iKnow: an Intent-Guided Chatbot for Cloud Operations with Retrieval-Augmented Generation

Junjie Huang[1], Yuedong Zhong[1], Guangba Yu[1], Zhihan Jiang[1], Minzhi Yan[2], Wenfei Luan[2], Tianyu Yang[2], Rui Ren[3], Michael R. Lyu[1]

[1]The Chinese University of Hong Kong, [2]HCC Lab, Huawei Cloud, [3]Computing and Networking Innovation Lab, Huawei Cloud

Read this experience paper!

ARISE Automated Reliable Intelligent Software Engineering

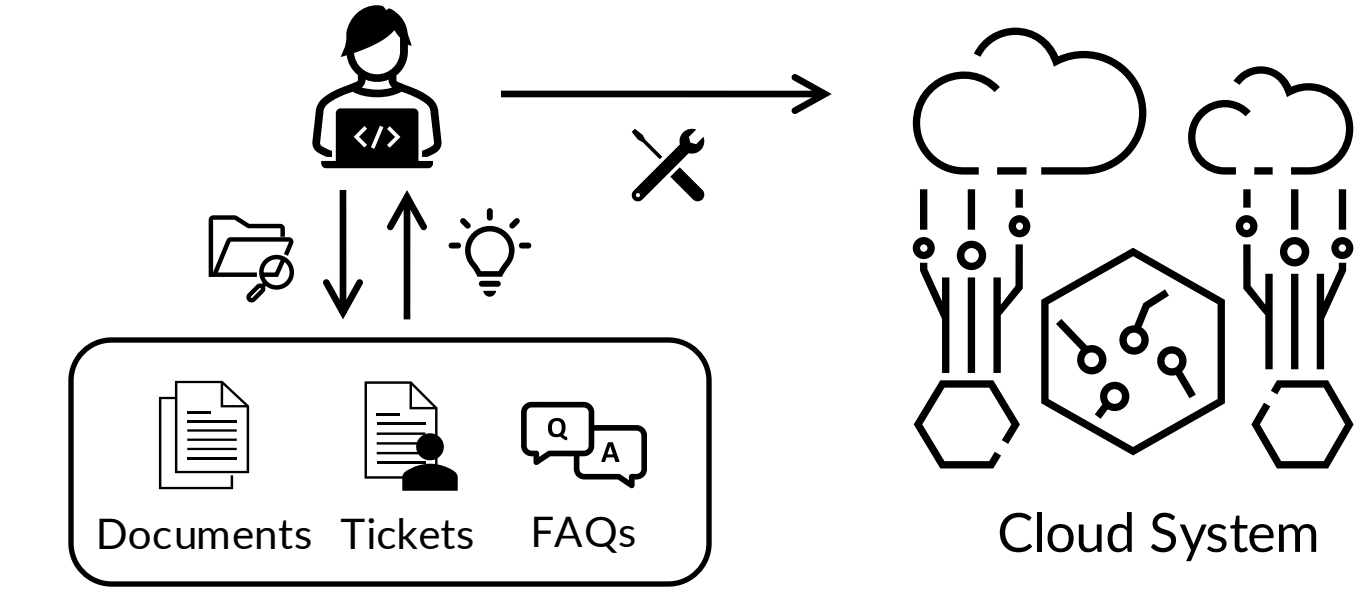
香港中文大學 The Chinese University of Hong Kong

HUAWEI

HUAWEI CLOUD

Reliable Cloud Operation Demands Information

Reliable Operation: Engineers need to check the docs and follow the procedures for operation → **Information Need**



