

BankCARE: Advancing Bank Services with Enhanced LLM and Retrieval Generation

Deyu Chen¹ and Xiaofeng Zhang*¹

¹ University of Wollongong , Compute Science, Australia

² Shanghai Jiao Tong University, Electronic Information and Electrical Engineering, China
dc321@uow.edu.cn

Abstract. In the rapidly evolving fintech landscape and the ongoing digital transformation of banking, this study explores the application of a Retrieval Augmented Generation (RAG)-based approach to enhance bank customer service efficiency and quality. Facing challenges such as complex requirements, personalized solutions, data privacy, and dialogue system intelligence, we implement an existing RAG framework using the LangChain toolkit. By integrating vector indexing and similarity searches via FAISS, along with multiple embedding models for data processing and chunking, our approach efficiently captures and responds to customer needs. This real-time processing of external information allows the system to tailor responses to the query context, thereby improving response accuracy and adaptability. Validation on a real bank customer service dataset shows that this implementation outperforms existing techniques, enhancing response speed and quality, thereby boosting customer satisfaction and contributing to fintech innovation.

Keywords: Retrieval Augmented Generation, LangChain Framework, Customer Satisfaction Enhancement, Banking Online.

1 Introduction

With the rapid development of financial technology, banking services are gradually shifting to online.[1], and customers' demands for banking services are becoming increasingly diversified. In order to improve customer satisfaction and reduce labour costs, natural language processing technology plays an increasingly important role in bank customer service. The current situation and challenges of bank customer service With the rapid development of financial technology, banking business has experienced a shift from traditional counter services to online intelligent services. This shift has not only broadened the channels for customer service, but also greatly increased customer expectations for service quality and responsiveness. Natural Language Processing (NLP)[2] technology, in this context, has become a key driver of

¹ Corresponding Author

customer service innovation in banking. However, despite significant advances, current bank customer service systems still face challenges in understanding complex customer needs and providing personalised solutions, especially in terms of data privacy protection and the level of intelligence of the dialogue system.

In banking, customers often pose vague questions that mask their true intentions. Recognizing this, we propose a semantic-based approach to better discern and address these underlying inquiries. For instance, while a customer might simply ask, "Why was my account debited?" their real concern could involve specific transactions or account activities. This shift reflects a movement from static to dynamic customer demands.[3]Moreover, distinguishing between public domain knowledge (information widely available or easily accessible) and private domain knowledge (specific to an individual or organization) presents a challenge. For instance, a customer's question about their account details might require bank staff to access sensitive information like transaction history or account balances, which necessitates careful handling to protect privacy. Thus, in addition to automated systems, bank employees may need to consult experts to effectively respond to customer queries.

The swift progress of session- and chat-based language models has markedly propelled advancements in artificial intelligence, particularly in large language models (LLMs). [4]These models benefit from extensive pre-training on vast datasets and enhance their capabilities through reinforcement learning from human feedback. Despite their potential, the application of LLMs in the financial sector is limited due to a lack of domain-specific knowledge[5] and challenges such as generating misleading information and struggling with complex logical reasoning. Moreover, issues like computational cost and lack of transparency in model workings further complicate their use in finance.

This study introduces a novel Retrieval-Augmented Generation (RAG) [6]approach tailored for financial customer service, aiming to enhance service quality and efficiency. By merging retrieval and generative techniques, this method swiftly and accurately pinpoints customer needs. Utilizing the LangChain's JSON loader[7] for embedding expert-curated, specialized banking datasets, and the text_splitter library for data processing and chunking, our approach effectively indexes banking data, ensuring high relevance and minimal noise.[8] Integration of FAISS for vector indexing and similarity searches with various embedding models like piccolo, text2vec, and BGE substantially refines the accuracy of retrieval processes. This methodology not only mitigates the limitations posed by training data but also adeptly handles external information in real-time, adapting responses to specific queries' contexts and enhancing its applicability across scenarios. Our research leverages the benefits of the retrieval-enhanced generation method to significantly uplift the intelligence of bank customer services and contributes positively to the fintech sector's growth.

