AI_TLfanClub at FinMMEval 2026 Task 2: FiSCO-RAG for Cross-Lingual Evidence Recall in Financial Question Answering

CLEF 2026 FinMMEval Task 2 Working Notes

Do Thanh Duc[1], Nguyen Nhat Minh[1] and Le Thanh Huong[1]

[1]School of Information and Communication Technology, Hanoi University of Science and Technology, Vietnam

Abstract

This working notes paper describes the AI_TLfanClub submission to CLEF 2026 FinMMEval Task 2, Multilingual Financial Question Answering. We focus on the cross-lingual evidence recall: answers must be grounded in long contexts mixing English statements with English, Chinese, Japanese, Spanish, and Greek news, and ROUGE-1 rewards verbatim evidence tokens. FiSCO-RAG is an inference-only RAG pipeline with layout-aware chunking, entropy-calibrated CJK BM25, submodular partition-aware packing, schema-conditioned prompting, verifier filtering, and specificity-aware candidate selection. No fine-tuning or external web retrieval was used. The official run used a Qwen3.6-A3B-35B MoE backbone; controlled methodological ablations used Qwen3.5-4B, so we do not isolate model-scale effects. AI_TLfanClub ranked fourth of 12 completed submissions with ROUGE-1 F1 30.39%, precision 0.3647, recall 0.3070, and coverage 100%.

Keywords

multilingual financial QA, RAG, BM25, submodular context selection, FinMMEval, CLEF 2026

1. Introduction

FinMMEval Task 2 evaluates multilingual evidence-grounded financial question answering on PolyFiQA [1, 2, 3, 4]. Each question is paired with a long context containing financial statements and financial news. The financial statements are in English, while news evidence may appear in English, Chinese, Japanese, Spanish, and Greek. The system must produce an answer of at most 100 words in a fixed two-section format:

```
Answer: <short financial conclusion>.
<Evidence Section>: <supporting evidence or None.>
```

The task differs from ordinary open-domain generation in three ways. First, the contexts are long. In our local copy, public-gold development queries have median lengths above 63k characters, and local blind-test queries can approach 80k characters. Second, the evidence is sparse and mixed-format: the relevant facts may be in a table row, a cash-flow statement, or a non-English news sentence. Third, ROUGE-1 is the primary metric, so translating or paraphrasing a source-language quote can reduce score even when the meaning is correct.

Our approach is therefore retrieval-first. We do not rely on a model to read the entire raw context. Instead, FiSCO-RAG constructs a compact evidence context before generation. The system is designed around five principles:

- preserve financial-table layout and language-specific news blocks;
- make Chinese and Japanese evidence visible to lexical retrieval;
- reserve context budget for required evidence partitions such as news language and financial-statement type;

CLEF 2026 Working Notes, 21–24 September 2026, Jena, Germany

duc.dt242095m@sis.hust.edu.vn (D. T. Duc); minh.nn252184m@sis.hust.edu.vn (N. N. Minh); huong.lethanh@hust.edu.vn (L. T. Huong)

- force the answer into the expected two-section schema;
- verify that the evidence section is supported by the selected context.

We instantiate these principles as C2 entropy-calibrated n-grams, C1 submodular partition-aware context selection, and C3 specificity-aware MBR.

1.1. Acronyms and Notation

For clarity, we use the following abbreviations throughout the paper. RAG means retrieval-augmented generation. BM25 refers to Okapi BM25 lexical retrieval. CJK denotes Chinese, Japanese, and Korean scripts; in this task it mainly affects Chinese and Japanese news evidence. SCCS denotes our Schema-Conditioned Coverage Selection objective for partition-aware context packing. IDF denotes inverse document frequency. MBR denotes minimum Bayes risk candidate selection. MoE denotes mixture-of-experts model routing. NF4 is normal-float 4-bit quantization, used only in small-GPU development profiles, whereas bfloat16 is the 16-bit floating-point format used in the submitted 35B-profile run. We use C1, C2, and C3 as contribution labels for submodular partition-aware context selection, entropy-calibrated retrieval, and specificity-aware candidate selection, respectively.

This paper reports the submitted system, the official leaderboard result, development ablations, and the main failure modes observed during the competition.

2. Related Work

Financial question answering benchmarks have evolved from numerical reasoning over financial reports in FinQA [5], conversational financial reasoning in ConvFinQA [6], and hybrid table-text reasoning in TAT-QA [7] toward multilingual and multimodal evaluation in FinMMEval and MultiFinBen [1, 3, 4]. Task 2 differs from many earlier financial QA settings because the supplied context already contains both financial statements and multilingual news; the system must select evidence from that context rather than retrieve documents from an external corpus.

Retrieval-augmented generation is a standard approach for grounding neural answers in retrieved passages [8, 9]. However, long-context models can still underuse relevant passages depending on where evidence appears in the prompt [10]. We therefore use retrieval and context packing to present a shorter, more task-focused context to the generator.

Our retriever is lexical rather than fully dense. BM25 remains a strong and transparent baseline for passage retrieval [11], and it aligns well with ROUGE-style evaluation when answers are expected to copy financial terms, numbers, and source-language evidence. We tested dense reranking with multilingual BGE embeddings [12], but used it as an optional development component because the gains were not uniform across Easy and Expert questions. For candidate selection experiments, we also tested small-pool minimum Bayes risk decoding, following prior work on alternatives to single-candidate MAP decoding [13, 14].

The context-packing component is related to monotone submodular maximization under a knapsack budget, a classical framework for coverage-oriented selection [15, 16, 17]. Our tokenization rule is motivated by the information-theoretic view that roughly 2^{nH} strings are distinguishable under an entropy rate H [18]; this gives a compact way to choose language-dependent n-gram lengths without adding many free parameters.

