JOURNAL OF INFORMATION SCIENCE AND ENGINEERING XX, XXXX-XXXX (2024)
DOI: XX.XXX.XXXX.XX

Enhancing Traditional Chinese Enterprise Customer Service with a Hybrid Framework

YEN-WU TI[1], WUN-HAO WANG[2], TIAN-SHYR DAI[2]+, CI-SIANG HUANG[3] AND HSIN-YUN CHANG[4]
[1]*Department of Computer Science and Information Engineering*
Ming Chuan University, Taoyuan, 333321, Taiwan
E-mail: yenwuti@mail.mcu.edu.tw
[2]*Department of Information Management and Finance*
[3]*Department of Computer Science*
National Yang Ming Chiao Tung University, Hsinchu, 30093, Taiwan
E-mail: cameldai@nycu.edu.tw (T.-S. Dai); huanglin214@gmail.com (C.-S. Huang)
[4]*MetaEdge Corporation Taiwan, Taipei, 105407, Taiwan*

Deploying large language models (LLMs) for customer service automation in resource-constrained Traditional Chinese enterprises faces critical challenges: computational limitations, data privacy concerns, and linguistic misalignment specific to Traditional Chinese processing. This study presents a hybrid Zero-Shot Retrieval-Augmented Generation (RAG) framework designed for this environment, integrating keyword-based (BM25) and semantic retrieval models (chuxin-embedding and the LdIR-Qwen2 reranker) with the TAME-8B LLM. The framework is developed with a provider of regulatory-reporting software whose users are commercial banks in Taiwan, and the service cases concern statutory report filing to the financial regulator and the national credit bureau. Evaluated with 93 expert-annotated queries against a datastore of 964 historical service cases, the first stage identifies the originating case among all 964 with 100% top-1 accuracy, an upper bound since the queries were written with the resolving entries in view, and in our original experiments the second stage retains more than 95% of annotated reference snippets at TOP-7 while reducing the response text passed to the generator by 30.46%. Adding structured guiding words produces the largest changes in the overlap-based generation metrics among the factors we varied, raising them when an explicit system prompt is present and lowering them when it is absent. The findings support the feasibility of this expert system in a single-enterprise pilot-scale validation for Traditional Chinese customer service automation, and state the conditions under which it is effective.

Keywords: Expert Systems, Customer Service Automation, Retrieval-Augmented Generation (RAG), Traditional Chinese NLP, Hybrid Information Retrieval, Regulatory Reporting

1. INTRODUCTION

Traditional customer service models struggle to meet demands for instant, accurate responses. While Large Language Models (LLMs) have shown promise in automation—with successful deployments by companies like Salesforce [1], AT&T [2], and Bank of America [3]—their application to Traditional Chinese enterprise environments faces critical challenges. Most LLMs are trained on Simplified Chinese corpora [4], leading to grammatical errors [5] and unnatural responses when processing Traditional Chinese text [6].

+Corresponding author

Enterprise customer service systems possess a unique advantage: extensive historical service records documenting how agents have successfully resolved similar customer inquiries [7]. These records contain valuable domain-specific knowledge, standard operating procedures, and proven response patterns that are often absent from general-purpose LLM training data [8]. However, effectively leveraging this historical knowledge presents significant challenges: (1) service records are typically lengthy documents containing multiple interactions, requiring precise extraction of relevant segments [9]; (2) responses must maintain chronological coherence to reflect the problem-solving process; and (3) Traditional Chinese linguistic nuances demand specialized retrieval mechanisms beyond simple keyword matching [10].

While retrieval-augmented generation (RAG) [11] has emerged as a promising approach to incorporate external knowledge into LLMs, existing RAG systems face three critical limitations when applied to Traditional Chinese customer service: First, most retrieval models are optimized for Simplified Chinese or English, struggling with Traditional Chinese semantic patterns. Second, standard RAG approaches retrieve entire documents rather than relevant snippets, introducing noise and exceeding context window limits. Third, conventional retrieval methods ignore the temporal structure of multi-turn service interactions, losing important contextual dependencies.

To address these challenges, this study proposes a hybrid framework that effectively leverages historical service records for Traditional Chinese customer service automation, making three key contributions:

First, we develop a two-stage retrieval architecture that combines BM25 [12], chuxin-embedding [13], and the LdIR-Qwen2 reranker [14] through a weighted combination of normalised relevance scores. Section 4 evaluates whether this combination improves retrieval over the individual models, and finds that the answer depends on the retrieval depth.

Second, we introduce a Time-Sorted Snippet Ranking Algorithm that preserves the chronological structure of customer service interactions, maintaining temporal integrity and contextual coherence in response generation. Combined with TAME (Taiwan Mixture of Experts) 8B [10]—a language model fine-tuned specifically for Traditional Chinese operating with full precision (FP32)—this ordering aims to keep responses grounded and chronologically coherent (Section 3.3); its effect on the metrics is mixed (Section 4).

Third, we implement a sampling-based response generation strategy inspired by Cappy [15] that produces multiple candidate responses and employs the LdIR-Qwen2 reranker to select the most appropriate output. This mechanism is intended to reduce the risk of returning an irrelevant or internally inconsistent sample; we did not run an ablation isolating its effect.

Evaluated on 93 expert-annotated queries against the 964-case datastore, the framework retains 95.7% of the annotated reference snippets at TOP-7 retrieval while removing 30.46% of the response text in our original experiments, and in the adopted configuration reaches a BERT Score of 0.84 and ROUGE-L of 0.70, against 0.83 and 0.67 with document-level retrieval, differences small relative to their dispersion. We report these results as a single-enterprise pilot validation; the scope of what they do and do not establish is stated in Section 3.1 and revisited in the Limitations subsection of Section 5.

2. Literature Review

Large Language Models and Retrieval-Augmented Generation. Modern language models build on the Transformer architecture, whose attention mechanism processes sequences in parallel and captures long-range semantic dependencies [16]. Text is tokenized with subword schemes such as Byte-Pair Encoding [17] and mapped to dense vectors by embedding methods including Word2Vec [18], FastText [19], and GloVe [20]; most deployed models accept inputs of roughly 4,000 to 8,000 tokens [21]. Encoder-based models such as BERT [22] dominate understanding tasks, decoder-based models such as GPT [23] dominate autoregressive generation, and alignment with human intent is refined through reinforcement learning methods such as PPO [24].

Retrieval-Augmented Generation (RAG) extends these models with external knowledge at inference time and without retraining. Unlike pure in-context learning [25, 26], a RAG system retrieves relevant material from an external store, which matters in enterprise settings where the operative knowledge is internal and absent from any pre-training corpus. REPLUG [27], for instance, retrieves several documents and combines them with the input to form the token distributions used for sampling. System quality therefore rests on retrieval quality: lexical methods such as BM25 [12] match surface forms, neural retrievers match semantics, and hybrid combinations of the two generally outperform either alone, particularly in cross-lingual and domain-specific scenarios.

For Chinese, C-Pack [28] released the BGE embedding family together with C-MTEB, now the standard benchmark for Chinese retrieval, and BGE-M3 [29] extends it across more than a hundred languages with dense, sparse and multi-vector retrieval in one model. Reranking has moved in parallel: Sun et al. [30] show that a suitably instructed large language model can match supervised rerankers, and that the ability distils into smaller models. The LdIR-Qwen2 reranker we adopt is an LLM-backbone cross-encoder and reports 70.45 on the C-MTEB reranking tasks. Our contribution is accordingly not a new retriever but the decomposition of a multi-turn service record into a two-stage retrieval problem.

Prompting and Parameter-Efficient Adaptation. Where retrieval supplies knowledge, prompting governs how that knowledge is used [31]. Few-shot prompting improves performance on previously unseen tasks using only a handful of in-prompt examples [25], with roughly three examples often sufficient [26], while chain-of-thought prompting [32] and its zero-shot variant [33] elicit explicit step-by-step reasoning. When prompting is insufficient, parameter-efficient fine-tuning adapts a model at a fraction of the cost of full fine-tuning: adapter modules train 0.5% to 8% of the original parameters [34], prompt tuning learns virtual tokens [35], LoRA decomposes weight updates into low-rank factors [36], and QLoRA adds quantization to reduce memory further [37].

