Computer Standards & Interfaces 94 (2025) 103995

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Application of retrieval-augmented generation for interactive industrial knowledge management via a large language model

Lun-Chi Chen [a], Mayuresh Sunil Pardeshi [b], Yi-Xiang Liao [c], Kai-Chih Pai [a,*]

[a] College of Engineering, Tunghai University, Taichung, Taiwan
[b] School of Computer Sciences, UPES, Dehradun, India
[c] AI Center, Tunghai University, Taichung, Taiwan

ARTICLE INFO

Keywords:
Large language models
Retrieval-augmented generation
Interactive industrial knowledge management system

ABSTRACT

Industrial data processing and retrieval are necessary for adoption in Industry 5.0. Large Language Model (LLMs) revolutionize natural language process (NLP) but face challenges in domain-specific applications due to specialized terminology and context. Artificial Intelligence (AI) assistants for industrial-related work enquiry and customer support services are necessary for increasing demand and quality of service (QoS). Our research aims to design a novel customized model with a retrieval-augmented generation (RAG)-based LLM as a sustainable solution for industrial integration with AI. The goal is to provide an interactive industrial knowledge management (IIKM) system that can be applied to technical services: assisting technicians in the search for precise technical repair details and company internal regulation searches: personnel can easily inquire about regulations, such as business trips and leave requirements. The IIKM model architecture consists of BM25 and embedding sequence processing in the chroma database, where the top k-chunks are selected by the BAAI ranker to respond effectively to the queries. A group of documents of 234 MB size and pdf, pptx, docx, csv and txt formats are used for the experimental analysis. The designed interactive knowledge management system has a mean reciprocal rank (MRR) of 88 %, a recall of 85 % and a mean average precision (mAP) of 75 % in technical service. The internal regulatory documents have a generation-based retrieval evaluation prediction of recall of 91.62 %, MRR of 97.97 % and mAP of 91.12 %. We conclude with insights gained and experiences shared from IIKM deployment with Sakura incorporation, highlighting the importance of the hybrid approach integrating RAG-based generative pretrained transformer (GPT) models for customized solutions.

1. Introduction

An industrial document management system is an important asset for industry 5.0 [1,2]. There are numerous challenges in managing the domain-specific nonrecorded-based database. Relational databases are quite popular and easy to evaluate for statistical results. However, document-based data require sequence/sentence-based data processing for vector/keyword chunks, which can be evaluated as a respective response on the basis of its ranking. Owing to the privacy of industrial data as assets, limited privileges are given to customer-based queries for service center-based response. This interactive knowledge management approach is purposefully designed to reduce the discussion time of industrial employees with customers by directly responding through a chat-based service system. The documents updated within the system database can be used efficiently to answer queries with updated responses. Most customer-based queries are for maintenance of the product, part replacement, process guidance, complaint about device faults, etc. Nevertheless, this model can be beneficial to many industrial systems that consider product management and maintenance for good service and marketing strategies. The time saved by employees responding to redundant queries and processes benefits both parties.

Recent advances in OpenAI models are popularly known as the ChatGPT application [3,4,5], which produces human-like texts as a response to user queries. The ChatGPT is trained on a vast collection of databases and can also fix errors by itself by generating code and algorithms. Deductive and abrupt reasoning are proficient in the ChatGPT. Furthermore, the accuracy can be improved by the use of private knowledge collection for specific applications. The GPT-3.5 turbo performance surpasses that of LLM zero-shot learning with applications such as conversation, translation, programming, training, etc. The RAG

* Corresponding author.
E-mail address: kcpai@thu.edu.tw (K.-C. Pai).

https://doi.org/10.1016/j.csi.2025.103995
Received 30 September 2024; Received in revised form 23 December 2024; Accepted 1 March 2025
Available online 6 March 2025

helps to overcome the LLM challenges of transparency, knowledge updates and hallucinations. It is used to add domain-specific knowledge for the application, which increases its importance [6,7]. Thus, a non-parameterized knowledge database is added for a semi parametrized approach model with a parameterized model. The RAG algorithm is popularly known for combining pretrained seq2seq models with pretrained retrievers and fine tuning for knowledge capture to be used by generators later. Enhancing knowledge management usage via intelligent AI algorithms for interacting with thousands of users simultaneously reduces the workload in the organization. Therefore, the IIKM helps to resolve the interactive agent problem in the industrial environment for high service availability. In practice, providing high-quality service to industrial clients is a challenge [8,9]. Various limitations faced during service availability include delays in issuing responses, handling error messages, unavailability of the resource person, loss of request, etc. In real time, it is difficult to handle many services center calls at peak hours with limited human service providers. The motivation for the IIKM system is as follows: how can quality service be provided to industrial customers in terms of product manuals, product details, queries, instant complaints and reducing human workload? In 2020, N. Vo, S. Liu, X. Li, and G. Xu conducted research on leveraging unstructured calls via logcall-by-log data for churn prediction [10], which presented a churn risk prediction algorithm for customer segments and personality traits to develop retention strategies.

However, the passive approach does not help reduce the service center load or provide active services. Therefore, a new model is necessary to address the limited human availability of services. The inclusion of the Chinese language for Chinese document processing and conversation allows the model to be customized. The IIKM system consists of intelligent interactions with multiple customers as LLM-based services. Recently, LLM has become popular for AI assistant-based services. Using the RAG with Best Matching 25 (BM25), a ranking function commonly used in information retrieval systems to evaluate the relevance of documents based on search queries, in conjunction with embedding for LLM, offers a distinct approach for industrial services. The model is further improved by the BAAI/bge-reranker-base (BAAI) ranker to obtain top responses as well as meaningful context. The BAAI model has been trained in both English and Chinese languages, with some of the issues in the dataset being written in Chinese. Successively, the GPT turbo then constructs human-friendly responses that are easy to understand and provides details of the required process. Ultimately, high-quality responses are provided to multiple clients by the GPU-based processing system. In IIKM, the research gap is related to the industrial multiclient friendly interaction by the AI assistant by using LLM and parallel processing, which has rarely been applied until now. The IIKM system model is exclusively designed to provide AI assistant-based interactive services for multiple product manuals, maintenance, errors and related details. The IIKM system is rational because it uses a multicient-based customized RAG GPT approach to solve the interactive knowledge system issue. The experimental section presents highly accurate results that provide practical deployment in a real-world environment. The recent interactive models are compared in Table 1.

Considering prior references for interactive knowledge system applications [8,9], the results are insufficient to solve the scalability for client services and achieve high recall. Several methods for embedding, machine learning, reinforcement learning, and data mining are not sufficient independently to provide significant improvements. Therefore, an IIKM system with a custom RAG GPT architecture and data mining algorithms provides higher recall.

The key contributions of this study are outlined as follows:

(1). **To design a document knowledge retrieval system for domain-specific data via the RAG and LLM.**

Industry 5.0 integrates the work of humans in cooperation with automation and efficiency. Therefore, an intelligent system is necessary to respond as an assistant by summarizing multiple reports in real time. The work efficiency is highly improved, and human error is reduced.

(2). **Modeling an effective approach for selecting high-score chunks.**

The RAG architecture includes a generator module that can answer the user queries with reliability. The retrieval-augmented module subsequently helps update the server storage data. Therefore, an updated response is obtained every time a query is submitted.

(3). **Rank and respond to complex queries for product management.**

A single query can be related to multiple sentences, paragraphs or documents. Thus, an effective ranking algorithm is required to provide high relevance to the query response. The BAAI ranker acts as a cross-encoder that can rerank the top-k documents to deliver prominent responses.

(4). **To evaluate the design system model in detail, practical experience and real-world usage are incorporated.**

The current model is designed on the basis of the requirements of the Sakura Incorporation. The model functions to evaluate managerial queries for employee management as an AI assistant and as a customer service center to respond to product issues on the basis of knowledge management.

2. Literature survey

In Fig. 1, we review the various components related to knowledge management systems via LLM, including embedding, evaluation, datasets, rankers, and the RAG.

The dataset is the initial and important part of the knowledge

Table 1
Comparison of the Recent LLM based Applications.

Reference Paper	Domain Knowledge	Prompt Configuration	Prompt Engineering Process	Performance Parameters
Zhiling Zheng, et al. 2023 [11]	ChatGPT Chemistry Assistant	GPT-3.5 and GPT-4.	(a) Synthesis Conditions Summarization, (b) Synthesis Paragraph Classification, and (c) Text Embedding's for Search and Filtering.	Class weighted accuracy, precision, recall, and F1 score.
Herbold S., et al. 2023 [12]	Essay Writings in Education	Human vs. GPT-3 and GPT-4.	Language Skills and Confidence evaluation by two-sided Wilcoxon-rank-sum tests.	Arithmetic mean (M), standard deviation (SD), and Cronbach's α for the ratings.
Koziolek H., et al. 2023 [13]	Control Logic Code Generation from Images	GPT-4	Logic generation for all control loops, interlocks and sequences.	Use cases, manual syntax and plausibility checks.
Keiser F., et al. 2023 [14]	Quantitative reasoning and Force Concept Inventory (FCI) in Physics.	GPT-4	Probing conceptual understanding in physics, Simulating conceptual understanding of different cohorts and Simulating students' preconceptions with ChatGPT.	Statistics for Mean, Median, SD, Variance and Kruskal–Wallis test.
IIKM	IIKM as AI Assistant for Sakura Inc.	GPT-3.5 Turbo and GPT-4.	Keywords and Vectors Retrieval by TF-IDF/BM25/ Embedding, Augmentation with BAAI *Re*-Ranker and GPT based Generation and retrieval of the query response.	Mean Reciprocal Rank (MRR), Mean Average Precision (mAP), and Recall.

