Development of an Expert Chatbot for Digital Forensics Using RAG Model Implementation

Minsoo Kim
Sungkyunkwan University
Seoul, Republic of Korea
pinggoo001@g.skku.edu

Doyoun Kim
Dongguk University
Seoul, Republic of Korea
zkvpdls1@dgu.ac.kr

Yunji Park
Sungkyunkwan University
Seoul, Republic of Korea
yun.jiggle@g.skku.edu

Doowon Jeong*
Sungkyunkwan University
Seoul, Republic of Korea
doowon@g.skku.edu

2024 International Conference on Platform Technology and Service (PlatCon) | 979-8-3503-6787-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/PLATCON63925.2024.10830748

***Abstract*—Digital forensics is an important evidence collecting technique in criminal justice system. However, due to the invisibility of digital evidence, the results of analysis vary depending on the expertise of the investigator. Though there has been a trend to use commercially available chatbots to minimize the divergence of investigator's expertise, the effect was limited by unprofessional response and lack of standardized information sharing system. Therefore, this study proposed the data construction method for constructing expert chatbot based on web crawling of digital forensics related domains. Then, the collected data was used to build the new digital forensics knowledge DB. In addition, this research suggested the RAG based chatbot model that can address the problems of existing chatbots, such as generating misinformation and limitation in updating train data, and generate specialized answers based on the digital forensics knowledge DB. As a result, this research presented the possibility of establishing information sharing system of digital forensics and utilizing the specialized chatbot based on the system. Additionally, it provided the foundation of a multimodal chatbot for analyzing various forms of digital evidence.**

***Index Terms*—Digital Forensics, ChatGPT, RAG, Information System, Chatbot**

I. Introduction

Digital forensics is the study of searching and collecting evidences required in criminal justice system based on engineering such as computer science and ICT. The data of digital devices, the subject of digital forensics, is invisible. Invisibility is a characteristic of digital data that human cannot see the contents of data with unaided eye and must use the decoding devices to recognize the contents. Because of the invisibility, results of forensic investigation can differ depending on the investigator's understanding and expertise of data. This directly affects the overall reliability of the investigation, including evidence collection and analysis.

To resolve the variation of expertise, standardized information sharing system that can present digital forensics information is needed. To date, most of the information sharing systems including 'Open Source DFIR(OSDFIR)', 'Forensics Wiki' have been managed in a form of web pages [1], [2]. However, existing websites are not regularly maintained. Moreover, the lack of accessibility to information limits its function as a standardized information sharing system.

Recently, as technology has evolved, there has been a trend of using chatbots with large language model to build knowledge bases. However, the train data was based on generalized knowledge domains so that the generated answers has lacked speciality. It also has been difficult to update the train data and generated misinformation.

Therefore, this study proposed two steps to design a chatbot with expertise as a new information sharing system. First, it suggested a method to collect the specialized train data in the field of digital forensics. Next, it presented the structure of chatbot model that improves the reliability of answers to digital forensics domain by using RAG architecture. In the process, potential to build information sharing system of digital forensics and utilize specialized chatbot were confirmed. In addition, foundation for the multimodal chatbot that can analyze various types of digital evidence was provided.

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2024-00398745, Proofs and response against evidence tampering in the new digital environment)

*Corresponding author.

II. Related Works

A. Information Sharing System of Digital Forensics

There are two digital forensics-related information sharing systems available online: namely, Open Source DFIR and Forensics Wiki. Open Source DFIR is the websites that posts information of digital forensics. It especially posts the usage and patch history of TIMESKETCH, PLASO, LOG2TIMELINE which are used to analyze timeline and shares other information of analyzing digital evidence. The website has been continuously posting and managed, with the last post published on April 18, 2024. Still, it is difficult to read posts, as posts are not tagged to identify the topic of each, and it is not available to find a list of entire posts. Besides, most of the posts are related to specific tools and the publisher is not clearly identified. This cause the lack of versatility of information so that Open Source DFIR is not capable to use as standardized information sharing system.