Traditional Chinese Language Models. Although tens of millions of people use Traditional Chinese in daily communication, most large models are trained predominantly on Simplified Chinese corpora. The divergence is not a character mapping: grammatical patterns, idiomatic expressions, and cultural references differ, so models trained on Simplified text mis-parse Traditional input, misread grammatical structure, and produce unnatural responses. Taiwan LLM [10] addresses this gap directly through culturally aligned language processing, and Chinese retrieval models such as chuxin-embedding [13] and the LdIR-Qwen2 reranker [14] are trained predominantly on Simplified Chinese,

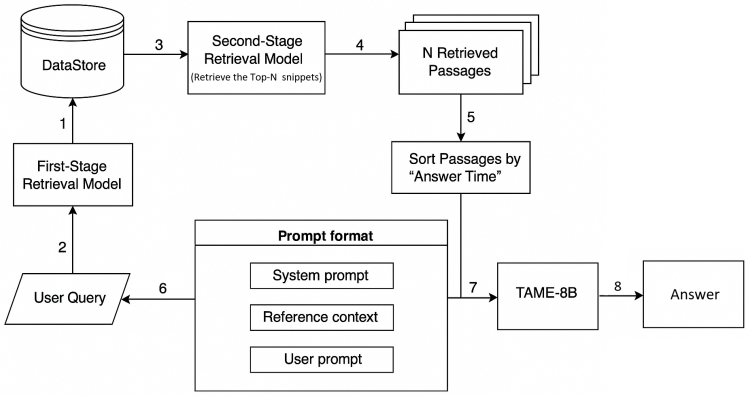


Fig. 1. The complete workflow of the proposed hybrid retrieval-augmented generation framework for Traditional Chinese customer service.

and their effectiveness on Traditional Chinese input varies with the task and the corpus, as Section 4 shows. Capturing the semantic nuance and cultural context of Traditional Chinese users remains an open problem [5].

LLM Applications in Customer Service. Applications of LLMs to customer service have attracted sustained attention [38], and several domain-specific Chinese systems illustrate the design space. ChatLaw [39] couples knowledge graphs with a mixture-of-experts model in a sensing–thinking–acting loop to suppress hallucination in legal consultation. Ma et al. [7] combined ChatGLM2-6B with the LangChain framework for telecommunications support, adding a knowledge decomposer that segments and structures the knowledge base. Wang et al. [8] adapted models to business scenarios through non-negative matrix factorization and low-rank adaptation, demonstrated on medical advisory systems. Wu et al. [40] integrated ChatGPT with specialized tourism databases to deliver region-specific answers, and Xie et al. [41] built an insurance service system from over 1,000 recorded sales interactions using dynamic prompting.

These systems share three limitations in the setting we address. They are optimized for Simplified Chinese or English, so their retrieval mechanisms do not capture the linguistic nuances of Traditional Chinese usage. They retrieve whole documents rather than the passages that answer the question, which injects noise and consumes the context window. And they treat a service record as an unordered pool of text, ignoring the chronological structure through which a multi-turn case actually reaches its resolution.

3. Model Framework

To address these limitations, we propose a hybrid retrieval-augmented generation framework specifically designed for Traditional Chinese enterprise customer service systems. The framework tackles three critical challenges: (1) ineffective retrieval mechanisms for Traditional Chinese linguistic patterns, (2) noise and redundancy from document-level retrieval, and (3) loss of temporal coherence in multi-turn interactions.

Figure 1 illustrates the complete system architecture. The workflow begins when a

user submits a query (②), which triggers a *two-stage retrieval process*. In the **first stage** (occurring between steps ② and ③), BM25 identifies relevant cases from the DataStore (①)—which contains historical service records with case-level aggregated text—by matching the query against complete case descriptions. The retrieved cases are then passed to the **second stage** (③), where a hybrid retrieval model (combining BM25, Chuxin, and Qwen) extracts precise snippets from the response content field and returns the top N most relevant passages (④). These snippets are sorted chronologically by their original "Answer Time" (⑤) to preserve temporal coherence, then combined with the user query (⑥) into a structured prompt format containing the system prompt, reference context, and user prompt. Finally, the TAME-8B language model (⑦) processes this structured input to generate the response (⑧).

Three design elements characterize the framework: *two-stage retrieval*, which filters at both case and snippet levels; the *combination of lexical and semantic scorers* used in the implementation reported here; and *temporal ordering* of the retrieved snippets, which preserves the chronology of a multi-turn interaction.

3.1 Dataset Format

Source and domain. The records are supplied by a software provider whose regulatory-reporting platform is operated by commercial banks in Taiwan; the cases concern the filing of statutory reports to the financial regulator and the national credit bureau, among them capital-adequacy returns, outstanding-credit monthly reports and overseas-exposure filings. Two properties of the domain shape the retrieval problem: the terminology is narrow and statutory, combining platform-specific operations with regulatory vocabulary absent from general-purpose corpora, and an incorrect answer carries a compliance cost rather than an inconvenience, so retrieval completeness is weighted above brevity throughout. All interactions are in Traditional Chinese through a ticket-based workflow.

Case structure. Each case carries a *Summary*, the *Customer-reported Content*, the *Response Content* holding an ordered sequence of response entries, and an *Answer Time* for each entry. Entries take one of three statuses: *internal analysis*, never sent to the customer; *external release*, messages issued to the customer; and *customer reply*. A case therefore interleaves customer-facing solutions with internal deliberation and administrative closure, so retrieving whole cases necessarily surfaces content that is irrelevant, or unsuitable, for customer-facing generation.

Construction. The issue-tracking system exported 974 cases; after records with missing required fields were discarded, **964** remained, and this is the corpus over which all retrieval reported here is performed. A case record is long—its concatenated fields average 2,394 characters—which is why the second stage extracts individual entries rather than passing the whole case to the language model.

Staff of the partner enterprise then constructed the evaluation set: for each entry they wrote the question in the form a customer would ask it, recorded the expected answer, and marked the response entries of the originating case that carried it. We call the marked entries *reference snippets* and the remaining entries of the same case non-reference candidates. The set comprises **93 queries** drawn from 77 source cases, since one case may give rise to more than one question, and **130 reference snippets**, an average of 1.40 per query with a maximum of 10; 77 of the 93 queries are answered by a single entry. Every reference snippet is drawn verbatim from its own case, so the ground truth is directly

comparable with what the retrieval stage returns. Section 4.2.1 examines how this construction affects the first-stage result.

De-identification. The raw records name bank staff and service agents and identify the client institutions. Personal identifiers were replaced with role placeholders and institution names with generic labels before any processing; no identifying information enters the retrieval index, the prompts, or any reported result. The dataset is not publicly released.

Scope of validation. All results below come from this single-enterprise, single-domain, text-only Traditional Chinese corpus, and the size of the annotated set is bounded by the cost of expert annotation rather than by data availability. We present the findings as a pilot-scale validation and state the limits on generalisation in the Limitations subsection of Section 5.

To support our two-stage retrieval architecture, we construct two complementary datasets from these historical records:

Case-Level Dataset: For the first retrieval stage, we consolidate all response entries for each case by chronologically ordering multiple "Response Content" fields based on their timestamps and merging them into a single text using newline delimiters. This creates a complete case representation preserving the temporal sequence of the entire service interaction.

Snippet-Level Dataset: For the second retrieval stage, we create individual records for each response entry, preserving the "Summary," "Customer-reported Content," and individual "Response Content" fields. This granular structure enables precise extraction of relevant response segments, minimizing redundancy while maintaining sufficient context.

3.2 Two-Stage Retrieval Framework

Our retrieval architecture employs a two-stage coarse-to-fine strategy that progressively narrows the search scope from complete cases to specific response snippets, as illustrated in Figure 1. The first stage (steps ② → ③) performs case-level retrieval to identify documents containing relevant information, while the second stage (step ③ → ④) performs snippet-level extraction to obtain precise, actionable content. This hierarchical approach addresses two key challenges: (1) efficiently identifying relevant cases from a large database, and (2) extracting precise information while minimizing noise and redundancy.

Stage 1: Case-Level Retrieval In the first stage, we identify the relevant case among all 964 cases in the historical service datastore using BM25 [12]. Each case is represented by concatenating three fields—summary, customer-reported content, and response content—into a single text using newline delimiters. This aggregated representation preserves the complete context of each service interaction while enabling efficient keyword-based matching.

BM25's effectiveness for this coarse-grained retrieval stems from three factors: (1) its ability to handle longer documents without degradation, (2) strong performance on keyword-rich queries typical of technical support scenarios, and (3) computational efficiency that keeps a single pass over the entire case database practical. The model computes relevance scores between the user query and each case's aggregated text, returning

cases ranked by similarity.

Stage 2: Snippet-Level Extraction The second stage focuses on precise snippet extraction from cases retrieved in Stage 1. Unlike the first stage, which considers all case components to maximize recall, this step processes only the *response content* field to extract actionable solutions and filter administrative or contextual information.