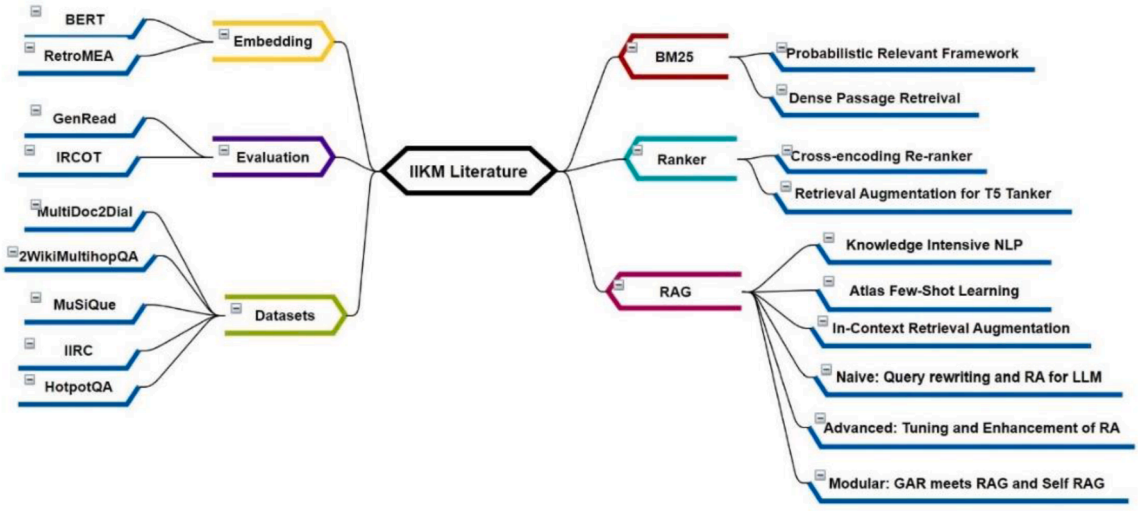

Fig. 1. Literature Analysis.

generation of the LLM model. Therefore, a complete library is recommended for topic-based information related to every detail of the subject considered. In the telecom field, complete records, reports and history operations are used to provide effective inputs via data parsing, cleaning, deduplication and extraction. This process forms the input to the LLM for processing and provides customer service and data search in Telecom by Cai H. and Wu S. [15].

BM25/F has served as a popular corporate and web search algorithm. The probabilistic relevance framework (PRF), as a conceptual model for functions of term weighting and document scoring, was presented by Robertson S and Zaragoza [16]. Open domain-based question-answering by dense-pass retrieval was demonstrated by Karpukhin V. et al. [17]. Dense representation is utilized with a dual-encoder framework used for learning from passages and question-answering for retrieval component design to index from low-dimensional and continuous space for top-k passages. The model is evaluated using the top-5, top-20 and top-100 outputs with different model comparisons and datasets.

Knowledge intensive-based task implementation with retrieval-augmented generation (RAG) was presented by Lewis P. et al. [18]. The RAG sequence generates top-k approximations from the seq2seq probability, retrieves the top-k documents and uses the maximum inner product search (MIPS), document index and generator for the output response. Reducing conversation hallucination via retrieval augmentation was demonstrated by Shuster K. et al. [19].

Chunking is essential in RAG, where breaking down large texts into smaller segments allows for more focused processing. Different chunking strategies cater to specific use cases, enhancing the model's ability to retrieve and generate relevant information effectively [20]. However, research on effective chunking techniques, especially for large input sizes, remains limited, and few studies have explicitly addressed the challenges associated with chunking in RAG systems. When selecting the best chunking strategy for a specific use case in RAG systems, it's crucial to consider several factors. These include the complexity of user queries (short and precise or long and complex), the nature of indexed content (long format like documents or short format like text messages), the intended use case for retrieved results (such as question answering, summarization, or semantic search), and the performance metrics of the model being used. By taking these factors into account, you can determine the most suitable chunking approach for your application [20].

This study presents multiple architectures consisting of rankers, neural retrievers, and encoder-decoders to improve conversation and knowledge. A benchmark comparison of different methods and datasets is well presented, with models trained in ParlAI. A language model with retrieval augmentation designed via few-shot learning (ATLAS) was presented by Izacard G. et al. [21]. The designed ATLAS model is a RAG-based pretrained language model with 50x fewer parameters but performs better than the standard large-parameter model. Retrieval augmented language models (RALMs) for in-context performance were demonstrated by Ram O. et al. [22]. The purpose of in-context RALM is to provide a retrieval method that uses a clean probe and has high bonding with custom LMs, which drastically improves its performance. A cross-encoding reranker with a prediction of a grounded span for dialog generation was presented by Li K. et al. [23]. This is an optimized pipeline for retrieving, reranking and generating, consisting of a passage dropout and regularization for improving model performance. The retrieval augmentation of the T5 reranker was demonstrated by Hui K. et al. [24]. Here, the performance improvement KL-divergence for the T5 reranker using in-domain and zero-shot out-of-domain methods is trained on Wikipedia and commercial web-search engines for results. Retrieval-oriented language pretraining was presented by Liu Z. et al. [25].

An evaluation of the mask autoencoder, asymmetric model structure and asymmetric masking ratios with multiple datasets and a model comparison are given for supervised and zero-shot evaluations. Augmentation in LLM by query retrieval was demonstrated by Ma X. et al. [26]. The naïve RAG model is designed to rewrite-retrieve-read concepts to improve the performance over previous models, where the rewriter is trained over previous feedback by reinforcement learning. It consists of data indexing, chunking, embedding/creating indices, retrieval and generation, as described by Gao Y. et al. [27].

A fine-tuning and open foundation for chat models was presented by Touvron H. et al. [28]. The advanced RAG is better in terms of retrieval generation, addressing indexing issues, fine-grained segmentation and metadata. Successively, alignment optimization, mixed retrieval, fine tuning/dynamic embedding, etc., can also be used. Iterative retrieval-generation (Iter-RetGen) for augmented LLM was demonstrated by Shao Z. et al. [29]. The Iter-RetGen algorithm has less overhead, improves performance, uses parametric/nonparametric question answering, and provides better self-asks for question answering. The modular RAG extends the functions of the search module, improves fine-tuning for retrieval and combines multiple modules.

Zero-shot-based information retrieval by the GAR meets the RAG criterion was presented by Arora D. et al. [30]. In this zero-shot setting, the multimodel architecture for GAR (generate additional context for query and retrieve) and RAG (retrieve relevant documents for context and generate answers) is iteratively improved to obtain a three-stage IR pipeline that improves recall and precision. Retrieval, generation and critique by learning (Self-RAG) was demonstrated by Asai A. et al. [31].

Algorithm
IIKM Algorithm.

Input: *Query*, $q_1 \in P$, prompt P
k, number of k chunks with the highest retrieval similarity scores.
n, number of n chunks with the highest rank similarity scores.
candidate Set ∈ D, Document Collections D.
Output: *Response*, IIKM prompt based statement.
Initialize Response, SimilarityScore =∅
While (Authentication Succeed):
BM25 (*Query, candidate Set, k*):
SimilarityScore = BM25 (Query, candidate Set) for each candidate.
Select Top k-chunks (SimilarityScore)
Return Top k-chunks
Embedding (*Query, candidate Set, k*):
SimilarityScore = Cosine Similarity (Query, candidate Set) for each candidate.
Select Top k-chunks (SimilarityScore)
Return Top k-chunks
If (User Authorized):
Top retrieval-chunks = BM25(*Query, candidate Set, k*) ∪ Embedding (*Query, candidate Set, k*)
Top n-chunk = BAAI Ranker (n, Top retrieval-chunks)
$Prompt_{Generation}$ = Prompt (Top-n chunks)
Response = GPT-3.5 Turbo ($Prompt_{Generation}$)
Return *Response*

Self-reflection and retrieval are improved for language model factuality and quality. Compared with the conventional RAG approach, on-demand passage retrieval via reflection tokens is feasible for diverse tasks. Language understanding via deep bidirectional transformer pre-training was presented by Devlin J. et al. [32]. An unlabeled text with a Mult conditional context is designed, and the model can be fine-tuned for several sentence- and task-level applications. LLM as context generators for knowledge-intensive tasks was demonstrated by Yu W. et al. [33]. The knowledge-intensive tasks are solved for multiple diverse and relevant contexts by clustering-based prompts and the generation of a read pipeline. Here, the experiments are shown with supervised and zero-shot settings. Knowledge-intensive multistep queries by interleaving with chain-of-thought (CoT) reasoning were presented by Trivedi H. et al. [34]. Interleaf retrieval with CoT improves recursively for better reasoning and reduces hallucinations in few-shot QA processing of multistep open-domain QA datasets.

The paper plan includes a methodology section in which the detailed operational model is presented, including the functioning, system model, flowchart, and algorithms with descriptions. Next, the experiment section will demonstrate different configuration-based results with a detailed description for evaluation. Ultimately, conclusions for the complete research and references are presented. The paper is presented as follows:

(1). The IIKM system provides a practically applicable system in industrial environments. Thus, applied research work is presented by this system.
(2). The customized RAG LLM architecture is designed to model and optimize the source domain database.
(3). An evaluation is provided in detail with model parameter settings and multiple independent experiments for industrial and customer usage.
(4). The discussion section discusses practical system deployment on industrial premises with the necessary transfer experience.