3. Task and Data

3.1. Task Setting

Following the official FinMMEval Task 2 definition [2, 4], each instance contains:

- `task_id`: stable identifier;

- `query`: long context with instructions, financial statements, news, and the question;
- `question`: extracted natural-language question;
- `tier`: Easy or Expert in the local blind-test file;
- `answer`: gold answer only in public-gold development data.

The official output is a JSONL file with one object per row:

```
{"task_id": "easy_0001", "answer": "Answer: ... News Evidence: ..."}
```

The local blind test used for leaderboard submission has 256 rows: 128 Easy and 128 Expert. The `Answer Format:` block embedded in each query specifies the expected evidence section. In the blind test, Easy rows require `News Evidence`, while Expert rows require `Financial Statements Evidence`.

3.2. Data Statistics

The following statistics were computed from the repository data used during development and submission packaging.

Table 1
Data statistics for the development and blind-test files used in this submission.

Split	Rows	Median chars	Max chars	zh/ja/es/el	Tables
Public Easy gold	76	63,288	77,699	76/76	76/76
Public Expert gold	76	63,367	77,716	76/76	76/76
Local blind test	256	52,372	79,492	256/256	256/256

All rows contain table-like financial statements and multilingual news. This motivated a retrieval and context-packing approach rather than raw long-context prompting.

4. System Overview

FiSCO-RAG is an inference-only RAG system. The submitted run used the local leaderboard pipeline with a Qwen3.6-A3B-35B MoE deployment model on an RTX 6000-class Kaggle environment. Controlled development ablations used a Qwen3.5-4B profile on smaller GPU settings. These models belong to the Qwen3 family, which supports multilingual generation and controllable thinking/non-thinking modes [19]. The paper treats the 35B model as the final deployment backbone and attributes methodological findings to retrieval, packing, verification, and candidate selection rather than to model scale.

Figure 1 summarizes the end-to-end pipeline.

4.1. Formal Design Model

Let q be the question and let $\mathcal{C} = \{c_1, \ldots, c_m\}$ be the chunks obtained from the supplied context. Each chunk has length l_j, normalized retrieval score $\tilde{s}(c_j, q) \in [0, 1]$, language label ℓ_j, and optional financial-statement type t_j. The prompt budget is B, so context construction selects a subset $S \subseteq \mathcal{C}$ with $\sum_{c_j \in S} l_j \leq B$. Let $\mathcal{L}_q$ be the set of languages that may contain relevant news evidence, and let $\mathcal{T}_q$ be the set of financial-statement types required by the question.

The main retrieval failure in this task is not only low relevance but cross-lingual coverage failure. If Cov^{ℓ} denotes the event that the selected context contains answer-bearing evidence in language ℓ, then the evidence event for a multilingual answer satisfies

$$\Pr[\mathrm{Cov}] = \Pr\left[\bigwedge_{\ell \in \mathcal{L}_q} \mathrm{Cov}^{\ell}\right] \leq \min_{\ell \in \mathcal{L}_q} \Pr[\mathrm{Cov}^{\ell}]. \tag{1}$$

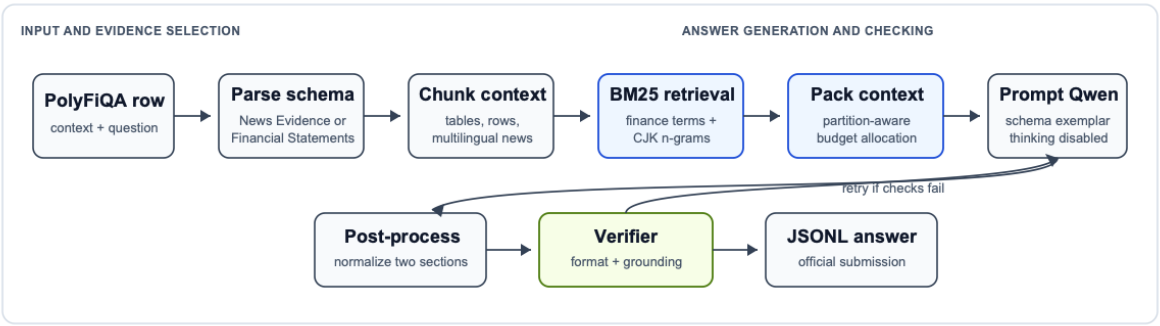

Figure 1: FiSCO-RAG parses the expected evidence schema, chunks the supplied context, retrieves evidence with multilingual BM25, packs a partition-aware context, generates with Qwen in non-thinking mode, and verifies the final two-section answer before writing the JSONL submission.

Equation 1 makes the weakest language the bottleneck: if one CJK or Greek evidence block is invisible to retrieval or evicted by top-k packing, the downstream generator cannot recover the gold evidence verbatim. FiSCO-RAG therefore uses C2 to make language-specific lexical matches observable, C1 to keep required evidence partitions inside the budget, and C3 to preserve specific evidence tokens during candidate selection.

4.2. Layout-Aware Chunking

The raw query is split into retrieval units while preserving financial structure:

- Markdown/ASCII financial tables are kept whole when possible.
- Oversized tables are split on row boundaries.
- News blocks are kept by language marker.
- Remaining narrative text is split with paragraph windows.

This avoids common truncation failures, such as separating a table value from its row label or cutting a multilingual quote in the middle.

4.3. Multilingual BM25 Retrieval

BM25 [11] is used because the official metric rewards lexical overlap and many gold answers quote evidence verbatim. The tokenizer combines:

- whitespace/Unicode word tokens for English, Spanish, Greek, numbers, and financial terms;
- character unigrams and bigrams for Chinese and Japanese.