Forensics Wiki is another website that share the digital forensics related information, and uses tags to categorize the topics and organize the posts. Additionally, it manages the posts by using github for ease of writing and editing the posts. The website also continuously creating the articles, with the last post published on May 18, 2024. However, academic

979-8-3503-6787-4/24/$31.00 ©2024 IEEE

research data is not sufficient so that it is hard to find the academic trend through Forensics Wiki.

As existing information sharing system in the field of digital forensics has limitations of universality and periodic management, new information system is needed.

B. Development of Chatbot

Chatbots, also defined as conversational artificial intelligence, are categorized into three types based on the technology those use. The earliest form, 'pattern-based model(rule-based model)', generates answers according to a certain format or patterns of input designed by the developer. ELIZA is typical of the pattern-based model and characterized by a low degree of freedom in asking and answering questions [3], [4].

The next generation of chatbot is the 'retrieval-based model' [5]. This model learns from train data consists of multiple Q-A pairs. The train data is converted in to vector values through embedding procedure which translate the various formats of data into numeric data that can be computed easily by machine. When user enter the query, model retrieve the most similar question data from the Q-A pairs in train data. Caculation of similarity uses methods like TF-IDF and cosine similarity [6]. Given the similarity between user query and question data, model provides answer data matches with the most similar question in the pair of Q-A as the answer of user's query. Since the model presents answers within the train data, its performance is highly relying on the quantity and quality of the data. A representative example is DocChat, which has a higher degree of freedom in answering compared to patterned models, still has the limitation of structured answers [7].

The most recent model is the 'geneartion-based model'. This model is possible to generate various type of data and the model specialized in processing language data is defined as 'language model'. Language model is divided into 'Seq2Seq' and 'Transformer' [8]. Each model generates the sentence by learning the order of word placement or learning the contextual importance of words based on train data. These generative models have advantage over the previous models in that they have high degree of freedom to generate sentence [9]. There have been previous researches that utilize the characteristic of generation-based model to build chatbots in digital forensics field, such as comparing the applicability of various LLMs and model architectures or measuring the applicability of ChatGPT in the digital forensics field [10], [11].

C. Usage of RAG

Although generation-based model is highly versatile, it does not always generate correct answer and hallucinations can frequently occur during the generation. Consequently, users are misled because of the hallucination [12]. Also, due to high cost for training, it is it is challenging to update the train data, resulting in a lack of up-to-date data [13].

There is chatbot architecture that applies fine-tuning to generative models, but their use is limited by the characteristics of the digital forensics field. Digital forensics has a fast cycle in which new concepts and data advent through updates and development of OS, SW, HW, etc. For this reason, a model that can continuously update data is required for effective utilization of digital forensics chatbots.

Fine-tuning involves updating parameters inside the model in response to train data and the entire re-training process is required whenever the train data changes. This process requires a large amount of computing resources and is expensive in terms of time and resources. In other words, fine-tuning-based models are not flexible enough to handle changes in train data, which is a limitation when applied to digital forensics, where data change cycle is fast.

Retrieval Augmented Generation(RAG) is a new model that has emerged to compensate for the limitations of generation-based model [14]. The basic form consists of the retriever that searches for documents similar to the user's query from train data and the generator that generates answers based on the retrieved documents. The retriever retrieves similar documents based on the retrieval DB, which is a set of documents embedded as vector values. During the retrieval, retriever searches for the k documents that have the highest similarity to the user's input question, the same as the retrieval-based model. In this case, k is an arbitrary value that can be set by the user and represents the number of documents to be retrieved in order of highest similarity. Then, generator applies noise techniques to the retrieved documents and learns to infer the correct answer from the masked documents. As a result, generator generates new sentences based on the retrieval DB, but not identical to the retrieved documents.

One application of RAG is RAPTOR, which is a technique for efficiently embedding long data into the search DB within RAG [15]. RAPTOR summarizes the data to create an embedding storage file in a tree structure. The tree structure is used to cluster similar content using a summary of the embedded sentences. Each LLM-based summary of a partitioned document becomes a node, and nodes are organized from leaf to root.

