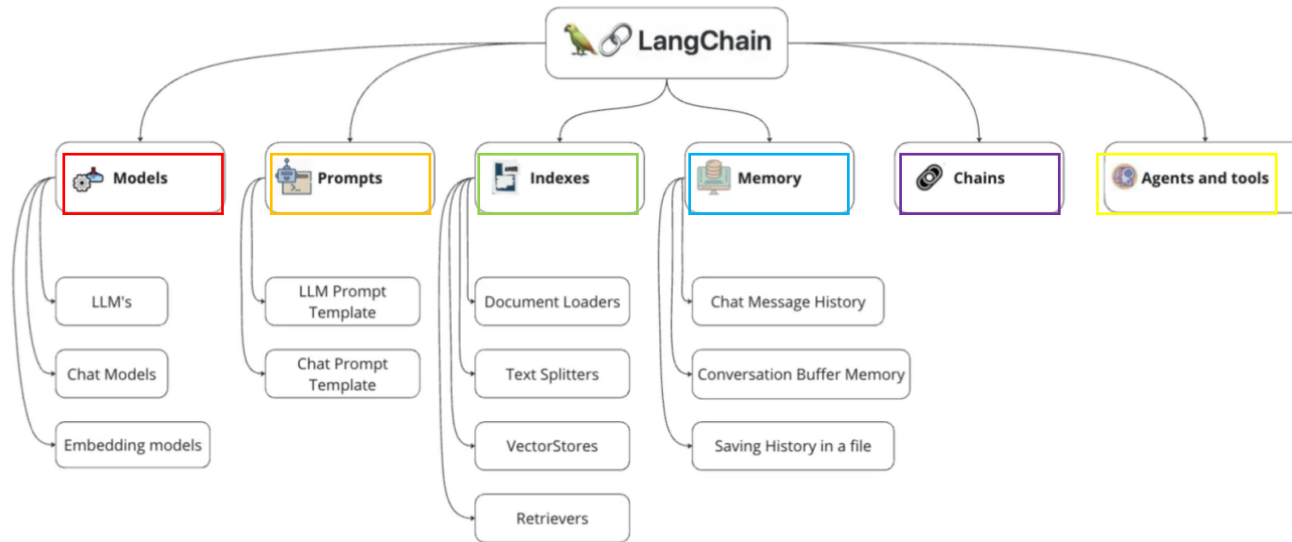
- Retrieval-Augmented Generation (RAG)
 - Addresse the limitations of LLMs in **domain-specific knowledge**
 - Allow to **retrieve relevant documents** from an **external database**
 - Provide **more precise** and **contextually relevant answers**
- **Workflow** of the RAG pipeline.

- Query
- Retrieve
- Augment
- Generate
- Response



Development Framework and Tools **that support RAG**

- Overview of development frameworks
 - **Langchain**: a **framework** designed for building applications powered by LLMs



- **Langsmith**: **debugging, testing, and evaluating** the performance of these applications.
- **Programming Language**: Python, HTML
- **Development environment**: Pycharm

The dataset that support RAG

Vehicle User Manuals as a Knowledge Base

BYD ATTO 3 OM	2024/2/22 20:00	WPS PDF 文档	36,377 KB
BYD DOLPHIN OM	2024/2/22 19:59	WPS PDF 文档	31,632 KB
BYD HAN EV OM	2024/2/22 19:58	WPS PDF 文档	57,767 KB
BYD SEAL OM	2024/2/22 19:59	WPS PDF 文档	62,985 KB
MAZDA CX-5 OM	2024/3/10 21:33	WPS PDF 文档	17,789 KB
OPEL-AMPERA E OM	2024/2/22 20:06	WPS PDF 文档	6,857 KB
OPEL-MOKKA X OM	2024/2/22 20:09	WPS PDF 文档	7,023 KB
TANG EV OM	2024/2/22 19:57	WPS PDF 文档	63,090 KB
TESLA Model 3 OM	2024/2/22 20:25	WPS PDF 文档	8,365 KB
TESLA Model S OM	2024/2/22 20:24	WPS PDF 文档	8,570 KB
TESLA Model X OM	2024/2/22 20:26	WPS PDF 文档	10,452 KB
TESLA Model Y OM	2024/2/22 20:25	WPS PDF 文档	9,944 KB
VOLVO OM	2024/2/22 19:54	WPS PDF 文档	27,538 KB

