MAIS: A Multi-Agent Iterative Framework for Scientific Query-Focused Summarization

Avreymi Asraf and **Hillel Darshan**
The Hebrew University of Jerusalem
{avraham.asraf, hillel.darshan}@mail.huji.ac.il

Abstract

Scientific Query-Focused Summarization (Sci-QFS) is a critical but challenging task for researchers, as current Large Language Models (LLMs) often produce factually inconsistent summaries and require extensive training data. To address this, we propose MAIS (Multi-Agent Iterative Summarization), a zero-shot framework that requires no training. MAIS employs a team of four specialized agents that collaboratively generate, critique, and progressively refine a summary. Its core contribution is a self-correcting mechanism where a Judge agent uses a ReAct paradigm to evaluate the summary against the source document and provides targeted feedback for revision, integrating evaluation directly into the generation process. We evaluated MAIS on a dataset of 61 paper-query pairs, demonstrating that the system successfully converged on a judge-approved summary in 90% of cases, requiring an average of only 2.87 iterations. Analysis of per-iteration accuracy reveals a dynamic, non-monotonic refinement process, confirming that the feedback loop actively improves summary faithfulness over time. Our work shows that agent-based iterative refinement is a promising approach for creating more reliable and trustworthy summarization tools for scientific research.

1 Introduction

Efficiently reviewing scientific literature is a critical but time-consuming challenge for researchers. This necessitates tools for **Scientific Query-Focused Summarization (Sci-QFS)** to answer targeted questions about dense academic papers. Current Large Language Models (LLMs) struggle with this task due to the high demand for factual accuracy, the complexity of technical content, and the lack of specialized training data. To address this, we propose MAIS (Multi-Agent Iterative Summarization), a zero-shot framework of four agents that collaboratively generate and refine a summary in an iterative loop. Its key contribution is a self-correcting architecture that mimics critical reading by integrating evaluation directly into the generation process to produce faithful, query-focused summaries without supervision.

2 Related Work

Prior work in QFS addresses challenges in data, complexity, and evaluation. To get over lack of data issues, systems like LMGQS (Xu et al., 2023) synthetically generate training data. To tackle complexity, zero-shot architectures like BRD's (Zhang et al., 2024) retrieve-then-generate pipeline or Multi²'s (Cao et al., 2025) parallel agent aggregation have been proposed. MAIS is also a zero-shot, multi-agent system but is novel in its use of an **iterative refinement loop** to improve a single summary. This iterative approach extends to evaluation. While systems often use post-hoc metrics ranging from ROUGE (Lin, 2004) to modern LLM-based evaluators that check for "atomic content units (ACU)" (Liu et al., 2023), MAIS integrates evaluation **within the generation loop**. Its Judge agent provides real-time feedback, transforming evaluation from a final score into an active refinement mechanism.

3 The MAIS Framework

MAIS is a zero-shot, multi-agent framework designed for Sci-QFS. It operates without training data and consists of four specialized LLM agents working in a collaborative loop to generate and refine a summary. Our implementation uses a hybrid-model approach: we leverage the powerful Gemini 2.5 Pro for the critical, one-time question generation task, while the iterative components of the loop (Summarizer, QA Agent, and Judge) use the faster and more efficient Gemini 2.5 Flash-Lite model. The overall workflow is illustrated in Figure 1. The process begins with an initialization step:

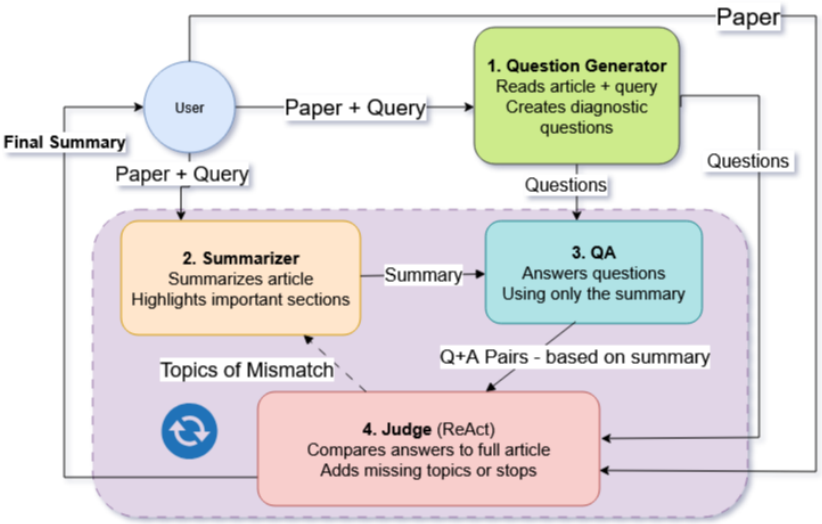

Figure 1: The iterative refinement loop of the MAIS framework. The process begins with question generation, then enters a loop of summarizing, self-assessing (QA), and judging until the summary is approved.

1. Question Generation: The process begins with a critical initialization step performed by the **Question Generator**. Recognizing that the quality of the diagnostic questions underpins the entire framework's success, we employ the more powerful Gemini 2.5 Pro model for this specific task. The agent analyzes the source article and user query to produce a static set of up to seven high-quality diagnostic questions. While this model is more computationally intensive, its one-time execution at the start of the process justifies the cost, as it establishes a robust evaluation benchmark for all subsequent iterations.

Once the questions are set, the system enters the iterative refinement loop:

2. Summarization: The **Summarizer** agent generates a summary. On the first pass, it creates a baseline summary. On subsequent passes, it receives "Topics of Mismatch" from the Judge and is tasked with revising the summary to address these specific deficiencies.

3. QA Self-Assessment: The **QA Agent** attempts to answer the diagnostic questions using *only the current summary* as its source of truth. This produces a set of QA pairs that reflect the summary's current knowledge and gaps.

4. Judgment and Refinement: The **Judge** agent is the core controller of the loop, employing a **ReAct (Reasoning and Acting)** (Yao et al., 2023) paradigm to guide the refinement process. Its **Reasoning** step involves critically comparing the QA pairs (from the summary) against the ground-truth source article to identify any factual discrepancies or omissions. Based on this reasoning, the Judge then takes one of two **Actions**:

- **Iterate:** If discrepancies are found, it generates "Topics of Mismatch" and sends them back to the Summarizer for another refinement cycle.
- **Terminate:** If the summary is judged faithful and complete, it approves the summary, the loop concludes, and the final version is returned to the user.

This self-correction mechanism ensures that the final summary has undergone a rigorous internal validation process for faithfulness and completeness before being finalized.

4 Experimental Setup

Our evaluation is designed to demonstrate the primary strengths of the MAIS framework: its ability to generate high-quality, query-adaptive summaries and the effectiveness of its self-refinement loop. The evaluation combines a detailed qualitative case study with a quantitative analysis of the system's internal convergence metric.

4.1 Dataset

To create a comprehensive dataset, we began with a set of 13 scientific papers from the field of Artificial Intelligence. For each paper, we crafted 5 distinct queries, resulting in a total of 65 initial paper-query pairs. During the experimental runs, a subset of these pairs was excluded due to technical failures of non-recoverable LLM API errors. After this filtering process, we obtained a final, clean dataset of **61 successful runs across 13 unique papers.**

4.2. Evaluation Methodology Our evaluation focuses on two key aspects:

System-Internal Metric: Convergence Rate. To quantitatively measure the efficiency and effectiveness of our self-correction loop, we analyzed the results from the 61 successful runs in our final dataset. For each run, we recorded the number of iterations required before the Judge agent approved the summary. We also tracked the termination status: whether the process concluded with a completed status (Judge's approval) or by reaching the maximum iteration limit (max_iterations_reached). A high completion rate with a low average number of iterations serves as strong evidence that our iterative refinement mechanism is both effective and efficient.