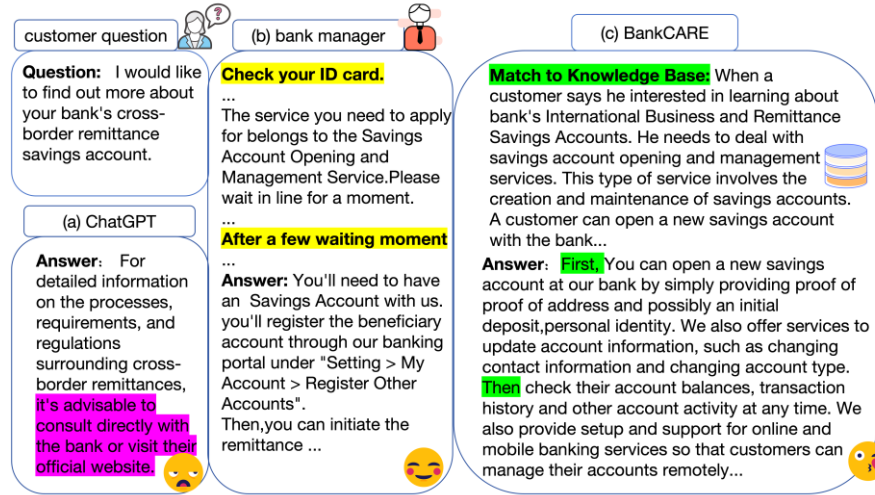


Image 1 Comparison of three methods in responses to a customer's query about bank services.

(a) ChatGPT, (b) bank manager, and (c) BankCARE shows a different approach to answering the question but the help is limited. ChatGPT provides a direct suggestion, and the bank manager section details steps to follow Professional training is taking a lot of time, whereas BankCARE which matches the query to a knowledge base has a comprehensive and helpful response.

In this work, we show that :

Specialization in Bank Customer Question Tasks: BankCARE is tailored specifically to understand and respond to inquiries related to banking customer services. This specialization ensures that the model is finely tuned to address the unique terminology language and context often encountered in banking-related queries, thereby enhancing the accuracy and relevance of its responses in this domain.

Enhanced Response Speed and Quality: Leveraging advancements in vectorization techniques and similarity search algorithms, BankCARE improves both the speed and quality of its responses. By efficiently processing and retrieving relevant information from its knowledge base, BankCARE can swiftly provide accurate and insightful answers to customer questions, thereby enhancing user satisfaction and minimizing the occurrence of illusionary or irrelevant responses.

2 Relational Work

2.1 The problem of the large language model

Google introduced the pre-trained language model BERT[9] in 2018, marking a significant advancement in natural language processing (NLP). BERT excels in various NLP tasks, propelling a surge in research based on pre-trained models and solidifying the pre-training paradigm in NLP. Although this shift greatly impacted the industry, it didn't fundamentally change how models address specific problems. In 2020,

OpenAI released GPT-3[10], which performed impressively in text generation and NLP tasks.

The launch of ChatGPT[11] in November 2022 was particularly noteworthy, showcasing the vast potential of large language models. ChatGPT excels in understanding user needs and providing tailored responses. Its versatility extends beyond everyday communication to executing complex tasks like article writing and question answering. Remarkably, ChatGPT often surpasses task-specific algorithms across various tasks, using a single model. This capability is not only a milestone in AI but also significantly influences NLP research.

However, large Language Models may never completely eliminate the "illusion", as Sam Altman puts it: "Illusion and creativity are equivalent". This concept has been understood for a long time, just like biological evolution[12], where only unstable randomness can lead to diversity, and diversity allows species to evolve.

In terms of technical principles, the illusion problem stems from a probabilistic selection at the output of a large model. The big model will form a probability table during training and choose the probability of the next token based on the input's antecedent, which carries a certain amount of randomness. This means that the model does not always choose the token that is ranked first in the probability table, but rather chooses randomly among a fraction of the higher probabilities. However, this randomness can also lead the model to generate inaccurate or generic answers.[13] In addition to this, there are some other reasons for large language models to cause phantom problems, such as overconfidence, lack of reasoning ability, insufficient world knowledge, and anthropomorphism. Together, these factors affect the quality and veracity of the output of large language models and require further research and improvement to address.