III. System Architecture

A. Construction of Retrieval Data

1) Criteria and Scope of Data collection: To build a RAG-based digital forensics chatbot, retrieval DB for the retrieval of similar documents was constructed. For this process, data for retrieval DB was collected based on three criteria. First, data with authoritativeness was collected. The documents published by academic institutions were considered authoritative. Documents officially published by organizations or companies were also included as authoritative data. Additionally, open-sourced documents that can be verified by a large number of people were considered to be authoritative. The second criterion is accessibility. In this research, accessibility is defined as the ability to read and collect the content of a document or data online. Finally, possibility of automating collection was used as criteria. Documents with a uniform data structure that can be automated to extract the content were collected.

Based on the three criteria, Forensics Wiki was selected as the starting point for data collection in this study. As

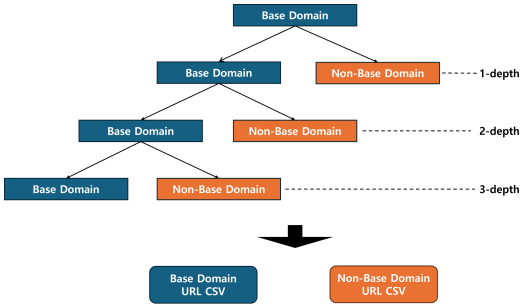

Fig. 1. Structure of URL Crawler

mentioned, Forensics Wiki was selected to collect data because it is regularly managed among the existing information sharing systems, and it is easy to access and generalize posts using tags. Starting from the tag list of Forensics Wiki, contents were crawled and collected up to 3 depths. In this case, the depth increases by 1 while moving from one page to the attached links. The process is shown in figure 1. There are two main types of pages that are crawled. One is the base domain, which are pages in the Forensics Wiki domain that are linked from the tag list of Forensics Wiki.

The other is non-base domain, which are pages in domains other than Forensics Wiki that are referred in each page of Forensics Wiki. Non-base domain pages include official documentation published by specialized companies such as Microsoft, Exterro, Magnet Forensics, Opentext, and others. It also includes webpages that provide tools for forensics, such as Github, Mitec, Sourceforge, etc. The last type of pages in the Non-Base Domain are academic articles published in conferences and journals.

There are also pages excluded from the collection because they do not meet the criteria. Excluded pages include Wikipedia and blogs and articles written by individuals. Wikipedia is a web domain that contains a conceptual definition and basic description. It is characterized by its ease of access online and the capacity for an unspecified number of people to edit the content. However, this leads to a limitation in collecting the content of each page, as it is impossible to verify its expertise. Personal blogs and posts consist of data which people working in the field of digital forensics explain concepts and phenomena related to the field. These pages contains an amount of data explaining concepts and basic definitions, but lack timeliness due to the absence of management. Moreover, it is difficult to verify the expertise of individuals who wrote the article.

2) Collection of Retrieval Data: In order to crawl the content that falls within the scope of the search data collection, the URL of each page was first collected. 'https://forensics.wiki/tags/' was specified as the root page to crawl and collect URLs up to 3 depths. The structure of the URL crawler is shown in figure 1.

The collected URLs are saved in separate files for each domain and utilized for crawling. The elements corresponding to the body of each domain are then analyzed to determine which elements to collect for each URL domain. To collect the specified elements, four document loaders (WebBaseLoader [16], ArxivLoader [17], OnlinePDFLoader [18], PyPDFLoader [19]) provided by Langchain was used, which provides a framework for developing LLM-based apps. To improve limitation of Forensics Wiki, additional academic data was collected from digital forensics-related conferences such as Digital Forensics Research Conference (DFRWS), Forensic Science International: Digital Investigation, etc. using ArxivLoader.

Furthermore, the collected data was built in the form of a DB to be utilized as a new information sharing system. The database consists of a domain table and multiple URL tables. Domain table is a table to manage the collected URLs by domain. URL table is a table that manages URLs of pages belonging to each domain as multiple records. In the DB, Domain ID is the primary key and foreign key, which gives the connection between the Domain table and each URL table. The schema of the DB on which the information sharing system is based is shown in figure 2.