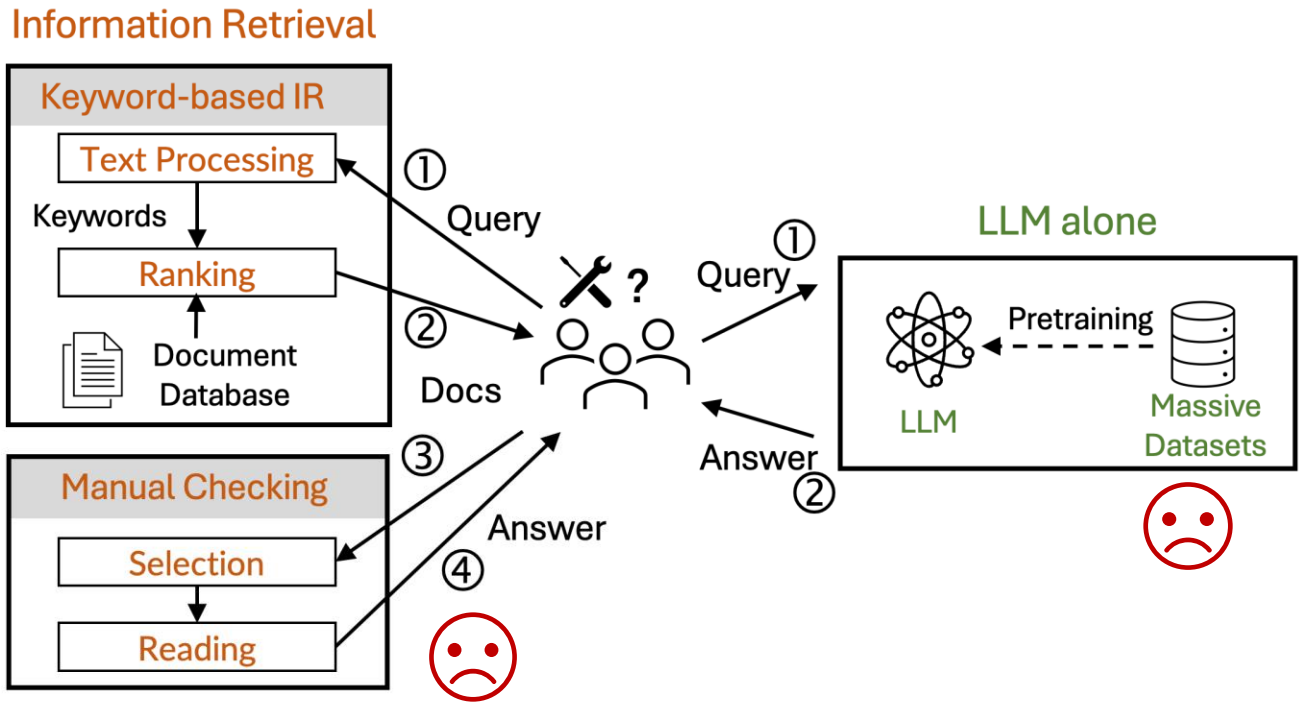
Comprehensive Documentation in Cloud Companies

IR or LLMs can provide information, but…

Traditional information retrieval (IR) system:

Input: a user query
Output: relevant documents.
Problems:

- Users still need to check manually to get the desired answer



Large language models (LLMs) alone:

Input: a user query.
Output: answer to the query.
Problems:

- **Hallucinations** can risk high-stakes operational environment
- **Confidential** documents are absent from the training data of LLMs

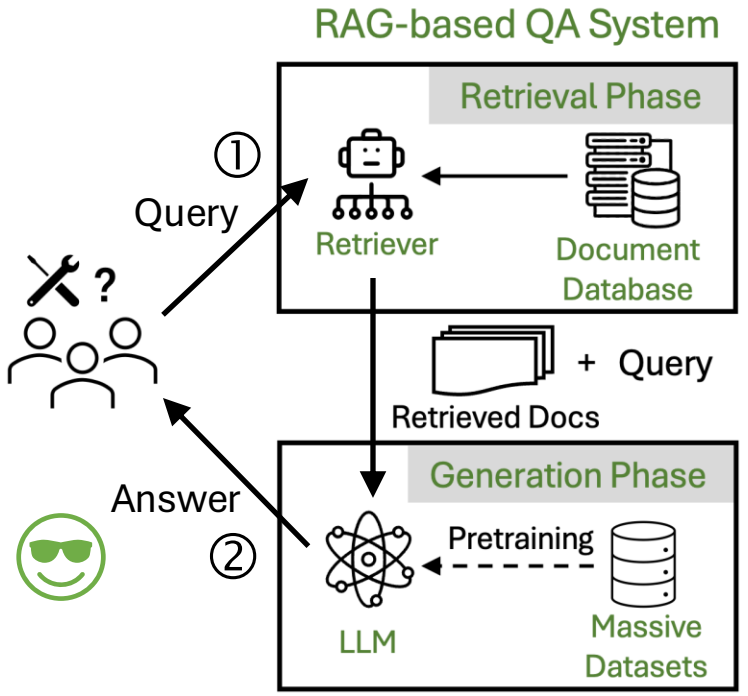
RAG for cloud operation question answering (OpsQA)