- Comprehensive documents provided by manufacturers
 - operation
 - maintenance
 - safety features

Contents	
Overview.....	3
Exterior.....	3
Interior Overview.....	4
Touchscreen.....	6
Interior Electronics.....	11
Car Status.....	13
Voice Commands.....	16
Cameras.....	18
Opening and Closing.....	19
Keys.....	19
Doors.....	22
Windows.....	24
Storage Areas.....	25
Rear Trunk.....	25
Front Trunk.....	27
Interior Storage.....	29
Seating and Safety Restraints.....	30
Front and Rear Seats.....	30
Seat Belts.....	33
Child Safety Seats.....	36
Airbags.....	42
Connectivity.....	48
Mobile App.....	48
Wi-Fi.....	50
Bluetooth.....	51
Phone, Calendar, and Web Conferencing.....	54
Smart Garage.....	56
Driving.....	59
Starting and Powering Off.....	59
Steering Wheel.....	61
Mirrors.....	65
Shifting.....	67
Lights.....	70
Wipers and Washers.....	73
Braking and Stopping.....	74
Park Assist.....	77
Vehicle Hold.....	79
Traction Control.....	80
Acceleration Modes.....	81
Driver Profiles.....	82
Trip Information.....	84
Rear Facing Camera(s).....	85
Pedestrian Warning System.....	86
Autopilot.....	87
About Autopilot.....	87
Autopilot Features.....	91
Traffic Light and Stop Sign Control.....	99
Full Self-Driving (Beta).....	106
Autopark.....	110
Summon.....	112
Smart Summon.....	114
Limitations and Warnings.....	116
Active Safety Features.....	122
Lane Assist.....	122
Collision Avoidance Assist.....	125
Speed Assist.....	128
Cabin Camera.....	129
Dashcam, Sentry, and Security.....	130
Safety & Security Settings.....	130
Dashcam.....	132
Sentry Mode.....	134
USB Drive Requirements for Recording Videos.....	136
Climate.....	137
Operating Climate Controls.....	137
Adjusting the Front and Rear Vents.....	142
Cold Weather Best Practices.....	144
Hot Weather Best Practices.....	147
Navigation and Entertainment.....	148
Maps and Navigation.....	148
Media.....	154
Theater, Arcade, and Toybox.....	156
Charging and Energy Consumption.....	159
Electric Vehicle Components.....	159
High Voltage Battery Information.....	161
Charging Instructions.....	163
Scheduled Charging and Scheduled Departure.....	169
Getting Maximum Range.....	170
Maintenance.....	172
Software Updates.....	172
Maintenance Service Intervals.....	174
Tire Care and Maintenance.....	176
Cleaning.....	183
Windshield Wiper Blades, Jets and Fluid.....	187
Jacking and Lifting.....	189
Parts and Accessories.....	190
Do It Yourself Maintenance.....	193
Specifications.....	194
Identification Labels.....	194
Vehicle Loading.....	195
Dimensions.....	197
Subsystems.....	199
Wheels and Tires.....	201