The success of ChatGPT has encouraged more developers to use OpenAI's proprietary models or APIs[14] to develop applications based on large language models. Nonetheless, the invocation of the Big Language Model still requires a lot of custom development work, including API integration, interaction logic, and data storage. In order to rapidly create end-to-end applications or processes based on the Big Language Model, a number of open source projects have been launched by organisations and individuals since 2022, most notably the LangChain framework[15]}, an open source framework designed to simplify the development of Big Language Model applications by providing a common interface to a wide range of Big Language Model applications. It enables language models to connect with other data sources and allows language models to interact with the environment.

And on top of all that, Lewis introduced the RAG in 2020. Particularly, the RAG comprises of an initial retrieval stage[17] in which the LLM searches external data sources for pertinent information prior to providing text or answering inquiries.[18] This procedure guarantees that the responses are founded on recovered evidence, which significantly enhances the output's correctness and relevancy. It also provides information for the process during the next generation phase. The RAG addresses the "illusion"[19] of generation by the dynamic retrieval of data from the knowledge base during the inference phase.

2.2 Benefits and challenges of Large Language Models for the Financial Services Industry

Large Language Models (LLMs) have evolved from mere auxiliary tools to central technologies in the financial services sector. Initially focused on tasks like text categorization and customer intent recognition, LLM applications now encompass sentiment analysis and intricate transaction monitoring. LLMs, leveraging massive datasets, can generate actionable insights for investors making critical stock decisions without needing code adjustments once they're trained to identify patterns and predict trends.[21]

In a financial landscape marked by rapid changes and competition from fintech and peer-to-peer firms, technology emerges as a crucial differentiator. It enables forward-thinking institutions to introduce new services, optimize existing ones, and enhance customer acquisition and retention strategies. To capitalize fully on LLMs, financial institutions must be at the forefront of digital innovation, utilizing these models for complex financial calculations and analysis of unstructured data, thus enhancing decision-making processes.[22]The efficiency of LLMs in processing vast volumes of data can significantly improve without human intervention, aiding analysts in making swift, informed decisions. This capability is especially vital in the competitive banking sector, where quick and flexible computing power is essential. Investments in high-performance cloud technologies and robust processing resources are therefore crucial, though there remains scope for improvement in processing speed, cost, and quality.[23]

While the deployment of LLMs has streamlined operations and introduced new financial products, it has also increased the demand for high-quality data annotations, highlighting a limitation in their real-world effectiveness. Furthermore, the financial language's complexity poses challenges to model accuracy. Nevertheless, the automation of routine tasks by LLMs enhances productivity and customer satisfaction, supporting organizations in a dynamic, collaborative, and competitive environment. Moreover, with growing regulatory pressures, LLMs can automate data collection for compliance processes, improving the accuracy and speed of decision-making. This not only helps organizations meet their compliance obligations but also reduces costs associated with fines and litigation while maintaining brand integrity. Thus, continued investment in new technologies is essential for financial institutions to adapt to evolving regulations and harness the potential of AI growth.

3 Method

3.1 Overview

This paper proposes a model based on Retrieval Augmented Generation (RAG)[24], which includes a retrieval module and a generation module. This research employs a langchain-based Retrieval Augmented Generation (RAG) framework specifically designed for understanding and generating responses to bank customer enquiries. RAG is responsible for generating accurate, context-aware responses by combining a large language model with external knowledge retrieval. LangChain facilitates the

knowledge base management, text processing, [26] and model integration by providing the basic components for the RAG implementation. It allows seamless integration of large language models with external data sources, thus enhancing the generation of context-aware responses to customer enquiries.

