Interdisciplinary Journal of Information, Knowledge, and Management

An Official Publication of the Informing Science Institute InformingScience.org

IJIKM.org

Volume 20, 2025

A METHOD FOR DETECTING KNOWLEDGE CONFLICTS IN CHINESE INTELLIGENT AGENT INTERACTIONS

Huanyu Cheng*	State Grid Jiangsu Electric Power Co., Ltd. Information and Communication Branch, Nanjing, China	chenghuanyu2020@163.com
Yingcheng Gu	State Grid Jiangsu Electric Power Co., Ltd. Information and Communication Branch, Nanjing, China	guyc3@js.sgcc.com.cn
Mengting Xi	State Grid Jiangsu Electric Power Co., Ltd. Information and Communication Branch, Nanjing, China	ximt@js.sgcc.com.cn
Qiuyuan Zhong	State Grid Jiangsu Electric Power Co., Ltd. Information and Communication Branch, Nanjing, China	zhongqy1@js.sgcc.com.cn
Liu Wei	State Grid Jiangsu Electric Power Co., Ltd. Information and Communication Branch, Nanjing, China	weil18@js.sgcc.com.cn

* Corresponding author

ABSTRACT

Aim/Purpose	This study aims to address the knowledge conflict issues encountered in multi-agent collaboration, particularly when agents based on large language models (LLMs) provide inconsistent answers or recommendations due to varied knowledge sources or errors caused by hallucinations.
Background	The paper tackles the limitations of intelligent agents that cannot dynamically detect or resolve knowledge conflicts. The accuracy of agent responses is enhanced by introducing an automated conflict detection and resolution method.
Methodology	We propose a Knowledge Conflict Resolution (KCR) method that leverages prompt engineering and fine-tuned LLM agents for conflict detection and resolution. The method is evaluated in task-oriented dialogue scenarios, comparing

Accepting Editor Geoffrey Liu | Received: February 5, 2025 | Revised: April 13, April 17, April 19, 2025 | Accepted: April 20, 2025.
Cite as: Cheng, H., Gu, Y., Xi, M., Zhong, Q., & Wei, L. (2025). A method for detecting knowledge conflicts in Chinese intelligent agent interactions. *Interdisciplinary Journal of Information, Knowledge, and Management, 20,* Article 12. https://doi.org/10.28945/5497

(CC BY-NC 4.0) This article is licensed to you under a Creative Commons Attribution-NonCommercial 4.0 International License. When you copy and redistribute this paper in full or in part, you need to provide proper attribution to it to ensure that others can later locate this work (and to ensure that others do not accuse you of plagiarism). You may (and we encourage you to) adapt, remix, transform, and build upon the material for any non-commercial purposes. This license does not permit you to use this material for commercial purposes.

	it against baseline models in terms of consistency, task success rate, and user satisfaction.
Contribution	This study proposes a novel approach to detecting knowledge conflict in intelligent agents. Our innovative approach offers three key advantages over existing solutions: *Higher Accuracy:* Achieves 97.3% conflict detection rate compared to 85-91% in current methods. *User-Friendly Design:* Simplifies complex coordination between agents without technical expertise. *Practical Implementation:* Works effectively across different LLM platforms without requiring system overhauls.
Findings	Experimental results show that our KCR method significantly outperforms existing approaches in resolving conflicts and maintaining coherent multi-agent conversations, notably improving user-perceived reliability.
Recommendations for Practitioners	Incorporating conflict detection mechanisms in intelligent agents can improve knowledge management and enhance user satisfaction, particularly in knowledge-intensive industries.
Recommendations for Researchers	Ensuring the consistency and accuracy of knowledge from different sources is crucial. This paper proposes an effective knowledge conflict detection method to enhance the consistency of knowledge.
Impact on Society	Enhances the reliability and accuracy of intelligent agents in professional fields, facilitates organizational knowledge consistency, and promotes the practical adoption of large language models in complex scenarios.
Future Research	Future research should broaden the scope of this method to include English, investigate its applicability in multimodal large models, and further develop strategies to ensure organizational knowledge consistency in intelligent agents.

Keywords: intelligent agent, large language model, knowledge conflict, ChatGPT

INTRODUCTION

Large language models (LLMs) such as ChatGPT (Wang et al., 2023), Gemini (Gemini Team, 2023), LLAMA (H. Xu et al., 2023), Qwen (Qwen Team, 2024), DeepSeek (Bi et al., 2024), and ERNIE (Sun et al., 2021) have significantly reshaped how people acquire and utilize information. As of October 2024, ChatGPT alone had over 250 million weekly active users. Despite their wide adoption, these models face critical limitations. They often suffer from hallucination issues – generating content that deviates from factual information – and cannot perceive or reflect real-time updates due to the static nature of their training data (Dahl et al., 2024; Huang et al., 2024; R. Xu, Qi, et al., 2024).

Intelligent agents, defined as entities capable of perceiving and acting upon their environment (Agha, 1986; Genesereth & Ketchpel, 1994; Green et al., 1997), have recently benefited from the incorporation of LLMs as their reasoning engines (Lu et al., 2023; Qin et al., 2024; Schick et al., 2023; Yao et al., 2023). These agents are designed to integrate perception, action, and communication modules to assist users in daily and professional settings. With improvements in interaction mechanisms (Park et al., 2023; Qian et al., 2024), multi-agent collaboration has become more effective.

However, LLMs and intelligent agents serve different roles. While LLMs emphasize generality and broad applicability, intelligent agents require task-specific precision (Lu et al., 2023; Qin et al., 2024;

Yao et al., 2023). Due to this difference, general-purpose LLMs often struggle in industry-specific domains unless fine-tuned with expert knowledge – an expensive and technically demanding process (Gao et al., 2024; Wu et al., 2023). Consequently, intelligent agents frequently rely on user feedback and corrections to enrich their internal knowledge. Yet, as user knowledge evolves or multiple users contribute overlapping but differing information, the potential for knowledge conflict arises.

Currently, most intelligent agents handle such conflicts passively. Agents based on Qwen2.5 often prioritize their own internal knowledge, ignoring recent user inputs, while ChatGPT-like agents replace old information without validating it. Neither approach actively notifies users of potential inconsistencies nor provides mechanisms to resolve them. This can lead to confusion, misinformation, and reduced user trust – especially in organizations where knowledge consistency is critical.

Moreover, current conflict detection methods are limited. They typically focus on identifying outdated or inconsistent parametric knowledge within English-language LLMs (Li et al., 2024; Mündler et al., 2024; Zhang & Choi, 2023) while overlooking interactive, user-driven conflicts in intelligent agent environments. Many approaches rely on binary prompts (Li et al., 2024), factual temporal filters (Zhang & Choi, 2023), or chain-of-thought reasoning (Mündler et al., 2024), which may not generalize to multi-user, multi-session scenarios or non-English settings such as Chinese.

To address these challenges, this study proposes a Knowledge Conflict Resolution (KCR) technique tailored for Chinese-language intelligent agents. This technique empowers users to make informed decisions when conflicts arise rather than allowing the system to silently override prior inputs. This study made the following contributions. First, we proposed a lightweight and scalable conflict detection framework for intelligent agents that does not require model retraining. Second, we introduced an algorithm capable of handling multiple knowledge sources, time-sensitive contradictions, and language-specific considerations in Chinese. Third, we constructed a benchmark dataset for conflict detection and evaluated performance across several LLMs, including Llama3.2, Qwen2.5, and DeepSeek-r1.

The rest of this paper is organized as follows. The next section introduces the proposed KCR method and agent framework. Then, the experimental setup and results are presented. Then, the current research gaps and the characteristics of our method are discussed. Finally, directions for future work are outlined.

CONFLICT DETECTION AND RESOLUTION

The method for detecting and resolving knowledge conflicts in intelligent agent interactions due to issues such as hallucinations, the inability to update training datasets in real-time, domain-specific knowledge barriers, and inadequate alignment with human language, or because some users base their inputs on personal cognition, experiences, and mental positions, the responses from large language models can sometimes be erroneous or inconsistent with the user's understanding. At this point, it becomes necessary to address the problem of knowledge conflicts in interactions between intelligent agents and users. The method for resolving conflicts proposed in this paper can be divided into the following three steps: user knowledge discovery, knowledge conflict detection, and knowledge conflict resolution. We first briefly describe the structure of the agent and then provide a detailed introduction to the method we propose.

THE FRAMEWORK OF INTELLIGENT AGENT

The framework of our intelligent agent is shown in Figure 1.