We employ a *hybrid retrieval strategy* combining three complementary models, each addressing different aspects of semantic matching:

- **BM25 [12]:** Provides lexical matching for queries with exact keywords or technical terms, ensuring strong baseline performance on literal matches.
- **chuxin-embedding [13]:** A Chinese text-embedding model built on bge-m3-retromae and further pre-trained, fine-tuned and refined on Chinese retrieval data. It encodes queries and candidate snippets independently into a shared vector space.
- **LdIR-Qwen2 reranker (1.5B) [14]:** A Chinese reranking model built on the Qwen2 architecture. Unlike the two retrievers above, it scores a query and a candidate snippet jointly rather than through independent representations, which makes it effective on paraphrased or conceptually similar queries at a higher computational cost.

We combined these models because they score different aspects of query–snippet relevance: BM25 relies on lexical overlap, chuxin-embedding compares dense representations, and the LdIR-Qwen2 reranker scores query–snippet pairs jointly. Their scores are aggregated with the weights defined below.

Hybrid Weighting Strategy The three models are combined through a weighted sum, where each model's contribution is determined by weights W_i intended to balance retrieval quality against computational cost:

$$\text{Score}(q,s) = \sum_{i \in \mathscr{M}} W_i \cdot \text{Sim}_i(q,s) \tag{1}$$

where q denotes the user query, s denotes a candidate snippet, $\mathscr{M} = \{\text{BM25}, \text{Chuxin}, \text{Qwen}\}$, and $\text{Sim}_i(q,s)$ represents the similarity score computed by model i.

Accuracy matters more than latency in this setting: an incorrect answer carries a compliance cost, while a difference of a few seconds does not. The weights are therefore determined by

$$W_i = \frac{F_i^\alpha}{T_i^\beta} \Big/ \sum_{j \in \mathscr{M}} \frac{F_j^\alpha}{T_j^\beta} \tag{2}$$

where F_i is the average retrieval quality (F1-score) of model i across retrieval thresholds, T_i is its average query time, and α and β control the relative importance of quality and

speed. We set $\alpha = 4$ to prioritise retrieval quality and $\beta = 0.1$ to limit the influence of query-time differences. The resulting weights are reported in Section 4.2.2, together with what this formulation does and does not achieve on our data.

Snippet Selection Extraction proceeds in three steps. The response content of the retrieved case is split into snippets at newline boundaries, which matches the natural structure of the records, where each entry addresses one aspect of the inquiry. Each snippet is then scored against the query by the weighted hybrid model of Equation (1), each component model computing its similarity independently before the scores are combined. Finally the top N snippets are passed to the generation stage, with N chosen empirically to balance recall of the reference snippets against the redundant content that would otherwise distract the language model.

The rationale for selecting N involves analyzing the trade-off between *retrieval completeness* (measured by recall) and *information density* (measured by compression rate—the percentage reduction in content length compared to full case documents). Our selection criterion requires achieving high recall ($\geq 95\%$ of manually annotated reference snippets) while maximizing compression to reduce noise. The specific value of N and resulting performance metrics are reported in Section 4.2.4.

3.3 Text Generation Stage

Following the two-stage retrieval process (steps ②–④ in Figure 1), the text generation stage transforms the retrieved snippets into coherent, contextually appropriate responses through three key components: temporal snippet ranking (step ⑤), structured prompt construction (step ⑥), and response generation with sampling-based reranking (step ⑦).

Language Model Configuration We employ the 8B-parameter version of the TAME (Taiwan Mixture of Experts) model [10], leveraging its full FP32 precision as our primary language model. Built upon Llama 3.1, TAME-8B has been fine-tuned using datasets sourced from multiple Taiwanese enterprises that communicate in Traditional Chinese, ensuring better alignment with our objective of developing a framework tailored for Traditional Chinese customer service applications. We configure the key sampling parameters as follows: temperature = 0.1, top-k = 50, and max new tokens = 1024.

Time-Sorted Snippet Ranking (Step ⑤) After retrieving the top-N snippets from the second-stage retrieval (step ④), we apply a Time-Sorted ranking algorithm to preserve the chronological structure of the customer service interaction. As shown in step ⑤ of Figure 1, instead of presenting snippets in order of textual relevance scores, we reorder them according to their original "Answer Time" timestamps. This approach maintains the temporal integrity of the problem-solving process, which is critical in customer service scenarios where solutions often build upon previous troubleshooting steps.

The rationale for temporal sorting is twofold: (1) it preserves the natural progression of how service agents resolved similar issues in the past, and (2) it ensures that earlier diagnostic steps appear before subsequent solutions, helping the language model under-

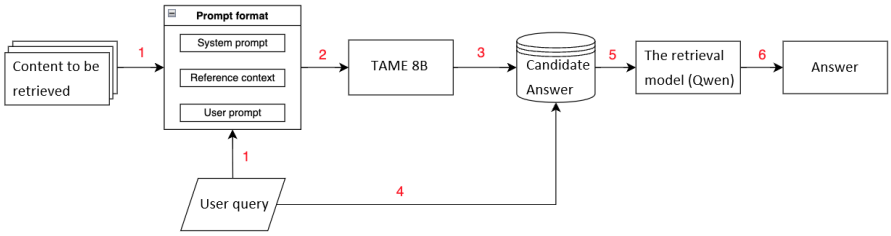


Fig. 2. Sampling-based reranking process for response generation. Steps ①–③ are repeated K times to generate diverse candidate responses, which are then evaluated by the Qwen reranker (steps ④–⑤) to select the final answer (⑥).

stand the causal relationships between different response entries. This temporal coherence enables TAME-8B to generate responses that reflect the logical flow of effective customer service interactions.

Structured Prompt Construction (Step ⑥) The time-sorted snippets are then integrated with the user query through a structured prompt template comprising three components, as illustrated in step ⑥ of Figure 1:

- **System Prompt:** Defines the assistant's role and behavior. We employ the prompt: "You are a knowledgeable, efficient, and direct AI assistant," which encourages concise, accurate responses without unnecessary elaboration.
- **Reference Context:** Contains the time-sorted retrieved snippets from step ⑤, providing domain-specific knowledge and historical solutions relevant to the user's query.
- **User Prompt:** The original customer query from step ②, ensuring the model addresses the specific question asked.

Response Generation with Sampling-Based Reranking (Step ⑦) Rather than generating a single response, we follow Tan et al. [15] in producing several candidates and reranking them, as illustrated in Figure 2. The structured prompt is passed to TAME-8B K times with different sampling seeds (typically $K = 5$), which reduces the risk that a single low-quality sample is returned because of sampling variance or an ambiguous context. Each candidate is then paired with the original query and scored by the LdIR-Qwen2 reranker for relevance to the query; the highest-scoring candidate is returned as the final answer. The design is intended to reduce the risk of returning an irrelevant or internally inconsistent sample, though we do not isolate its effect experimentally.

4. Experimental Results and Analysis

This section evaluates our proposed hybrid framework through systematic experiments on Traditional Chinese customer service data. Section 4.1 introduces the evaluation

metrics for both retrieval and generation tasks. Section 4.2 presents comprehensive experimental results: we first assess the two-stage retrieval architecture (Section 4.2.1), then examine how the hybrid model weights were obtained (Section 4.2.2), compare the second stage against current open-source baselines (Section 4.2.3), select the snippet count (Section 4.2.4), and evaluate text generation quality under different configurations (Section 4.2.5). Section 4.3 reports a human evaluation of the generated responses and Section 4.4 the per-query latency.

4.1 Evaluation Method

4.1.1 Stage 1: Case-Level Retrieval Metrics

For the first stage, we evaluate retrieval performance using two key metrics: **Accuracy**, which measures the precision of the top-1 retrieval result, and **Average Query Time (Avg Time)**, which measures the response speed of models.

4.1.2 Stage 2: Snippet-Level Retrieval Metrics

After obtaining the best retrieval results from the first stage, we proceed with the extraction of precise case snippets. We define manually annotated standard answer snippets as *reference snippets*, while the snippets retrieved by the model are referred to as *candidate snippets*. The "response content" from the cases retrieved in the first stage is segmented into individual snippets using newline characters, which serve as the basic units for retrieval in this stage.

To determine the most relevant information, we calculate the similarity between each candidate snippet and the user's query. Based on these similarity scores, we extract the top N snippets (where N is a positive integer) for further evaluation. To systematically assess retrieval performance, we vary N from 1 to 10 and conduct experiments to compare the effectiveness of different retrieval models under various N configurations. Snippet retrieval is evaluated by Precision, Recall, and F1-score, along with Avg Query Time. We also report Compression Rate (CR) to quantify how much the original response content is reduced while retaining relevant evidence: $\mathrm{CR} = \left(1 - \frac{\sum_{i=1}^{N} L_{\mathrm{snippet}_i}}{L_{\mathrm{original}}}\right) \times 100\%$, where N is the number of retrieved snippets, L_{snippet_i} denotes the character length of the i-th retrieved snippet, and L_{original} represents the total character length of the complete response content field from the retrieved case. A higher compression rate indicates more effective filtering of redundant information while maintaining retrieval completeness.