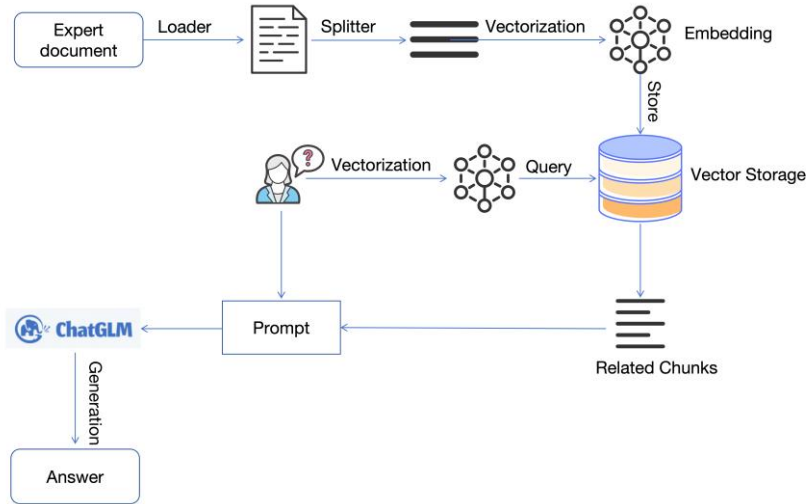


Image 2 The model architecture of BankCARE.

3.2 Data Pre-processing

The data is the raw material of LLM, which can be in various forms (including non-altered and structural data). In this study, the JSON format prepared by the banking industry, including customer inquiries and related knowledge bases (such as service details, service flow). To cope with the challenges brought by the lengthy text, we use the Langchain's JSON loader to load. Once the document is successfully loaded, parsed and converted into text, the document split process begins. The core activity of this stage involves decomposing this article into a manageable block. This process is also called text splitting or block. This becomes crucial when dealing with a large number of documents. Considering the context limitations of many LLM (for example, GPT-3.5 approximately 2048 [26]), Given the potential size of the document, the division of text becomes essential. The selected text splitting method depends to a large extent on the unique nature and requirements of the data. In this study, we use Langchain's text splitter [27] library to effectively index the content in the vector database to minimize the noise in the data to effectively retrieve during the rag process. In addition, effective block strategies are essential to ensure the semantic integrity of the document. By subdividing JSON documents into semantic independence blocks, we can promote more accurate semantic search and accurately match with bank user needs query.

Therefore, in this study, we chose a fixed size method. We only need to determine the number of tokens in the block, and whether there should be any overlap between them. We keep some overlap between blocks to ensure that the semantic environ-

ment between blocks will not be lost. Compared with other forms of blocks, this decomposition method is more economical and easy to use in computing, which helps reduce inaccurate search results due to improper size.

3.3 Embedding Models and Vector Storage

Vectorisation is a process of transforming textual data into vector matrices, which can have a direct impact on subsequent retrieval. The document is divided into parts that make sense on a semantic level and then it is converted into a space of vectors. The resultant embeddings are then kept in a vector store. In addition to handling some aspects of store and vector management, vector stores are special search databases created to facilitate vector search. Intrinsicly, a database that enables quick searches for comparable vectors is called a vector store. An effective vector store or index to keep the changed document chunks and their corresponding IDs is necessary for an RAG model to execute productively. The amount of the data and the available processing power are two factors that influence the choice of vector storage. Several prominent vector stores include FAISS: FAISS is a library created by Facebook AI that is well-known for maintaining enormous sets of high-dimensional vectors in an efficient manner, as well as for carrying out similarity searches and grouping in a highly dimensional environment. It employs progressive Pinecone designed to optimise memory usage and query duration: next, various embedding models are utilised to convert textual data into vector form and store it in a vector database. models such as piccolo, text2vec, and BGE are utilised based on their applicability to banking including proprietary terminology and optimisation requirements. These models strike a balance between out-of-the-box usability and customisation, which is essential to improve the accuracy of our retrieval system.[28]

Following the document splitting process, text blocks are transformed into vector representations so that semantic similarity may be quickly assessed. Every block has been encoded by this "embedding" so that similar blocks are clustered together in vector space. Modern artificial intelligence models are not complete without vector embeddings. They entail transferring data from intricate, unstructured formats, such as text or images, which usually has low dimensions. This vector space allows for productive computation and, critically, significant aspects of the raw data are captured by the spatial connections in this space. For example, Embeddings are used to capture semantic information in text data. Texts convey analogous meanings, even with a different wording, map to closures in the embedding space.