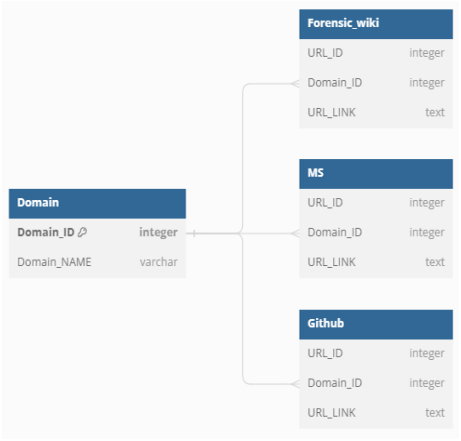

Fig. 2. DB Schema of information system

B. Construction of Model

In this study, the digital forensic chatbot is composed of RAG using RAPTOR as shown in figure 3. First, it crawled

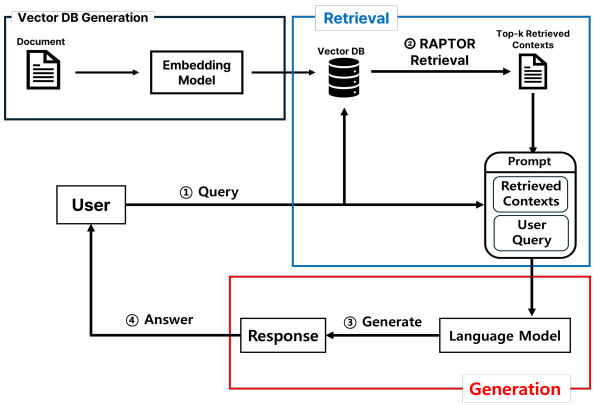

Fig. 3. Model Architecture

the contents form URLs of each domain. The crawled contents were stored in the form of retrieval DB to generate answers to user queries. Then, RAPTOR organized the retrieval DB as a tree structure with multiple nodes. OpenAI's 'text-embedding-3-small' was used for embedding retrieval DB, and the API of the GPT-4o model worked as generator to generate answers based on the retrieved documents.

IV. Experiments

The experiment was conducted to measure the answer reliability and accuracy of the RAG chatbot built in this study, and the same questions were entered into the ChatGPT-4o model, which is the same as RAG generator, for performance comparison. The k of the RAG retriever was set to 10 to search the top 10 documents in the retrieval DB based on similarity. 1,933 URLs was collected through the crawling steps presented in Chapter 3, consisting of 1,305 base domain URLs and 628 non-base domain URLs, but the retrieval DB built for the experiment was based on 5 web pages of Forensics Wiki related to memory analysis. In addition, prompts were adjusted to provide detailed answers to each question, but if there is no matching answer, it is output as unknown.

To test the performance of the model built in this study, the questions were categorized into abstract questions, concrete questions, and equivocal questions and presented to the model. Abstract question referred to questions about a wide range of content, while specific question were narrower in scope. Equivocal question asked for the definition of the same word or terms with different meanings. The types of questions entered into the model are as follows.

- Abstract Question: Give me techniques to perform anti-forensics on tools that can acquire memory.
- Detailed Question: Where can I get some sample images of windows memory data?
- Equivocal Question: What is SIMCon?

A. Evaluation

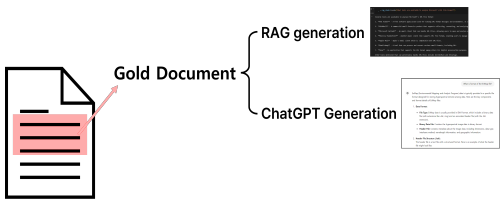

Fig. 4. Evaluation Method

The evaluation of the model was performed using Gold Document as shown in figure 4. A Gold Document is a document that contains the correct answer to an arbitrary question among the documents that comprise retrieval DB. In this study, Gold Documents were selected from the retrieval DB based on the three questions presented above. Then, BLEU score of the answers generated by the RAG chatbot and ChatGPT-4o was measured to check the extent to which the generated sentences were included in the Gold Document.