RAG: Retrieval-Augmented Generation

- A Retriever + A Generator

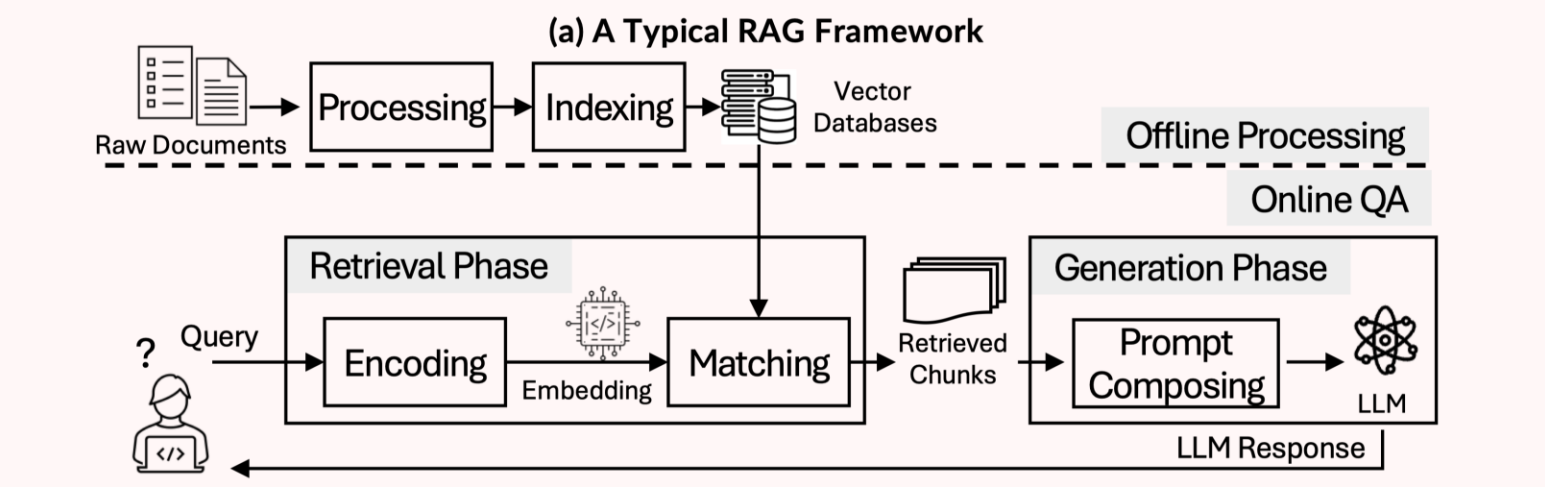
Advantage of RAG:

- Summarized answers with source reference



How to develop a RAG-based chatbot in OpsQA?

We built an RAG for OpsQA at Huawei Cloud



Diverse Documents

- Product manuals
- FAQs
- Incident tickets
- Failure libraries

Model Settings

- Retriever: BGE-M3
- VecDB: FAISS
- Reader: Qwen2.5-32B-Instruct

Settings

- Framework: langchain
- Chunk size: 100 tokens
- Top-k context: 5 docs

Naïve RAG still produce unsatisfactory outputs…

Initial Empirical Study

Research Questions (RQs):

- **RQ1**: What questions do cloud operators frequently ask?
- **RQ2**: How well can existing RAG-based chatbot answer these questions?
- **RQ3**: Why does existing RAG-based chatbot fail?