Dataset preparation

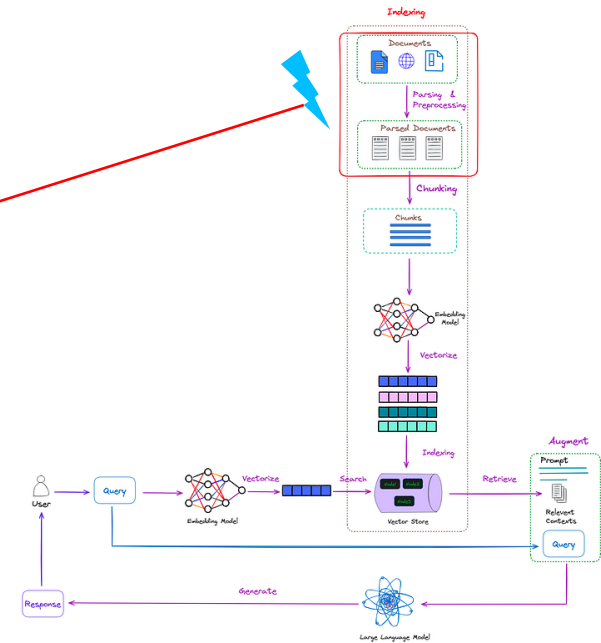
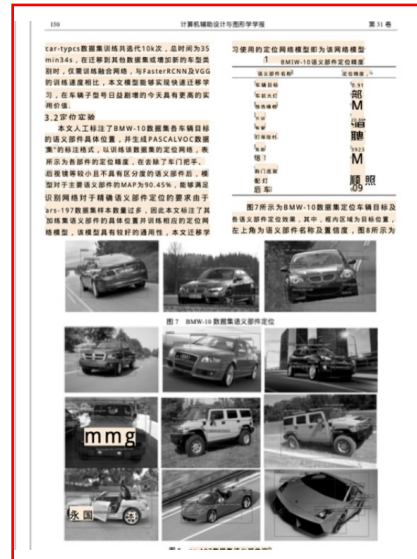
Challenges in PDF Parsing

Common Methods for Parsing PDF

Chunks of Raw Corpus

Challenges in PDF Parsing

- Inaccuracies in text extraction and layout recognition.
 - text misalignment
 - incorrect table parsing
 - loss of structural information during extraction



Common Methods for Parsing PDF

- Rule-based Approach:
 - rely on rules to extract content from PDFs.
 - for simple layouts
 - struggle with complex documents.
 - E.g.:
 - **pypdf, pdfplumber, ReportLab**
- Based on Deep Learning Models:
 - Leverage object detection and OCR models
 - better understand the structure of a document
 - require significant computational resources.
 - E.g.:
 - **Unstructured, Layout-parser, PP-StructureV2**
- Based on Multimodal Large Models:
 - combine text and image processing capabilities
 - provide a more comprehensive solution for parsing complex documents.
 - E.g.:
 - **GPT4-V, OCR model with GPT4/GPT3.5**

Chunks of Raw Corpus

- Import libraries:
 - PyPDF2, langchain_community.document_loaders, RecursiveCharacterTextSplitter
- Define functions:
 - 'get_pdf_text', 'get_text_chunks'
- Process the PDF
 - print chunks

```
1739 445
MODEL S
2021 + Notes take precedence.
OWNER'S
Software ILL
Europe
YOUR OWN The content in the owner's manual on how to use your vehicle conflicts with information in the release notes' the release
For the ver pl choose to Strip Mode is Ready to Launch.
touchscreen dependi
All information g
BEFORE NOTES
vehicle'
information f
6. Once you see "Cheetah Stance Enabled" and "Ready
to launch" on the instrument panel, release the brake
pedal to launch the vehicle. Acceleration Modes
93 DrivingTrack Mode, available only on Plaid Model S vehicles, is
designed to modify the stability control, traction control,
regenerative braking, and cooling systems to increase
performance and handling while driving on closed
circuit courses. Track Mode improves cornering ability by
```

```
1 from PyPDF2 import PdfReader
2 from langchain_community.document_loaders import PyPDFLoader
3 #from langchain.text_splitter import CharacterTextSplitter
4 from langchain.text_splitters import RecursiveCharacterTextSplitter
5 from langchain_community.embeddings import OpenAIEmbeddings
6 from langchain_community.vectorstores import FAISS
7 import os
18 def get_pdf_text(pdf_docs):
19
20     text = ''
21     for pdf in pdf_docs:
22         pdf_reader = PdfReader(pdf)
23         pages = pdf_reader.pages
24         for page in pages:
25             text += page.extract_text()
26
27     return text
28
29 # usages
30 def get_text_chunks(text):
31
32     text_splitter = RecursiveCharacterTextSplitter(chunk_size=500, chunk_overlap=50)
33     chunks = text_splitter.split_text(text)
34
35     return chunks
46 pdf_docs = ['TESLA Model S OM.pdf']
47 text = get_pdf_text(pdf_docs)
48 chunks = get_text_chunks(text)
49 print(len(chunks), len(chunks[0]))
50 print(chunks[0])
```