4.1.3 Text Generation Metrics

For evaluating the performance of text generation, the following metrics are used to compare the standard reference texts in the test dataset with the texts generated by the language model:

- **ROUGE-N** [42]: This metric assesses the degree of content similarity between the candidate text and the reference text.
- **ROUGE-L** [42]: This method captures not only token overlap but also keyword and word order similarity, providing a more nuanced evaluation.

- **BLEU** [43]: This metric is commonly used to assess machine translation quality, focusing on the similarity between machine-generated translations and reference translations, with an emphasis on sentence accuracy and precise alignment.
- **BERT Score** [44]: Unlike traditional metrics, this approach captures latent semantic information that would otherwise be difficult to quantify, offering a more comprehensive and multi-dimensional evaluation of text generation.
- **LLM Score:** This method leverages the TAME-8B model to determine whether the reference text and candidate text are semantically aligned in meaning.

4.2 Experimental Results

4.2.1 Two-Stage Retrieval Experiments

We systematically evaluate retrieval performance at both stages of our framework, examining different model configurations and text combinations to identify the optimal setup.

Stage 1: Case-Level Retrieval. The first-stage experiment evaluates BM25, Chuxin, and Qwen on different combinations of case text fields. For neural models (Chuxin and Qwen), we test performance using both Traditional Chinese and Simplified Chinese input formats to assess their effectiveness in our target linguistic context. We employ top-1 retrieval and analyze performance differences between Traditional Chinese and Simplified Chinese across models, matching cases with the highest similarity scores to assess retrieval capabilities.

Table 1 presents the experimental results, revealing a positive correlation between text length and retrieval accuracy. BM25 is the fastest model in every setting and the most accurate on the response and combined fields, though not on the summary alone. That response content alone (93.55%) outperforms customer-reported content alone (86.02%) reflects the construction of the evaluation set, discussed below.

Most notably, when all case-related text fields are combined (summary + customer-reported content + response content), BM25 achieves the highest observed top-1 retrieval accuracy, 100%. Two properties of the evaluation set bear on how this figure should be read. The originating case is always present in the datastore, which is the intended deployment condition—a new inquiry is matched against past cases—but it does mean the target is never absent. And the queries were written by annotators who had the resolving entries in view, while each response entry restates the problem in its problem-analysis field, so a query shares vocabulary with the indexed content of its own case to a degree a genuine inquiry would not. Indexing the response content is legitimate, since historical replies are available at query time; the inflation comes from how the evaluation set was constructed. Regarding computational efficiency, BM25 ranks first (8.42s average query time), followed by Chuxin (30.36s), while Qwen demonstrates relatively lower efficiency (71.03s) due to its larger model size. These results identify BM25 with aggregated text fields as the selected operating configuration among those evaluated for case-level retrieval.

Stage 2: Snippet-Level Extraction. Building on the selected first-stage configuration—BM25 with combined text fields—we proceed to snippet-level extraction, evaluating how effectively retrieval models identify reference snippets. Given that a case typically contributes fewer than ten candidate entries and a query carries 1.40 reference

Model	Text fields	Acc. (%)	Time (s)	Text fields	Acc. (%)	Time (s)
BM25		52.69	0.23		93.55	7.47
Chuxin (Traditional)		54.84	0.26		82.80	30.23
Chuxin (Simplified)	Summary	55.91	0.33	Response	84.95	30.20
Qwen (Traditional)		62.37	3.28		86.02	70.53
Qwen (Simplified)		61.29	4.37		87.10	70.41
BM25		86.02	1.23		**100.00**	8.42
Chuxin (Traditional)	Customer-reported	83.87	2.34	Combined (all)	91.40	30.36
Chuxin (Simplified)		83.87	2.46		91.40	30.43
Qwen (Traditional)		84.95	15.06		94.62	71.03
Qwen (Simplified)		86.02	16.29		97.85	70.80

Table 1. Stage 1 retrieval performance across text field combinations and models. Two panels share one Model column: the "Summary" and "Customer-reported" field settings on the left, "Response" and "Combined" on the right. (Traditional) and (Simplified) denote the input text format used for evaluation, not different model versions. Acc. = top-1 accuracy; Time = average query time.

snippets on average (maximum 10), we systematically vary the retrieval threshold from TOP-1 to TOP-10.

Table 2 presents the experimental results. As expected, Precision declines significantly as TOP-N increases, reflecting the imbalanced distribution of reference snippets. However, the neural retrieval models—Chuxin and Qwen—generally outperform BM25 by identifying a higher proportion of relevant reference snippets within smaller retrieval scopes. This performance advantage stems from two key factors:

- **Lexical gap in Q&A scenarios**: User queries and solution descriptions often lack exact keyword overlap, as solutions describe problems from different perspectives.
- **Deep semantic understanding**: Pre-trained neural models excel at capturing semantic associations between questions and answers, overcoming lexical matching limitations and accurately identifying relevant content early in retrieval rankings.

Notably, Chuxin and Qwen exhibit alternating performance advantages across different TOP-N thresholds (particularly within TOP-5), suggesting complementary retrieval characteristics. This observation motivates our hybrid retrieval strategy, which assigns weights to each model based on Equation (2).

Selection of Operating Configuration. Given our practical application requirements, the evaluation criteria must balance two objectives: ensuring comprehensive retrieval of reference snippets while minimizing redundant content. Given the compliance-sensitive nature of this application, we first restrict attention to configurations meeting our operational threshold of Recall $\geq 95\%$, then select, for the hybrid model, the retrieval depth maximizing both F1-score and Compression Rate.

Our original experiments adopted the following two-stage operating configuration. The first stage applies BM25 over the aggregated Summary, Customer-reported Content and Response Content fields, achieving perfect case-level accuracy (100%) at an average query time of 8.42 seconds. The second stage retrieves the TOP-7 snippets with the weighted hybrid model at the weights $0.25/0.35/0.40$ used in our original experiments;

Model	TOP N	Prec. (%)	Recall (%)	F1 (%)	CR (%)	TOP N	Prec. (%)	Recall (%)	F1 (%)	CR (%)
BM25	1	32.26	29.21	30.06	88.15	6	22.71	91.36	35.12	36.72
Chuxin(Traditional)		41.94	38.89	39.73	87.45		23.42	94.77	36.22	37.70
Chuxin(Simplified)		40.86	38.35	39.02	87.48		23.42	95.09	36.30	37.51
Qwen(Traditional)		45.16	42.11	42.96	85.69		23.24	92.80	35.79	35.56
Qwen(Simplified)		48.39	44.80	45.83	86.44		23.24	93.33	35.83	35.07
Our model		43.01	39.43	40.45	86.63		23.24	92.80	35.79	35.66
BM25	2	25.27	45.88	31.99	75.92	7	22.09	94.23	34.39	31.04
Chuxin(Traditional)		32.80	60.93	42.03	74.23		22.25	95.30	34.66	31.90
Chuxin(Simplified)		31.18	57.71	39.87	74.74		22.25	95.30	34.66	31.77
Qwen(Traditional)		32.26	60.39	41.49	72.49		22.55	96.56	35.09	29.68
Qwen(Simplified)		32.26	59.86	41.31	72.91		22.55	96.56	35.09	29.46
Our model		31.18	57.71	39.87	72.94		22.55	95.70	35.00	30.46
BM25	3	25.99	69.75	37.21	62.75	8	21.52	95.48	33.61	27.03
Chuxin(Traditional)		27.78	74.23	39.60	62.17		21.65	96.59	33.86	27.41
Chuxin(Simplified)		27.06	72.08	38.52	62.94		21.65	96.59	33.86	27.49
Qwen(Traditional)		28.85	76.92	41.11	58.91		21.65	96.56	33.85	25.78
Qwen(Simplified)		28.14	74.77	40.03	59.91		21.78	97.63	34.09	25.35
Our model		29.21	78.89	41.94	60.76		21.78	96.77	34.01	26.07
BM25	4	23.92	78.03	35.78	52.22	9	21.17	97.63	33.23	23.73
Chuxin(Traditional)		24.73	80.36	36.86	53.45		21.29	97.85	33.38	23.79
Chuxin(Simplified)		24.73	80.36	36.86	53.27		21.17	97.67	33.24	24.69
Qwen(Traditional)		25.27	82.51	37.72	48.76		21.17	97.63	33.23	22.51
Qwen(Simplified)		25.27	82.51	37.72	49.71		21.17	97.63	33.23	22.43
Our model		25.54	83.58	38.15	50.95		21.29	97.31	33.36	22.45
BM25	5	23.17	85.23	35.39	43.84	10	20.89	97.85	32.82	20.58
Chuxin(Traditional)		23.82	88.10	36.35	45.68		20.89	97.85	32.82	21.49
Chuxin(Simplified)		23.17	84.87	35.28	45.10		20.78	97.67	32.68	21.96
Qwen(Traditional)		24.03	87.74	36.50	41.68		20.89	98.17	32.85	20.04
Qwen(Simplified)		24.03	89.18	36.71	41.87		20.89	98.17	32.85	20.06
Our model		24.25	90.25	37.07	42.67		20.99	98.39	32.99	19.96

Table 2. Stage 2 snippet-level retrieval performance. The table is split into two panels sharing one Model column: TOP 1–5 on the left, TOP 6–10 on the right. (Traditional) and (Simplified) denote the input text format, not different model versions. "Our model" is the hybrid of BM25, Chuxin and Qwen at the fixed weights $0.25/0.35/0.40$ used in our original experiments. Prec. = precision, CR = compression rate; all values are percentages.