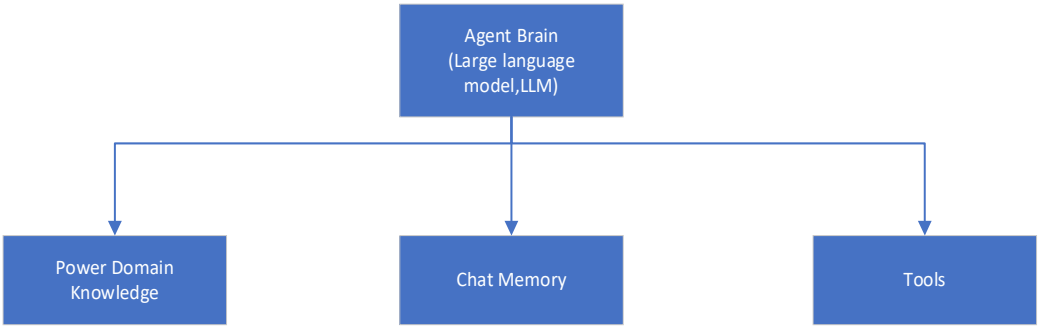


Figure 1. The framework of intelligent agent

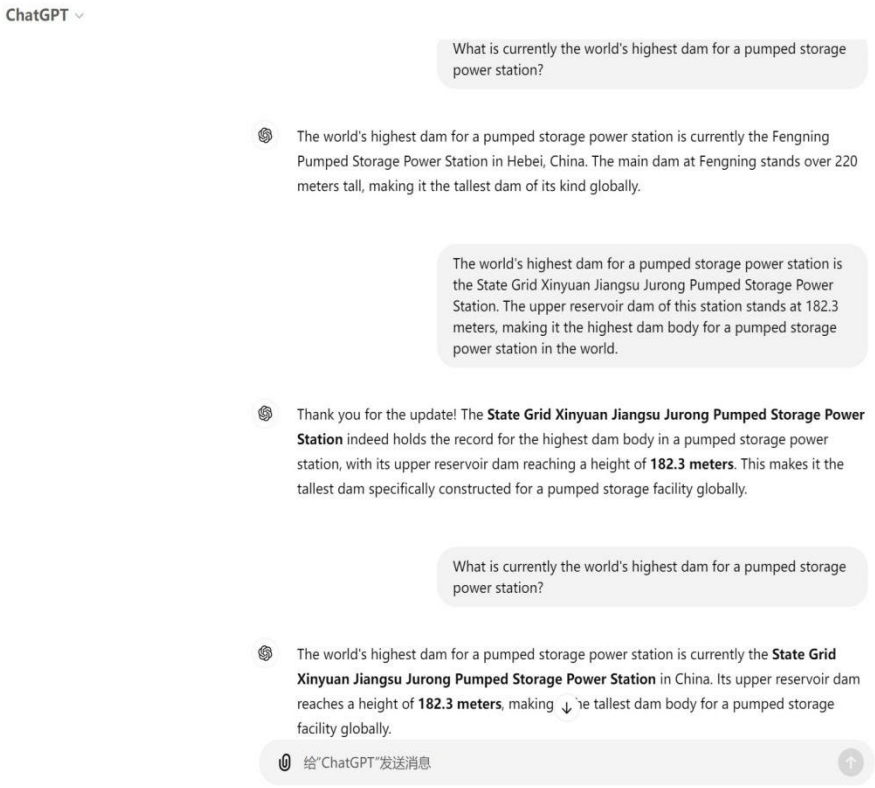
Usually, each user is assigned an individual agent, with different users having distinct agents. The role of each agent is to receive user input, manage user history information such as basic personal details and past conversation records, and provide personalized services based on a large language model. These agents are not limited to consultation; they can also call tools to perform specific actions and even control hardware to influence and modify the surrounding environment.

The intelligent agent has a "brain," typically powered by a large language model. This brain is responsible for receiving user inputs, breaking down and analyzing user intentions, and using its own abilities or external tools to complete tasks and provide results back to the user. When analyzing and solving problems, the agent can refer to an external knowledge base related to the power industry. Knowledge in this repository is generally stored in files such as CSV, PDF, and TXT. The intelligent agent parses the information in these files and stores it in a vector database, making it accessible to the "brain" – that is, the large language model – for queries. Additionally, some of the user's personal information and conversation history are stored in chat memory. The intelligent agent analyzes this information and provides it as prompts to the large language model, ensuring that the agent's responses better meet the user's personalized needs. Developers can also provide tools for the language model to call upon, allowing it to interact with the environment and work collaboratively to accomplish the user's tasks. For example, users can utilize search tools to find the latest information online to compensate for the shortcomings of untimely knowledge updates in large models.

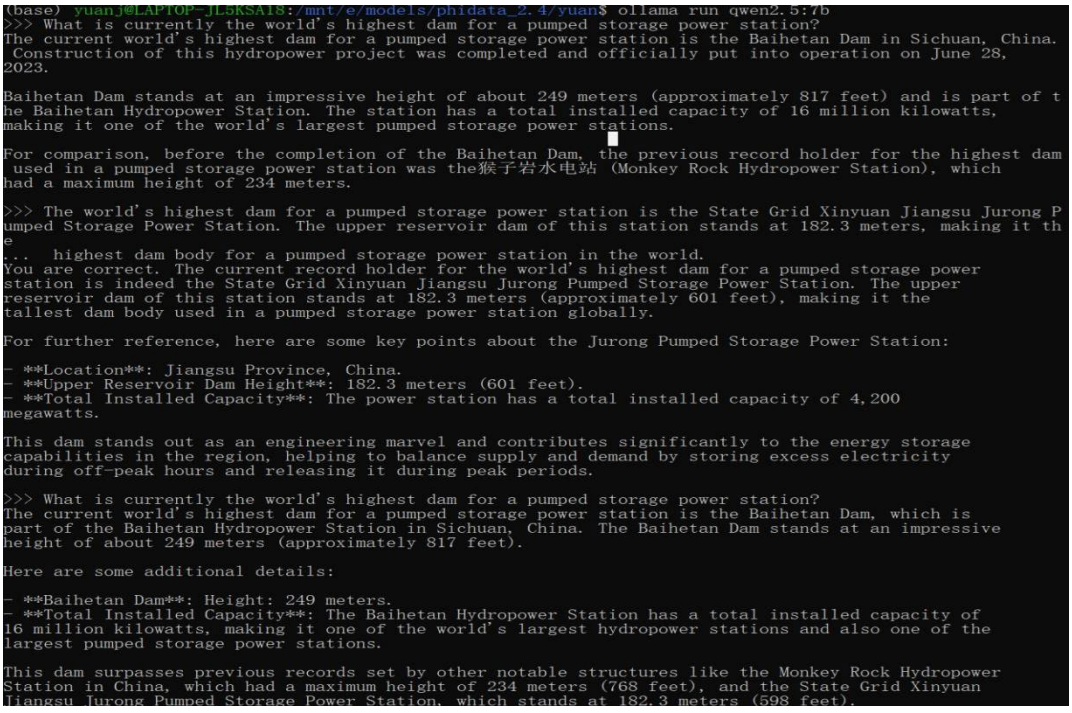
User Knowledge Discovery

In typical usage patterns, users generally consult agents with questions, after which the agents call upon their large language models to answer and return the responses to the users. Therefore, what users typically initiate are interrogative or imperative sentences. However, when the agent's returned answer does not meet the user's expectations and the user has obtained what they believe to be the correct answer from other sources, the user also expects to use their newly acquired answer to replace the one provided by the agent's large language model. This allows the agent to provide the user's expected answer in future consultations.

Therefore, we can assume that when a user inputs an interrogative or imperative sentence, they expect the agent to answer their question. Conversely, when a user inputs a declarative sentence, they more likely want the agent to remember it as knowledge. For example, in the case shown in Figure 2, when the user inputs "Which is the highest dam of a pumped storage power station in the world currently?" the user expects the intelligent agent to answer the question. When the user inputs "The highest dam of a pumped storage power station in the world is the State Grid Xinyuan Jiangsu Jurong Pumped Storage Power Station, the user expects the Agent to remember this piece of knowledge, so that in future inquiries, the Agent will answer with the Jiangsu Jurong Pumped Storage Power Station rather than another station.



(a)



(b)

Figure 2. Conflict handling of user knowledge information by ChatGPT and Qwen 2.5 (essentially, neither performs conflict analysis)

To correctly distinguish whether the user's intention is to ask a question or to correct a knowledge conflict in the agent, we provide the agent with a prompt. Assuming the user's input is *input_str*, the following statement is generated: "Is the sentence '*$input_str*' a piece of knowledge?" *$input_str* represents the actual value of *input_str* being inserted at this position. This statement is then submitted to the large language model in the agent for a response. Regular expressions are used based on the agent's answer to judge whether it is an interrogative or imperative sentence. If not, it is considered a piece of knowledge that the user expects the agent to remember. The reason for using regular expressions for judgment in this case is that they are relatively convenient and effective and can meet the target requirements in most situations.

Conflict Detection

There are four types of knowledge conflicts. Specifically, the first type is a conflict between the knowledge provided by the client and the knowledge inherent in the large language model within the agent; the second type is a conflict between the newly provided knowledge and previously stored knowledge in the agent's knowledge base; the third type is conflicts between different documents in the knowledge base of the same user; and the fourth type is a conflict between the knowledge in the client's agent and the knowledge of other individuals with similar backgrounds within the same organization. The following sections explain each type in detail.