RAG Chatbot

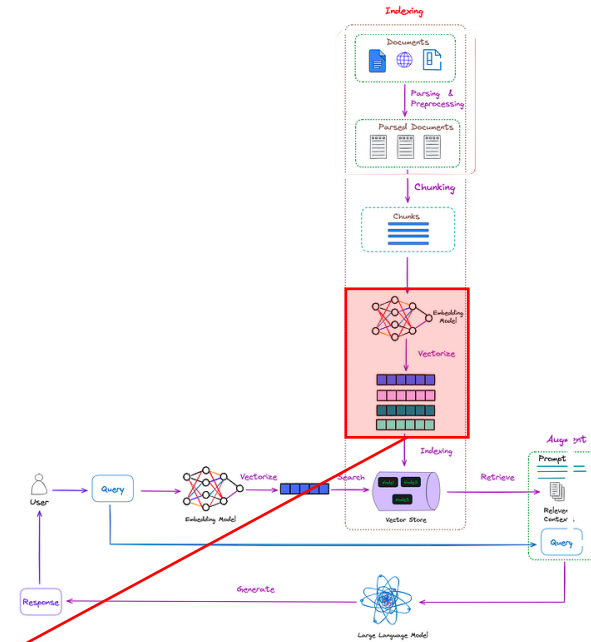
Embedding in Vector Database

Retriever Implementation

Prompt Engineering

Embedding in Vector Database

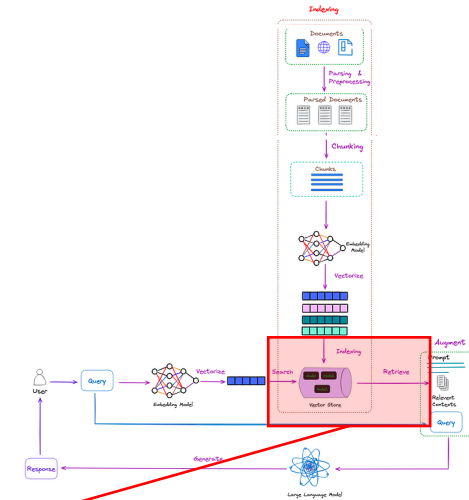
- Defining Function:
 - 'get_vectorstore'
 - Embeddings Model:text-embedding-3-small
 - Creating the Vector Store:FAISS
 - Return Statement



```
def get_vectorstore(chunks):  
    embeddings_model = OpenAIEmbeddings(openai_api_key=openai_api_key)  
    vectorstore = FAISS.from_texts(texts=chunks, embedding=embeddings_model)  
  
    return vectorstore  
vectorstore = get_vectorstore(get_text_chunks(get_pdf_text(['TESLA Model S OM.pdf'])))
```