The CJK n-grams are important because whitespace tokenization under-represents Chinese and Japanese news. The retriever also expands common finance terms; for example, “cash from operations” is linked to “net cash provided by operating activities”.

C2: Entropy-calibrated n-grams. For a language ℓ, let H_ℓ be its empirical character entropy rate in bits per character and let L_0 be the target collision scale for a retrieval unit. A compact information-theoretic rule chooses the shortest n-gram length whose typical set has enough capacity:

$$n_\ell^\star = \left\lceil \frac{\log_2 L_0}{H_\ell} \right\rceil . \tag{2}$$

We use $L_0 = 2^{10} = 1024$, a one-kilo-unit collision target chosen before development scoring to match the order of magnitude of candidate retrieval units inside one PolyFiQA row. The intent is local discrimination among chunks within a long query, not corpus-level language modelling. This value gives $\lceil 10/H_\ell \rceil = 2$ for Chinese and Japanese and 3 for the alphabetic-script languages in Table 2; it was not tuned on the blind test or on held-out leaderboard feedback. High-entropy scripts such as Chinese and Japanese therefore need fewer characters to form discriminative lexical units, while lower-entropy alphabetic scripts are better handled with word tokens. Table 2 shows the design values used to set the tokenizer.

Table 2
Entropy-calibrated token lengths used as retrieval design targets. Latin and Greek text is primarily represented with word tokens, while CJK text receives character n-grams.

Language	H_ℓ	$n_\ell^\star$	Implementation
Chinese	9.6	2	char unigrams+bigrams
Japanese	7.5	2	char unigrams+bigrams
Greek	3.8	3	word tokens
Spanish	4.0	3	word tokens
English	4.5	3	word tokens

This rule gives a retrieval-side fidelity bound for verbatim evidence. If a chunk c_ℓ contains a source-language phrase p^ℓ with $|p^\ell| \geq n_\ell^\star$, then the tokenizer produces at least $|p^\ell| - n_\ell^\star + 1$ matching n-grams. For a normalized BM25 score $\tilde{s}$, this yields the lower-bound form

$$\tilde{s}(c_\ell, q) \geq \kappa \frac{|p^\ell| - n_\ell^\star + 1}{|\tau_{n_\ell^\star}(c_\ell)|}, \tag{3}$$

where $\tau_{n_\ell^\star}(c_\ell)$ is the n-gram multiset and $\kappa > 0$ absorbs BM25 IDF and length-normalization constants. The bound is not used as a runtime score; it explains why CJK n-grams make answer-bearing chunks visible to the subsequent context selector.

4.4. Controlled Cross-Lingual Semantic Alignment

Lexical retrieval is deliberately the default retrieval signal because the primary metric rewards copied numbers, names, and source-language evidence. However, Task 2 is still cross-lingual at the query–evidence interface: the question is usually written in English, while answer-bearing news may be in Chinese, Japanese, Spanish, or Greek. We therefore treat semantic alignment as a constrained auxiliary signal rather than as a replacement for verbatim retrieval.

In development experiments, top BM25 chunks can be mapped into a multilingual embedding space with BGE-M3 [12]. The dense ranking is fused with the BM25 order by reciprocal rank fusion, and only the fused candidate pool is passed to SCCS. This design has two safeguards. First, semantic alignment can add recall for paraphrased questions or translated financial concepts, but it cannot rewrite the evidence because the final context still contains the original source-language chunks. Second, the verifier and the specificity-aware selector operate on the generated answer after retrieval, so dense alignment is accepted only when the output preserves grounded, high-specificity evidence tokens. This is why the paper reports dense alignment as an optional development component: it improved Easy-style news questions but was not uniformly beneficial for Expert questions, where exact statement wording often matters more than broad semantic similarity.

4.5. Partition-Aware Context Packing

Independent top-k ranking can select several chunks from the same source type while missing the evidence partition required by the answer format. Our packer therefore combines relevance scoring with task partitions:

- if the expected section is `News Evidence`, reserve high-scoring news chunks across source languages when they fit the budget;
- if the question asks for financial statements, reserve chunks matching the requested statement type: income statement, balance sheet, or cash-flow statement;
- fill remaining budget greedily by relevance, table/numeric value, and question intent.

C1: Submodular cross-lingual context selection. The practical packer is modelled as a knapsack-constrained submodular selection problem. Define language and statement-type coverage functions

$$\phi_\ell(S) = \min\left(1, \sum_{c_j \in S} \mathbf{1}[\ell_j = \ell]\mathbf{1}[\tilde{s}(c_j, q) > 0]\right), \quad \psi_t(S) = \min\left(1, \sum_{c_j \in S} \mathbf{1}[t_j = t]\right). \tag{4}$$

The SCCS objective is

$$f(S) = \alpha \sum_{\ell \in \mathcal{L}_q} \phi_\ell(S) + \beta \sum_{t \in \mathcal{T}_q} \psi_t(S) + \gamma \sum_{c_j \in S} \tilde{s}(c_j, q), \qquad \sum_{c_j \in S} l_j \le B, \tag{5}$$

where α controls language coverage, β controls financial-statement type coverage, and γ controls lexical relevance. The functions ϕ_ℓ and ψ_t are truncated coverage functions, and the last term is modular; hence f is monotone submodular. We optimize it with a cost-benefit greedy approximation to the standard knapsack setting [15, 16, 17].