Conflict with knowledge in the large language model

When the knowledge provided by the client conflicts with the knowledge inherent in the large language model within the agent, it could be due to the client's active correction of the model's knowledge because of perceived inadequacies, or it could stem from erroneous information provided by the client. In this paper, we do not delve into the underlying causes; instead, upon detecting a conflict, we alert the user and allow them to choose the answer they consider correct.

To detect conflicts when the client's current input, represented as $paragraph_{input}$, is a declarative statement, we construct a simple question in the following format: "Is the statement '$\$paragraph_{input}$' correct?" This question is then submitted to the large language model within the agent for an answer. Using regular expressions, we analyze the response to determine whether it affirms or denies the statement, thereby assessing if it conflicts with the user's knowledge. For instance, if the user inputs "The approximate value of π is 3.4," we would construct the question "Is the statement 'The approximate value of π is 3.4' correct?" and submit it to the model for evaluation.

Conflict with prior knowledge in the agent

The user's inputted paragraph is represented as $paragraph_{input}$, and the declarative knowledge paragraph represented as $paragraph_{stored}$ previously inputted by the user and stored in the agent is segmented into several sentences; for example, using punctuation marks "." or "?" for segmentation. For question sentences marked with a question mark"?", they are directly deleted. For each sentence, an embedding algorithm is used to obtain its embedding vector v. Here, we use the xiaobu-embedding-v2 model (https://huggingface.co/lier007/xiaobu-embedding-v2), which is a Chinese embedding model based on the piccolo-embedding (Huang et al., 2024) algorithm that maps Chinese tokenization into a 1024-dimensional vector. This can be represented by the formula as follows:

$$v = \text{Linear}(\text{pool}(\text{BERT}(\text{text}))) \quad (1)$$

where BERT refers to the BERT encoding of the text, pool, and linear denote pooling and linear transformation, respectively.

Thus, $\text{paragraph}_{\text{input}} = \{v^1_{input}, v^2_{input} \ldots\ldots v^M_{input}\}$ which means it's segmented into M sentences, $\text{paragraph}_{\text{stored}} = \{v'^1_{stored}, v'^2_{stored} \ldots\ldots v'^N_{stored}\}$ which means it's segmented into N sentences, and then the similarity between each pair of embedding vectors in $\text{paragraph}_{\text{input}}$ and $\text{paragraph}_{\text{stored}}$ is calculated as follows:

$$simi_{i,j} = (v^i_{input} \cdot v'^j_{stored})/|v^i_{input}| \cdot |v'^j_{stored}|, 1 \le i \le M, 1 \le j \le N \quad (2)$$

Assuming that the corresponding sentence groups of v^i_{input} and v'^j_{stored} be the *i-th* sentence embedding vector of $paragraph_{input}$ and the *j-th* sentence embedding vector of $paragraph_{stored,}$ respectively. Take the two groups of sentences with the highest and second highest similarity, where each group contains a sentence from the user's input $paragraph_{input}$ and a sentence from the previously stored knowledge $paragraph_{stored}$ in the agent. When their similarity is greater than or equal to the threshold *T*, further construct three questions to determine if they contradict each other; when their similarity is less than the threshold, it suggests that they are not closely related and, therefore, do not contradict each other, in which case no further action is required. It should be noted that the selection of threshold T is related to the language category. In this paper's Chinese knowledge conflict detection task, an empirical attempt showed that a T value of 0.6 is appropriate. However, the optimal T value for English or other languages needs to be further determined through additional experiments. The conflict between knowledge and its correctness is related to time, as some things that appear to be conflicting or contradictory at the same time may not be so at different times. For example, the statements "The capital of China is Beijing" and "The capital of China is Nanjing" are contradictory, but the statements "The capital of China after 1949 is Beijing" and "The capital of China from 1927 to 1949 was Nanjing" are not contradictory. Considering this, we first assess whether the knowledge involves time constraints. If both pieces of knowledge are compared and involve time constraints, the method of determining conflict differs slightly from other situations.

Let the inputted knowledge statement be $sentence_{input}$ and the knowledge statement stored in the agent's large model be $sentence_{stored}$. We first construct two questions:

(1) "Does the sentence "' + $sentence_{input}$ + "' involve time constraints? Please answer with only one word, 'Yes' or 'No'."

(2) "Does the sentence "' + $sentence_{stored}$ + "' involve time constraints? Please answer with only one word, 'Yes' or 'No'."

If the answers to both questions are "Yes", set the question mode to *ask_mode* = 1, and construct the question: *con_question*="Are the sentences "' + $sentence_{input}$ + "' and "' + $sentence_{stored}$ + "' contradictory? Please answer with only one word, 'Yes' or 'No'." If either of the answers is "No", set *ask_mode* = 2 and construct the question: *con_question*="Can the situations described in "' + $sentence_{input}$ + "' and "' + $sentence_{stored}$ + "' coexist? Please answer with only one word, 'Yes' or 'No'."

Then, the question *con_question* is submitted to the agent's large language model, and its answer *con_answer* is obtained. If *ask_mode* == 1 and *con_answer* == 'Yes', or if *ask_mode* == 2 and *con_answer* == 'No', then $sentence_{input}$ and $sentence_{stored}$ are considered to conflict. Otherwise, they do not.

The algorithm is shown in Figure 3.

For example, if the input sentence is "Shanghai is the biggest city of China," and the stored sentence is "The biggest city in China is Chongqing," the two questions posted to the agent's model would be:

(1) "Does the sentence 'Shanghai is the biggest city of China' involve time constraints? Please answer with only one word, 'Yes' or 'No'."

(2) "Does the sentence 'The biggest city in China is Chongqing' involve time constraints? Please answer with only one word, 'Yes' or 'No'."

The answer to both questions is "No," so the next question is: "Can the situations described in 'Shanghai is the biggest city of China' and 'The biggest city in China is Chongqing' coexist? Please answer with only one word, 'Yes' or 'No'." If the answer is "No", they are considered to conflict.

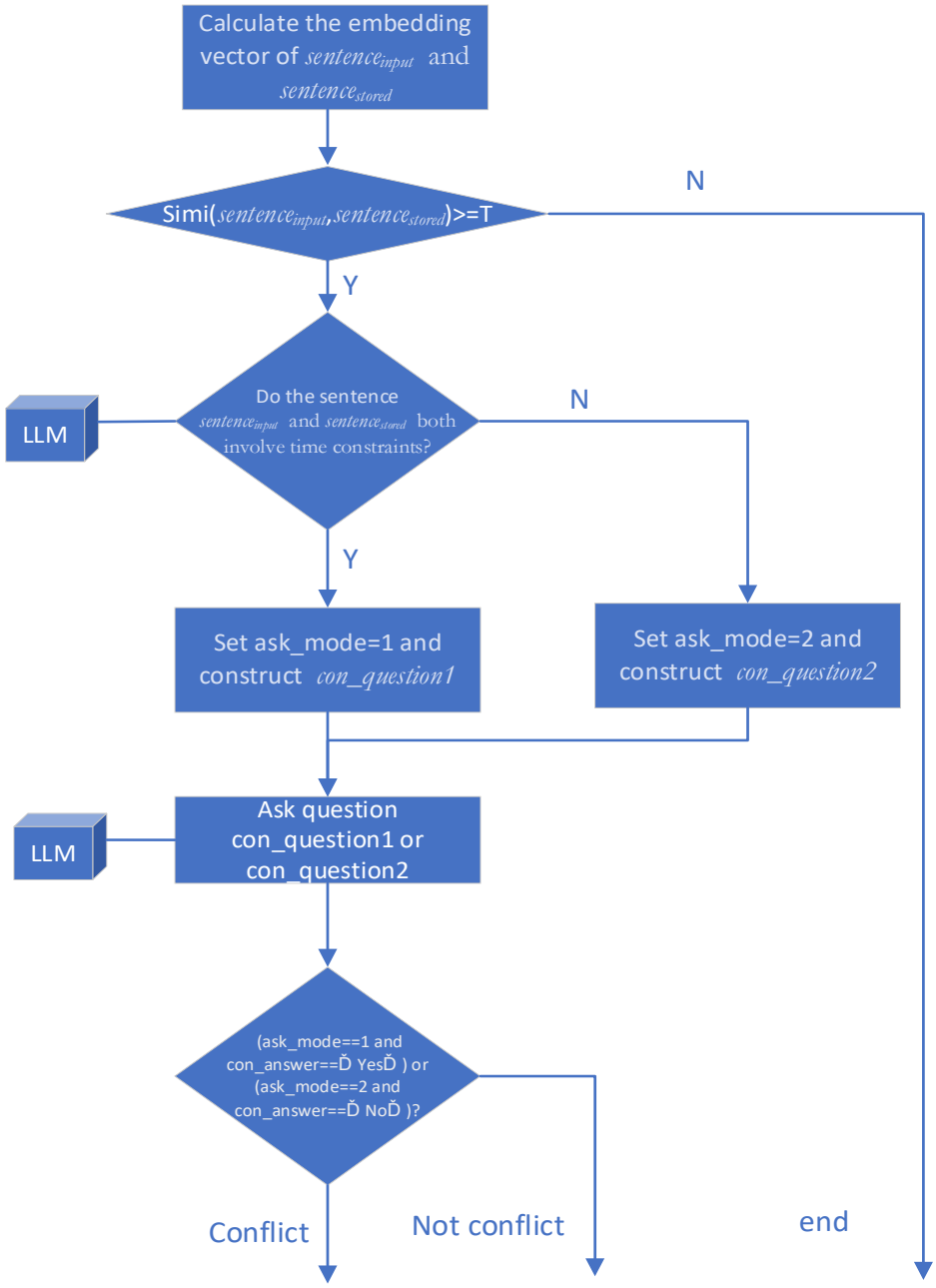


Figure 3. Algorithmic flow of detecting conflict with prior knowledge in the agent

For another example, if the inputted sentence is "The capital of China after 1949 is Beijing" and the stored sentence is "The capital of China from 1927 to 1949 was Nanjing," the two questions asked would be:

(1) "Does the sentence 'The capital of China after 1949 is Beijing' involve time constraints? Please answer with only one word, 'Yes' or 'No'."