3. IIKM system architecture

3.1. RAG IIKM system model design

The input given to the IIKM system is for the industrial system operations process and service enquiry. The IIKM model consists of three main stages: retrieval, augmentation, and generation (RAG). RAG is used to provide relevant responses from high-level document collections by generating text or by question-answering. Therefore, the need for retraining of the complete model is eliminated, and a high-quality prediction is made. RAG methods are designed to be compatible with knowledge-intensive tasks. RAG features can be given as external knowledge association accuracy, provide the latest information, and are transparent, customizable, secured, trustworthy and scalable systems [35]. The details of each phase are presented in the following section.

Fig. 2 presents the detailed function of the RAG framework of the IIKM system model. For documents preprocess, we focused on extracting the textual content from the pptx, pdf, docx, csv and txt files. For the flow charts and architecture diagrams, we opted to represent them primarily as tables in the document, which allowed us to extract and present the information in a structured and clear way. The tables were converted into Markdown format for better readability and accessibility.

The input given by the user in the form of a query is compared with the database tokens and forwarded to the system for retrieval. In the retrieval phase, the proposed approach simultaneously searches vector features and retrieves keyword information from the metadata. The process of keyword extraction is based on the Jieba, a word segmentation library written in Python. Furthermore, a dictionary is established for proper names, with the objective of improving the accuracy of the relevant documents retrieved. BM25 performs query score calculations on the basis of the keywords from the documents while interacting with the Chroma database. Similarly, in parallel, the embedding provides vectors from the chroma database. The top k chunks from both the retrieval functions of BM25 and embedding are subsequently provided as inputs to the BAAI ranker [36]. The top n chunks are then stored in the database as the generation phase and utilized by GPT-3.5 to provide the top response to the user.

3.2. Retrieval-Augmented generation (RAG)

Retrieval: The purpose of retrieval is to retrieve the top-k relevant documents from the large knowledge base. We used TF-IDF, BM25, and embedding for the retrieval operation. The term frequency–inverse document frequency (TF-IDF) is used to process documents and measure their importance. The term t contained in document d is the relative frequency given by the term frequency tf(t,d).

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \epsilon d} f_{t',d}}$$

Here, $\sum_{t' \epsilon d} f_{t',d}$ is the total number of terms in document d.

The information provided by words is counted by the inverse document frequency (IDF), which is a logarithmically scaled document-to-word inverse function.

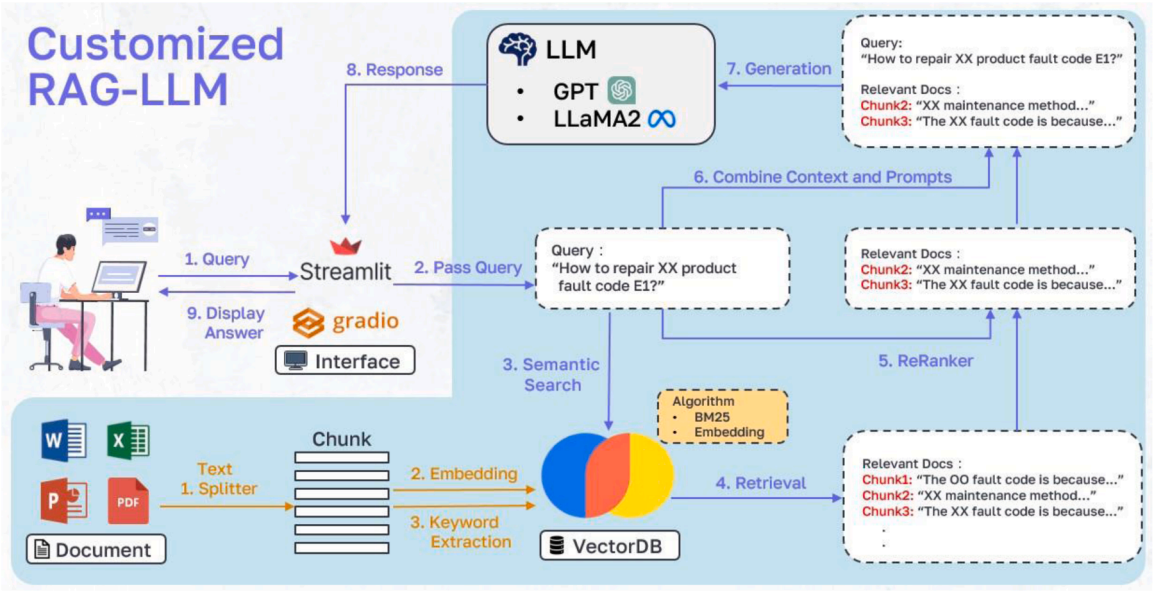

Fig. 2. IIKM System Model Design for the Sakura Corporation.

$$idf(t,d) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

where N = the total number of documents in the corpus $N = |D|$. $|\{d \in D : t \in d\}|$ = term t occurring in the number of documents. Thus, tf-idf(t,d, D)= tf(t,d)*idf(t,D). The greater the number of terms occurring in the document is, the closer the log ratio is to 1, leading to a tf-idf near 0 [37].

The best matching (BM25) function provides search query-based relevant documents. It is a bag-of-words retrieval function that uses query terms to rank the document collections.

$$Score(D,Q) = \sum_{i=1}^{n} IDF(q_i).\frac{f(q_i,D).(K_1+1)}{f(q_i,D) + K_1.\left(1 - b + b.\frac{|D|}{avgdl}\right)}$$

The above BM25 is used to calculate scores with query Q and keywords $q_1, \ldots q_n$. Here, the keyword q_i occurrence in document d is counted by $f(q_i, D)$. $|D|$ *and avgdl* are the document length and average document length, respectively. K_1, b are free parameters for optimization.

$$IDF(q_i) = ln\left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1\right)$$

Here, the document collection count is given by N, and $n(q_i)$ represents the total number of documents with keyword q_i.

Augmentation: Retrieval augmented generation (RAG) is a successor to natural language processing (NLP) foundational models that are utilized for information extraction from data collection with pertinent recovered data in the target context. RAG pipelines are usually employed as BAAI general embedding (BGE) embedding models by BAAI. The retrieval model performance is further boosted by the reranking models. In this case, the input user query is taken in conjunction with the context document being executed with full attention. Thus, the proposed method achieves high accuracy for retrieval in comparison with the embedding dot product. Henceforth, the BGE embedding model is reranked to obtain the most relevant data used for obtaining the top 10 documents and, successively, the top 3 results.

Generation: The generative pretrained transformer (GPT) - 3.5 is a large language model (LLM) decoder-based transformer of a deep neural network (DNN) with an attention architecture [38]. It is used for focusing on relevant text prediction by selectively inputting text. The model can be configured for a 2048 token-long context with few-shot and zero-shot learning abilities on many tasks. The GPT 3.5, which was designed by OpenAI in 2022, has the ability to fine-tune its own data. It is a supervised training model that can be fine-tuned for business and can handle 4000 tokens at a time, which is a pay-per-use model.

3.3. System flowchart and algorithm

Fig. 3. The IIKM flowchart presents the process execution steps in detail. At the beginning, the authentication for the valid user is confirmed. After successful authentication, the user input for the query is in the form of a question. The query is then checked to determine

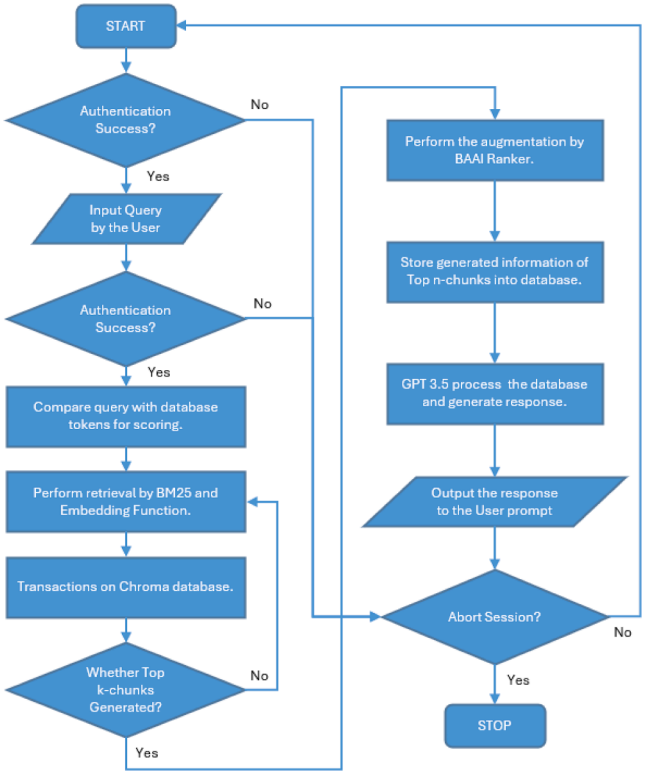

Fig. 3. IIKM Flowchart.

whether it is authorized to transact confidential information. If allowed, then the query is compared with the database tokens from the document collection stored in the industrial data server. The tokens relate to the keywords or sentences present in single or multiple documents. BM25 then performs document ranking on the basis of the query tokens and embedding function for vector normalization from the chroma database to produce the top k chunks. All the documents are processed until the top-k chunks are obtained. The BAAI ranker reranks the top-k documents from the embedding model. The top-n chunks are then stored in the database as outputs. The final output is then provided by GPT-3.5 from the updated database in a grammatically constructed detail response sentence. Next, if the user continues to submit queries, then the responses are generated; otherwise, the system aborts the session.