Equation 5 explains the role of the coverage weights. Let $s^\star = \max_j \tilde{s}(c_j, q)$. If an uncovered language ℓ has a positive-scoring chunk that fits the remaining budget and $\alpha > \gamma s^\star$, then adding that missing-language chunk has larger marginal coverage value than any purely relevance-driven single chunk. Under a sufficient budget condition, this makes the selector prefer at least one evidence candidate per required language before spending the remaining budget on relevance. In practice, this condition is used as a design criterion for balancing coverage and relevance rather than as a hard runtime certificate.

This component keeps the selected context faithful to the output schema, while still following the evidence aggregation principle used in retrieval-augmented QA [9].

4.6. Schema-Conditioned Prompting

The prompt always places the question before the context and repeats it near the generation point. This protects the question from right truncation and reduces position-sensitivity in long prompts [10]. The system message requires:

- the two-section answer format;
- exact figures and dates from context;
- source-language quotes copied verbatim;
- no commentary beyond the answer and evidence sections;
- total answer length under 100 words.

A one-shot exemplar is selected according to the expected evidence section.

4.7. Generation, Post-Processing, and Verification

Generation uses the Qwen chat template with `enable_thinking=False` [19]. This was important because thinking traces consume the output budget and reduce format adherence.

Post-processing then removes residual preambles or thinking blocks, anchors on the first `Answer:`, normalizes the evidence-section header to the expected section, removes duplicate answer blocks, and truncates the output to 100 words.

The verifier checks the answer prefix and expected evidence-section header, nonempty answer and evidence body, word-count compliance, character n-gram overlap between evidence text and selected context, and source-language presence for news-evidence answers. If verifier checks fail, the local runner can resample with a small temperature and keep the best verified candidate.

C3: Specificity-aware MBR candidate selection. When multiple candidates are available, the verifier first filters them into $\mathcal{Y}^+$. A vanilla MBR selector chooses the candidate with highest average similarity to the rest of the candidate set:

$$\bar{\rho}(y) = \frac{1}{|\mathcal{Y}^+| - 1} \sum_{\substack{y' \in \mathcal{Y}^+ \\ y' \neq y}} \rho(y, y'), \tag{6}$$

where ρ is ROUGE-1 or a hybrid lexical-semantic similarity. This objective can drift toward generic consensus answers when the candidate pool is large: short, common English tokens agree with many candidates, while foreign-script quotes and rare financial identifiers may reduce self-overlap.

We therefore regularize MBR with an answer specificity term. Let $T(y)$ be the set of evidence tokens in candidate y, and let $\mathrm{idf}(w) = \log((|\mathcal{D}| + 1)/(\mathrm{df}(w) + 1))$ be computed from the retrieved context collection $\mathcal{D}$. Define

$$\mathrm{Spec}(y) = \frac{1}{|T(y)|} \sum_{w \in T(y)} \mathrm{idf}(w). \tag{7}$$

The specificity-aware selector is

$$\hat{y} = \arg\max_{y \in \mathcal{Y}^+} \left[\bar{\rho}(y) + \lambda \, \mathrm{Spec}(y) \right]. \tag{8}$$

The parameter λ has a direct interpretation: $\lambda = 0$ recovers vanilla MBR, while larger values favor candidates that preserve rare evidence tokens such as foreign-script quotes, dates, named entities, and financial figures. The verifier is kept before Eq. 8 so that specificity does not reward unsupported rare tokens. In the reported final-answer experiments, the measured effect is the verifier-filtered MBR candidate pool itself; the IDF specificity term in Eq. 8 is the formal C3 regularizer and is not claimed as a separately isolated leaderboard gain.

5. Submitted Configuration

The official leaderboard submission was generated with the local Task 2 runner on the 256-row blind test file. The submitted configuration uses the settings in Table 3.

Development and smoke tests used a Qwen3.5-4B P100-compatible profile. The final submission used the larger RTX6000 profile to allow longer retrieved contexts. Since we did not run a controlled 35B-versus-4B ablation, the leaderboard score should be interpreted as the result of the full submitted deployment stack, not as evidence that model scaling alone caused the gain.

6. Results

6.1. Official Leaderboard Result

AI_TLfanClub ranked fourth among completed Task 2 submissions. Table 4 shows the top five official results available from the leaderboard snapshot provided by the organizers.

The precision-recall profile shows that our system produced relatively precise answers compared with higher-recall teams. This is consistent with our design: the system aggressively controls format and evidence grounding under the 100-word cap, which improves precision but can miss some reference evidence.

Table 3
Submitted configuration.

Component	Submitted value
Backbone	Qwen3.6-A3B-35B MoE
Precision	bfloat16
Quantization	none
Context length	24,576 tokens
Packed context	18,000 characters
Input token cap	13,500 tokens
Attention	eager
Retrieval/packing	C2 BM25 retrieval + C1 SCCS packing
Prompting	schema-conditioned prompt with section-aware exemplar
Generation	max 256 new tokens, stop strings enabled
Candidate selection	verifier-filtered generation; C3 MBR in multi-sample runs
Verifier	enabled, max 2 attempts, min verbatim ratio 0.35
Training	none
External web retrieval	none

Table 4
Official Task 2 leaderboard snapshot, top five completed submissions. The primary metric is ROUGE-1 F1.

Rank	Team	F1	Precision	Recall	Coverage	Updated
1	Calibrated Signals	31.18%	0.3025	0.3764	100.00%	2026-05-14
2	pranshu rastogi	30.96%	0.3208	0.3537	100.00%	2026-05-21
3	IGT	30.71%	0.2821	0.4044	100.00%	2026-05-20
4	AI_TLfanClub	30.39%	0.3647	0.3070	100.00%	2026-05-15
5	PooRi	29.72%	0.2781	0.3798	100.00%	2026-05-14

6.2. Internal Public-Gold Results

On the 76+76 public-gold development data, the retrieval improvements produced large gains over a structured-prompt baseline.