V. Results

The generation outputs of RAG and ChatGPT-4o were compared based on BLEU scores, as shown in Table I. For the abstract question, the generation output of RAG and ChatGPT-4o respectively received BLEU scores of 0.4761 and 0.3326. Next, for the detailed question, RAG generation and ChatGPT-4o generation each scored 0.6356 and 0.1757. Finally, RAG and ChatGPT-4o generation individually scored 0.6699 and 0.3721 for the equivocal question. This indicates that RAG generates responses based on data from the search database, and the variability in generated responses can be attributed to the differing content of the database.

TABLE I
Comarison of BLEU Score between RAG and ChatGPT Generation

Question Type	RAG Generation	ChatGPT Generation
Abstract	0.47613	0.33267
Detailed	0.63566	0.17579
Equivocal	0.66991	0.37219

The following subsections A-C are examples of RAG generation,which were produced by referencing the Gold Document, and ChatGPT generations for the three questions.

A. Abstract Question

It was observed that to the abstract question, 'Give me techniques to perform anti-forensics on tools that can acquire memory', the RAG generation formulated a response by referencing the Gold Document. Table II and figure 5 present examples of the Gold Document, RAG Generation, and ChatGPT-4o Generation.

Performing anti-forensics on memory acquisition tools involves techniques aimed at hindering or misleading the process of capturing or analyzing volatile memory (RAM). These techniques can include:

1. Obfuscation
Code Obfuscation: Use techniques to make the code of running processes more difficult to analyze. This can involve using packers or crypters to encrypt and compress the executable files. Function Obfuscation: Rename or scramble function names and variable names to prevent easy identification by memory analysis tools.

2. Anti-Debugging Techniques
Detecting Debuggers: Implement checks within your code to detect the presence of debuggers. If a debugger is detected, the program can alter its behavior or terminate. Debugger Evasion: Use timing checks and anti-debugging API calls (e.g., IsDebuggerPresent, CheckRemoteDebuggerPresent) to evade memory acquisition tools that rely on debugging methods.

3. Memory Manipulation
Memory Cleansing: Periodically overwrite sensitive information in memory to ensure it is cleared before an acquisition can take place. Direct Kernel Object Manipulation (DKOM): Alter kernel structures directly to hide processes, threads, or other objects from being detected by memory acquisition tools.

Fig. 5. ChatGPT Generation of abstract question

B. Detailed Question

RAG generated an answer based on Gold Document for the detailed question, 'Where can I get some sample images of windows memory data?'. Compared to the abstract question, the content of Gold Document was more likely to be used. Table III and figure 6 are example of Gold Document, RAG Generation, and ChatGPT Generation.

TABLE II
COMPARISON BETWEEN GOLD DOCUMENT AND RAG GENERATION ABOUT ABSTRACT QUESTION

index	Gold Document	RAG Generation
1	"We present a novel memory acquisition technique, based on direct page table manipulation and PCI hardware introspection, without relying on operating system facilities - making it more difficult to subvert"	"Page Table Manipulation: By directly manipulating the page tables, an attacker can control which memory regions are accessible or visible to the memory acquisition tool. PCI Hardware Introspection: Using hardware-based methods to introspect and manipulate memory contents."
2	"This approach is especially effective against memory acquisition, since the volatility of the evidence does not permit the investigator to reacquire the memory under the same conditions"	"Since memory evidence is volatile and cannot typically be reacquired under the same conditions, an attacker can exploit this by rapidly changing the memory contents or causing system reboots to disrupt the acquisition process."
3	"We incorporated this novel acquisition technique into the open source "WinPmem" memory acquisition tool (Cohen, 2012a), making it resilient to current published anti-forensic attacks."	"The development of resilient memory acquisition tools, such as the novel technique integrated into the "WinPmem" tool, is crucial for countering these anti-forensic strategies."