(2) "Does the sentence 'The capital of China from 1927 to 1949 was Nanjing' involve time constraints? Please answer with only one word, 'Yes' or 'No'."

The answers are both "Yes," so the next question is: "Are the sentences 'The capital of China after 1949 is Beijing' and 'The capital of China from 1927 to 1949 was Nanjing' contradictory? Please answer with only one word, 'Yes' or 'No'." If the answer is "No," they are considered not to conflict

Knowledge conflicts between different documents in the knowledge base of the same user agent

Sometimes, users upload multiple documents containing conflicting information to the knowledge base. If a user inquires about this conflicting information, the intelligent agent may return incorrect responses. Therefore, it is necessary to detect knowledge conflicts across different documents in the knowledge base. Directly detecting conflicts across all documents in the knowledge base can be time-consuming when the uploaded documents are rich in content, which may not meet user expectations. Hence, this paper decentralizes the knowledge conflict detection process, integrating it into each user query interaction.

When a user submits a query to the agent, represented as $paragraph_{input}$, the intelligent agent searches across multiple knowledge base documents uploaded by the user. If no relevant information is found, the agent uses its internal large language model to generate a response directly. When relevant information is retrieved from the knowledge base, multiple results, such as $answer_1, answer_2,...answer_n$, are combined into a single long sentence as $answer_{con} = answer_1+answer_2+...answer_n$. Here, the operator "+" denotes string concatenation. A question is then constructed: "Are there any contradictions in the following statements: ${$answer_{con}$}?" and it is submitted to the large language model for answering. If the large model determines that there are no contradictions, it generates an answer and delivers it directly to the user. Otherwise, the embedding algorithm is applied to obtain the embedding vectors $v_{doc}^1, v_{doc}^2, ...v_{doc}^n$ of the results, and their pairwise similarities are calculated. Vectors with a similarity greater than or equal to a threshold T are identified. Using the method mentioned in the previous section, these vectors are evaluated to determine whether they conflict with one another. If conflicts are found, the user is prompted to address them. When the number of vectors with a similarity greater than or equal to T is large, further analysis can be limited to only the top few vectors with the highest similarity scores to streamline conflict detection.

Knowledge conflicts between different user agents

Since any two users may have significant cognitive differences due to their profession, field, or other factors, it is often difficult to determine the correctness of their knowledge, making the detection of knowledge conflicts less meaningful. For example, when two people from different countries say the same sentence, "I went to the capital this weekend," a Chinese person refers to Beijing as the capital, while an American refers to Washington, D.C., and they naturally differ.

However, since the users' knowledge backgrounds and working environments are similar within the same organization, detecting knowledge conflicts within their respective agents is still meaningful. It can help eliminate potential risks arising from inconsistent knowledge recognition in the workplace. Therefore, in this paper, the discussion of knowledge conflicts between different user agents primarily refers to conflicts between users within the same small group.

The detection of knowledge conflicts between different user agents is based on the assumption that there are no conflicts within the knowledge of a single user's agent. Suppose there are r users, represented as $u_1,u_2 ... u_r$, and all user agents' knowledge bases contain a total of L statements. The following algorithm is used to detect knowledge conflicts between users.

The pseudocode for the knowledge conflict detection algorithm between different user agents is as shown in Figure 4.

Algorithm for Detecting Knowledge Conflicts Among Different User Agents

STEP 1. Sequentially select a user u_i and its knowledge base $KL(u_i)=\{v_1, v_2...,v_M\}$

STEP 2. Start from u_{i+1}, sequentially select a user u_j and its knowledge base $KL(u_j)=\{v'_1, v'_2...,v'_N\}$

STEP 3. Sequentially select a vector $v_p(i)$ $1<=i<=M$ from $KL(u_i)$, calculate its similarity with any vector $v'_q(j)$ $1<=q<=N$ from $KL(u_j)$, and obtain the maximum similarity value between $v_p(i)$ and $v'_q(j) \in KL(u_j)$, i.e.,

STEP 4. When $simi(v_p(i), v'q(j))>T$, use the same method as conflict detection with prior knowledge in the agent to determine whether the corresponding sentences $s_p(i)$ and $s'_q(j)$ contradict each other. If they are found to contradict, proceed to STEP 5; otherwise, return to STEP 3.

STEP 5. Select the sentence corresponding to $v_p(i)$ and the sentence preceding it in the knowledge base $KL(u_i)$ If they are in the same paragraph, combine them into a longer sentence $ss_p(i)=\{s_{p-1}(i), s_p(i)\}$, otherwise, $ss_p(i)=s_p(i)$

STEP 6. Select the sentence corresponding to $v'_q(j)$ and the sentence preceding it in the knowledge base $KL(u_j)$, If they are in the same paragraph, combine them into a longer sentence $ss'_q(j)=\{s_{q-1}(j), s_q(j)\}$, otherwise $ss'_q(j)=s'_q(j)$

STEP 7. Use the same method as conflict detection with prior knowledge in the agent to determine whether the sentences $ss_p(i)$ and $ss'_q(j)$ contradict each other. If they do, prompt users u_i and u_j to handle the conflict.

Figure 4. Algorithm for detecting knowledge conflicts among different user agents

The reason why the algorithm checks for contradiction again when $s_p(i)$ and $s'_q(j)$ are found to contradict is that some statements that appear contradictory in isolation may not actually contradict each other in their broader context. For example, $\pi=3$ and $\pi=3.14$ may seem contradictory on their own, but if the former requires retaining one significant digit and the latter requires retaining three significant digits, they are not actually contradictory.

Conflict resolution

Essentially, large language models do not conduct conflict analysis and thus do not handle conflicts. Objectively, ChatGPT directly replaces its own knowledge with the information provided by the user, which can be shown in Figure 2(a). At the same time, the Qwen2.5 model does not pay attention to the user's input information and always uses its own knowledge to respond, as shown in Figure 2(b).

When our method detects a conflict, it hands the decision over to the user, alerting them that their recently entered information conflicts with the stored historical knowledge. This can, to some extent, draw the user's attention. Some meticulous users will analyze the cause of the conflict and further verify the accuracy of the information, thus preserving relatively more accurate information after thoughtful consideration, improving the efficiency of subsequent interactions. When preserving relatively more accurate information, the retained information should replace the other information. If the stored historical knowledge is more accurate, the historical knowledge sentence obtained in the previous step should replace the sentence in the declarative paragraph recently input by the user; whereas, if the knowledge recently input by the user is more accurate, the sentence in the declarative paragraph recently input by the user should replace the corresponding sentence in the historical knowledge.

EVALUATION EXPERIMENT

We use Phidata2.4 as the development framework for the intelligent agent, with Python version 3.10 and the large language model base for the agent being Qwen2.5. The localization deployment tool for the large language model is Ollama. The main relevant computer hardware includes a CPU of AMD Ryzen 9 5900HX, a GPU of 3080 Laptop with 16GB VRAM, and 64GB of RAM.

In Phidata, the agent comes with a storage module, and the historical dialogue records with the large model influence the agent's future responses. Since all questions posed to the agent are answered using the inherent capabilities of the agent's large model, such question records are cleared and not saved in this method. For declarative professional knowledge from users that needs to be retained, it is saved in the agent's chat history. The threshold for the similarity of sentence embedding vectors is set at 0.6.

Data Analysis

This study involves the analysis of knowledge conflicts in different scenarios, requiring different datasets for testing. For the first scenario, where the knowledge provided by the client conflicts with the knowledge inherent in the large language model within the agent, we do not expand further as the agent's large language model is inherently capable of detecting this type of conflict, meaning our algorithm and the model's algorithm yield the same result.