Visualization of text embeddings helps intuitively understand semantic relationships. These embeddings, represented in 2D or 3D spaces, cluster similar words or phrases together, showcasing their semantic closeness. Initially, models like Word2Vec and GloVe[29] advanced semantic understanding by analyzing term co-occurrences in large text corpora. However, the field has evolved to transformer-based models like BERT, RoBERTa[30], ELECTRA[31], T5[32], and GPT. These newer models consider entire sentence contexts, enabling richer semantic information gathering and better ambiguity resolution. The quality of these context-sensitive embeddings is crucial for effectively finding semantically relevant documents and generating accurate, context-appropriate responses.

3.4 Retrieval By Similarity Search

At the core of the RAG framework, we utilise FAISS (Facebook Artificial Intelligence Similarity Search)[34] because of its unparalleled efficiency in vector indexing and similarity search. The advanced algorithms of FAISS quickly retrieve the IDs of the most relevant documents from our banking knowledge base, which helps to narrow the search space by recognizing sections or chunks that may include related information. Upon receiving a user query, the system converts the input into a vector representation as in the indexing phase. It then calculates the similarity score between the query vector and the vectorised blocks in the indexed corpus. The system prioritises the retrieval of the top K blocks that are most similar to the query. The search latency is greatly reduced while maintaining high accuracy. This integration ensures that our RAG system is able to handle the complex demands of bank customer queries, providing accurate, contextually relevant information by quickly identifying and generating responses using the closest vector representation.

"Similarity Search"[35] identifies documents that are similar to the query based on cosine similarity. Vector stores support another type of search, Maximum Marginal Relevance (MMR)[36], which ensures that documents are retrieved that are not only pertinent to the question, but are also various and qualitative, thus removing redundant information and enhancing the variety of retrieved results. The Similarity Search method, on the other hand, simply takes semantic similarity into account. A similarity score threshold search strategy was also employed in this investigation. Only papers with scores higher than 3 are returned by this procedure, which sets a similarity score threshold of 3. When searching for similar documents, the "k" parameter is usually used to specify the first "k" documents to be retrieved.

3.5 Optimisation and Evaluation of Answer Generation

In the final stage in the BankCARE system, the quality of the generator usually determines the quality of the final output. It is responsible for converting the retrieved information into a coherent and fluent text. Unlike traditional language models, the generator improves accuracy and relevance by integrating the retrieved data. In RAG, when generating text, the LLM can query this database to retrieve relevant information based on the context of the prompt. This retrieved information then serves as additional input to the LLM, guiding it to produce more accurate and consistent output. This comprehensive input allows the generator to deeply understand the context of the query and thus produce more informative and contextually relevant answers.[37] In addition, the generator is guided by the retrieved text to ensure that the generated content is consistent with the acquired information. The user query and the selected document are synthesised into a coherent prompt for a large language model to make a response. The way in which the model responds may vary according to task-specific criteria, either allowing it to exploit its inherent knowledge of the parameters or restricting its response to the document information provided. In an ongoing dialogue, any existing dialogue

history can be integrated into the prompt, allowing the model to effectively engage in multiple rounds of dialogue interaction.

4 Experiment

In order to gain a deeper understanding of the impact of the various components in the methodology, we conducted an ablation study in terms of model, embedding, and prompt. The evaluation was carried out on the same dataset, including the actual customer requirement documents of the bank prepared by the bank experts and the test set. For model, we chose chatgpt-3.5, chatglm3, and qwen to retrieve on this dataset for generating and reporting measurements of the average similarity of the final results to the answers; For embedding, we used word2vec, bge, piccolo’s embedding model respectively, and used the same dataset for embedding to measure the text similarity of the retrieved results. For prompt, we used one-sho[39], step-by-step[38], least-to-most to measure the average similarity between the final result and the answer by using different prompts for the retrieved results. This ablation study, with fewer iterations than the main experiment, aimed to control variables within a constrained budget.