Table 5
Internal public-gold development results. Scores are ROUGE-1 F1.

Configuration	Easy	Expert	Overall
Structured evidence prompt, no RAG	0.2310	0.1639	0.1975
BM25 retrieval only	0.4046	0.2720	0.3383
BM25 retrieval + BGE reranker	0.4173	0.2685	0.3429
Mixed dev configuration	0.4194	0.2724	0.3452

The mixed dev configuration combines BM25 retrieval, optional BGE reranking for Easy-style news cases, partition-aware packing, and verifier-based output selection. The reranker helped Easy, where question wording and news evidence were better aligned, but slightly hurt Expert. We therefore treated dense/reranker components as optional rather than mandatory.

6.3. Retrieval-Side Ablation

To isolate C1 and C2 without rerunning expensive LLM decoding, we also measured retrieval-side coverage on all 152 public-gold rows. Each variant selected an 18k-character context, and we compared the selected context with the gold answer. Gold-token recall is a ROUGE-style recall proxy over content tokens; script recall measures non-English source-script characters from the gold answer that survive in the selected context. Easy news coverage is the average fraction of {zh,jp,es,el} news languages present

in the selected context, and Expert statement hit is the fraction of questions whose target statement type appears in the selected context.

Table 6
Retrieval-side ablation on the 152 public-gold rows. These are context-coverage proxies, not final generation ROUGE scores.

Selector	Gold recall	Easy recall	Expert recall	Script recall	Coverage hit
Word-only BM25 top-k	0.4241	0.4083	0.4398	0.1302	0.0329 / 0.3947
CJK-aware BM25 top-k	0.4324	0.4160	0.4488	0.2370	0.1184 / 0.3947
C2 BM25 + C1 SCCS	0.4451	0.4690	0.4212	0.9513	0.9079 / 1.0000

The proxy confirms the expected behavior. CJK-aware tokenization improves source-script retention over word-only BM25, while SCCS strongly increases schema-relevant coverage: Easy multilingual news coverage rises from 0.1184 to 0.9079, and Expert target-statement coverage rises from 0.3947 to 1.0000. The lower Expert gold-token recall for SCCS is a useful warning rather than a contradiction: under a fixed budget, statement reservation can evict nearby lexically overlapping text while preserving the financial-statement partition needed by the prompt and verifier.

6.4. MBR and Sampling Ablation

We also tested candidate selection strategies on a 40-item development subset (20 Easy + 20 Expert) with a Qwen3.5-4B P100 profile. MBR denotes minimum Bayes risk candidate selection [13, 14].

Table 7
MBR and sampling ablation on a 40-item development subset. Scores are ROUGE-1 F1.

Stack	Easy	Expert	Overall	Runtime
Greedy + verifier(2)	0.3023	0.1868	0.2445	948 s
Qwen sampling recipe + verifier(2)	0.2946	0.1989	0.2467	1007 s
MBR-N=4	0.3075	0.2053	0.2564	3154 s
MBR-N=4 + Qwen recipe	0.3109	0.2056	0.2583	3085 s
MBR-N=8	0.2988	0.1906	0.2447	5198 s

The main observation is that moderate candidate selection helps, but larger candidate pools can drift toward generic consensus answers and increase runtime. This empirical regression is the motivation for the IDF-regularised selector in Eq. 8: the consensus term keeps stable answers, while the specificity term protects rare multilingual evidence from being averaged away.

6.5. Component-Level Evidence and Remaining Gaps

Table 8 summarizes the component-level evidence available at submission time. Retrieval and candidate-selection gains are measured directly. C1 and C2 are additionally isolated with the retrieval proxy in Table 6; C3 still lacks a clean final-answer comparison between vanilla MBR and IDF-regularised MBR under the same candidate pool.

This table makes the empirical limitation explicit: the strongest measured effect is retrieval; C1 and C2 now have retrieval-side isolation, while C3 is implemented in the final stack but still needs a clean one-factor final-answer ablation against vanilla MBR.

7. Discussion

7.1. What Worked

Layout-aware BM25 retrieval was the largest gain. The jump from structured prompting to BM25 retrieval on public-gold development data was much larger than gains from decoding changes. This

Table 8
Component-level evidence from development experiments. Scores are ROUGE-1 F1 unless otherwise stated.

Question	Current evidence	Remaining isolation needed
Does retrieval help?	No-RAG 0.1975 → BM25 0.3383 overall	Already measured
Does C2 tokenization help?	Word-only top-k 0.1302 → CJK top-k 0.2370 script recall	Retrieval proxy; generation not rerun
Does C1 packing help?	CJK top-k 0.1184 → SCCS 0.9079 Easy news coverage; 0.3947 → 1.0000 Expert statement hit	Retrieval proxy; generation not rerun
Does dense reranking help?	BM25 0.3383 → BM25+BGE 0.3429 overall	Already measured
Does full packing/verification help?	BM25+BGE 0.3429 → mixed stack 0.3452 overall	Split C1 from verifier
Does MBR help?	Greedy 0.2445 → MBR-N=4 0.2564 on 40-item subset	Split vanilla MBR from IDF-MBR
Does larger MBR help?	MBR-N=4 0.2564 → MBR-N=8 0.2447	Already measured

confirms that evidence selection is the main bottleneck.

CJK-aware tokenization was necessary. Chinese and Japanese news are present in all development and blind-test rows. Character n-grams give these chunks a lexical retrieval signal without translating the evidence.