For the second scenario—conflicts between newly provided knowledge and previously stored knowledge in the agent's knowledge base, we designed 100 sets of dialogues. Among them, 90 are sets of dialogues, each with two contradictory declarative statements, one correct and one incorrect. These 90 sets include both the aforementioned cases of knowledge errors due to outdated information and cases where the knowledge itself is incorrect. There are 10 sets of dialogues, each of which has no contradictory declarative statements. Although the content within each of those 10 sets of dialogues is not inherently conflicting, they are quite misleading and can easily be perceived as contradictory. For example, one set contains the following two declarative statements: (1) The price of a notebook is not expensive, usually not exceeding dozens of yuan; (2) The price of a laptop is relatively high, generally over 2000 yuan. These two sentences actually refer to two different objects: a paper notebook and a laptop. Therefore, they are not contradictory. However, LLMs might easily misidentify both as referring to a laptop, thus incorrectly concluding that the two statements are in conflict. For the third scenario, the handling method is somewhat similar to the approach used in the second scenario, so it was not tested separately.

For the fourth scenario, due to the difficulty of simulating multi-user agents, we used an alternative approach. We prepared 50 documents, each containing 1-3 knowledge segments of varying lengths, with each segment comprising one or more sentences. Each document is treated as a separate user, with the user's knowledge base consisting of knowledge segments, and each segment made up of sentences. Detecting conflicts within these documents can thus be viewed as simulating multi-user knowledge conflict detection. For instance, one document might contain the following statement: "Beijing is a beautiful city; it is the capital of China. It is located at the northernmost part of the North China Plain, around 40 degrees north latitude." In contrast, another document might include: "The capital of the People's Republic of China is Shanghai, located at the mouth of the Yangtze River." The documents contain only declarative knowledge, excluding interrogative, imperative, or other subjective statements.

Performance Measurement

In this study, we use four performance metrics, Precision (*P*), Accuracy (*A*), Recall (*R*), and *F1*-score, to measure and evaluate the experimental results. Their definitions are as follows:

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \tag{3}$$

$$\mathrm{A} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}} \tag{4}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{5}$$

$$F_1 = \frac{2\times P\times R}{P+R} \tag{6}$$

Where *TP* represents the number of knowledge pairs correctly predicted as contradictory, *TN* represents the number of knowledge pairs correctly predicted as non-contradictory, *FP* represents the

number of knowledge pairs incorrectly predicted as contradictory, and *FN* represents the number of knowledge pairs incorrectly predicted as non-contradictory. The F1-score provides a comprehensive single-value metric for evaluating the effectiveness of conflict detection.

RESULTS

To demonstrate the effectiveness of our method, we chose for comparison three similar methods that were relatively recent. One is the detection method proposed by Wang et al. (2024), which breaks down the entire context into individual sentences and classifies each sentence to determine whether it conflicts with the knowledge in the knowledge base. Another approach was proposed by Li et al. (2024), where they concatenated the two sentences that need to be judged into a single paragraph and then submitted this paragraph to a large model, asking whether there is any self-contradictory content within it. The third method was proposed by M ndler et al. (2024), where they leverage chain-of-thought prompting to ask the LLM to first provide an explanation and then, in conjunction with this explanation, determine whether the two sentences are contradictory. For automatic testing, we conducted experiments on five open-source large models: Llama3.2, Llama3.2-vision, Qwen2.5, DeepSeek-r1, and Phi3-mini. Llama3.2 is the 3 billion parameter version, Llama3.2-vision has 11 billion parameters, Qwen2.5 has 7 billion parameters, phi-3-mini has 3.8 billion parameters, and DeepSeek r1 has 7 billion parameters. For cases where the same user submits conflicting knowledge to the agent, the overall experimental results are shown in Table 1.

Table 1. Experimental results of conflicting knowledge detection of the same agent

		Method	Precision (P)	Accuracy (A)	Recall (R)	F1 Score (F1)
The LLM contained in agent	Llama3.2	Wang	0.889	0.380	0.356	0.508
		Mündler	0.895	0.85	0.944	0.919
		Li	0.650	0.170	0.167	0.266
		this study	**0.947**	**0.950**	**1.000**	**0.973**
	Llama3.2-vision	Wang	0.891	0.530	0.544	0.676
		Mündler	0.853	0.34	0.322	0.468
		Li	0.868	0.380	0.367	0.516
		this study	**0.933**	**0.880**	**0.933**	**0.933**
	Qwen2.5	Wang	1.000	0.110	0.010	0.020
		Mündler	/	/	/	/
		Li	0.868	0.380	0.367	0.516
		this study	**0.947**	**0.950**	**1.000**	**0.973**
	DeepSeek r1	Wang	**1.000**	0.120	0.022	0.043
		Mündler	**0.851**	0.43	0.444	0.584
		Li	0.868	0.380	0.367	0.516
		this study	0.972	**0.770**	**0.767**	**0.857**
	Phi3-mini	Wang	0.750	0.200	0.167	0.273
		Mündler	1.000	0.33	0.256	0.407
		Li	0.868	0.380	0.367	0.516
		this study	**0.940**	**0.830**	**0.867**	**0.902**

As shown in Table 1, our method achieves the best results when using any of the five LLMs as the base model for the agent. When Llama3.2 is used as the large base model, our method can identify all conflicting knowledge statements, significantly improving the values of the A (accuracy) and R (recall) metrics and helping the F1 score increase by more than 0.05 relative to the other three methods.

When Llama3.2-vision or Qwen2.5 is used as the base model, our method also shows some improvement over the other three methods. When Qwen2.5 serves as the large base model, the methods proposed by Wang et al. only detected a set of conflicting knowledge pairs, resulting in primary metric scores of 0.01, whereas our method can still accurately detect conflicting knowledge pairs, thus achieving better metric scores. It should be noted that when using Qwen 2.5, M ndler' method assumes all knowledge pairs are non-contradictory, rendering the calculation of various metrics impossible. When DeepSeek r1 or Phi3-mini is used as the large base model, the F1 score of our method is more than 0.3 higher than that of the other three methods.

Additionally, another point that can be observed from the table is the detection effectiveness of the Qwen large model, which is more sensitive to the choice of instructions and prompts – incorrect choices can lead to poor performance in detection. In contrast, Llama3.2 and Llama3.2-vision are more robust with respect to the selection of instructions and prompts. Taking Qwen2.5 as an example, in the 90 sets of dialogues that contain conflicting knowledge, our algorithm detected all 90 conflicts. In the 10 sets of dialogues that do not contain conflicting knowledge, our method correctly identified five sets as non-conflicting while incorrectly detecting conflicts in the other five sets. For instance, our method erroneously identified the two statements "The price of a notebook is not expensive, usually not exceeding dozens of yuan" and "The price of a laptop is relatively high, generally over 2000 yuan" as being contradictory.

We can also briefly analyze the underlying reasons behind the experimental results in Table 1. Qwen2.5 and DeepSeek r1 are more proficient in handling Chinese, which leads to significant instability in the other three algorithms that primarily focus on English knowledge conflict detection some of them even fail to detect a single conflict with Qwen2.5. On the other hand, Llama3 is a commonly used open-source model in academic research, so other methods tend to unintentionally adopt prompt formats that are better suited for this model. As a result, the performance on Llama3 is generally better than on phi3.

Additionally, we employ inferential statistical methods to compare ratio scores and report the results of significance tests. Here, we opt for the macro sign test (S-test) and the macro t-test(T-test) proposed by Yang and Liu (1999) to do significance tests on metrics such as P, A, R, and F1 scores. The S-test is a non-parametric test suitable for situations where the data does not meet the assumption of normal distribution, whereas the T-test is a parametric test appropriate for situations where the data satisfies the assumption of normal distribution. For detailed explanations and calculation methods of the S-test and T-test, users can reference the literature proposed by Yang and Liu (1999). The test results are shown in Table 2 and Table 3.

Table 2. Significance testing (S-test) of performance difference

Method A	Method B	Precision (P)		Accuracy (A)		Recall (R)		F1 Score	
		S	*p*	S	*p*	S	*p*	S	*p*
Ours	Wang	2	0.5	0	0.03125	0	0.03125	0	0.03125
Ours	Li	1	0.1875	0	0.03125	0	0.03125	0	0.03125

Table 3. Significance testing (T-test) of performance difference

Method A	Method B	Precision (p)		Accuracy (a)		Recall (r)		F1 score	
		t	*p*	t	*p*	t	*p*	t	*p*
Ours	Wang	0.998	0.188	7.726	0.0005	7.17	0.0012	5.62	0.005
Ours	Li	1.971	0.059	8.09	0.0007	7.45	0.001	6.05	0.004

To interpret these results, we briefly explain the role of p-values: in both tests, a p-value less than 0.05 generally indicates a statistically significant difference between the two methods compared.

By using S-test, since our method outperforms the other three methods across all five categories on metrics A, R, and F1 value, the value of S is consistently 0, which yields identical and low p-values. This indicates that the performance improvements are statistically significant ($p<0.05$). Similarly, in the T-test, all p-values for Accuracy, Recall, and F1 score are also below 0.05, further confirming the statistical significance of our method's superiority. The slightly higher p-values for Precision suggest that the improvement on this metric, while present, is not statistically significant at the 0.05 level in some cases. These significance tests collectively support that our proposed method achieves consistently better performance compared to the baselines.