RQ	Study Factor	Description
RQ1	Query Intent	What is the information needed?
RQ2	Answer Correctness	What are the requirements of answers? Does the answer satisfy the requirements?
RQ3	Root Cause	What causes the incorrect answer?

Study Methods: Open Coding

- **Data:** 2000 queries and 3 datasets.
- **Inter-annotator agreement**: Krippendorf's alpha

	# Query	Document Source	Data Type	# Words	User Group
$\mathcal{A}$	300	Product docs, group Wikis, FAQ	Text, PDF, HTML	~1.2 M	External Ops Team
$\mathcal{B}$	1,350	Product docs, group Wikis, FAQ	Text, PDF	~0.6 M	Internal Ops Team
$\mathcal{C}$	350	Incident ticket, fault library, FAQ	Text, PDF, Figure	~0.4 M	On Call Team

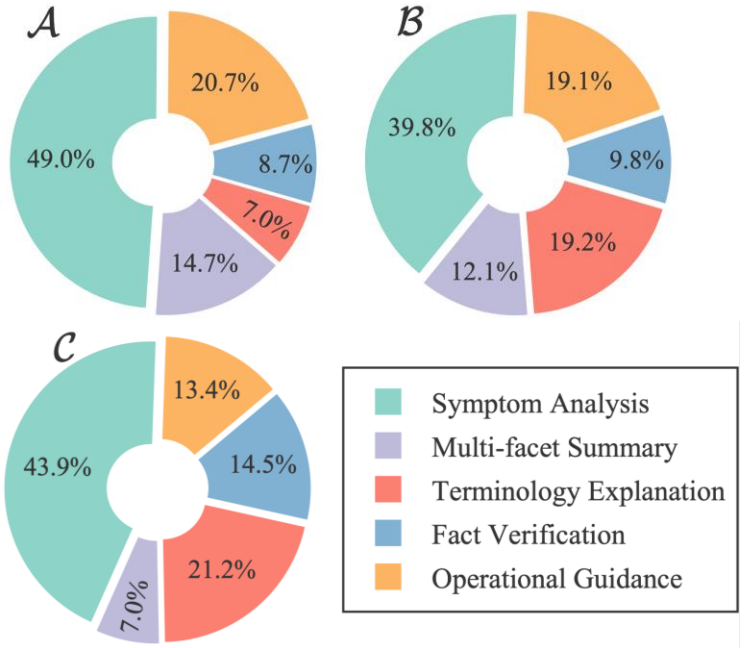
RQ1: What questions do cloud operators frequently ask?

Findings:

- Operational queries have **five** intent types, differing general programmers' questions [1, 2]
- Symptom analysis is the most common intent (avg. 40.6%)
- Different teams have similar information need, highlighting the generalizability of intent taxonomy.

Intent	Query Manifestation	Query Examples
Symptom Analysis	Describes observable or ambiguous system behaviors and anomalies.	"No performance data for VM disk usage?" "App Orchestration error: Slave has no response."
Multi-facet Summary	Solicits a comprehensive or summarized overview of operational concepts or issues, often using high-level terms.	"Flink troubleshooting summary?" "FAQ for SystemA backup?" "Comprehensive guide to $\mathcal{S}$ production issues"
Terminology Explanation	Asks for the meaning of a technical term, usually short or abbreviated.	"ESN?" "SystemA private key?" "Operator login password policy?"
Fact Verification	Seeks specific factual details, often posed as yes/no or what questions.	"Does ServiceA support capacity statistics?" "Can roles of existing users be modified?"
Operational Guidance	Requests step-by-step instructions for an action, often as a "how-to" question or verb–noun phrase.	"Check tenant operation logs?" "Add MySQL to access whitelist?" "How to request quota for the current VDC?"