4.1 Ablation Study of Embedding

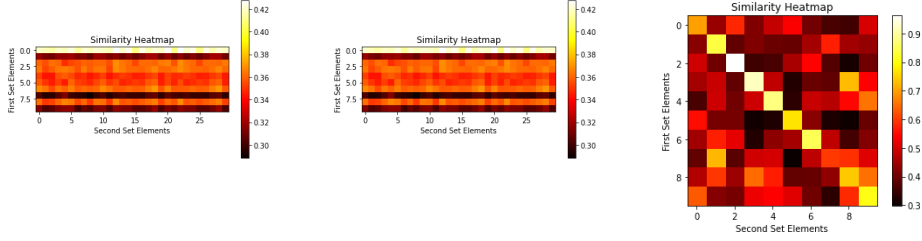
In the experiment, FAISS-text2vec, with a context size of 768, achieved a similarity score of 0.4465, suggesting its robust performance despite a smaller context size. FAISS-bge’s large context size of 8192 correlated with a lower similarity score of 0.2275, indicating room for optimization. Meanwhile, FAISS-piccolo offered a balance with a context size of 1024 and a score of 0.3722. This underscores the importance of context size in embedding models and its influence on retrieval quality within the RAG framework.

Model Name	Context Size	Release Data	Similarity score
Faiss-text2vec	768	20/09/2023	0.4465
Faiss-bge	8192	29/01/2024	0.2275
Faiss-piccolo	1024	24/10/2022	0.3722

Table 1 The model architecture of BankCARE.

4.2 Benefits and challenges of Large Language Models for the Financial Services Industry

In our second ablation study, heatmaps provide a visual analysis of outcomes from three distinct chatgpt-3.5, qwen, and chatglm3—employed to answer queries using a singular embedding model and vector database. Despite identical retrieval inputs, the resulting generation from each LLM varied, reflecting their unique generative capabilities. These differences were quantitatively measured against expert-crafted answers, offering a rigorous assessment of each model’s effectiveness. Notably, graph underscore the importance of llm for generating answer.

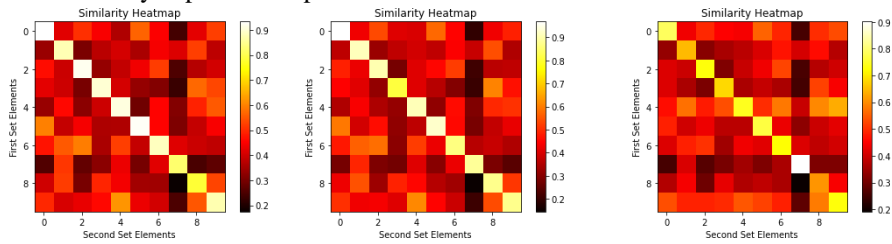


4.3 Benefits and challenges of Large Language Models for the Financial Services Industry

For further analysis, the ablation study of Prompt is conducted to investigate the role of prompt for BankCARE’s quality of response. Specifically, for the first test, select one-shot strategies while retaining the other processes[41], by providing a single comprehensive example, encourage the model to generalize from that instance to produce an answer in a similar style or structure

And for the second test, after obtaining the learned relational chunks from vector storage, Step-by-step prompts guide the language model through an iterative process to enhance the quality of the generated content by focusing on one part of the problem at a time. As for the third test, we utilized least-to-most prompts (LTM) [42] starting with the simplest components of a problem which is to break down the steps of the problem, and gradually building up to the useful for smaller tasks layered understanding.

These three tests are performed on all the benchmark datasets, and the comparison results in terms of average similarity score and hot graph with answer documents written by experts are reported.



5 Conclusions

This study applies a Retrieval-Augmented Generation (RAG) approach, utilizing a LangChain-based framework integrated with FAISS for vector indexing and

similarity searches, to meet banking customer service needs. This implementation enhances the efficiency and quality of responses to customer queries, showing significant improvements in speed and accuracy over existing technologies. Such enhancements boost customer satisfaction and contribute to the advancement of financial technology, demonstrating the effective adaptation of RAG in banking services. Nevertheless, the research encounters limitations due to a constrained dataset size and the high complexity of the model, which may impact the generalizability and scalability of the approach. Future work should aim to expand the dataset and optimize the model structure to improve its banking applicability. It will also be important to address inherent challenges related to the RAG approach, including its reliance on specific embedding and retrieval algorithms and the opaque nature of large models.