We also tested the impact of different similarity thresholds T in Figure 3 on the test results. We selected two LLMs – Llama3.2 and Qwen2.5 – that performed best in the previous experiments. As shown in Table 4, when T exceeds 0.6, the performance tends to decline, especially with a noticeable drop in recall (R) when T=0.8. The performance is similar for T=0.5 and T=0.6, but a lower threshold like T=0.5 results in more sentence pairs to be processed, which also leads to slower speeds. Therefore, considering all factors, T=0.6 is a better choice overall.

Table 4. Impact of different T values on experimental results

	Llama3.2			**Qwen 2.5**		
	P	**R**	**F1**	**P**	**R**	**F1**
T=0.5	0.956	1	0.978	0.947	1.000	0.973
T=0.6	0.947	1.000	0.973	0.947	1.000	0.973
T=0.7	0.944	0.967	0.955	0.946	0.967	0.956
T=0.8	0.960	0.589	0.730	0.945	0.600	0.734

Here, we illustrate the performance of our method compared to methods like ChatGPT and Qwen with a detailed example. The experimental design for one set is as follows. Currently, the pumped storage power station with the highest-rated head in the world is the Kanno River Pumped Storage Power Station in Japan. However, the Tiantai Pumped Storage Power Station in Zhejiang, China, is under construction and is expected to be preliminary completed by July 2026, at which point it will become the new pumped storage power station with the highest-rated head in the world. Therefore, we designed five sequential inputs for the large model or agent:

(1) Which pumped storage power station currently has the world's highest-rated head?

(2) The pumped storage power station with the highest rated head in the world is the Kanno River Pumped Storage Power Station in Japan.

(3) Which pumped storage power station currently has the world's highest-rated head?

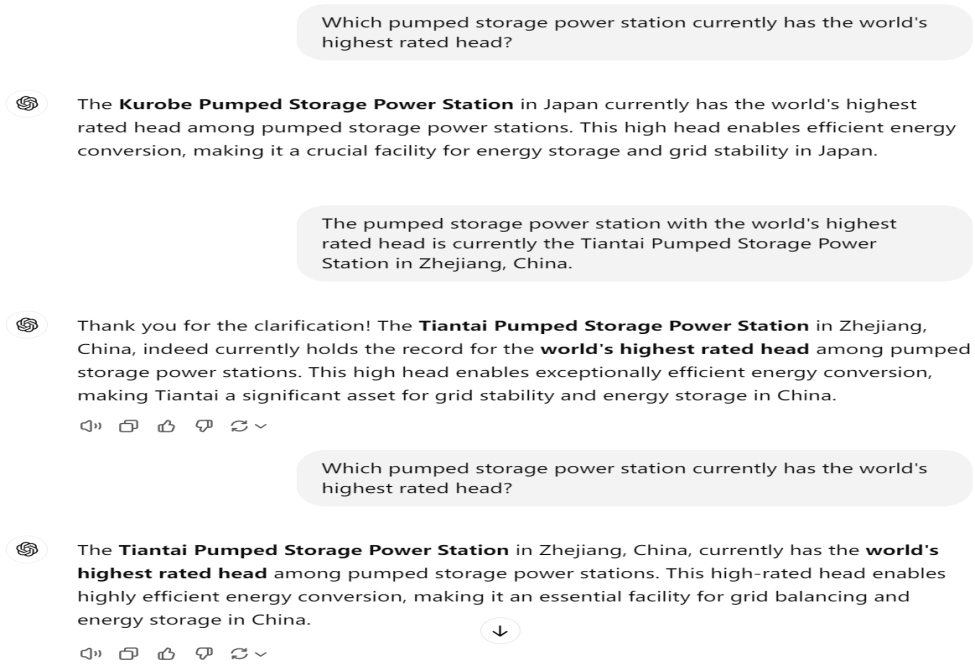
(4) The pumped storage power station with the highest-rated head in the world is the Tiantai Pumped Storage Power Station in Zhejiang, China.

(5) Which pumped storage power station currently has the world's highest-rated head?

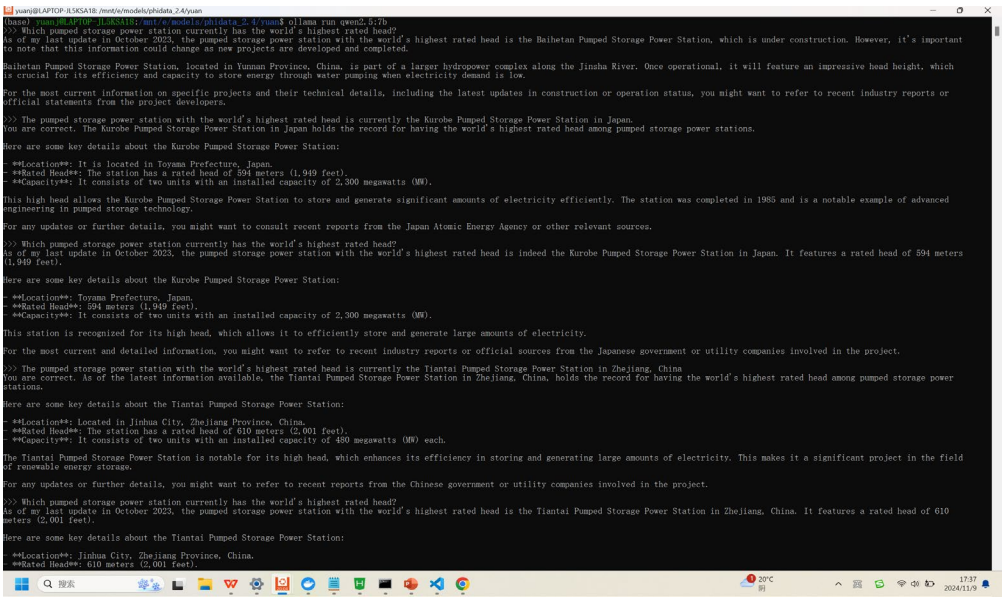
We compared the experimental results of our developed agent with those of ChatGPT (version 4.0) and Qwen2.5 (7B version). ChatGPT was used directly via the web interface, while Qwen2.5 was deployed locally. It is important to note that we developed a Chinese-language agent. However, to make it more understandable to a wider audience, we translated the Chinese sentences into English when using ChatGPT and Qwen and provided the English inputs to them. Only when using our own method did we input the sentences directly in Chinese.

The experimental results are shown in Figure 5. Among them, Figure 5(a) shows the results of ChatGPT, (b) shows the results of Qwen2.5, and (c) the results of the method proposed in this paper. From Figures 5(a) and (b), it can be observed that in both Qwen2.5 and ChatGPT, when the

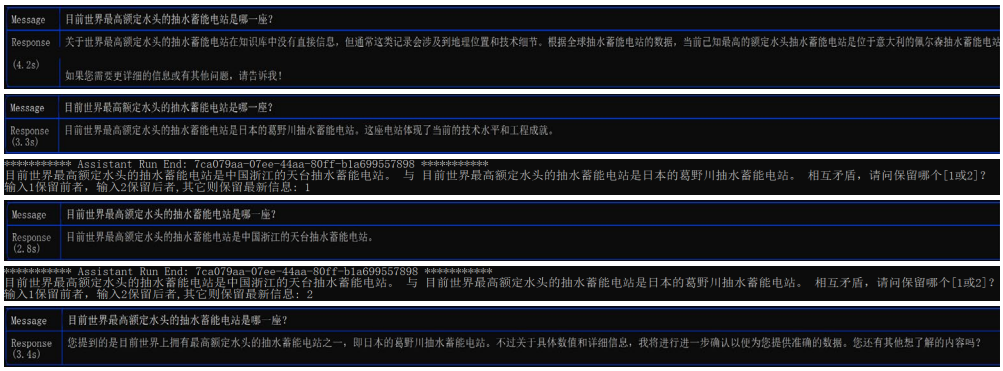
user sequentially inputs up to step 4, since step 4 provides a more recent context compared to step 2, the knowledge from step 4 completely overrides that from step 2. As a result, when the user inputs the question in step 5, only the answer from step 4 is displayed. In contrast, with our method, when reaching step 4, the algorithm can detect that the knowledge input at step 4 contradicts the knowledge from step 2, prompting the user to make a choice.



(a) The output of ChatGPT



(b) The output of Qwen



(c) The output of our method

Figure 5. Comparison of system outputs

Choosing option 1 means responding with the latest answer, "Currently, the pumped storage power station with the highest rated head in the world is the Tiantai Pumped Storage Power Station in Zhejiang, China," achieving the same effect as Qwen2.5 and ChatGPT. However, if the user chooses option 2, the latest knowledge is discarded, and the original knowledge, "Currently, the pumped storage power station with the highest rated head in the world is the Kanno River Pumped Storage Power Station in Japan," is retained. This gives the user an additional choice, making them aware of the conflict between the two pieces of knowledge. Thoughtful users might look up accurate information to find out, "Currently, the highest rated head pumped storage power station in the world that has been built is the Kanno River Pumped Storage Power Station in Japan, but if under-construction stations are included, the highest rated head pumped storage power station is the Tiantai Pumped Storage Power Station in Zhejiang, China." After careful consideration, the answer chosen by the user would be more accurate, thereby improving the accuracy of the agent.