The algorithm for the IIKM system presents the interaction with the AI assistant as follows. The input given to the system is a user query in the prompt section. Inputs k and n refer to the number of chunks with the highest retrieval and rank similarity scores, respectively. The candidate set refers to the industrial source documents from the data server. The output given by the system is in the form of a response statement with correct grammar for better understanding.

At the beginning, the response and similarity score variables are initialized to NULL. Authentication is confirmed for the organizational member or registered client. Private queries regarding product sales and profit are limited to organizational authority and public queries for service-related queries. Next, on the basis of the query input and database token calculation, the matching token score is calculated as a retrieval function. The retrieval function is defined for the BM25 algorithm, which takes inputs for the query, candidate set and k chunks, which are usually set to 5, where $k > n$. The similarity score is then calculated for each candidate via the BM25 algorithm on the basis of the query. Successively, the top-k chunks are taken by sorting the similarity score in descending order and selecting the highest scoring k chunks. Thus, the top k-chunks are then returned by the BM25 function.

Similarly, the embedding function takes the input as BM25. In this case, the similarity score is calculated by cosine similarity for every candidate from the candidate set. Next, the top k-chunks are calculated from the similarity score and returned by the function.

Retrieval1 stores the vector from the BM25 algorithm by processing it from the chroma database with respect to the token score. Similarly, Retrieval2 stores the vector from the embedding algorithm. The resulting top-k chunks are obtained by combining Retrieval1 and Retrieval2. Successively, the BAAI ranker takes the top n equal to 3 chunks for information generation and is then combined with a prompt. Finally, the response is generated by the GPT-3.5 Turbo via information generation and returned to the user. If the authentication or authorization does not succeed, then "Access Denied" is returned.

3.4. System frontend design

The front end of the IIKM system is presented in Fig. 4. The fields within the input prompt can be described as storage references for all the document database collection. The query asked here is a common question for workplace leave applications. The retrieval score can be used to select how many top n-chunks should be produced by the ranker for the final response evaluation. Finally, the response given by the IIKM system is in the form of detailed organizational rules and its summarization for the application process from the document collections. The relevant documents present the documents referred to from the collection as references. Here, the frontend images are recreated from the Chinese language version for ease of understanding.

Fig. 5 provides details from the IIKM system prompt about the confidence from the top instances given as confidence values of 1 and 2. The keywords are usually taken from the query words and are then compiled in detail from the document collections. The respective keywords related to those instances are given with successive related paragraphs.

In Fig. 6, the additional details of the industrial product query that can possibly be made are suggested by the systems example. The retrieval score standard is also graded on a scale of 5 ratings by the system to be (1) completely irrelevant, (2) barely relevant, (3) somewhat relevant, (4) mostly relevant and (5) fully relevant.

4. Experiments and evaluation design

In this section, the detailed implementation procedures for the experimental settings and the results of the system evaluation are presented.

Fig. 4. Input prompt for the Sakura AI Assistant.

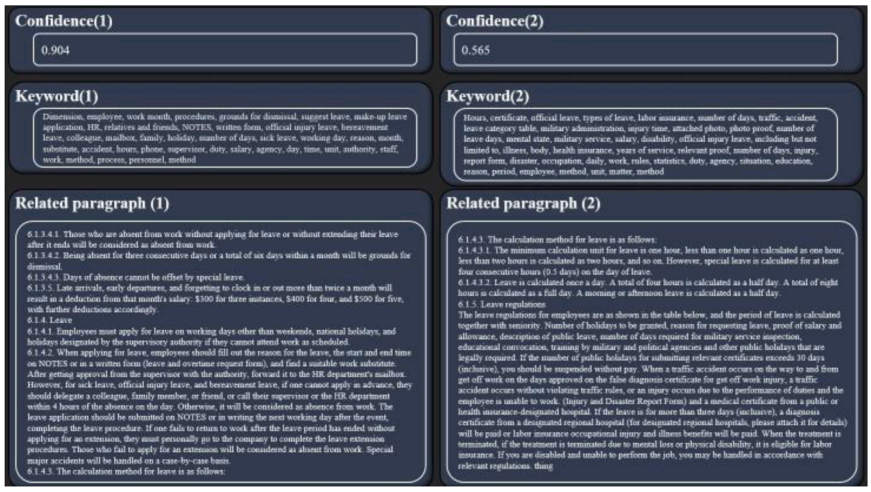

Fig. 5. Prompt based Detail Response Analysis.

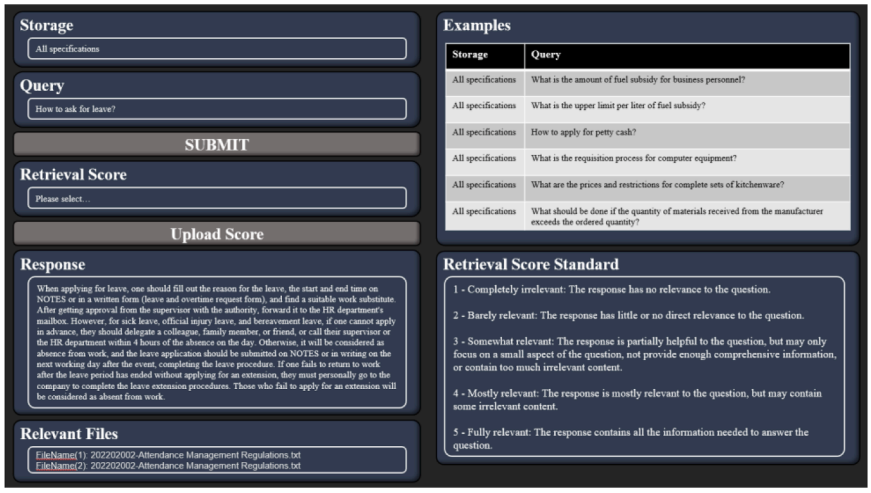

Fig. 6. Retrieval Score Standard.

4.1. Experimental settings and system configuration

The system configuration for setting up the IIKM system is shown in Table 2.

The following is a common method of RAG (retrieval augmented generation), which can be divided into three stages: retrieval, augmentation and generation. The retrieval phase involves searching for the most relevant documents to short list only the query-related documents. In this case, the potential methods include the use of TF-IDF, BM25 and the popular embedding model to calculate cosine similarity. The purpose of the augmentation stage is to update the order of the documents according to the query relevancy. Thus, the total number of files read by our system is reduced in the generation stage. As a result, the respective tokens are saved and improve the accuracy of the generator's response, leading to improved document content. In conjunction, the generation stage mainly addresses user issues after the document content is sorted through the augmented stage. This article mainly discusses the retrieval and augmented stages, with the aim of finding the most relevant documents possible in the limited number of searches and sorting them according to the relevance of the questions and documents. This article experiments the retrieval effects of the TF-IDF, BM25, and embedding methods, as well as the retrieval effect after adding the Reranker model, and proposes a customized system architecture. The total number of approaches involved is as follows: 1. TF-IDF, 2. TF-IDF + Reranker, 3. BM25, 4. BM25 + Reranker, 5. Embedding, 6. Embedding + Reranker, and 7. Our system (BM25+Embedding) + Reranker.

Table 3 provides the details regarding the dataset contents used for the model input. Table 4 shows the hypertuning parameter setup for the model chunk and overlap size, which is useful for achieving the results

Table 2
System Configuration.

System	Workstation (Ubuntu 20.04.6 LTS, 64-bit OS)
Processor	Intel(R) Core(TM) i7–8700 K CPU @ 3.70 GHz
Memory	Kingston 16G*4, DDR4–3200
Graphics Card (GPU)	Gigabyte AORUS RTX2080Ti 11G
Software Library	llama-index, langchain, openai, fastapi, gradio, transformers, chromadb, rank-bm25 and torch.

Table 3
Dataset Contents.

Dataset	Contents
Service Center	Product details, manuals, troubleshooting and maintenance information. Eg: ppt, csv, pdf, docx and txt.
Organizational Regulations	Employee benefits, leave regulations, customer management in record based format.

Table 4
Hyper Tuning Parameter Setup.

	Parameters	Value
Chunking	Chunk size	512
	Overlap size	50
OpenAI	temperature	0

given in the experiments. The algorithm and software versions include BM25, baai/bge-large-zh-v1.5, BAAI/bge-reranker-large, and GPT-3.5 Turbo. The IIKM system extracts 5 relevant documents and 10 in total by using the BM25 and embedding methods, calculates the similarity to the problem through a reranker, and selects the five highest documents.

4.2. Input queries and evaluation

The input queries given to the IIKM system for the experimental performance analysis are as follows:

(1). The consumers using the F series of water heaters were asked to see the Zhitong data.
(2). The consumer's water heater displays E7. What would you ask the service assistant to help determine subsequent maintenance?
(3). If a consumer's DH1683 water heater needs to replace the filter element, what would you ask the service assistant to provide guidance?
(4). If the DH1693 water heater pump of a consumer's home is broken, what would you ask the service assistant to help provide guidance?
(5). The consumer's digital water heater displays E7. What would you ask the service assistant to help determine subsequent maintenance?
(6). The consumer's digital water heater displays E1. What would you ask the service assistant to help determine subsequent maintenance?
(7). The consumer DH1693 water heater makes a loud sound when it is in use. What would you ask the service assistant to help determine subsequent maintenance?