TABLE III
COMPARISON BETWEEN GOLD DOCUMENT AND RAG GENERATION ABOUT DETAILED QUESTION

index	Gold Document	RAG Generation
1	"The Digital Forensics Tool Testing project has published a few Windows memory images.The CFReDS Project has created some downloadable memory images."	"Notably, the Digital Forensics Tool Testing project and the CFReDS Project have published some Windows memory images that you can download."
2	"A number of RAM images can be downloaded from https://belkasoft.com/x. Images include ones with Gmail emails, Skype activity, Paltalk chats, browser URLs etc."	"Additionally, the website https://belkasoft.com/x offers a variety of RAM images, including those with Gmail emails, Skype activity, Paltalk chats, browser URLs, and more."
3	"Getting started with memory analysis can be difficult without some known images to practice with."	"These resources provide valuable practice material for getting started with memory analysis."

C. Equivocal Question

To the equivocal question, 'What is SIMCon?', the RAG generation formulated a response by referencing the Gold Document. Unlike the response generated by ChatGPT-4o which introduced SIMCon as New York's longest running tabletop gaming convention, RAG generation introduced SIMCon, a tool used in the field of digital forensics.This indicates that the response provided by RAG is more specialized and expert-level. Table IV and figure 7 present examples of the Gold Document, RAG Generation, and ChatGPT-4o Generation.

You can find sample images of Windows memory data in several places, depending on the specific type of data you're looking for. Here are a few options:

1. **Official Documentation and Tech Blogs**:
   - Microsoft Docs often provides screenshots and sample images related to memory management and performance data.
   - Blogs from Microsoft or other tech experts that focus on system performance and memory management might have relevant images.

2. **Tech Forums and Communities**:
   - Websites like Stack Overflow, Reddit (r/sysadmin, r/windows), and Super User may have threads with users discussing memory data, often including screenshots.

3. **Educational Websites and Online Courses**:
   - Websites like Udemy, Coursera, and Pluralsight that offer courses on Windows administration or computer science topics often include sample data and screenshots in their materials.

4. **Research Papers and Technical Reports**:
   - ResearchGate, Google Scholar, and similar platforms might have research papers and technical reports that include sample images of memory data.

Fig. 6. ChatGPT Generation of detailed question

TABLE IV
COMPARISON BETWEEN GOLD DOCUMENT AND RAG GENERATION ABOUT EQUIVOCAL QUESTION

index	Gold Document	RAG Generation
1	"SIMCon is now Paraben's SIM Card Seizure. SIMCon is an application developed by Inside Out Forensics in Norway."	"SIMCon, now known as Paraben's SIM Card Seizure, is a specialized forensic software developed by Inside Out Forensics in Norway."
2	"SIMCon is a program that securely images all files on a GSM SIM Card with a standard PC-SC smart card reader, either serial or USB. Specific information regarding stored numbers, call history, and text messages are available."	"It is designed primarily for law enforcement to securely image all files on a GSM SIM card using a standard PC-SC smart card reader (either serial or USB)."
3	"Recover deleted text messages stored on the card. Manage PIN and PUK codes. Compatible with SIM Cards and USIM Cards"	"Key functionalities include recovering deleted text messages, managing PIN and PUK codes, and ensuring compatibility with both SIM and USIM cards."

SIMCon, or the Simulation Gaming Convention, is New York's longest-running tabletop gaming convention, held annually at the University of Rochester. Organized by the University's Simulation Gaming Association (URSGA), the convention brings together enthusiasts of various gaming genres, including board games, trading card games, tabletop RPGs, and wargames.
The 45th edition of SIMCon will take place from March 22nd to March 24th, 2024, at the Frederick Douglass Commons on the University of Rochester campus. This year's theme is "Time Travel," offering a variety of themed events, tournaments, and a game library with over 300 board games. The convention also features local vendors and special guests, providing a comprehensive gaming experience for attendees of all ages.
For more information and to register for the event, you can visit the official SIMCon website (SA Rochester) (CCC Rochester) (Coscove | Cosplay marketplace).

Fig. 7. ChatGPT Generation of equivocal question

VI. Conclusion

The chatbot presented in this paper was able to provide expert and reliable answers in the forensic field by utilizing the RAG technique. Even when compared to ChatGPT-4o, a widely accessible generative language model, the results from the chatbot presented in this paper were found to be more forensically professional and had reliable sources for the responses. Based on the results obtained in this study, two critical aspects essential for the development of a domain-specific chatbot have been identified.