Intent taxonomy of real user queries in OpsQA



Query distribution of different intents

[1] How do programmers ask and answer questions on the web? In *ICSE-NEIR'11*
[2] A manual categorization of android app development issues on stack overflow. In *ICSME'14*

RQ2: How well can RAG-based chatbot answer questions

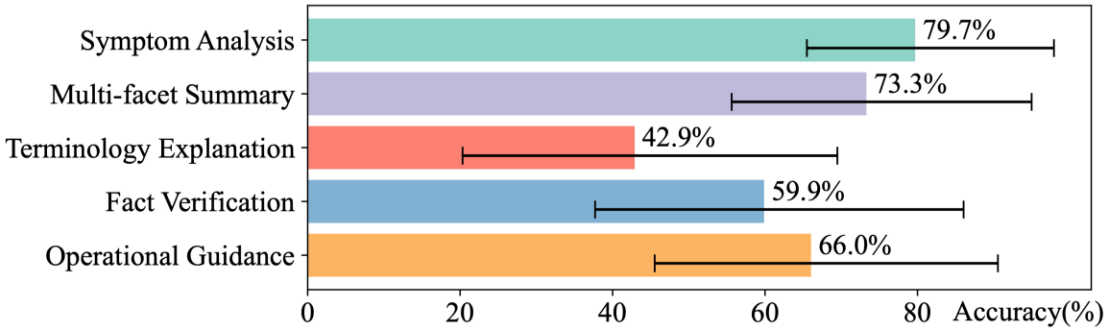
Findings:

- Queries of different intents have distinct answer requirements, leading to intent-specific failures.
- RAG varies in performance in OpsQA, struggling with factual and terminology-related queries.

Intent	Answer Requirement	Symptoms of Failed Answer
Symptom Analysis	Diagnoses the issue, identifies root cause, and provides actionable suggestions, grounded in the retrieved sources.	**Hallucination**: Fabricates causes or suggestions not in sources. **Intent-conflicting**: Misidentifies or ignores the issue. **Overly Generic**: Lacks concrete diagnosis or actionable steps. **Deficiency**: Provides misleading or incomplete analysis.
Multi-facet Summary	Aggregates and synthesizes relevant aspects to deliver a comprehensive overview, highlighting key perspectives.	**Hallucination**: Fabricates aspects not supported by sources. **Intent-conflicting**: Summarizes non-targeted concepts. **Overly Generic**: Offers vague summary without specifics. **Deficiency**: Omits key dimensions or offers misleading synthesis.
Terminology Explanation	Provides a clear and accurate definition of the targeted term with relevant details, as supported by sources.	**Hallucination**: Fabricates definitions unsupported by sources. **Intent-conflicting**: Explains unrelated concepts. **Overly Generic**: Gives vague or overly brief definitions. **Deficiency**: Gives incorrect or incomplete definitions.
Fact Verification	Explicitly confirms or refutes the queried fact with explanations and supporting evidence from the sources.	**Hallucination**: States facts not grounded in sources. **Intent-conflicting**: Fails to directly answer the question. **Overly Generic**: Provides vague explanations for the verification. **Deficiency**: Gives incorrect or incomplete explanations.
Operational Guidance	Delivers thorough, step-by-step instructions with actionable details, ensuring completeness and practical applicability.	**Hallucination**: Suggests unverified or incorrect procedures. **Intent-conflicting**: Fails to provide steps or targets wrong task. **Overly Generic**: Lacks concrete actions or operational details. **Deficiency**: Misses critical steps or details.

Intent-specific answer criteria

Need intent-specific improvements?



Response accuracy of different intent types

RQ3: Why does existing RAG-based chatbot fail?

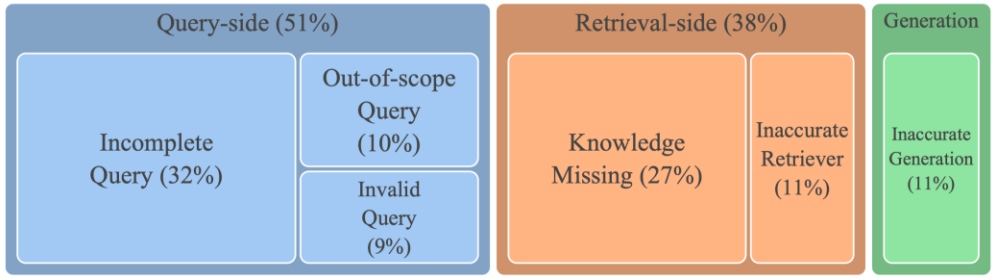
Findings:

- 51% failures originate from user query issues (e.g. incomplete query 32%). → Need clarification
- 27% root causes are knowledge missing (the second largest) → Need gap detection

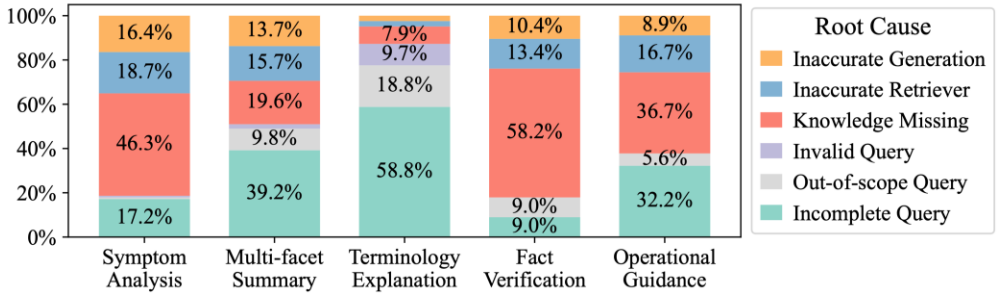
Improve query clarity and detect knowledge gaps

Root Cause	Definition	Example
Incomplete Query	The query lacks sufficient context or specificity, often containing only a noun/term without signal words (*e.g.*, what, why, how), making the intent ambiguous.	"ESN?"; "Snapshot Residue"; "SSL_ERROR_SYSCALL"
Out-of-Scope Query	The query is unrelated to IT operations or outside the domain covered by the system.	"What's the financial status of customerA?"
Invalid Query	The query contains nonsensical text, corrupted input, or severe misspellings, rendering it unanswerable.	"sdfa"; "Snaphot Resdieu"
Knowledge Missing	The query is relevant and well-formed, but the required information is absent from the document corpus, resulting in no retrievable evidence.	Asking about a newly released feature not yet documented in the knowledge base.
Inaccurate Retriever	The retriever fails to return all necessary documents due to model or data preparation issues (*e.g.*, improper chunking, poor embeddings, poor ranking methods).	Relevant documents exist in corpus but are not included in the top-k retrieved chunks.
Inaccurate Generation	The model generates incorrect or unsupported output despite receiving relevant context, due to decoding, prompt, or alignment issues (*e.g.*, misunderstanding terms or instructions, omitting details).	Misinterpreting "indicator" as a general term, or providing a generic summary when a step-by-step guide was requested.