Schema enforcement reduced avoidable loss. The fixed `Answer:` plus evidence-section format is part of the task. Post-processing that anchors the answer and normalizes the expected section removed common free-form generation failures.

Verifier-based filtering was useful but not sufficient. It catches unsupported quotes and wrong headers, but it does not prove that a financial claim follows from the evidence.

7.2. What Did Not Always Work

Dense reranking was mixed. It improved Easy but slightly reduced Expert ROUGE in development. One likely reason is that semantic relevance does not always align with the exact wording used in the gold answer.

Semantic alignment should remain constrained by verbatim evidence. Cross-lingual dense embeddings are useful for mapping English questions to non-English news, but an unconstrained semantic space can prefer a topically similar passage whose wording does not match the reference. For this reason, our alignment design is retrieval-side only: it expands the candidate pool, then SCCS, source-script preservation, verifier overlap, and specificity-aware MBR decide whether the evidence is specific enough to survive into the final answer.

Large MBR pools regressed. MBR-N=8 was slower and had lower overall ROUGE-1 than MBR-N=4 in the development subset (0.2447 versus 0.2564), and also lower than MBR-N=4 with the Qwen sampling recipe (0.2583). For this task, more candidates can increase generic consensus rather than improve evidence specificity.

High precision came with lower recall. The official score shows precision 0.3647 and recall 0.3070. Our answer control likely kept many predictions concise and on-format, but the 100-word cap and verifier pressure may have reduced the amount of reference evidence copied.

8. Error Analysis

We grouped observed errors into the following categories.

1. **Missing evidence in the packed context.** If the relevant table row or news quote is not retrieved, the generator falls back to partial answers or `None.` evidence.

2. **Evidence present but paraphrased.** The model sometimes translates or summarizes multilingual evidence rather than copying it. This hurts ROUGE even when the meaning is close.
3. **Numeric drift.** The model may round values, convert units, or attach a number to the wrong period.
4. **Wrong evidence granularity.** Long news chunks can contain several relevant facts; the answer may quote a nearby but not exact sentence.
5. **Expert mismatch and ambiguity.** Expert questions often require combining multiple statement lines, and some references appear sensitive to wording choices that are hard to infer from semantic retrieval alone.
6. **Over-short answers.** The official precision-recall profile suggests that our system was conservative. This helped precision but could leave out reference details.

As a lightweight quantitative diagnostic, Table 9 reports verifier reasons from the verbose blind-test output. These are not gold error labels, but they indicate where the pipeline spent correction effort.

Table 9
Automatic verifier diagnostics on 256 blind-test outputs.

Diagnostic reason	Count	Interpretation
MBR-selected candidate	199	Multi-sample selection changed or confirmed the candidate
Verifier passed directly	21	Initial candidate satisfied format and grounding checks
Low verbatim overlap	28	Evidence text was weakly supported by selected context
Missing language coverage	6	News evidence missed two or more detected source languages
Too long	2	Output exceeded the verifier length tolerance before truncation

9. Reproducibility Notes

9.1. Implementation Entry Point

All results were produced with the same local inference runner. The runner loads the blind-test JSONL file, constructs the retrieved context, generates an answer, applies post-processing and verification, and writes the official two-field submission JSONL containing only task identifiers and answers. We avoid reporting machine-specific file names or local storage paths because they are not part of the method.

9.2. Environment

Table 10 summarizes the execution environment and the main implementation constraints relevant to reproducibility. We report the software stack, the development and submission hardware profiles, and the fact that the submitted system remained inference-only.

The lexical retriever used the `rank-bm25` Python package [20].

9.3. Hyperparameters

The SCCS weights α, β, and γ are best read as design weights rather than dataset-fitted parameters: α rewards language coverage, β rewards financial-statement type coverage, and γ rewards lexical relevance. In the submitted implementation these roles are realized through reservation budgets and boost magnitudes: news chunks receive a strong positive boost for news questions, statement-type matches receive the largest filing-side boost, and BM25 scores are normalized before structural boosts

Table 10
Execution environment and implementation notes.

Item	Value
Python	3.11-class Kaggle environment
Core libraries	`torch`, `transformers`, `datasets`, `rank-bm25`, `tqdm`, `pyyaml`
Optional libraries	`sentence-transformers` for dense/reranker experiments
Development GPU	P100 16GB for Qwen3.5-4B control runs
Submission GPU	RTX 6000-class GPU for Qwen3.6-A3B-35B MoE
Random seed	42 for development splits/sampling where applicable
Fine-tuning	none
External retrieval	none

are applied. This keeps the number of tuned free parameters small while making the sufficient-condition discussion in Eq. 5 operational.

Tables 11 and 12 summarize the concrete submitted settings for model loading, prompt budgeting, verification, retrieval, packing, and candidate selection.

Table 11
Model, prompt, generation, and verifier hyperparameters in the submitted RTX6000 profile.