References

- [1] Asai, A., Wu, Z., Wang, Y., Sil, A. and Hannaneh Hajishirzi (2023). Self-rag: Learning to retrieve, generate, and critique through self-reflection.
- [2] Barnett, S., Kurniawan, S., Srikanth Thudumu, Zach Brannelly and Abdelrazek, M. (2024). Seven failure points when engineering a retrieval augmented generation system.
- [3] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C. and Hesse, C. (2020). Language models are few-shot learners. In: H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan and H. Lin, eds. [online] Curran Associates, Inc., pp.1877–1901. Available at: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf.
- [4] Bujard, H., Gentz, R., Lanzer, M., Stueber, D., Mueller, M., Ibrahim, I., Haeuptle, M.-T. and Dobberstein, B. (1987). [26] A T5 promoter-based transcription-translation system for the analysis of proteins in vitro and in vivo. Elsevier, pp.416–433.
- [5] Chase, H. (2022a). LangChain. [online] Available at: <https://github.com/langchain-ai/langchain>.
- [6] Chase, H. (2022b). LangChain. [online] Available at: <https://github.com/langchain-ai/langchain>.
- [7] Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D. and Liu, Z. (2023). BGE m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation.
- [8] Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., Stoica, I. and Xing, E.P. (2023). Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality. [online] Available at: <https://lmsys.org/blog/2023-03-30-vicuna/>.
- [9] Church, K.W. (2017). Word2Vec. *Natural Language Engineering*, 23, pp.155–162.
- [10] Clark, K., Luong, M.-T., Le, Q.V. and Manning, C.D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555.
- [11] Corley, C.D. and Mihalcea, R. (2005). Measuring the semantic similarity of texts. pp.13–18.

- [12] Dipietro, L., Sabatini, A.M. and Dario, P. (2008). A survey of glove-based systems and their applications. *Ieee transactions on systems, man, and cybernetics, part c (applications and reviews)*, 38, pp.461–482.
- [13] Florin Cuconasu, Trappolini, G., Siciliano, F., Filice, S., Cesare Campagnano, Yoelle Maarek, Tonello, N. and Silvestri, F. (2024). The power of noise: Redefining retrieval for RAG systems.
- [14] Gautier Izacard, Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S. and Grave, E. (2023). Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, [online] 24, pp.1–43. Available at: <http://jmlr.org/papers/v24/23-0037.html>.
- [15] Gregory, P.A., Bert, A.G., Paterson, E.L., Barry, S.C., Tsykin, A., Farshid, G., Vadas, M.A., Khew-Goodall, Y. and Goodall, G.J. (2008). The miR-200 family and miR-205 regulate epithelial to mesenchymal transition by targeting ZEB1 and SIP1. *Nature cell biology*, 10, pp.593–601.
- [16] Hagendorff, T., Fabi, S. and Kosinski, M. (2023). Human-like intuitive behavior and reasoning biases emerged in large language models but disappeared in ChatGPT. *Nature Computational Science*, 3, pp.833–838.
- [17] He, Z., Zhong, Z., Cai, T., Lee, J.D. and He, D. (2023). REST: Retrieval-based speculative decoding.
- [18] Huang, A.H., Wang, H. and Yang, Y. (2023). FinBERT: A large language model for extracting information from financial text. *Contemporary Accounting Research*, 40, pp.806–841.
- [19] Jeong, C. (2023). Generative AI service implementation using LLM application architecture: based on RAG model and LangChain framework. *Journal of Intelligence and Information Systems*, 29, pp.129–164.
- [20] Jiang, H., Wu, Q., Lin, C.-Y., Yang, Y. and Qiu, L. (2023). LLMingua: Compressing prompts for accelerated inference of large language models. In: H. Bouamor, J. Pino and K. Bali, eds. [online] Association for Computational Linguistics, pp.13358–13376. doi:<https://doi.org/10.18653/v1/2023.emnlp-main.825>.
- [21] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D. and Yih, W. (2020). Dense passage retrieval for open-domain question answering. In: B. Webber, T. Cohn, Y. He and Y. Liu, eds. [online] Association for Computational Linguistics, pp.6769–6781. doi:<https://doi.org/10.18653/v1/2020.em%20nlp-main.550>.
- [22] Komeili, M., Shuster, K. and Weston, J. (2021). Internet-augmented dialogue generation. arXiv preprint arXiv:2107.07566.
- [23] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S. and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In: H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan and H. Lin, eds. [online] Curran Associates, Inc., pp.9459–9474. Available at: https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.
- [24] Li, J., Chen, X., Hovy, E. and Jurafsky, D. (2015). Visualizing and understanding neural models in NLP. arXiv preprint arXiv:1506.01066.
- [25] Li, M., Song, F., Yu, B., Yu, H., Li, Z., Huang, F. and Li, Y. (2023). Api-bank: A benchmark for tool-augmented llms. arXiv preprint arXiv:2304.08244.
- [26] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