These queries show a real-life user scenario provided by Sakura and ask 5 professional technical maintainers with three years experiences to ask questions in these 7 scenarios. However, the user input for one of the questions does not conform to the default scenario; it is not included in the statistical range of this experiment. Therefore, this experiment collects a total of 34 questions asked by users in real situations. The seven scenarios are related to technical problems such as water heater maintenance and inspection, which includes 5 professional technical maintainers, and 7 different maintenance scenarios consisting of a query by each user.

4.3. IIKM system retrieval ranking evaluation

To evaluate the performance of the IIKM system, three ranking evaluation indicators were used: recall, mean reciprocal rank (MMR) and mean average precision (mAP). Recall represents the retrieved relevant instances from the document collections, also known as sensitivity.

$$Recall = \frac{\text{Relevant retrieved instances}}{\text{All relevant instances}}$$

The mean reciprocal rank (MRR) provides a probability of correctness by evaluating a procedure for a query that produces a list of possible responses as a statistical measure.

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$

Here, Q refers to a set of queries or query instances. Specifically, Specifically, |Q| denotes the number of queries in the dataset or evaluation set. Each query is typically associated with a list of ranked items (such as search results, recommendations, or other ranking tasks) based on their relevance to the query. $rank_i$ is the i th query rank from the top relevant document. The correct answer is the rank-based multiplicative inverse given as $1, \frac{1}{2}$ and $\frac{1}{3}$ for the first, second, and third iterations, respectively. Therefore, MRR is the average of the reciprocal rank solutions for a specific query.

The mean average precision (mAP) is specifically used for document retrieval performance analysis.

$$mAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q}$$

Here, the number of queries is given as Q, which is used to evaluate the complete set of query mAPs for average precision scores.

4.4. Human evaluation

We further evaluate our system in terms of response generation and document retrieval via human scoring in the Organizational Regulations document dataset. Fig. 7 shows that the x-axis represents the user code, which is detailed in Table 5, and the y-axis represents the number of questions collected. The test time duration is 3 weeks, with 14 users and 197 documents in the total amount of data. The rules consist of users completing the scoring, including the generation score, the retrieval file 1 score, and the retrieval file 2 score. The scoring standard uses 5-point Likert scales: 5 - completely relevant, 4 - most relevant, 3 - somewhat relevant, 2 - barely relevant and 1 - not relevant at all.

5. Results

Fig. 8 presents the evaluation of the embedded and ranked results of the dataset from the service center. The mean reciprocal rank (MRR) provides a probability of correctness by evaluating a procedure for a query that produces a list of possible responses as a statistical measure. The BAAI embedding approach was used in our model. Thus, the figure shows the recall, MMR and mAP for seven question groups. The problem of error codes such as questions 2, 5 and 6 has poor retrieval results because the documents of the fault codes are very similar. Therefore, in terms of embedding methods, it is easy to increase retrieval efficiency because of different user problems. Therefore, we later added BM25's keyword search method to compensate for the problem of poor retrieval of similar documents.

Fig. 8(a) presents the MRRs for the 7 questions given in Section 4.2. The embedding and ranking function compared here shows that the embedding function may provide high confidence in false positives, but the ranking function can help in evaluating true positive results. In the case of Fig. 8(b) Recall, the embed and ranker have completely the same response for the results. Fig. 8(c) shows that the mAP performance is similar to that of MRR, where the ranker improves the results that were lost by the embedding function.

In Fig. 9, the x-axis represents the retrieval method, and the y-axis represents score measurements. The graph shows the comparison between the use of the Retrieval method and the addition of the Ranker method. The ranker model can effectively improve the sorting of related files. The chart shows that adding the ranker model can significantly improve MRR and mAP. Therefore, the IIKM system performs better in comparison. The bar graph shows that IIKM systems can lead other algorithms in terms of recall, MRR and mAP. The IIKM system combines the BM25 and embedding methods to search for keywords (BM25) and file content (embedding) simultaneously, allowing the system to search more comprehensively. When the recall reaches 0.85, most questions can be retrieved effectively. When the MMR reaches 0.88, the IIKM system can effectively determine the correlation between documents

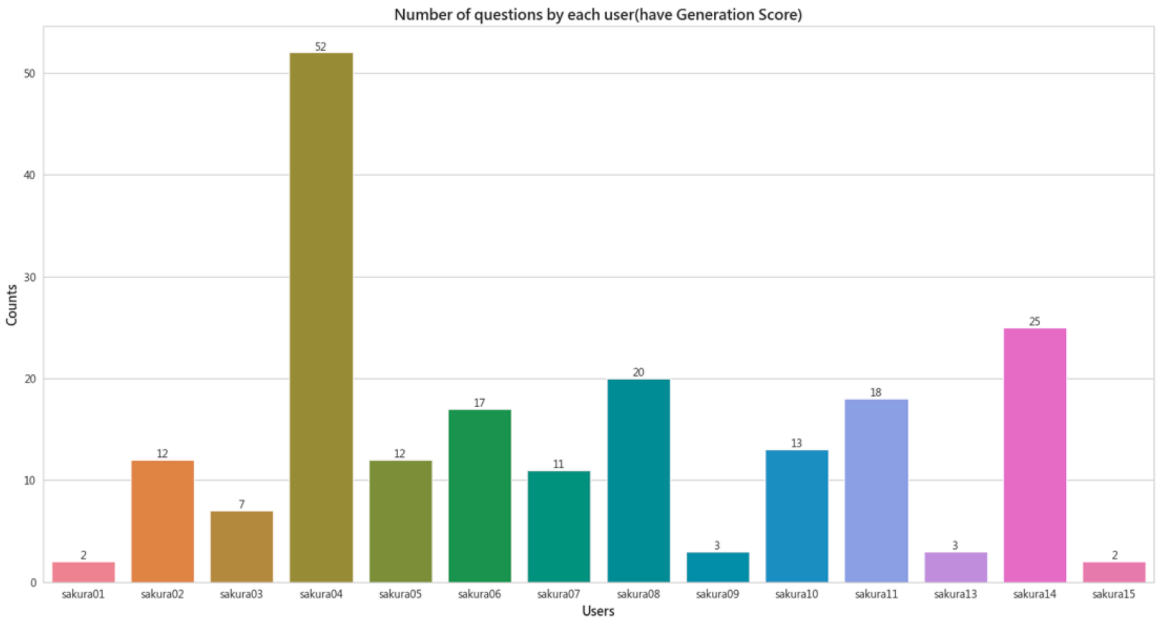

Fig. 7. Generation and Retrieval Evaluation for Multiple Users.

Table 5
Departmentwise User Queries.

User	Storage	Counts
Sakura01	Human Resources Department, Audit Office	2
Sakura02	Overseas Business Office	12
Sakura03	Manufacturing Management Office	7
Sakura04	R&D Center	52
Sakura05	Import Business Office	12
Sakura06	Integrated kitchen business office	17
Sakura07	Planning Department, Integrated Kitchen Business Department, R&D Center, Business Planning Office	11
Sakura08	Marketing Management Office	20
Sakura09	Digital Application Department	3
Sakura10	Finance Department	13
Sakura11	Human Resources Department	18
Sakura13	Yashi Zhuangtou North	3
Sakura14	Service Management Office	25
Sakura15	Sakura Home Furnishing	2

and questions, thus reducing the number of documents that the system needs to input in the generation stage. mAP takes into account both the recall and MRR indicators. The figure shows that our system can effectively retrieve more relevant documents and sort them according to the similarity between documents and questions.

We further examined the performance of the IIKM system in the Organizational Regulations document dataset. In Fig. 10, the x-axis represents the generation score standard, where the points are as follows: 5 - completely relevant, 4 - most relevant, 3 - somewhat relevant, 2 - barely relevant and 1 - not relevant. The y-axis represents the number of GPT-generated scores. The completely relevant score reached 75.63 % (149/197), the most relevant score reached 15.23 % (30/197), the least relevant score reached 10 % (18/197), the barely relevant score reached 75.63 % (149/197), and the number of questions with scores of 1 and 2 points was only 5.

Similarly, in Fig. 11, the initial experiments are performed for the retrieval phase. The first document after retrieval is highly related. (b) The second document will be worse because there is only one document that can answer some questions, so the second document will find irrelevant documents. Fig. 11(c). presents the top score obtained by the Top2 search. Thus, our system can effectively search files and find highly relevant files. The IIKM system has a high score of 98.98 % (195/197) in the retrieval phase. Therefore, we believe that it can find documents effectively on the basis of user problems in the retrieval phase.

Moreover, Fig. 12 presents the heatmap for the complete retrieval and generation scores. Here, the x-axis represents the retrieval score, and the y-axis represents the generated score. The heatmap shows that it mainly falls into the first quadrant, which means that our system can effectively solve user problems in both retrieval and generation. There are only a few questions that GPT cannot answer correctly even if the correct file content is provided.

Table 6 shows that the results for the generation-based retrieval evaluation predict the recall as 91.62 %, the MRR as 97.97 % and the mAP as 91.12 % are better because of complete documentation availability and properly defined rules.