Retriever Implementation

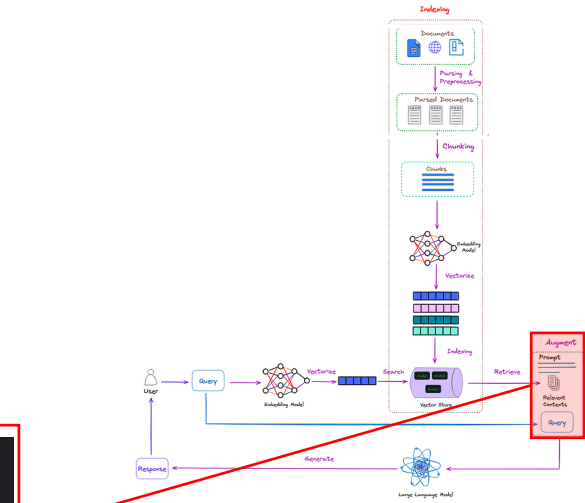
- Conversational Retrieval Chain
- Parameters
 - Chat_model:gpt3.5,llama2:7B,mistral:7B,llama3:8B
 - Search Type:Cosine Similarity
 - Search Parameters:return the **top 5** most similar results
 - Memory:remember previous interactions
 - Verbose:enables detailed logging or output



```
17 conversation_chain = ConversationalRetrievalChain.from_llm(LLM=chat_model,  
18 retriever=vectorstore.as_retriever(search_type='similarity', search_kwargs={'k':5}),  
19 memory=memory,  
20 verbose=True)
```

Prompt Engineering

- How prompts are structured for a question-answering system related to car user manuals?
- System Prompt
 - Role Definition
 - Use of Context:
 - Handling Unknowns
 - Answer Length
 - Context Insertion
- User Prompt



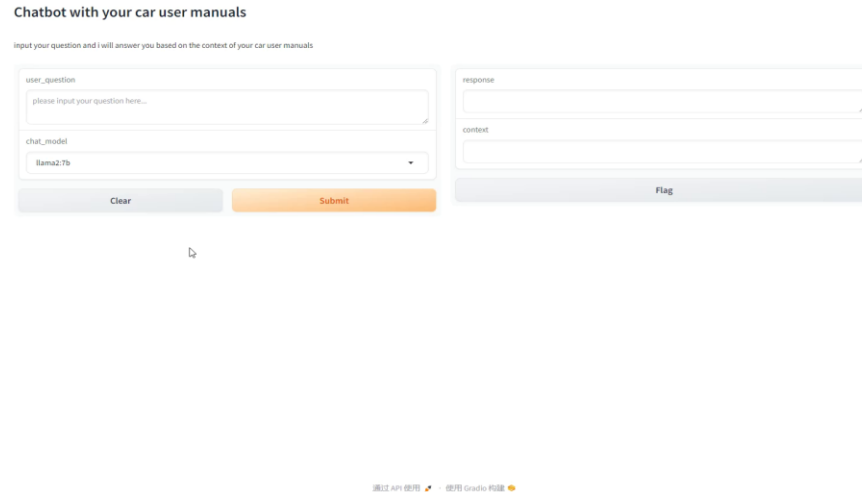
```
10 system_prompt = f"""you are an assistant for question-answering tasks related to car user manuals.
11 use the following pieces of retrieved context to answer the user's question.
12 If you don't know the answer, just say 'I don't know'.
13 Keep your answer to 150 words or less.
14 \n\n
15 {context}"""
16 user_prompt = f"""user question is: {user_question}"""
```