The first aspect is the collection of domain-specific professional data. It is important that collected data is sourced not merely from simple searches but from reliable and authoritative references. Another crucial aspect is the preprocessing of the collected data. Given that preprocessing can significantly influence the final outcomes, even when utilizing identical datasets, attention must be paid to this step. It is also essential to explore strategies, like the RAPTOR method used in this study, to efficiently utilize the acquired data in the model.

Furthermore, by collecting data not only from 'Forensics Wiki,' which served as a key source for digital forensic-related data in this study, but also from various academic journals, papers, and other diverse knowledge sources, a more effective domain-specific chatbot could be developed. Therefore, future research will aim to enhance these aspects.

References

[1] J. Metz, "Open source dfir," osdfir.blogspot.com. Accessed: Aug. 7, 2024. [Online]. Available: https://osdfir.blogspot.com/

[2] ——, "Welcome to the forensics wiki," forensics.wiki. Accessed: Aug. 7, 2024. [Online]. Available: https://forensics.wiki/

[3] J. Weizenbaum, "Eliza—a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966.

[4] M. Wahde and M. Virgolin, "Conversational agents: Theory and applications," in *HANDBOOK ON COMPUTER LEARNING AND INTELLIGENCE: Volume 2: Deep Learning, Intelligent Control and Evolutionary Computation*. World Scientific, 2022, pp. 497–544.

[5] Z. Jia, Z. Lub, and H. Lib, "An information retrieval approach to short text conversation," *arXiv preprint arXiv:1408.6988*, 2014.

[6] X. Li, L. Mou, R. Yan, and M. Zhang, "Stalematebreaker: A proactive content-introducing approach to automatic human-computer conversation," *arXiv preprint arXiv:1604.04358*, 2016.

[7] Z. Yan, N. Duan, J. Bao, P. Chen, M. Zhou, Z. Li, and J. Zhou, "Docchat: An information retrieval approach for chatbot engines using unstructured documents," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 516–525.

[8] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong and M. Strube, Eds. Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 1577–1586. [Online]. Available: https://aclanthology.org/P15-1152

[9] S. Pandey and S. Sharma, "A comparative study of retrieval-based and generative-based chatbots using deep learning and machine learning," *Healthcare Analytics*, vol. 3, p. 100198, 2023.

[10] M. Scanlon, F. Breitinger, C. Hargreaves, J.-N. Hilgert, and J. Sheppard, "Chatgpt for digital forensic investigation: The good, the bad, and the unknown," *Forensic Science International: Digital Investigation*, vol. 46, p. 301609, 2023.

[11] A. Wickramasekara, F. Breitinger, and M. Scanlon, "Sok: Exploring the potential of large language models for improving digital forensic investigation efficiency," *arXiv preprint arXiv:2402.19366*, 2024.

[12] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.

[13] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[14] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[15] P. Sarthi, S. Abdullah, A. Tuli, S. Khanna, A. Goldie, and C. D. Manning, "Raptor: Recursive abstractive processing for tree-organized retrieval," *arXiv preprint arXiv:2401.18059*, 2024.

[16] Langchain, "Webbaseloader," python.langchain.com. Accessed: Aug.7, 2024. [Online]. Available: https://python.langchain.com/v0.2/docs/integrations/document_loaders/web_base/

[17] ——, "Arxivloader," api.python.langchain.com. Accessed: Aug.7, 2024. [Online]. Available: https://python.langchain.com/v0.2/docs/integrations/document_loaders/arxiv/

[18] ——, "langchain_community.document_loaders.pdf.onlinepdfloader," api.python.langchain.com. Accessed: Aug.7, 2024. [Online]. Available: https://api.python.langchain.com/en/latest/document_loaders/langchain_community.document_loaders.pdf.OnlinePDFLoader.html

[19] ——, "langchain_community.document_loaders.pdf.pypdfloader," api.python.langchain.com. Accessed: Aug.7, 2024. [Online]. Available: https://api.python.langchain.com/en/latest/document_loaders/langchain_community.document_loaders.pdf.PyPDFLoader.html