In terms of operational time costs, tasks that previously necessitated manual interaction through a search application to retrieve information required approximately two minutes. These tasks can now be accomplished simply by typing a query. The response time is approximately three to five seconds, which results in a significant reduction in the time required for query operations.

6. Discussion

6.1. LLM model setup and hardware

LMs are well known for AI assistant services that can be applied in the industrial domain with the help of GPU server-based processing as the necessary component. Interaction with the LLM frontend by the Gradio and Fast API provides easy access to LLM-based services [39]. The purpose of this LLM-configured model is to serve as a service call center so that thousands of clients can simultaneously interact with the IIKM system. Therefore, the use of LLM with our approach provides better model features for Chinese language support. The system configuration should be the exclusive server setup with high-configuration GPU hardware. The IIKM architecture can quickly respond to user questions through high-speed computing support and provide the function of changing LLMs, allowing users to choose the LLM that suits them according to their own language and field.

6.2. Integration of open ai and parallel computing

Open AI's GPT 3.5 turbo used within the IIKM system is an AI-based language model, which works by interacting with human-

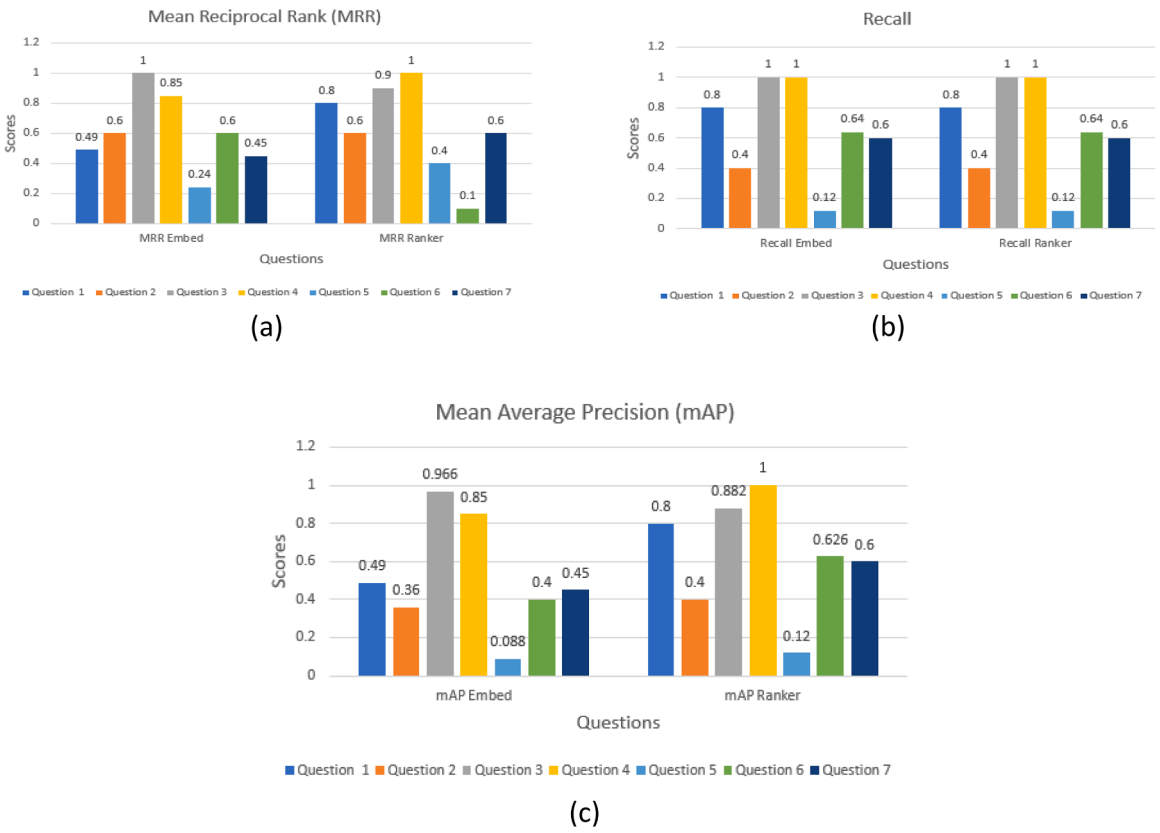

Fig. 8. Comparison of Embed and Ranker Results for a) MRR, b) Recall and c) mAP.

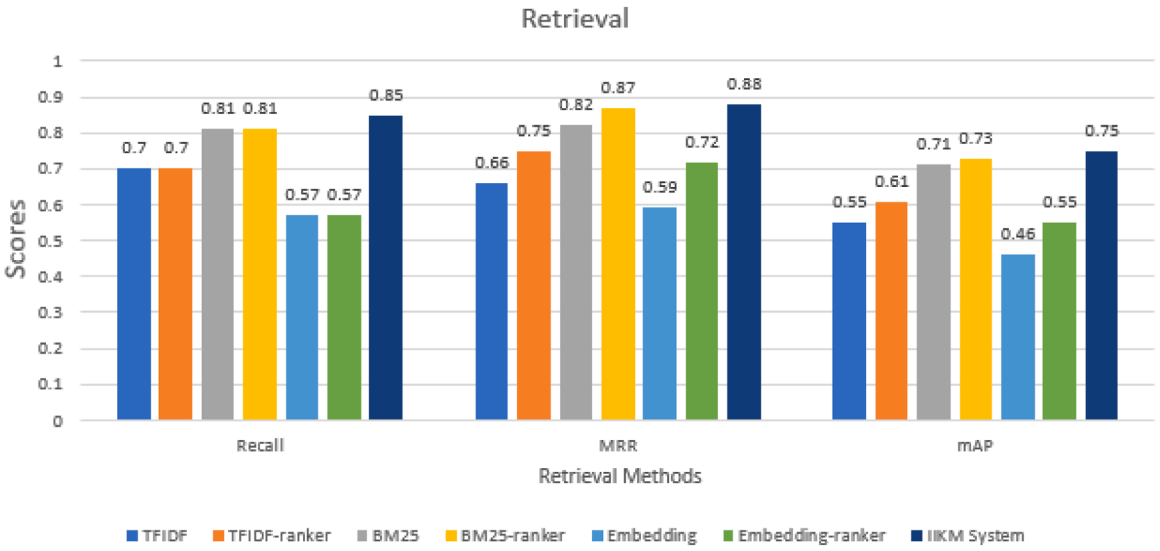

Fig. 9. Comparison of different models with retrieval.

understandable grammatically correct language [40]. The IIKM architecture includes GPU-based parallel computing for efficient facilities. The purpose of GPU processing is to provide services to thousands of clients from different regions, time zones and around the clock. Moreover, as the industrial products, rules and regulations are updated, the updated database can immediately impact the token-generating reference documents for an updated response outcome.

6.3. *Performance modeling of the llm*

The AI assistant service performance for the LLM is evaluated via multiple experiments, settings and results. The performance evaluation is based on the query-based relative response given to the user from knowledge management/database containing all the updated industrial documents. The IIKM uses the retrieval methods of BM25 and the embedding function to generate vectors from the database for the issue query to generate the best response later. The performance for the relative measures is given by the mean reciprocal rank (MRR), recall and

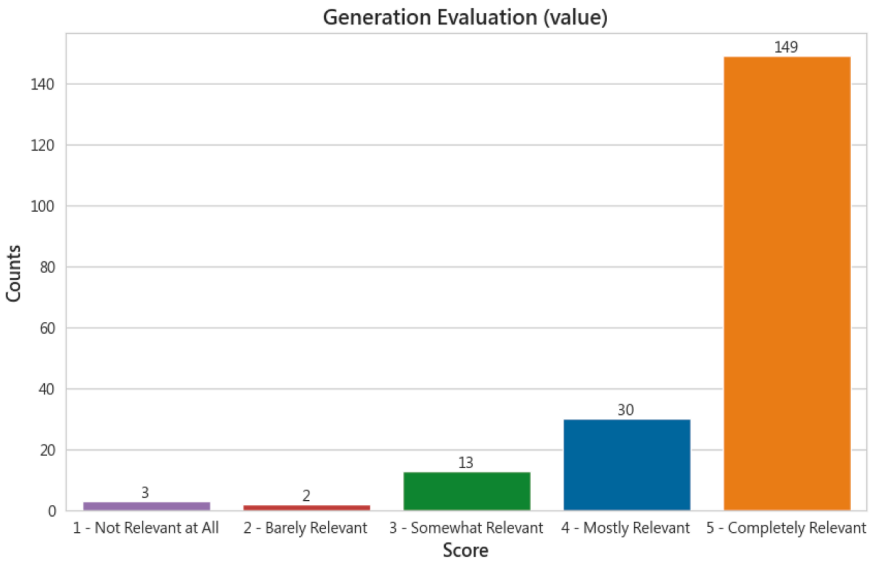

Fig. 10. Generation Evaluation for the Query Response.