Achievements

User Application Interface of Chatbot

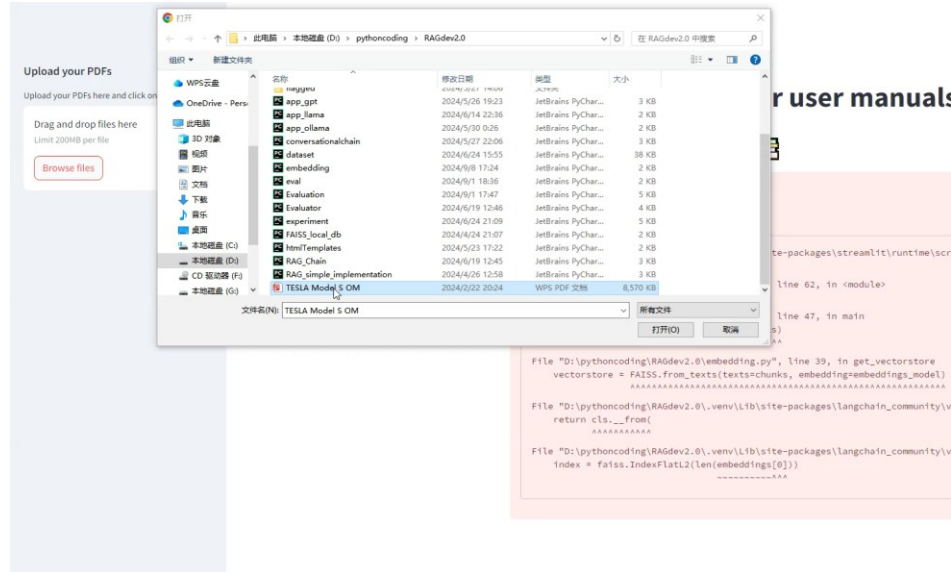
User Application Interface

- Tools Used: Streamlit and Gradio for interface development.
- Chatbot developed using Gradio
 - User Input Section:
 - Text Input Box
 - Model Selection Dropdown
 - Submit and Clear Buttons
 - Response Section:
 - Response Box
 - Context Box



User Application Interface

- Tools Used: Streamlit and Gradio for interface development.
- Chatbot developed using Streamlit
 - PDF Upload Section:
 - upload PDFs
 - File Management
 - Chat Interface:
 - Question Input
 - Response Display



Evaluation and Result Analysis

Evaluation Dataset preparation

Evaluation Indicators

Overall Performance Scores of Chatbots







Performance Scores of Chatbots in 3 Querytypes

Low Score Attribution

Evaluation Indicators




- Faithfulness (**Result correctness**):
 - Measures the factual consistency of the generated answer given the context.
- Answer Relevancy(**Stick to the topic**):
 - How relevant the answer is to the question.
- Context Relevancy(**Retrieve hit rate**):
 - The relevance of the retrieved context to the original question
- Human evaluation: Overall evaluation based on answer helpfulness.

Overall Performance Scores of Chatbots

LLM integrated in chatbot	Result correctness (Faithfulness)	Stick to the topic (Answer_relevancy_score)	Human_evaluation_score
GPT3.5	0.67	0.70	0.56 
Llama2:7B	0.57	0.87 	0.67
Mistral:7B	0.73 	0.70	0.58
Llama3:8B	0.47 	0.40 	0.69 

- Complicated & controversial evaluation result
- Result correctness:mistral:7B scored the highest
- Stick to the topic: llama2:7B performed best
- Human evaluation:gpt3.5 were surprisingly not good

Performance Scores of Chatbots in 3 Querytypes

	Querytype	Result correctness	Stick to the topic	Retrieve hit rate
GPT3.5	unanswerable question	0.49	0.38	0.04
	half-answerable question	0.72	0.79	0.06
	answerable question	0.88	0.88	0.11
Llama2:7B	unanswerable question	0.34	0.74	0.04
	half-answerable question	0.64	0.90	0.06
	answerable question	0.71	0.95	0.12
Mistral:7B	unanswerable question	0.59	0.49	0.04
	half-answerable question	0.80	0.68	0.06
	answerable question	0.66	0.88	0.12
Llama3:8B	unanswerable question	 0.24	0.12	0.04
	half-answerable question	 0.54	0.56	0.06
	answerable question	 0.49	0.53	0.12

Low score attribution

- Low score attribution and Manual labeling
 - Not retrieved(R1): the relevant information is not present
 - Missed the top ranked documents(R2):the relevant information exists but ranks too low
 - Not used by generative model(R3):correct information have not been used to produce responses
 - Noise in retrieved information(R4): irrelevant or low-quality content retrieved