Section 4.2.2 revisits how those weights were obtained, maintaining 95.70% recall at a 35.00% F1-score and a 30.46% compression rate. This configuration balances retrieval completeness, information density and computational cost, and is the one used for the text-generation experiments that follow.

4.2.2 Hybrid Model Weight Determination

F_i in Equation 2 is the average F1-score of model i over TOP 1 to TOP 7, the practical operating region; T_i is the average time it takes to score the candidate snippets of one retrieved case. T_i is a *second-stage* cost: first-stage retrieval runs once, by BM25 alone, and does not enter the calculation. The values are $F = 34.28, 37.92, 38.67$ and $T = 0.009, 0.803, 0.704$ s for BM25, chuxin-embedding and the LdIR-Qwen2 reranker, with F1 on a 0–100 scale. With $\alpha = 4$ and $\beta = 0.1$ these give $W = 0.333, 0.318, 0.349$ — close to uniform, because $\beta = 0.1$ compresses an 89-fold difference in query time to a factor of 1.57 and the three F1-scores span about 4.4 points. Because the three scores are on different scales—an unbounded term-weight sum, a cosine similarity, a classification logit—each is min–max normalised over the candidate set before the weighted sum is taken.

Two caveats apply. The hybrid row of Table 2 was produced with the weights $0.25/0.35/0.40$ used in our original experiments, not with the values derived here, and we have not re-run that experiment with the corrected weights in its original software environment; the corrected weights are evaluated instead in Section 4.2.3, where every configuration is run on an equal basis. And the weights were not derived on held-out data; we cannot say how much a held-out derivation would change the ranking, since scores that lie close together can reorder under a small change of weights. Table 2 also does not show the hybrid outperforming its strongest component: at TOP-7 it attains an F1 of 35.00 against 35.09 for the LdIR-Qwen2 reranker alone, and a recall of 95.70% against 96.56%. Our contribution therefore lies in the two-stage decomposition, not in the specific weights.

4.2.3 Comparison against Current Retrieval Baselines

The three retrievers above were selected in 2024. To place them against current open-source models we repeat the second-stage comparison with BGE-M3 [29] used as a dense retriever and bge-reranker-v2-m3 [45], a cross-encoder from the same family, holding the first stage, the candidate pool, the segmentation and the reference snippets fixed so that only the snippet scorer varies. Table 3 also reports the hybrid at three weightings and three further equal-weight combinations.

The comparison was run in a current software environment, in which the tokenizer, the embedding library and the model runtime all differ from the original experiments. Re-running the configurations of Table 2 under these conditions broadly reproduces their ordering but not their absolute values: F1 shifts by -0.4 to $+1.2$ points at the depths shown, rising for every configuration and depth except the hybrid at TOP-3, where it falls by 0.44, while recall at TOP-7 falls by 2.6 to 3.0 points, so the hybrid no longer meets the 95% threshold there. Orderings that turn on differences of a tenth of a point are not stable between the two tables: the hybrid and the LdIR-Qwen2 reranker lie within 0.15 F1 points of each other at TOP-7 in both, and exchange places between them. Both

shifts are consistent with the change in the candidate pool: the entries of a retrieved case are deduplicated before scoring in this run, which removes repeated entries and so raises precision while costing recall where a reference snippet occurred more than once. The change of software environment may also contribute. Compression rate is likewise computed against that deduplicated pool, a different base from Table 2. The pool is held fixed across every configuration within Table 3, so the comparison there is like for like, but comparisons should be made within that table rather than across tables.

Configuration	**F1@3**	**F1@5**	**F1@7**	**R@7**	**CR@7**	**Time (s)**	N_{95}
BM25	38.13	36.24	35.19	91.25	11.18	0.012	9
chuxin-embedding	40.35	37.43	35.64	92.28	11.70	0.311	9
LdIR-Qwen2 reranker	41.50	36.81	36.01	93.75	11.00	5.378	9
BGE-M3 (dense)	40.75	37.48	35.40	91.74	11.60	0.295	—
bge-reranker-v2-m3	**43.51**	**38.66**	**36.93**	**95.25**	11.19	0.331	**7**
Original hybrid, at three weightings							
Hybrid, 0.25/0.35/0.40	41.50	38.25	36.15	93.06	11.18	5.701	9
Hybrid, uniform $1/3^{\dagger}$	42.04	37.53	36.02	92.94	11.21	5.701	9
Hybrid, 0.333/0.318/0.349	41.50	37.53	36.15	93.06	11.25	5.701	9
Further combinations, at equal weight							
BM25 + chuxin + BGE-M3†	41.11	37.35	36.05	93.47	11.37	0.618	9
BM25 + BGE-M3 + bge-rr†	42.07	38.07	36.42	94.13	11.46	0.638	9
BM25 + bge-rr†	42.65	38.37	36.68	94.21	11.28	0.343	8

Table 3. Second-stage comparison against current baselines, over the same 93 queries, first-stage results and 130 reference snippets. F1@N is the F1-score at retrieval depth N; R@7 and CR@7 are recall and compression rate at TOP-7. N_{95} is the smallest retrieval depth at which average recall reaches 95%, the criterion by which N is selected in Section 4.2.4; — means the threshold is not reached by TOP-10. †marks the four configurations that weight their components equally. Time is the mean per-query scoring cost on an NVIDIA T4 GPU, measured in the rerun described above and not comparable with Table 6; figures for combinations are the sums of their components. Run in a current software environment, so absolute values are not comparable with Table 2.

What the Comparison Shows bge-reranker-v2-m3 achieves the highest observed F1 at every retrieval depth reported, and is the only configuration that reaches 95% average recall by TOP-7. BM25 + bge-reranker reaches the threshold at TOP-8, the remaining configurations at TOP-9, and BGE-M3 used alone does not reach it by TOP-10. Since N is chosen in Section 4.2.4 as the smallest depth satisfying that threshold, the same rule applied here would select TOP-7 with the cross-encoder and TOP-9 with the hybrid. The cross-encoder also costs 0.33 s per query against the hybrid's 5.70 s, a 568M-parameter model replacing a 1.5B one.

Two further points. The three weightings of the hybrid are not distinguishable from one another—36.15, 36.02 and 36.15 at TOP-7, with the uniform assignment best at TOP-3—so the choice of weights has little effect. And in these configurations the benefit of combining appears to diminish as the performance gap among the constituents widens. Four entries in Table 3 weight their components equally (marked †), and measured against

each combination's own strongest constituent, BM25 + chuxin + BGE-M3 gains 0.41 F1 points at a spread of 0.45, the uniform hybrid of BM25, chuxin and LdIR-Qwen2 gains 0.01 at a spread of 0.82, and the two combinations containing bge-reranker-v2-m3 lose 0.51 and 0.25 at a spread of 1.74. One caveat applies to the combinations themselves: chuxin-embedding, BGE-M3 and bge-reranker-v2-m3 all derive from the BGE-M3 line, so a combination drawn from them supplies less independent signal than its component count suggests. Given differences of this size on 93 queries we treat this as a descriptive pattern rather than a general result, but it suggests that our original design was reasonable for the models then available (three retrievers within 0.82 points of one another), and that the difference now comes from a stronger single model. Compression rate is almost invariant throughout, spanning 11.00 to 11.70 at TOP-7, so no configuration here holds a compression advantage over the others.

This result concerns the choice of second-stage scorer, not the two-stage architecture: the scorer can be replaced by a stronger model without changing the rest of the pipeline. bge-reranker-v2-m3 is the component we would now recommend, with the caveat that the 512-token maximum input length used in our runs truncates a few long entries, which may affect its figures.