Component	Parameter	Submitted/main value or rule
Data	Dataset split	PolyFiQA-Easy/Expert test split; blind-test runner used 256 local rows
Data	Random seed	42 for deterministic development splits and sampling setup
Model	Backbone	Qwen3.6-A3B-35B MoE; no adapter; no fine-tuning
Model	Precision/quantization	bfloat16, no quantization in the submitted profile
Model	Context/attention	24,576-token context, `eager` attention, `device_map=auto`
Prompt	Strategy	`bm25_rag`; question-first prompt; question repeated before generation
Prompt	Exemplar and output cap	Section-aware one-shot exemplar enabled; answer capped at 100 words
Prompt	Packed context budget	18,000 characters and 13,500 tokenized input tokens
Generation	Decoding recipe	max 220–256 new tokens; temperature 0.7, top-p 0.8, top-k 20 when sampling is enabled; repetition penalty 1.0
Generation	Stop/bad strings	Stop at next-question/chat terminators; block common hedge preambles such as "As an AI"
Verification	Attempts and resampling	Verifier enabled; maximum 2 attempts; retry temperature 0.4
Verification	Grounding threshold	Evidence must reach 0.35 character-7-gram verbatim overlap with the selected context
Candidate selection C3	MBR pool	Submitted local profile used verifier-best answer plus up to 6 sampled candidates when MBR was enabled
Candidate selection C3	MBR metric	Hybrid self-consensus: ROUGE-1 agreement with optional multilingual embedding cosine fallback/fusion
Candidate selection C3	Specificity weight	Formal IDF term in Eq. 8; $\lambda = 0$ recovers vanilla MBR and larger values favor rare evidence tokens

9.4. Training Data and External Data

The submitted system is inference-only. We did not train or fine-tune on PolyFiQA, and we did not use external supervised data. PolyFiQA Easy and Expert public HuggingFace releases provide test/gold data but no official train split. The one-shot exemplars are fixed public/development-format examples used only to specify the required answer schema; they were not selected from blind-test rows or from model predictions on the blind test.

Table 12
Retrieval, packing, and candidate-selection hyperparameters.

Component	Parameter	Submitted/main value or rule
Chunking	Oversize split	Tables/news chunks larger than the working budget are row/head split; large chunks target about 3,500 characters
Retrieval C2	BM25 implementation	`BM25Okapi` with default $k_1 = 1.5, b = 0.75, \epsilon = 0.25$
Retrieval C2	Tokenization	Unicode word tokens for Latin/Greek text; CJK character unigrams and bigrams
Retrieval C2	Query expansion	Finance lexicon for revenue, earnings, balance sheet, cash flow, capital allocation, periods, and entities
Packing C1	Initial retrieval width	`top_k`=10; final packed context may contain up to `top_k`×4 chunks
Packing C1	News reservation	For news questions, reserve one high-scoring news chunk per detected language; Easy profile uses up to 75% of the context budget
Packing C1	Statement reservation	For filing questions, reserve the target statement type and then fill other statement types if budget permits
Packing C1	Greedy stopping	Stop after coverage is satisfied, at least two chunks are selected, 85% of budget is used, and two consecutive chunks fall below the value floor
Candidate selection C3	MBR metric	Hybrid self-consensus: ROUGE-1 agreement with optional multilingual embedding cosine fallback/fusion
Candidate selection C3	Specificity weight	Formal IDF term in Eq. 8; $\lambda = 0$ recovers vanilla MBR and larger values favor rare evidence tokens

9.5. Prompt Template

The listing below shows the inference prompt skeleton used by the submitted `bm25_rag` strategy. The concrete evidence section name and one-shot exemplar are selected before inference from the expected answer format.

Listing 1: Schema-conditioned user-message template used during inference.

```
Question: {question}

Required output format:
Answer: <facts with exact figures>.
{section_name}: <verbatim quote(s); preserve original language for non-English; or
    "None.">

Example (different question, same format):
Question: {ex_question}
{ex_answer}

Now answer this question using ONLY the financial context below. Keep it <= 100
    words.

[Context]
{context}

[End of Context]

Question (repeat): {question}

Your answer:
```

The system message fixes the role as an expert multilingual financial analyst, restricts evidence to the supplied context, requires exactly two sections (`Answer:` plus the resolved evidence-section name), requires exact figures and dates, asks the model to copy Chinese/Japanese/Spanish/Greek quotes verbatim with language labels, permits `None.` only when no relevant evidence exists, caps the total output at 100 words, and forbids extra commentary or duplicate answer blocks. The Qwen chat template

is then applied with `enable_thinking=False`.

10. System Card Summary

Model. The submitted configuration used a Qwen3.6-A3B-35B MoE model as the generation backbone. Development experiments used a Qwen3.5-4B profile [19]. We do not report a controlled model-size ablation.

Retrieval. All evidence was retrieved from the supplied query. No web search or external financial database was used.

Training. No task-specific training or LoRA adapter was used for the submitted run.

Known risks. The system can still miss evidence in very long contexts, mis-copy multilingual text, round numbers incorrectly, or output evidence that overlaps the context but does not fully support the financial conclusion.

Licenses. PolyFiQA is distributed by the FinMMEval organizers. Qwen models and retrieval libraries retain their original licenses and citation requirements.

11. Conclusion

AI_TLfanClub submitted an inference-only RAG system to FinMMEval 2026 Task 2 and ranked fourth on the official leaderboard with ROUGE-1 F1 30.39% and full coverage. The main contribution is not a model-scaling claim, but a cross-lingual evidence pipeline: layout-aware chunking, multilingual BM25, partition-aware context packing, strict schema prompting, lightweight verification, and specificity-aware candidate selection.

The competition highlighted a practical lesson for multilingual financial QA: when references require verbatim evidence under a short answer cap, retrieval and output control dominate. The remaining headroom lies mainly in numeric verification, semantic support checking, and recall-oriented answer expansion that preserves format precision.

Acknowledgments

The authors would like to express their gratitude to Associate Professor Dr. Le Thanh Huong (Hanoi University of Science and Technology) for her invaluable guidance and support. Furthermore, we thank the CLEF 2026 FinMMEval organizers for preparing the shared task, evaluation resources, and official leaderboards.

Declaration on Generative AI

During the preparation of this work, the author used large language model tools for text preparation assistance, language editing, and LaTeX formatting. The author reviewed and edited the content and takes full responsibility for the publication's content.