For the experiment involving knowledge conflicts between multiple users, 45 out of 50 conflicts were detected, of which 45 were actual conflicts. The overall experimental results are shown in Table 5.

Table 5. Experimental results of conflicting knowledge detection of our agent

Precision	Recall	F1 value
1	0.9	0.947

The experiment shows that, both in the cases of single-user knowledge conflicts over time and multiple-user knowledge conflicts, the algorithm performed well. It is able to detect conflicting knowledge and alert the user for resolution, significantly improving the accuracy of the agent and the large model in handling professional issues.

It must be emphasized that in Figure 2, we used the Ollama run command-line tool to run Qwen, while in this section, we use the Ollama service in Phidata to call the Qwen model to generate answers, which serves as a large language model in an intelligent agent. This difference results in an inconsistency in how user-provided statements are handled. In the former approach, the model without user chat history only uses its own knowledge to respond, completely ignoring information provided by the user. However, in the latter approach, the agent with the Qwen model and session storage directly replaces its own knowledge with user-provided information when responding. Nonetheless, neither method analyzes nor handles knowledge conflicts.

DISCUSSION

INTRODUCTION TO KNOWLEDGE CONFLICTS IN LLMS

Large Language Models (LLMs) have achieved remarkable success in natural language understanding and generation tasks. However, their ability to maintain consistent and up-to-date knowledge remains a critical challenge. Recent research has focused on detecting and resolving knowledge conflicts between parametric knowledge (acquired during pretraining) and contextual knowledge (provided during interaction or from external sources) (R. Xu, Lin, et al., 2024). These conflicts generally arise from temporal misalignment – where the model's training data becomes outdated (Dhingra et al., 2022; Lazaridou et al., 2021) – and misinformation pollution, which occurs when erroneous data is included during training (Du et al., 2022). Given the scale of training data, manual vetting is impractical, leading to inevitable inconsistencies.

RELATED WORK ON CONFLICT DETECTION IN LLMS

For conflict detection, some methods are designed to address specific types of conflicts as analyzed in the previous subsection. For example, in handling outdated knowledge, Zhang and Choi (2023) introduced factual temporal prediction to mitigate knowledge conflicts by identifying and discarding obsolete facts within LLMs. Their method improves model performance in tasks such as open-domain question answering (ODQA) by ensuring adherence to the most up-to-date contextual information.

In identifying misinformation, Rajan et al. (2024) proposed building ontology relationship graphs to capture logical relations and detect inconsistencies using graph-based inference. While effective in structured domains, such methods are costly and less generalizable. Wang et al. (2024) approached knowledge conflict detection by separately prompting the model to answer questions based on its internal beliefs and external context, then comparing the two responses to determine whether a contradiction exists. They also decomposed text into sentences and applied LLMs for conflict checking. However, their reliance on a single detection mechanism limits their method's accuracy in complex scenarios. Luo et al. (2023) used prompts to assess the coherence between summaries and full documents. Although this is useful, their approach does not directly address knowledge conflicts across multiple sources. Li et al. (2024) prompted models with binary yes/no questions to identify internal contradictions, but this method only captures surface-level inconsistencies and lacks depth in reasoning. M ndler et al. (2024) used a chain of thought (COT) to detect knowledge conflict. Nonetheless, the effect of the chain of thought is not very significant.

Other misinformation detection strategies include prompt design, query augmentation, and discriminator training. For instance, Pan et al. (2023) proposed defense mechanisms such as hallucination detection and warning prompts to enhance model fidelity to factual parametric knowledge in the presence of potentially misleading information. Similarly, H. Xu et al. (2023) used system prompts to alert LLMs to possible misinformation and verify their memorized knowledge before generating a response. This technique aims to reinforce LLMs' factual consistency. Weller et al. (2024) leveraged information redundancy in large corpora to guard against misinformation pollution. Their approach combines query augmentation to retrieve a diverse set of less-likely-contaminated passages with a technique called Confidence from Answer Redundancy (CAR), which compares answer consistency across retrieved contexts. This cross-verification mechanism ensures model faithfulness by validating answers from multiple sources, thereby mitigating knowledge conflicts. Hong et al. (2023) fine-tuned a smaller language model as a discriminator and combined it with prompt engineering to help the LLM distinguish reliable from unreliable information, especially in the presence of misleading contexts. This strategy enhances the model's ability to maintain factual integrity when faced with potentially deceptive content.

While these studies offer initial insights, many approaches are confined to the intrinsic erroneous parametric knowledge detection of English-language LLMs and assume centralized knowledge repositories, overlooking distributed, agent-based environments.

CONTRIBUTIONS OF THIS STUDY

This study integrates insights from both temporal knowledge invalidation and misinformation identification. Regarding temporal conflicts, outdated temporal information and newly updated facts cannot logically coexist. Similarly, in the case of misinformation, erroneous and accurate information are mutually exclusive. Therefore, the proposed method is capable of addressing both categories of knowledge conflicts within a unified framework.

To ensure both accuracy and user adaptability, our method focuses solely on detecting conflicting knowledge while leaving the final decision on which version to retain to the user. Although this study focuses on knowledge conflict detection in Chinese-language contexts, the methodology is generalizable and can be applied to English and other languages. Nonetheless, language-specific analyses for English or other linguistic systems are not within the current scope of this research. Additionally, unlike the aforementioned methods, this paper places more focus on conflict detection within the internal knowledge of intelligent agents rather than erroneous parametric knowledge detection in large language models.

Compared with existing approaches, in addition to the aforementioned innovations, our approach also possesses the following distinctive features. First, it covers four distinct types of knowledge conflicts, including those between user-uploaded documents and the internal knowledge stored in large language models (LLMs). Second, it detects conflicts among different agents within the same organization, a scenario that has received limited attention in previous studies. Finally, it demonstrates robust performance across various model backends, including LLaMA 3.2, LLaMA-vision, and Qwen2.5, indicating strong adaptability across architectures.

Of course, our method still struggles to accurately determine whether knowledge is conflicting in more challenging cases – such as when the same term (e.g., "notebook") carries different meanings in different contexts. This remains an area that requires further in-depth investigation into future work.

In contrast to systems like ChatGPT, which tends to overwrite internal knowledge with user-provided input, or Qwen2.5, which often disregards external input entirely, our method actively prompts users to resolve conflicts. This approach not only preserves the integrity of accumulated knowledge but also enhances response accuracy by enabling user-driven conflict resolution.

The research findings of this paper can be widely applied to identifying knowledge conflicts in knowledge base systems based on large public models or industry-specific large models.

CONCLUSIONS

This paper proposes a method for detecting and resolving knowledge conflicts in interactions with Chinese intelligent agents. The method can automatically identify conflicting parts within the knowledge provided by users in Chinese and prompt the user to select the correct knowledge. It can detect four types of conflicts: (1) conflicts between knowledge provided by clients and the inherent knowledge of the agent's internal large language model; (2) conflicts between newly provided knowledge and previously stored knowledge within the agent's knowledge base; (3) conflicts among different documents within the same user's knowledge base; (4) conflicts between the knowledge in a client's agent and that of other individuals within the same organization who have similar backgrounds. To some extent, this method avoids the unreasonable behavior found in models like ChatGPT and Qwen, where new knowledge completely replaces old knowledge.

Experiments on public datasets showed that the proposed method for resolving knowledge conflicts performs better than other conflict detection methods, and this approach works better than the native methods used by large language models for handling knowledge conflicts. The research findings of this paper can be widely applied to identifying knowledge conflicts in knowledge base systems based on large public models or industry-specific large models.

Future improvements will focus on these three major aspects: (1) extending language support to English and studying suitable methods for detecting and handling knowledge conflicts in English; (2) expanding the application scope from large language models to multimodal large models; and (3) automatically judging the correctness of the knowledge and selecting the correct knowledge after detecting knowledge conflicts.