4.2.4 Optimal Snippet Count Selection

The number of snippets N passed to the generator balances retrieval completeness against redundant content. For this compliance-sensitive application, where an incomplete answer carries a regulatory cost, we adopt $\geq 95\%$ average recall as an operational design threshold, meaning the retrieved snippets must contain at least 95% of the manually annotated reference snippets on average. Table 2 shows that TOP-7 through TOP-10 all meet it. Beyond TOP-3 both F1-score and compression rate decrease monotonically in N, so the configuration maximising both among those meeting the threshold is simply the smallest such N: TOP-7, at 95.70% recall, an F1-score of 35.00% and a compression rate of 30.46%.

Practical Implications The choice of $N = 7$ has three practical consequences. Since a case typically contributes fewer than ten candidate entries, nearly all available snippets are considered for short cases, while for long ones the selection discards roughly two thirds of the content. Seven snippets of 150–200 tokens occupy about 1,000–1,400 tokens of context, well within TAME-8B's window and leaving ample room for the query and the generated response.

This is the snippet count used in the text-generation experiments of Section 4.2.5.

4.2.5 Text Generation Model Experiments

To evaluate the impact of prompt engineering on text generation quality, we compare three system prompt configurations with TAME-8B, which differ in how much task specification they supply. **TAME** is the model's default prefix, "You are an AI assistant called Twllm, created by the TAME (Taiwan Mixture of Experts) project", which establishes identity but states no task requirement. **Expert** gives an explicit instruction, "You are a knowledgeable, efficient, and direct AI assistant"; this is the configuration our framework adopts. **Without System Prompt** is a control that omits system-level instructions

entirely, testing whether explicit prompting is needed at all.

We cross these with three ways of integrating the retrieved material. **Passage** supplies the top-*N* snippets ranked by the hybrid model's relevance score alone. **Time Sorted** takes the same snippets and reorders them by their original "Answer Time" (Section 3.3), preserving the sequence in which the case was resolved; this is the configuration our framework adopts. **Document** is a control that bypasses snippet extraction and passes the complete "Response Content" field of the retrieved case to the language model, testing whether fine-grained filtering helps at all.

Table 4 presents experimental results across different system prompt and retrieval configurations, with standard deviations in parentheses describing dispersion across the 93 evaluation queries. We generate five candidates per query at two temperature settings (0.1 and 0.6), selecting the best via Qwen-based reranking.

System Prompt Impact. Without a system prompt, inference times nearly double (10–12s vs. 5–7.5s) owing to lengthy, repetitive sequences, and the LLM score is generally lower; ROUGE scores are higher, as TAME-8B mimics retrieved Q&A patterns rather than synthesizing focused responses. TAME and Expert prompts perform comparably, with neither consistently ahead across metrics, indicating that explicit role definition appears more important than the specific wording of the system prompt.

Temperature and Retrieval Effects. Varying temperature from 0.1 to 0.6 produces comparatively small changes in the reported metrics, suggesting that prompt and retrieval choices have greater effects within the configurations tested. Among retrieval methods, Document-based generation exhibits longer inference times (6.5–7.5s) and generally slightly lower BLEU scores compared to Passage and Time Sorted approaches. However, no single configuration consistently outperforms others across all metrics, suggesting that relevance-based ranking (Passage) and temporal ordering (Time Sorted) capture complementary quality dimensions.

To further enhance generation quality, we incorporate structured guiding words that prompt TAME-8B to organize responses systematically. Specifically, we prepend the instruction: *"Based on the provided context, the core information directly related to '{user query}' is as follows:"* to guide the model's response structure. The rows marked "0.1 + GW" in Table 4 report the results of this modification, applied across all system prompts and retrieval configurations.

Impact of Guiding Words. The addition of structured guiding words produces contrasting effects depending on the presence of system prompts. When explicit task instructions are provided (TAME or Expert), guiding words improve the overlap-based metrics: BERT Score increases by 0.01–0.04 points, BLEU by 0.02–0.10 and ROUGE-L by 0.09–0.19 compared to the corresponding rows at temperature 0.1, although the LLM score falls in four of the six configurations. For instance, the Expert + Time Sorted configuration achieves BERT Score of 0.84 (vs. 0.82), BLEU of 0.35 (vs. 0.29), and ROUGE-L of 0.70 (vs. 0.59) at temperature = 0.1. These improvements indicate that explicit structural cues help TAME-8B organize retrieved information more coherently.

Conversely, without system prompts, guiding words cause comprehensive performance degradation: BERT Score drops by 0.05–0.06, BLEU decreases by 0.13–0.20, and inference time nearly doubles (from 10–12s to 20–22s). This counterintuitive result occurs because the retrieved Q&A pairs contain inherent structural patterns. Without task-oriented instructions, TAME-8B attempts direct pattern matching; the added guiding

System Prompt	Retrieval Type	Setting	BERT Score	BLEU Score	ROUGE-1 (Recall)	ROUGE-L (Recall)	LLM Score	Inference Time
TAME	Time Sorted	0.1	0.81 (0.11)	0.30 (0.29)	0.63 (0.32)	0.56 (0.34)	0.82	5.22 (6.14)
	Passage	0.1	0.83 (0.11)	0.34 (0.31)	0.67 (0.30)	0.60 (0.34)	0.85	5.53 (6.31)
	Document	0.1	0.80 (0.12)	0.26 (0.27)	0.58 (0.32)	0.50 (0.34)	0.76	6.45 (8.48)
	Time Sorted	0.6	0.82 (0.11)	0.28 (0.27)	0.62 (0.31)	0.54 (0.34)	0.85	5.65 (6.78)
	Passage	0.6	0.83 (0.11)	0.32 (0.29)	0.63 (0.30)	0.56 (0.33)	0.84	5.47 (5.97)
	Document	0.6	0.81 (0.12)	0.30 (0.29)	0.63 (0.32)	0.56 (0.34)	0.76	7.20 (10.16)
	Time Sorted	0.1 + GW	0.84 (0.12)	0.38 (0.31)	0.75 (0.29)	0.70 (0.33)	0.78	6.55 (6.56)
	Passage	0.1 + GW	0.84 (0.11)	0.36 (0.29)	0.76 (0.29)	0.70 (0.33)	0.78	7.19 (7.74)
	Document	0.1 + GW	0.84 (0.13)	0.36 (0.29)	0.74 (0.30)	0.69 (0.33)	0.77	9.08 (11.01)
Without System Prompt	Time Sorted	0.1	0.82 (0.12)	0.30 (0.31)	0.70 (0.32)	0.65 (0.34)	0.77	11.93 (13.84)
	Passage	0.1	0.82 (0.13)	0.34 (0.32)	0.73 (0.29)	0.66 (0.32)	0.76	10.20 (11.85)
	Document	0.1	0.81 (0.12)	0.29 (0.30)	0.69 (0.32)	0.64 (0.35)	0.72	12.06 (13.61)
	Time Sorted	0.6	0.82 (0.12)	0.29 (0.32)	0.68 (0.32)	0.62 (0.35)	0.73	10.12 (11.06)
	Passage	0.6	0.81 (0.12)	0.27 (0.29)	0.68 (0.31)	0.63 (0.34)	0.80	11.39 (12.02)
	Document	0.6	0.82 (0.12)	0.28 (0.29)	0.70 (0.30)	0.65 (0.33)	0.74	11.70 (12.76)
	Time Sorted	0.1 + GW	0.77 (0.11)	0.15 (0.19)	0.69 (0.34)	0.64 (0.35)	0.73	20.05 (14.78)
	Passage	0.1 + GW	0.76 (0.12)	0.14 (0.19)	0.67 (0.35)	0.63 (0.36)	0.62	21.58 (15.31)
	Document	0.1 + GW	0.76 (0.12)	0.16 (0.20)	0.67 (0.34)	0.63 (0.35)	0.60	21.40 (15.18)
Expert	Time Sorted	0.1	0.82 (0.11)	0.29 (0.27)	0.66 (0.31)	0.59 (0.33)	0.83	6.21 (5.73)
	Passage	0.1	0.82 (0.11)	0.31 (0.28)	0.66 (0.29)	0.58 (0.32)	0.87	5.68 (4.17)
	Document	0.1	0.82 (0.10)	0.27 (0.25)	0.66 (0.30)	0.58 (0.33)	0.85	7.33 (7.65)
	Time Sorted	0.6	0.82 (0.12)	0.31 (0.30)	0.65 (0.29)	0.58 (0.32)	0.78	5.91 (4.96)
	Passage	0.6	0.81 (0.12)	0.29 (0.29)	0.63 (0.32)	0.56 (0.35)	0.82	5.94 (6.42)
	Document	0.6	0.81 (0.11)	0.27 (0.27)	0.62 (0.31)	0.54 (0.33)	0.84	7.47 (9.41)
	Time Sorted	0.1 + GW	0.84 (0.11)	0.35 (0.28)	0.76 (0.27)	0.70 (0.30)	0.84	7.22 (7.13)
	Passage	0.1 + GW	0.84 (0.10)	0.33 (0.27)	0.76 (0.28)	0.71 (0.32)	0.77	7.59 (7.46)
	Document	0.1 + GW	0.83 (0.11)	0.31 (0.27)	0.72 (0.29)	0.67 (0.32)	0.80	10.03 (11.83)

Table 4. Text generation performance across system prompts, retrieval configurations, sampling temperatures, and guiding words. "Setting" gives the sampling temperature; rows marked "+ GW" prepend the structured guiding words described in the text below and use the default sampling temperature of 0.1. Values are mean (standard deviation) across the 93 evaluation queries.

words disrupt this strategy, forcing the model to generate longer, unfocused outputs that deviate from effective responses.