Low Score Attribution

- Low Score Attribution For All half-answerable question

	R1: Not retrieved	R2: Missed the top ranked documents	R3:Not used by generative model	R4: Noise in retrieved information
Score 1: Result correctness	0.82	not observed	0.96	0.96
Score 2: Stick to the topic	not observed	1.00	not observed	0.88
Score3:Retrieve hit rate	0.70	0.94	0.91	0.46

- Low Score Attribution For All answerable question

	R1: Not retrieved	R2: Missed the top ranked documents	R3:Not used by generative model	R4:Noise in retrieved information
Score 1: Result correctness	0.61	not observed	0.75	0.93
Score 2: Stick to the topic	not observed	0.90	not observed	1.00
Score3:Retrieve hit rate	0.52	0.95	0.86	0.45

Low Score Attribution

- Comparison of low score attribution for half-answerable question of each model

LLM integrated in chatbot	scoreclass	Not retrieved	Missed the top ranked documents	Not used by generative model	Noise in retrieved information
GPT3.5	Result correctness	0.80	not observed	1.00	1.00
	Stick to the topic	not observed	1.00	not observed	1.00
	Retrieve hit rate	0.63	0.94	0.88	0.31
Llama2:7B	Result correctness	1.00	not observed	1.00	1.00
	Stick to the topic	not observed	1.00	not observed	0.00
	Retrieve hit rate	0.80	0.93	0.93	0.47
Mistral:7B	Result correctness	0.83	not observed	1.00	1.00
	Stick to the topic	not observed	1.00	not observed	1.00
	Retrieve hit rate	0.69	1.00	0.94	0.50
Llama3:8B	Result correctness	0.89	not observed	1.00	1.00
	Stick to the topic	not observed	1.00	not observed	0.86
	Retrieve hit rate	0.69	0.94	0.88	0.56

Low Score Attribution

- Comparison of low score attribution for answerable question of each model

LLM integrated in chatbot	scoreclass	Not retrieved	Missed the top ranked documents	Not used by generative model	Noise in retrieved information
GPT3.5	Result correctness	0.50	not observed	1.00	1.00
	Stick to the topic	not observed	1.00	not observed	0.88
	Retrieve hit rate	0.33	0.87	1.00	0.13
Llama2:7B	Result correctness	0.88	not observed	0.88	0.88
	Stick to the topic	not observed	0.00	not observed	0.00
	Retrieve hit rate	0.67	0.73	0.93	0.40
Mistral:7B	Result correctness	0.63	not observed	0.50	1.00
	Stick to the topic	not observed	1.00	not observed	1.00
	Retrieve hit rate	0.60	1.00	0.53	0.53
Llama3:8B	Result correctness	0.40	not observed	0.80	1.00
	Stick to the topic	not observed	0.88	not observed	1.00
	Retrieve hit rate	0.54	1.00	1.00	0.77

Summary and Outlook

Summary and Outlook

- Summary of Achievements:
 - successfully verified the **effectiveness of RAG technology**
 - transforming static vehicle manuals into **a dynamic, interactive LLM chatbot**
 - paves the way for more innovative applications of LLMs **in specialized domains**
- Challenges and Future Work:
 - focus on improving the chatbot's ability.
 - enhance Contextual Understanding
 - dynamic Content Updating
 - feedback Loop Mechanism
 - improved Prompt Engineering

Question and Answer



University of Stuttgart
Institut of Industrial Automation
and Software Engineering

Thank you!

Juntao Lin

e-mail st176526@stud.uni-stuttgart.de

phone +49 (0) 711 685- 15773217446

fax +49 (0) 711 685-

University of Stuttgart
Institute of Industrial Automation and Software Engineering

Pfaffenwaldring 47, 70550 Stuttgart