References

[1] Z. Xie, Y. Dai, R. Elbadry, V. Jani, X. Peng, L. Qian, G. Georgiev, D. Dimitrov, F. Zhang, J. Huang, J. Geng, Y. Chen, Y. Yuan, H. Wu, Y. Wang, I. Koychev, V. Stoyanov, M. Song, Y. Chen, X. Liu, P. Nakov, Overview of FinMMEval 2026: Multilingual and multimodal financial evaluation, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction, Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Springer Lecture Notes in Computer Science LNCS, Jena, Germany, 2026.

[2] Z. Xie, X. Peng, G. Georgiev, D. Dimitrov, R. Elbadry, F. Zhang, L. Qian, J. Huang, V. Jani, Y. Dai, J. Geng, Y. Chen, Y. Yuan, H. Wu, Y. Wang, I. Koychev, V. Stoyanov, M. Song, Y. Chen, X. Liu, P. Nakov, Overview of the FinMMEval 2026 task 2: Financial question answering and summarization, in: CLEF 2026 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, Jena, Germany, 2026.

[3] X. Peng, L. Qian, Y. Wang, R. Xiang, Y. He, Y. Ren, M. Jiang, V. J. Zhang, Y. Guo, J. Zhao, H. He, Y. Han, Y. Feng, Y. Jiang, Y. Cao, H. Li, Y. Yu, X. Wang, P. Gao, S. Lin, K. Wang, S. Yang, Y. Zhao, Z. Liu, P. Lu, J. Huang, S. Wang, T. Papadopoulos, P. Giannouris, E. Soufleri, N. Chen, Z. Deng, H. Fu, Y. Zhao, M. Lin, M. Qiu, K. E. Smith, A. Cohan, X.-Y. Liu, J. Huang, G. Xiong, A. Lopez-Lira, X. Chen, J. Tsujii, J.-Y. Nie, S. Ananiadou, Q. Xie, MultiFinBen: Benchmarking large language models for multilingual and multimodal financial application, 2025. URL: https://arxiv.org/abs/2506.14028. doi:10.48550/arXiv.2506.14028. arXiv:2506.14028.

[4] Z. Xie, et al., FinMMEval: A multimodal, multilingual financial benchmark for large language models, arXiv preprint arXiv:2602.10886 (2026). URL: https://arxiv.org/abs/2602.10886.

[5] Z. Chen, W. Chen, C. Smiley, S. Shah, I. Borova, D. Langdon, R. Moussa, M. Beane, T.-H. Huang, B. Routledge, W. Y. Wang, FinQA: A dataset of numerical reasoning over financial data, in: Proceedings of EMNLP, 2021. URL: https://aclanthology.org/2021.emnlp-main.300/.

[6] Z. Chen, W. Chen, A. Gupta, T.-H. Huang, W. Y. Wang, ConvFinQA: Exploring the chain of numerical reasoning in conversational finance question answering, in: Proceedings of EMNLP, 2022. URL: https://aclanthology.org/2022.emnlp-main.421/.

[7] F. Zhu, W. Lei, Y. Huang, C. Wang, S. Zhang, J. Lv, F. Feng, T.-S. Chua, TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance, in: Proceedings of ACL-IJCNLP, 2021. URL: https://aclanthology.org/2021.acl-long.254/.

[8] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, in: Advances in Neural Information Processing Systems, 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

[9] G. Izacard, E. Grave, Leveraging passage retrieval with generative models for open domain question answering, in: Proceedings of EACL, 2021. URL: https://aclanthology.org/2021.eacl-main.74/.

[10] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, P. Liang, Lost in the middle: How language models use long contexts, Transactions of the Association for Computational Linguistics (2024). URL: https://aclanthology.org/2024.tacl-1.9/.

[11] S. Robertson, H. Zaragoza, The probabilistic relevance framework: Bm25 and beyond, Foundations and Trends in Information Retrieval (2009). URL: https://doi.org/10.1561/1500000019.

[12] 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). URL: https://arxiv.org/abs/2402.03216.

[13] B. Eikema, W. Aziz, Is map decoding all you need? the inadequacy of the mode in neural machine translation, in: Proceedings of COLING, 2020. URL: https://aclanthology.org/2020.coling-main.398/.

[14] M. Freitag, G. Foster, D. Grangier, V. Ratnakar, Q. Tan, W. Macherey, High quality rather than high model probability: Minimum bayes risk decoding with neural metrics, Transactions of the Association for Computational Linguistics (2022). URL: https://aclanthology.org/2022.tacl-1.36/.

[15] G. L. Nemhauser, L. A. Wolsey, M. L. Fisher, An analysis of approximations for maximizing submodular set functions–i, Mathematical Programming 14 (1978) 265–294. doi:10.1007/BF01588971.

[16] S. Khuller, A. Moss, J. S. Naor, The budgeted maximum coverage problem, Information Processing Letters 70 (1999) 39–45. doi:10.1016/S0020-0190(99)00031-9.
[17] M. Sviridenko, A note on maximizing a submodular set function subject to a knapsack constraint, Operations Research Letters 32 (2004) 41–43. doi:10.1016/S0167-6377(03)00062-2.
[18] C. E. Shannon, A mathematical theory of communication, Bell System Technical Journal 27 (1948) 379–423. doi:10.1002/j.1538-7305.1948.tb01338.x.
[19] Qwen Team, Qwen3 technical report, arXiv preprint arXiv:2505.09388 (2025). URL: https://arxiv.org/abs/2505.09388.
[20] D. Brown, Rank-BM25: A collection of BM25 algorithms in python, 2020. URL: https://github.com/dorianbrown/rank_bm25.