Configuration Adopted. Our proposed framework—combining Expert prompts, Time Sorted retrieval, and structured guiding words—reaches a BERT Score of 0.84, BLEU of 0.35, ROUGE-L of 0.70 and an LLM Score of 0.84, though no configuration leads on every metric. Time Sorted and Passage-based retrieval are close throughout: Time Sorted leads on BLEU and the LLM score in this configuration, Passage on ROUGE-L, and neither dominates across the table. We adopt Time Sorted for the reason given in Section 3.3—it preserves the order in which a multi-turn case was actually resolved—rather than on the strength of these margins. This configuration is adopted in the pilot system evaluated in this study.

4.3 Human Evaluation

To test whether these metrics agree with a service professional's judgment, a domain expert at the partner enterprise, who is not an author of this paper, rated 39 generated responses—produced by the adopted configuration of Section 4.2.5 and sampled from the 93 evaluation queries in proportion to the issue categories recorded by the platform—on five five-point scales: factual correctness, actionability, politeness, consistency with enterprise and regulatory policy, and safety, defined as whether sending the response as

written could cause material harm. The rater saw the query, the response and the original agent reply, but not the automatic scores. All five dimensions were judged in one pass over the responses; the safety ratings were returned on a separate sheet because the spreadsheet first supplied had omitted that column. Eight items were silently repeated, giving 40 paired dimension-level judgments (8×5), all of which agreed, although one repetition fell adjacent to its original, so that figure is an upper bound. Mean ratings were 4.13, 3.87, 3.82, 4.00 and 4.38 in the order above.

Table 5 reports Spearman correlations over the 39 items. The metrics track factual correctness closely (ρ 0.67 to 0.71), actionability, politeness and safety moderately (0.33 to 0.58), and policy consistency hardly at all, where only BERTScore reaches a nominal $\rho = 0.32$. Splitting by overlap with the reference answer—ROUGE-L of 0.8 or above, 20 items, against 19 below—the correlations in the low-overlap group fall further: BLEU correlates with politeness at $\rho = 0.03$ and with policy consistency at $\rho = 0.08$. The metrics thus reflect factual correctness reasonably well but say little about whether a response is suitable to send.

Expert dimension	BLEU	ROUGE-1	ROUGE-L	BERTScore
Factual correctness	0.705*	0.676*	0.667*	0.705*
Actionability	0.531*	0.382*	0.368*	0.569*
Politeness	0.484*	0.445*	0.451*	0.584*
Safety	0.371*	0.353*	0.374*	0.328*
Policy consistency	0.250	0.179	0.171	0.320*

Table 5. Spearman rank correlation between expert ratings and automatic metrics, over 39 generated responses. *denotes nominal two-sided $p < 0.05$; these are twenty tests, unadjusted for multiple comparisons, and are reported descriptively.

Failure Modes Six responses illustrate three failure kinds; five of them were rated at or below 2.5 on average. Two retrieved a case's administrative closing message rather than its resolution—both scored ROUGE-L below 0.06, so the metrics caught them—and two answered an adjacent sub-issue. The third kind the metrics miss: two responses scoring ROUGE-L of 1.00 were rated 2.5 and 3.0 because what they reproduced was an internal working record carrying instructions addressed to colleagues. This is the predicted consequence of the record structure of Section 3.1: an extraction stage selecting on relevance alone will sometimes select an internal-analysis entry. The expert flagged thirteen of the 39 responses as raising disclosure concerns—the practices of other client institutions, database objects and query syntax, unauthorised delivery commitments—while only two were factually wrong. The safety scale separates these from ordinary quality defects: the single response rated 1 proposed that the customer disable an encryption feature in order to work around a field-length limit, and was rated 3 on both factual correctness and policy consistency, so neither of those scales would have isolated it. In this setting, responses that are correct but unsuitable to send are a greater risk than factually wrong ones.

4.4 End-to-End Latency

Table 6 reports the cost of answering one query in our original experimental environment. First-stage retrieval searches all 964 documents at 8.42 s but runs once; second-

stage extraction, operating only on the entries of the retrieved case, adds 1.49 s; generation in the adopted configuration takes 7.2 s including the five candidates of Section 3.3, and 20 to 22 s without a system prompt when the guiding words are added. A query is answered in about 17 seconds.

This is within what the asynchronous, ticket-based workflow tolerates—an agent reviews the draft before it is sent—and we do not claim it would be acceptable in a synchronous chat setting. The figures were measured in our experimental environment rather than on the enterprise's own installation, which we did not benchmark, and should be read as indicative. What the enterprise's setting does require is that the framework run on hardware it operates itself—the records cannot be sent to an external service—and every component we use is small enough to do so.

Stage	**Component**	**Time (s)**
Stage 1: case retrieval	BM25 over 964 documents	8.42
Stage 2: snippet extraction	BM25 over the retrieved case	0.009
	chuxin-embedding	0.803
	LdIR-Qwen2 reranker	0.704
	combined (measured)	1.49
Generation	TAME-8B, adopted configuration	7.2
	TAME-8B, no system prompt, with guiding words	20–22
End to end		**about 17**

Table 6. Per-query latency. Stage-2 times are measured at TOP-7 on the candidate entries of one retrieved case; the combined figure is measured rather than summed. Generation times include the five candidates generated for sampling-based reranking; the adopted configuration is Expert prompt, Time Sorted retrieval and guiding words, whose mean inference time is given in Table 4. All figures come from our original experimental environment; the corresponding entries in Table 3 come from a later rerun on an NVIDIA T4 GPU under a different software environment, so the two differ, most visibly for the LdIR-Qwen2 reranker, and should be read within their own table.

5. Conclusion

This study presents a hybrid retrieval-augmented generation framework for Traditional Chinese customer service automation, where existing large language models struggle with linguistic and cultural nuance. The framework combines BM25, chuxin-embedding and the LdIR-Qwen2 reranker with TAME-8B in a zero-shot RAG pipeline.

The experiments lead to three observations. First, regarding prompt design, structured guiding words produce larger changes in the reported generation metrics than any other factor we varied. Second, the results support the practical feasibility of the two-stage decomposition here, though not its superiority over direct snippet retrieval or end-to-end alternatives; within the second stage the weighted combination is not shown to improve on its strongest component. Third, the claim that temporal ordering preserves the logical progression of a multi-turn interaction is supported by the generation metrics in some configurations but not uniformly.

The results indicate pilot-scale feasibility, with an agent reviewing each draft before it is sent, in a resource-constrained enterprise environment, using publicly available models without fine-tuning, each small enough to run on hardware the enterprise operates itself.

Limitations

Five limitations bound these results. Validation is confined to a single enterprise, a single regulatory domain and text-only interactions. The annotated set contains 93 queries, and differences between the better configurations are small relative to their dispersion, so we report them as such rather than as rankings. The queries were written by staff with sight of the resolving entries, which affects how the first-stage result should be read (Section 4.2.1). The human evaluation rests on one rater, so inter-rater agreement could not be computed and its correlations are indicative rather than established; the LLM-based score comes from the same model (TAME-8B) that generates the responses and is not independent evidence. Finally, the records interleave internal deliberation with customer-facing text—a third of the sampled responses raised disclosure concerns—and the snippet stage's filtering has not been audited against an adversarial standard.

Future research should expand validation across enterprises and industries, report inter-annotator agreement, stratify the analysis of temporal ordering by the number of turns in a case, compare TAME-8B against alternative Traditional Chinese and other locally deployable generation models under the same retrieved context, and compare against complete open-source question-answering pipelines rather than at the component level alone.

Acknowledgement

The authors used Claude (Anthropic) for language editing and grammar checking during manuscript preparation. All content was subsequently reviewed and revised by the authors, who take full responsibility for the accuracy and integrity of the published work.