Main root causes of OpsQA failures



Overall failure distribution

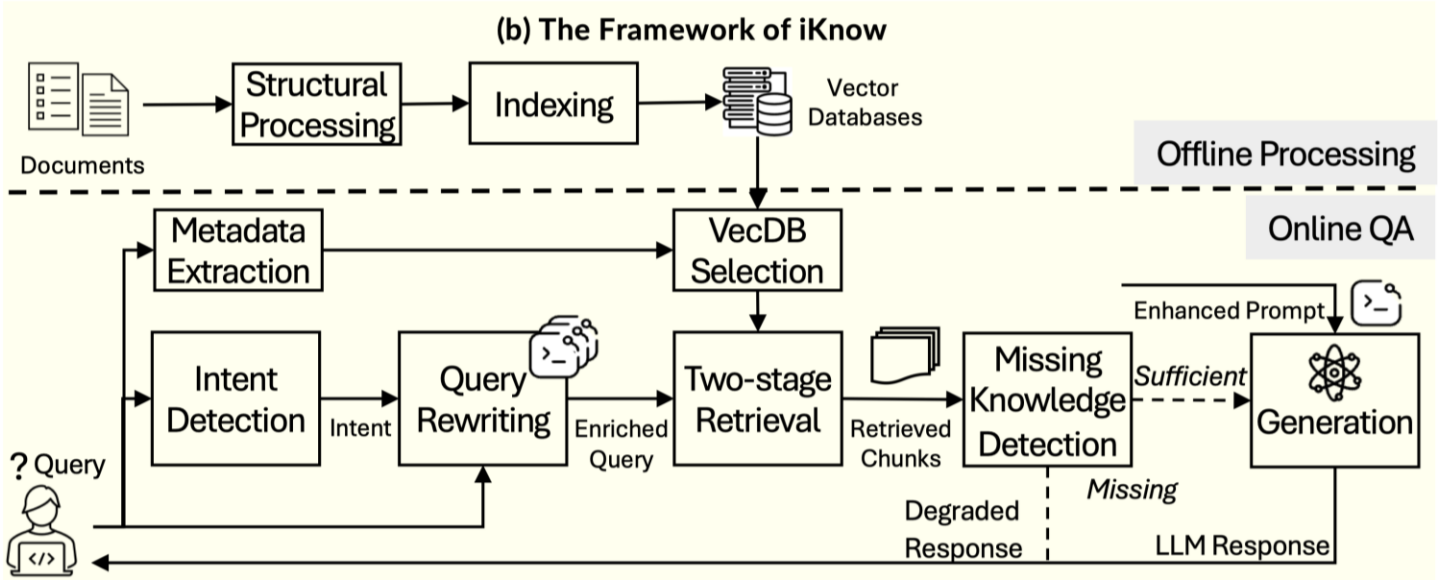


Failure distribution of varying intents

iKnow: an Intent-aware OpsQA RAG System

Improvements in iKnow:

- **Intent Detection**: Prototypical Network for efficient classification
- **Query Rewriting:** LLMs guided by intent-specific requirements
- **Missing Knowledge Detection**: LLMs to check knowledge existence



The framework of iKnow

RQ4: How effective is iKnow in OpsQA?

Findings:

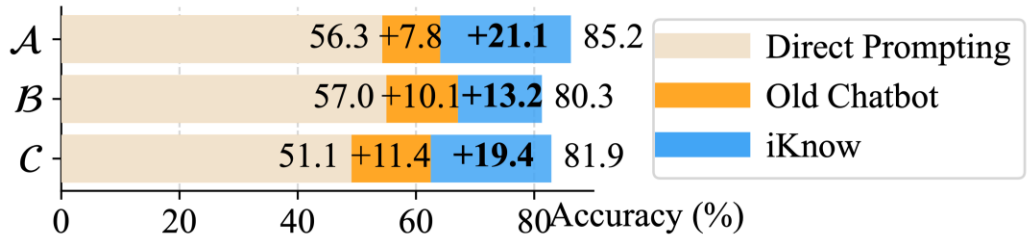
- iKnow outperforms direct prompting and naïve RAG.
- iKnow shows consistent improvement across intents.

Error Analysis:

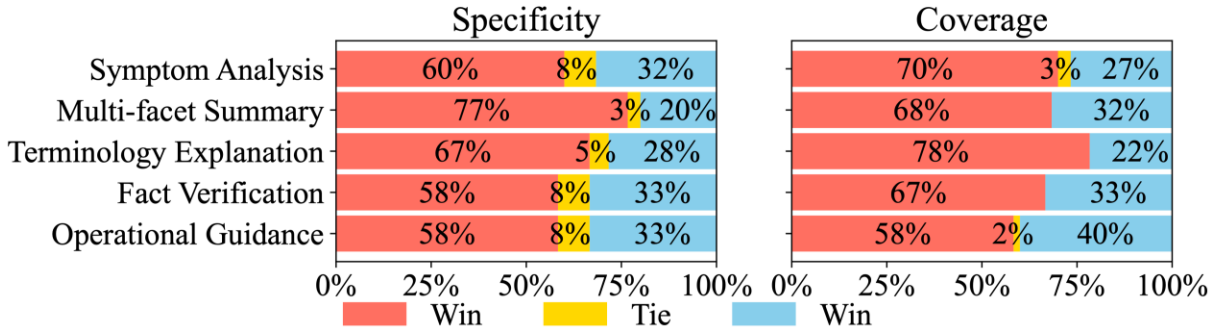
- Intent detection errors (23%)
- Retrieval errors (12.6%)
- Missing knowledge (37.2%)
- LLM judge mislabelling (16.6%)
- Other errors (9.6%)