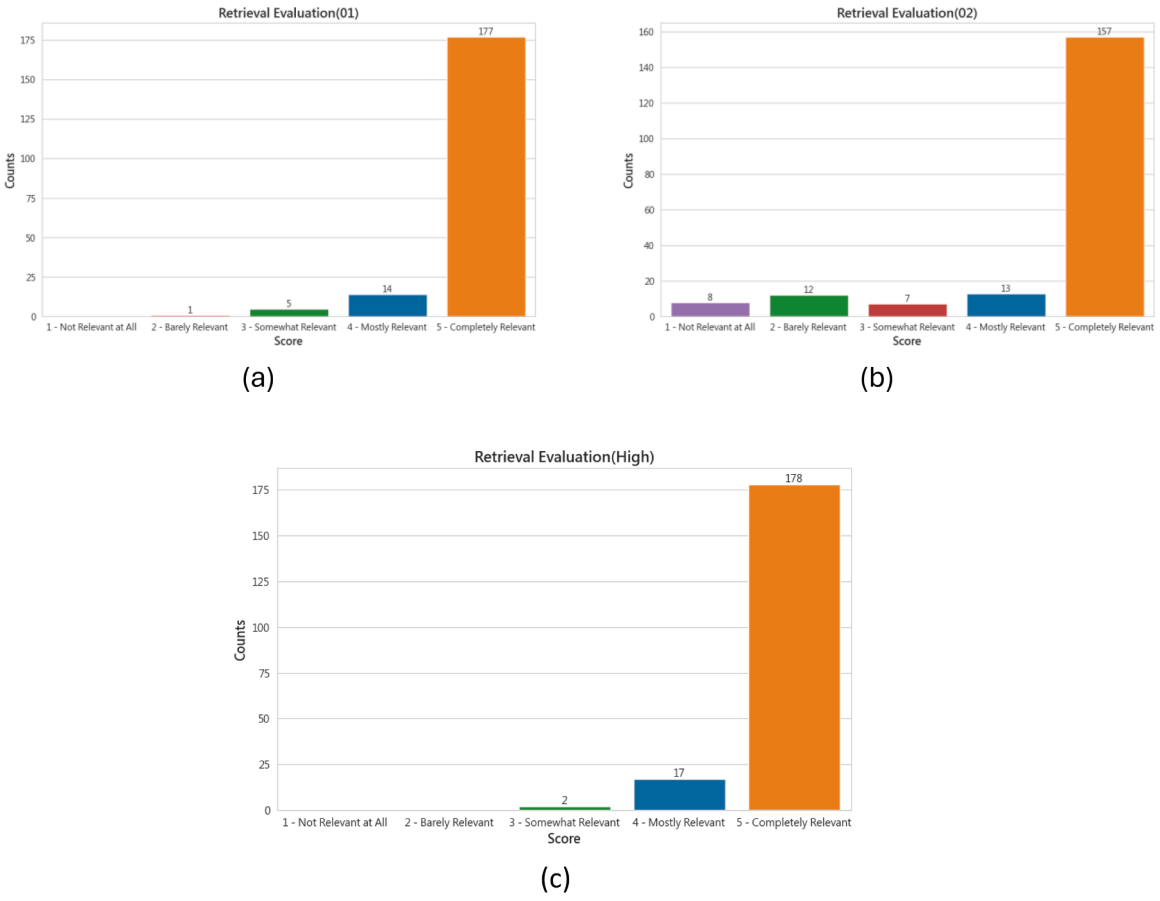

Fig. 11. Retrieval Evaluation for the Document Relevance (a) 1st High Relevance, and (b) 2nd High Relevance and (c) Top 1 search.

mean average precision (mAP). The augmentation phase helps to shortlist the top 3 documents to collect the best response.

6.4. Measures for quality control

The response to the user query determines the quality of the IIKM system. The precision and recall are considered useful output quality control parameters [41]. Recording the performance measures can help to determine the quality maintenance for the IIKM response system at regular monthly intervals. The retrieval methods used for BM25 and embedding are further improved by using the reranker method, which significantly improves the MRR and maP. Multiple tests on the system using a combination of TF-IDF, BM25, embedding and reranker can help to overcome different inaccuracies and can prove to be an effective

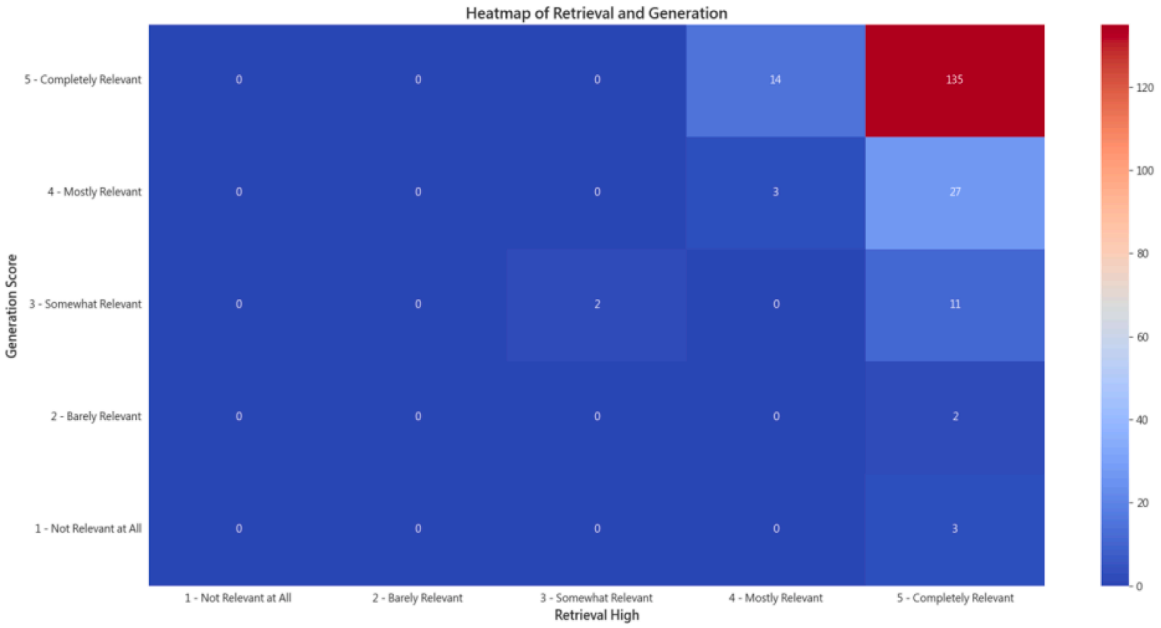

Fig. 12. Heatmap for the Complete Retrieval and Generation Scores.

Table 6
IIKM Resultsv.

Dataset	Mean Reciprocal Rank (MRR)	Recall	Mean Average Precision
Service Center	88 %	85 %	75 %
Organizational Regulations	91.62 %	97.97 %	91.12 %

system. The response quality can then be known from the usefulness of the relative content, which can be improved by updating the database collections regularly. The IIKM system provides a high-quality document reading function that can effectively convert text and tables into files to ensure the quality of the final LLM response. By adjusting Prompting, LLM's responses can best match the user's professional field, and the response style and content can be customized at the same time.

6.5. IIKM system deployment challenges and experiences

IIKM system research provides practical deployment experience sharing from the Sakura Incorporation for the use of a language model as an AI assistant for the service call center [42,43]. The deployment challenges are to convince industrial management of system upgrades with the latest GPU server hardware and open AI registration. Additionally, the memory required should be greater to hold the multiple instance conversation states. In conventional systems, the client's states are maintained only for a call or on a register. Nevertheless, if the call is aborted or the follow-up is missed, then the service center quality and feedback are degraded. Other challenges include training employees weekly/monthly and providing services 24/7. Therefore, addressing this type of challenge is achieved by the IIKM. Now, it is possible to keep the client state for several hours with no delayed response without costly training and available for all durations of server time. The AI assistant provides an exclusive chat window for every client and can provide updated information on any product usage/maintenance guide-based response. The hallucinations within the model are reduced by using data mining algorithms for processing and obtaining vectors. The IIKM system provides a scoring system and correction reply function for llm answers, allowing Sakura Company to quickly accumulate high-quality document content and discover error messages in past documents from each conversation. Through daily iterative dialog, the company's number of high-quality documents can be increased, and the IIKM system's answers can also be made more accurate.

6.6. Overcoming the system limitations

According to the open LLM and limited product guides, hallucinations within the IIKM system response were observed. Even though the general queries were responded to well, the technical queries were not helpful because of the correctness of document retrieval. To overcome the model issues, complete standard manual guides were used in data collection, and multiple methods for TF-IDF, BM 25, embedding and ranker were experimented with to shortlist the best match. High data collection and preprocessing for multiple clients with different categories of queries by different experts was evaluated. Our results prove that the retrieval quality can be effectively improved through a variety of retrievers and that a reranker model can be used to screen out relevant documents suitable for the user's problem effectively before being finally passed to the generation model (gpt-3.5-turbo). This can effectively reduce LLM hallucinations and improve the LLM correctness of answers.

6.7. Hypertuning the llm model

The response by the conversational system needs to be completely specific to the query. Every sentence in the response should be analyzed for relevancy, and when combining the subresponses from multiple documents, hypertuning is necessary to achieve the best orchestration of multiple parameters within the model. Ultimately, the reranker also helps to shortlist the top relevant documents for response augmentation. We set the temperature to 0 to improve the consistency of the GPT model responses. The quality of information retrieval can be improved by setting the chunk size that conforms to the embedding model. Finally, through experiments, we find that the reranker can meet the needs of users when the number of most relevant documents is 2, so we set the top $k = 2$.