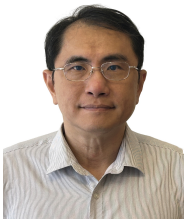
REFERENCES

1. D. Parmar, Enhancing customer relationship management with salesforce einstein gpt (2023).
2. G. Bovenzi, F. Cerasuolo, D. Ciuonzo, D. Di Monda, I. Guarino, A. Montieri, V. Persico, A. Pescapè, Mapping the landscape of generative ai in network monitoring and management, arXiv preprint arXiv:2502.08576 (2025).
3. R. VP, Role of ai in banking, Available at SSRN 3803749 (2021).
4. Y. Liu, J. Cao, C. Liu, K. Ding, L. Jin, Datasets for large language models: A comprehensive survey, arXiv preprint arXiv:2402.18041 (2024).
5. R. He, Y. He, L. Bai, J. Liu, Z. Sun, Z. Tang, H. Wang, H. Xia, R. Mihalcea, N. Deng, Chumor 2.0: Towards benchmarking chinese humor understanding, arXiv preprint arXiv:2412.17729 (2024).

6. T. Zhong, Z. Yang, Z. Liu, R. Zhang, Y. Liu, H. Sun, Y. Pan, Y. Li, Y. Zhou, H. Jiang, et al., Opportunities and challenges of large language models for low-resource languages in humanities research, arXiv preprint arXiv:2412.04497 (2024).
7. M. Xiaoliang, Z. RuQiang, L. Ying, D. Congjian, D. Dequan, Design of a large language model for improving customer service in telecom operators, Electronics Letters 60 (10) (2024) e13218.
8. W. Wang, Z. Zhao, T. Sun, Customizing large language models for business context: framework and experiments, arXiv preprint arXiv:2312.10225 (2023).
9. H. Takeuchi, L. V. Subramaniam, T. Nasukawa, S. Roy, Automatic identification of important segments and expressions for mining of business-oriented conversations at contact centers, in: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 2007, pp. 458–467.
10. Y. Lin, Y. Chen, Taiwan LLM: bridging the linguistic divide with a culturally aligned language model, CoRR abs/2311.17487 (2023). arXiv:2311.17487, doi:10.48550/ARXIV.2311.17487.
    URL https://doi.org/10.48550/arXiv.2311.17487
11. P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, Advances in neural information processing systems 33 (2020) 9459–9474.
12. J. Pokrywka, Passage retrieval of polish texts using okapi bm25 and an ensemble of cross encoders, in: 2023 18th Conference on Computer Science and Intelligence Systems (FedCSIS), IEEE, 2023, pp. 1265–1269.
13. Chuxin LLM, Chuxin-Embedding, Hugging Face model repository, built on bge-m3-retromae. Accessed 8 October 2026 (2024).
    URL https://huggingface.co/chuxin-llm/Chuxin-Embedding
14. neofung, LdIR-Qwen2-reranker-1.5B, Hugging Face model repository, accessed 6 October 2026 (2024).
    URL https://huggingface.co/neofung/LdIR-Qwen2-reranker-1.5B
15. B. Tan, Y. Zhu, L. Liu, E. Xing, Z. Hu, J. Chen, Cappy: Outperforming and boosting large multi-task lms with a small scorer, Advances in Neural Information Processing Systems 36 (2024).
16. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).
17. R. Sennrich, Neural machine translation of rare words with subword units, arXiv preprint arXiv:1508.07909 (2015).
18. T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013).
19. P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Transactions of the Association for Computational Linguistics 5 (2017) 135–146.
20. J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

21. J. H. Choi, How to use large language models for empirical legal research, Journal of Institutional and Theoretical Economics (Forthcoming) (2023).
22. J. Devlin, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
23. A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., Improving language understanding by generative pre-training (2018).
24. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347 (2017).
25. Q. Dong, L. Li, D. Dai, C. Zheng, J. Ma, R. Li, H. Xia, J. Xu, Z. Wu, B. Chang, et al., A survey on in-context learning, in: Proceedings of the 2024 conference on empirical methods in natural language processing, 2024, pp. 1107–1128.
26. T. B. Brown, Language models are few-shot learners, arXiv preprint arXiv:2005.14165 (2020).
27. W. Shi, S. Min, M. Yasunaga, M. Seo, R. James, M. Lewis, L. Zettlemoyer, W.-t. Yih, Replug: Retrieval-augmented black-box language models, arXiv preprint arXiv:2301.12652 (2023).
28. S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, D. Lian, J.-Y. Nie, C-Pack: Packed resources for general Chinese embeddings, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), 2024, arXiv:2309.07597.
29. J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, Z. Liu, BGE M3-Embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, arXiv preprint arXiv:2402.03216 (2024).
30. W. Sun, L. Yan, X. Ma, S. Wang, P. Ren, Z. Chen, D. Yin, Z. Ren, Is ChatGPT good at search? Investigating large language models as re-ranking agents, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2023, pp. 14918–14937. doi:10.18653/v1/2023.emnlp-main.923.
31. P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, ACM Computing Surveys 55 (9) (2023) 1–35.
32. J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al., Chain-of-thought prompting elicits reasoning in large language models, Advances in neural information processing systems 35 (2022) 24824–24837.
33. T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, Y. Iwasawa, Large language models are zero-shot reasoners, Advances in neural information processing systems 35 (2022) 22199–22213.
34. N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, S. Gelly, Parameter-efficient transfer learning for nlp, in: International conference on machine learning, PMLR, 2019, pp. 2790–2799.
35. B. Lester, R. Al-Rfou, N. Constant, The power of scale for parameter-efficient prompt tuning, arXiv preprint arXiv:2104.08691 (2021).
36. E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-rank adaptation of large language models, in: ICLR, 2022.
37. T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, Qlora: Efficient finetuning of quantized llms, Advances in Neural Information Processing Systems 36 (2024).

38. N. Xiao, C. Fan, L. Wang, T. Tao, W. Gao, Changes and applications of ai in the customer service industry, in: Proceedings of the 2024 7th International Conference on Computer Information Science and Artificial Intelligence, 2024, pp. 46–56.
39. J. Cui, Z. Li, Y. Yan, B. Chen, L. Yuan, Chatlaw: Open-source legal large language model with integrated external knowledge bases, CoRR (2023).
40. J. Wu, Z. Guo, Z. Wang, H. Zhang, X. Kang, X. Ma, Design and implementation of embedded qinghai tourism customer service question-and-answer robot based on chatgpt, in: Proceedings of the 2024 9th International Conference on Cyber Security and Information Engineering, 2024, pp. 871–882.
41. X. Xie, R. Cui, L. Fan, X. Yang, Large language models enhanced client simulation and feedback system for insurance advisors, in: 2024 4th International Conference on Big Data Engineering and Education (BDEE), IEEE, 2024, pp. 7–12.
42. C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: Text summarization branches out, 2004, pp. 74–81.
43. K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, BLEU: a method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), 2002, pp. 311–318.
44. T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, BERTScore: Evaluating text generation with BERT, in: International Conference on Learning Representations (ICLR), 2020.
45. BAAI, bge-reranker-v2-m3, Hugging Face model repository, accessed 6 October 2026 (2024).
URL `https://huggingface.co/BAAI/bge-reranker-v2-m3`



Yen-Wu Ti received the B.E. degree in mathematics from Tamkang University in 1995, the M.E. degree in applied mathematics from National Chiao Tung University, Hsinchu, Taiwan, in 1997, and the Ph.D. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 2009. He is currently an Associate Professor in the Department of Computer Science and Information Engineering, Ming Chuan University, Taoyuan, Taiwan. His research interests include machine learning and algorithms.



Wun-Hao Wang received the B.B.A. degree in information management from Yuan Ze University in 2023 and the M.S. degree in finance from National Yang Ming Chiao Tung University in 2025. His master's research developed the retrieval-augmented question-answering framework for Traditional Chinese enterprise customer service on which this work is based. His research interests include financial engineering and the application of large language models to enterprise information retrieval.



Tian-Shyr Dai received the Ph.D. degree in computer science from National Taiwan University. He is currently a full Professor in the Department of Information Management and Finance at NYCU, which he chaired from 2016 to 2019, and Executive Director of the EMBA Program. He is also a Research Member of the Risk and Insurance Research Center, NCCU, and a Senior Fellow of Advance HE. His research interests include financial engineering and financial technology.



Ci-Siang Huang received the B.S. degree in Finance from National Taiwan University in 2024, and is currently pursuing the M.S. degree in computer science at National Yang Ming Chiao Tung University. His research interests include AI applications in Finance.



Hsin-Yun Chang received the B.B.A. degree in information management from Fu Jen Catholic University. She has served at MetaEdge Corporation for 23 years and is currently Director of its Regulatory Department. Her expertise covers the planning and implementation of enterprise-level data processing systems, and her technical interests focus on core banking systems, data warehouse architecture, ETL processes, and regulatory compliance mechanisms.