Evaluation Settings

- LLM-as-a-judge accuracy
- Pairwise win-tie-loss evaluation
  - **Specificity**: amount of precise, concrete information provided
  - **Coverage**: degree to which all relevant aspects addressed



The LLM judge accuracy of iKnow and the old chatbot



Pairwise accuracy of iKnow and the old chatbot

RQ5 & RQ6: Ablation study and efficiency

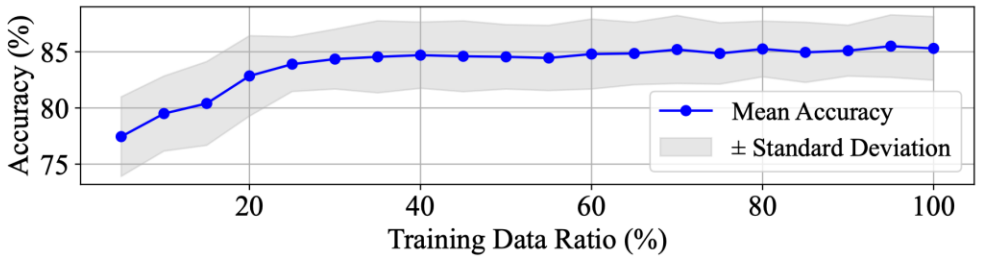
RQ5: Effects of different components

- **Intent detection**: Acc. 85.3%
- **Query rewriting**: 99% queries are enhanced
- **Missing knowledge detection**: F1 88.5%

RQ6: Efficiency of iKnow in OpsQA?

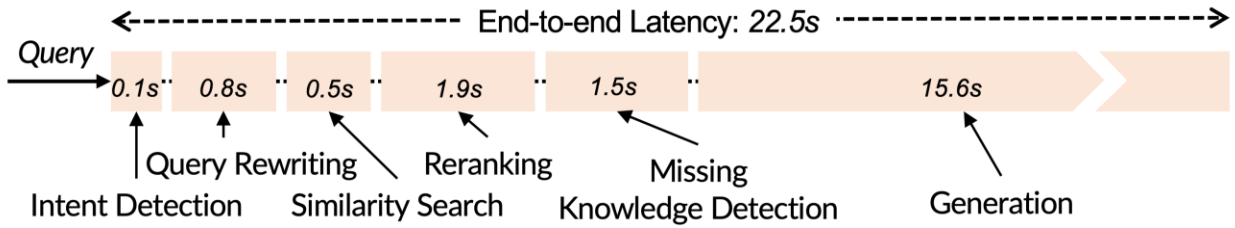
- **Acceptable time cost**: 4.3s cost by introduced modules, accounting for 19.1%

All components are useful!



Accuracy of intent detection

iKnow is efficient in OpsQA!



End-to-end and component-wise latency of iKnow

Lessons learned

For chatbot users

- Formulating a **complete question** with explicit information need and specific details is of great importance.
- Be aware of **knowledge boundary** of chatbots and ask relevant questions.
- Exercise caution and **verify** critical responses.

For OpsQA chatbot provider

- Be aware of deployment challenges and approach to **monitoring data**.
- OpsQA chatbot providers can benefit from integrating **intent detection and query rewriting**.
- Including **links to reference** documents in the answer enhances transparency.

For researchers

- Synthesizing domain-specific QA datasets for customized **training**.
- Incorporating **real-time monitoring data** for symptom analysis.
- Aggregating evidence for **multi-perspective questions**, e.g., GraphRAG

Conclusion

- **Empirical Analysis on OpsQA**:
  - Intents of OpsQA queries.
  - Answer requirements of different intents.
  - Root causes of failed answers.

- **iKnow: an intent-guided OpsQA RAG system**
  - Both *accurate* and *efficient*!
  - We share lessons learned through our experience.

Paper

Q & A