- [27] Liu, Z., Huang, D., Huang, K., Li, Z. and Zhao, J. (2021). Finbert: A pre-trained financial language representation model for financial text mining. pp.4513–4519.
- [28] Lohse, D., Milner, S., Fetters, L., Xenidou, M., Hadjichristidis, N., Mendelson, R., Garcia-Franco, C. and Lyon, M. (2002). Well-defined, model long chain branched polyethylene. 2. Melt rheological behavior. *Macromolecules*, 35, pp.3066–3075.
- [29] Matthijs Douze, Alexandr Guzhva, Deng, C., Johnson, J., Gergely Szilvasy, Pierre-Emmanuel Mazaré, Lomeli, M., Hosseini, L. and Hervé Jégou (2024). The faiss library.
- [30] Roumeliotis, K.I. and Tselikas, Nikolaos D (2023). Chatgpt and open-ai models: A preliminary review. *Future Internet*, 15, p.192.
- [31] Shu, Y., Yu, Z., Li, Y., Karlsson, B.F., Ma, T., Qu, Y. and Lin, C.-Y. (2022). TIARA: Multi-grained retrieval for robust question answering over large knowledge bases.
- [32] Wang, Y., Yao, Q., Kwok, J. and Ni, L.M. (2020). Generalizing from a few examples: A survey on few-shot learning.
- [33] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. and Zhou, D. (2023). Chain-of-thought prompting elicits reasoning in large language models.
- [34] Wu, S., Irsoy, O., Lu, S., Dabrovolski, V., Dredze, M., Gehrmann, S., Kambadur, P., Rosenberg, D. and Mann, G. (2023a). Bloomberggpt: A large language model for finance. arXiv preprint arXiv:2303.17564.
- [35] Wu, S., Irsoy, O., Lu, S., Dabrovolski, V., Dredze, M., Gehrmann, S., Kambadur, P., Rosenberg, D. and Mann, G. (2023b). Bloomberggpt: A large language model for finance. arXiv preprint arXiv:2303.17564.
- [36] Wu, T., He, S., Liu, J., Sun, S., Liu, K., Han, Q.-L. and Tang, Y. (2023c). A brief overview of ChatGPT: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica*, 10, pp.1122–1136.
- [37] Xiao, S., Liu, Z., Zhang, P. and Niklas Muennighoff (2023). C-pack: Packaged resources to advance general chinese embedding.
- [38] Yang, Y., Christopher, M. and Huang, A. (2020). Finbert: A pretrained language model for financial communications. arXiv preprint arXiv:2006.08097.
- [39] Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T.L., Cao, Y. and Narasimhan, K. (2023). Tree of thoughts: Deliberate problem solving with large language models.
- [40] Ye, J., Chen, X., Xu, N., Zu, C., Shao, Z., Liu, S., Cui, Y., Zhou, Z., Gong, C. and Shen, Y. (2023). A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. arXiv preprint arXiv:2303.10420.
- [41] Ye, X., Yavuz, S., Hashimoto, K., Zhou, Y. and Xiong, C. (2022). RNG-KBQA: Generation augmented iterative ranking for knowledge base question answering. In: S. Muresan, P. Nakov and A. Villavicencio, eds. [online] Association for Computational Linguistics, pp.6032–6043. doi:<https://doi.org/10.18653/v1/2022.acl-long.417>.
- [42] Zhang, B., Yang, H. and Liu, X.-Y. (2023). Instruct-fingpt: Financial sentiment analysis by instruction tuning of general-purpose large language models.
- [43] Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O. and Le, Q. (2022). Least-to-most prompting enables complex reasoning in large language models. arXiv preprint arXiv:2205.10625.