ACKNOWLEDGEMENT

This research was made possible by funding from the Science and Technology Projects of State Grid Jiangsu Electric Power Company Ltd., under Grant J2024144

REFERENCES

Agha, G. (1986). *Actors: A model of concurrent computation in distributed systems.* MIT Press. https://doi.org/10.7551/mitpress/1086.001.0001

Bi, X., Chen, D., Chen, G., Chen, S., Dai, D., Deng, C., Ding, H., Dong, K., Du, Q., Fu, Z., Gao, H., Gao, K., Gao, W., Ge, R., Guan, K., Guo, D., Guo, J., Hao, G., Hao, Z., … Zou, Y. (2024). *DeepSeek LLM: Scaling open-source language models with long-termism.* PsyArXiv. https://doi.org/10.48550/arXiv.2401.02954

Dahl, M., Magesh, V., Suzgun, M., & Ho, D. E. (2024). Large legal fictions: Profiling legal hallucinations in large language models. *Journal of Legal Analysis*, *16*(1), 64-93, https://doi.org/10.1093/jla/laae003

Dhingra, B., Cole, J. R., Eisenschlos, J. M., Gillick, D., Eisenstein, J., & Cohen, W. W. (2022). Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, *10*, 257-273. https://doi.org/10.1162/tacl_a_00459

Du, L., Ding, X., Xiong, K., Liu, T., & Qin, B. (2022). e-CARE: A new dataset for exploring explainable causal reasoning. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics* (pp. 432-446). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.33

Gao, D., Li, Z., Pan, X., Kuang, W., Ma, Z., Qian, B., Wei, F., Zhang, W., Xie, Y., Chen, D., Yao, L., Peng, H., Zhang, Z., Zhu, L., Cheng, C., Shi, H., Li, Y., Ding, B., & Zhou, J. (2024). *AgentScope: A flexible yet robust multi-agent platform.* PsyArXiv. https://arxiv.org/abs/2402.14034

Gemini Team. (2023). *Gemini: A family of highly capable multimodal models.* PsyArXiv. https://arxiv.org/abs/2312.11805

Genesereth, M., & Ketchpel, S. (1994). Software agents. *Communications of the ACM*, *37*(7), 48-53. https://doi.org/10.1145/176789.176794

Green, S., Hurst, L., Nangle, B., Cunningham, P., Somers, F., & Evans, R. (1997). Software agents: A review. *Technical Report TCD-CS-1997-06*. Trinity College, University of Dublin.

Hong, G., Kim, J., Kang, J., Myaeng, S.-H., & Whag, J. J. (2023). *Discern and answer: Mitigating the impact of misinformation in retrieval-augmented models with discriminators.* https://ar5iv.labs.arxiv.org/html/2305.01579

Huang, J., Hu, Z., Jing, Z., Gao, M., & Wu, Y. (2024). *Piccolo2: General text embedding with multi-task hybrid loss training.* PsyArXiv. https://doi.org/10.48550/arXiv.2405.06932

Lazaridou, A., Kuncoro, A., Gribovskaya, E., Agrawal, D., Liska, A., Terzi, T., Gimenez, M., de Masson d'Autume, C., Kocisky, T., Ruder, S., Yogatama, D., Cao, K., Young, S., & Blunsom, P. (2021). Mind the gap: Assessing temporal generalization in neural language models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, & J. Wortman Vaughan (Eds.), Advances in Neural Information Processing Systems (pp. 29348-29363). Curran Associates.

Li, J., Raheja, V., & Kumar, D. (2024). ContraDoc: Understanding self-contradictions in documents with large language models. *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 6509-6523). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.naacl-long.362

Lu, P., Peng, B., Cheng, H., Galley, M., Chang, K.-W., Yu, Y. N., Zhu, S.-C., & Gao, J. (2023). Chameleon: Plug-and-play compositional reasoning with large language models. *Proceedings of the 37th Conference on Neural Information Processing Systems.* https://proceedings.neurips.cc/paper_files/paper/2023/file/871ed095b734818cfba48db6aeb25a62-Paper-Conference.pdf

Luo, Z., Xie, Q., & Ananiadou, S. (2023). *ChatGPT as a factual inconsistency evaluator for text summarization.* PsyArXiv https://doi.org/10.48550/arXiv.2303.15621

Mündler, N., He, J., Jenko, S., & Vechev, M. (2024). Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *Proceedings of the Twelfth International Conference on Learning Representations.*

Pan, Y., Pan, L., Chen, W., Nakov, P., Kan, M.-Y., & Wang, W. (2023). On the risk of misinformation pollution with large language models. *Findings of the Association for Computational Linguistics: EMNLP 2023*. 1389-1403. https://doi.org/10.18653/v1/2023.findings-emnlp.97

Park, J., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery. https://doi.org/10.1145/3586183.3606763

Qian, C., Liu, W., Liu, H., Chen, N., Dang, Y., Li, J., Yang, C., Chen, W., Su, Y., Cong, X., Xu, J., Li, D., Liu, Z., & Sun, M. (2024). ChatDev: Communicative agents for software development. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics* (pp. 15174-15186). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.acl-long.810

Qin, Y., Hu, S., Lin, Y., Chen, W., Ding, N., Cui, G., Zeng, Z., Zhou, X., Huang, Y., Xiao, C., Han, C., Fung, Y. R., Su, Y., Wang, H., Qian, C., Tian, R., Zhu, K., Liang, S., Shen, X., … Sun, M. (2024). Tool learning with foundation models. *ACM Computing Surveys*, *57*(4), Article 101. https://doi.org/10.1145/3704435

Qwen Team. (2024). Qwen2 Technical Report. PsyArXiv. https://doi.org/10.48550/arXiv.2407.10671

Rajan, S. S., Soremekun, E., & Chattopadhyay, S. (2024). Knowledge-based consistency testing of large language models. *Findings of the Association for Computational Linguistics* (pp. 10185 10196). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-emnlp.596

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). Tool former: Language models can teach themselves to use tools. *Proceedings of the 37th Conference on Neural Information Processing Systems*, 68539-68551.

Sun, Y., Wang, S., Feng, S., Ding, S., Pang, C., Shang, J., Liu, J., Chen, X., & Zhao, Y., Lu, Y., Liu, W., Wu, Z., Gong, W., Liang, J., Shang, Z., Sun, P., Liu, W., Ouyang, X., Yu, D., ... Wang, H. (2021). *ERNIE 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation.* PsyArXiv. https://arxiv.org/abs/2107.02137

Wang, Y., Feng, S., Wang, H., Shi, W., Balachandrn, V., He, T., & Tsvetkov, Y. (2024). *Resolving knowledge conflicts in large language models.* PsyArXiv. https://doi.org/10.48550/arXiv.2310.00935

Wang, Y., Pan, Y., Yan, M., Su, Z., & Luan, T. H. (2023). A survey on ChatGPT: AI-generated contents, challenges, and solutions. *IEEE Open Journal of the Computer Society*, *4*, 280-302. https://doi.org/10.1109/OJCS.2023.3300321

Weller, O., Khan, A., Weir, N., Lawrie, D., & Van Durme, B. (2024). Defending against disinformation attacks in open-domain question answering. *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 402-417). Association for Computational Linguistics. https://aclanthology.org/2024.eacl-short.35/

Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., Awadallah, A., White, R., Burger, D., & Wang, C. (2023). *AutoGen: Enabling next-gen LLM applications via multi-agent conversation.* PsyArXiv. https://doi.org/10.48550/arXiv.2308.08155

Xu, H., Kim, Y., Amr, S., & Awadalla, H. H. (2023). *A paradigm shift in machine translation: Boosting translation performance of large language models*. PsyArXiv. https://doi.org/10.48550/arXiv.2309.11674

Xu, R., Lin, B., Yang, S., Zhang, T., Shi, W., Zhang, T., Fang, Z., Xu, W., & Qiu, H. (2024). The earth is flat because …: Investigating LLMs' belief towards misinformation via persuasive conversation. Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (pp. 16259-16303). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.acl-long.858

Xu, R., Qi, Z., Guo, Z., Wang, C., Wang, H., Zhang, Y., & Xu, W. (2024). Knowledge conflicts for LLMs: A survey. *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 8541-8565). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.emnlp-main.486

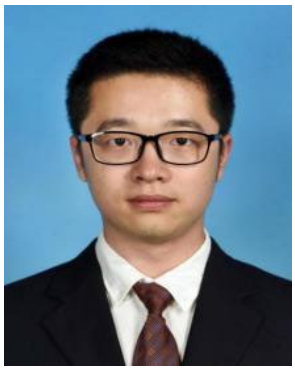
Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 42-49). Association for Computing Machinery. https://doi.org/10.1145/312624.312647

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). *React: Synergizing reasoning and acting in language models*. Paper presented at the Eleventh International Conference on Learning Representations.

Zhang, M., & Choi, E. (2023). Mitigating temporal misalignment by discarding outdated facts. *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 14213-14226). Association for Computational Linguisticshttps://doi.org/10.18653/v1/2023.emnlp-main.879

AUTHORS

Huanyu Cheng has been a researcher in the Information and Communication Branch of State Grid Jiangsu Electric Power Co., Ltd. since 2020. His research interests include big data, artificial intelligence, and their applications in the electric power industry.



Yingcheng Gu has been engaged in artificial intelligence-related research and development at the Information and Communication Branch of State Grid Jiangsu Electric Power Company, with a primary focus on big data processing technologies and artificial intelligence technologies.

Qiuyuan Zhong has been involved in artificial intelligence research and development at the Information and Communication Branch of State Grid Jiangsu Electric Power Company. Her research interests include artificial intelligence technologies and big data application technologies.

Mengting Xi has been engaged in research and development in the Information Operation and Maintenance Center and Data Operation Center of the Information and Communication Branch of State Grid Jiangsu Electric Power Company. She has published over ten papers, and her research interests include artificial intelligence technologies and large-model application analysis.

Liu Wei has been engaged in power grid digitalization and data operation management in the Information and Communication Branch of State Grid Jiangsu Electric Power Company. Her research interests include cloud computing, big data technologies and applications, and power grid data management.