7. Conclusion

Large language models (LLMs) are the best choice for AI assistant-

based interactions with the Sakura Corporation for furniture products. Many industries usually provide service call center facilities to resolve customer issues with proper treatment. However, increasing the customer base in multiple countries and providing services 24/7 is challenging for the human environment. Therefore, the IIKM system provides an automated AI assistant that can help resolve maintenance and product breakdown issues. Managing thousands of customers worldwide and 24/7 will reduce the human workload for service call centers. Ultimately, the service provided by the AI assistant will seek answers from the updated industrial server knowledge management. The IIKM system is designed to contain a RAG model and GPT 3.5 for answering queries with high-quality human understandable responses. The RAG model here consists of BM25 and embedding for the vector output generation from the query response by interacting with the chroma database, next sorting data from the BAAI ranker and providing the output to the GPT to generate the complete response. Our approach provides the processing of Chinese documents from the Sakura Industrial Corporation document collection to provide accurate query responses. The recall for the service center is 85 %, and for internal regulatory documents, it is 91.62 %. As the IIKM is effective for conversational applications on the basis of the experimental conclusions, it is recommended for use in industrial AI assistants and service centers. In future work, we will add the feature of a multilanguage support option and improve the LLM prompt engineering capability.

CRediT authorship contribution statement

Lun-Chi Chen: Writing – original draft, Methodology, Formal analysis, Conceptualization, Writing – review & editing. **Mayuresh Sunil Pardeshi:** Resources, Writing – original draft. **Yi-Xiang Liao:** Software, Formal analysis. **Kai-Chih Pai:** Writing – original draft, Supervision, Methodology, Formal analysis, Data curation, Conceptualization, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We would like to thank the DDS-THU AI center for supporting this project.

Data availability

The data that has been used is confidential.

References

[1] J. Yang, Y. Wang, X. Wang, X. Wang, X. Wang, F.Y. Wang, Generative AI empowering parallel manufacturing: building a "6S" collaborative production ecology for manufacturing 5.0, IEEE Trans. Syst. Man. Cybernet. Syst. (2024 Jan 30).
[2] A. Akundi, D. Euresti, S. Luna, W. Ankobiah, A. Lopes, I. Edinbarough, State of industry 5.0—Analysis and identification of current research trends, Appl. Syst. Innov. 5 (1) (2022 Feb 17) 27.
[3] K.I. Roumeliotis, N.D. Tselikas, Chatgpt and open-ai models: a preliminary review, Future Internet. 15 (6) (2023 May 26) 192.
[4] A. Koubaa, W. Boulila, L. Ghouti, A. Alzahem, S. Latif, Exploring ChatGPT capabilities and limitations: a survey, IEEe Access. 11 (2023) 118698–118721, https://doi.org/10.1109/ACCESS.2023.3326474.
[5] S. Rice, S.R. Crouse, S.R. Winter, C. Rice, The advantages and limitations of using ChatGPT to enhance technological research, Technol Soc 76 (2024 Mar 1) 102426.
[6] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, H. Wang, arXiv preprint arXiv: 2312.10997., 2023 Dec 18.
[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).
[8] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, A survey on evaluation of large language models, ACM. Trans. Intell. Syst. Technol. (2023 Dec.).
[9] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, J. Gao, arXiv preprint arXiv: 2402.06196., 2024 Feb 9.
[10] N.N. Vo, S. Liu, X. Li, G. Xu, Leveraging unstructured call log data for customer churn prediction, Knowl. Based. Syst. 212 (2021 Jan 5) 106586.
[11] Zhiling Zheng, Oufan Zhang, Christian Borgs, Jennifer T. Chayes, Omar M. Yaghi, J. Am. Chem. Soc. 145 (32) (2023) 18048–18062, https://doi.org/10.1021/jacs.3c05819.
[12] S. Herbold, A. Hautli-Janisz, U. Heuer, et al., A large-scale comparison of human-written versus ChatGPT-generated essays, Sci. Rep. 13 (2023) 18617, https://doi.org/10.1038/s41598-023-45644-9.
[13] H. Koziolek, A. Koziolek, arXiv preprint arXiv: 2311.10401., 2023 Nov 17.
[14] F. Kieser, P. Wulff, J. Kuhn, S. Küchemann, Educational data augmentation in physics education research using ChatGPT, Phys. Rev. Phys. Edu. Res 19 (2) (2023 Oct 25) 020150.
[15] H. Cai, S.TKG Wu, Telecom Knowledge governance Framework for LLM application, Res. Sq. (2024), https://doi.org/10.21203/rs.3.rs-3252192/v1.
[16] S. Robertson, H. Zaragoza, The probabilistic relevance framework: BM25 and beyond, Foundations and Trends® in Information Retrieval 3 (4) (2009 Dec 16) 333–389.
[17] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W.T Yih, arXiv preprint arXiv: 2004.04906., 2020 Apr 10.
[18] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.T. Yih, T. Rocktäschel, S. Riedel, Retrieval-augmented generation for knowledge-intensive nlp tasks, Adv. Neural Inf. Process. Syst. 33 (2020) 9459–9474.
[19] K. Shuster, S. Poff, M. Chen, D. Kiela, J. Weston, arXiv preprint arXiv: 2104.07567., 2021 Apr 15.
[20] A. Kshirsagar, Enhancing RAG performance through chunking and text splitting techniques, Int. J. Sci. Res. Comp. Sci. Eng. Inform. Technol. (2024).
[21] G. Izacard, P. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel, E. Grave, arXiv preprint arXiv: 2208.03299., 2022 Aug 5.
[22] O. Ram, Y. Levine, I. Dalmedigos, D. Muhlgay, A. Shashua, K. Leyton-Brown, Y. Shoham, arXiv preprint arXiv: 2302.00083., 2023 Jan 31.
[23] K. Li, T. Zhang, L. Tang, J. Li, H. Lu, X. Wu, H. Meng, Grounded dialog generation with cross-encoding reranker, grounding span prediction, and passage dropout, in: Proceedings of the Second DialDoc Workshop on Document-grounded Dialog and Conversational Question Answering, 2022 May, pp. 123–129.
[24] K. Hui, T. Chen, Z. Qin, H. Zhuang, F. Diaz, M. Bendersky, D. Metzler, arXiv preprint arXiv: 2210.05145., 2022 Oct 11.
[25] Z. Liu, Y. Shao, arXiv preprint arXiv: 2205.12035, 2022 May 24.
[26] X. Ma, Y. Gong, P. He, H. Zhao, N. Duan, arXiv preprint arXiv: 2305.14283., 2023 May 23.
[27] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, H. Wang, arXiv preprint arXiv: 2312.10997., 2023 Dec 18.
[28] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, arXiv preprint arXiv: 2307.09288., 2023 Jul 18.
[29] Z. Shao, Y. Gong, Y. Shen, M. Huang, N. Duan, W. Chen, arXiv preprint arXiv: 2305.15294., 2023 May 24.
[30] D. Arora, A. Kini, S.R. Chowdhury, N. Natarajan, G. Sinha, A. Sharma, arXiv preprint arXiv: 2310.20158., 2023 Oct 31.
[31] A. Asai, Z. Wu, Y. Wang, A. Sil, H. Hajishirzi, arXiv preprint arXiv: 2310.11511., 2023 Oct 17.
[32] J. Devlin, M.W. Chang, K. Lee, Toutanova K. Bert, arXiv preprint arXiv: 1810.04805., 2018 Oct 11.
[33] W. Yu, D. Iter, S. Wang, Y. Xu, M. Ju, S. Sanyal, C. Zhu, M. Zeng, M. Jiang, arXiv preprint arXiv: 2209.10063., 2022 Sep 21.
[34] H. Trivedi, N. Balasubramanian, T. Khot, A. Sabharwal, arXiv preprint arXiv: 2212.10509., 2022 Dec 20.
[35] H. Li, Y. Su, D. Cai, Y. Wang, L. Liu, arXiv preprint arXiv: 2202.01110., 2022 Feb 2.
[36] S. Xiao, Z. Liu, P. Zhang, Muennighof N. C-pack, arXiv preprint arXiv: 2309.07597., 2023 Sep 14.
[37] J. Beel, S. Langer, B. Gipp, TF-IDuF: a novel term-weighting scheme for user modeling based on users' personal document collections. 2017.
[38] K.S. Kalyan, A survey of GPT-3 family large language models including ChatGPT and GPT-4, Natural Language Processing Journal (2023 Dec 19) 100048.
[39] C. Jeong, arXiv preprint arXiv: 2309.01105., 2023 Sep 3.
[40] J. Ye, X. Chen, N. Xu, C. Zu, Z. Shao, S. Liu, Y. Cui, Z. Zhou, C. Gong, Y. Shen, J. Zhou, arXiv preprint arXiv: 2303.10420., 2023 Mar 18.
[41] H.D. Nguyen, P.H. Tran, T.H. Do, K.P. Tran, Quality control for smart manufacturing in industry 5.0. Artificial Intelligence for Smart Manufacturing: Methods, Applications, and Challenges, Springer International Publishing, Cham, 2023 Jun 2, pp. 35–64.
[42] Y. Wu, H.N. Dai, H. Wang, Z. Xiong, S. Guo, A survey of intelligent network slicing management for industrial IoT: integrated approaches for smart transportation, smart energy, and smart factory, IEEE Commun. Surv. Tutor. 24 (2) (2022 Mar 10) 1175–1211.
[43] M. Cichosz, C.M. Wallenburg, A.M. Knemeyer, Digital transformation at logistics service providers: barriers, success factors and leading practices, The International Journal of Logistics Management 31 (2) (2020 Jul 14) 209–238.