Qualitative Case Study: Query Adaptability To provide a concrete example of the system's query-focused capabilities, we performed an in-depth case study on a single cornerstone paper using three distinct queries.

The analysis revealed a high degree of adaptability. Each of the three generated summaries focused on different aspects of the source text, highlighting details and sections that were uniquely relevant to its specific query while omitting information that was central to the others. This confirms that MAIS successfully alters the content and focus of its output to align with the user's specific information need. The full text of the generated summaries is provided for review in Appendix A. (All other data is available on git).

5 Results and Analysis

In this section, we present the results of our experiments. We first analyze the overall convergence rate of the MAIS framework and then provide evidence for the effectiveness of the iterative refinement loop by tracking the improvement in summary accuracy over time.

5.1 Convergence Rate and Efficiency

The primary measure of our system's success is its ability to reach a satisfactory summary approved by the Judge agent. We analyzed the performance across **61 unique paper-query pairs** in our final dataset, with a maximum limit of 15 iterations per run. The overall results are summarized in Table 1.

Table 1: Overall performance of the MAIS framework across 61 runs.

Metric	**Count**	**Percentage**
Total Runs Evaluated	61	100%
Successful Completions (Judge Approved)	55	90.2%
Single-Iteration Completions	26	42.6%
Avg. Iterations (for completed runs)	2.87	–
Var. Iterations (for completed runs)	7.7	–

As shown in Table 1, the system successfully converged in **90 %** of cases. For these successful runs, the framework required an average of only **2.87** iterations to produce a high-quality, faithful summary. This demonstrates that our self-correction mechanism is both effective in its goal and efficient in its execution. The high variance of **7.7** shows the task difficulty varies between papers and between queries on the same paper. The few cases that reached the maximum iteration limit often involved highly complex or ambiguous source material or queries, a point we return to in our limitations.

5.2 Effectiveness of Iterative Refinement

To validate that our iterative loop improves summary quality, we measured how the internal accuracy scores evolved over successive iterations. We tracked the average accuracy of both the general diagnostic questions and the more challenging ACU-style questions across all runs.

The results, plotted in Figure 2, reveal the dynamic nature of the refinement process.

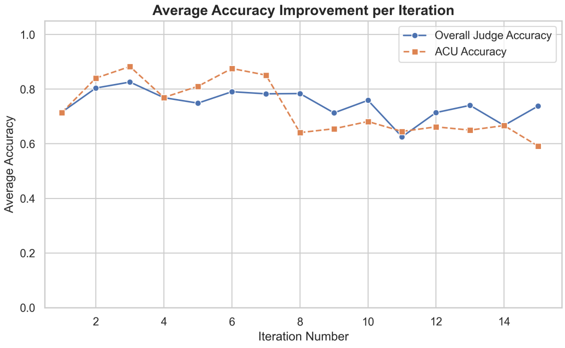

Figure 2: Average accuracy scores per iteration. The plot shows a general trend of improvement, characterized by a dynamic, non-monotonic refinement process where the system corrects and sometimes over-corrects in response to feedback.

The plot shows a significant improvement in the first few iterations, with both Overall and ACU accuracy peaking around iteration 3. This demonstrates the immediate and powerful impact of the initial feedback rounds. Following this peak, the process becomes more exploratory; the Summarizer attempts to integrate complex feedback, which can sometimes cause temporary dips in accuracy before recovering. Furthermore, as we pass the average iteration of termination, only the harder summarizations are left - and the accuracy is lower.

Crucially, the iterative process consistently guides the summary from a lower-quality initial state (average accuracy of $\tilde{0}.72$ at Iteration 1) to a generally higher-quality state in later iterations. The greater volatility of the ACU Accuracy line suggests that satisfying these fine-grained factual constraints is the most challenging aspect of the refinement task. This provides strong evidence that our framework is working as intended, actively engaging in a complex refinement process to make the summary more factually complete and correct.

6 Conclusion, Limitations, and Future Work

In this paper, we introduced MAIS, a zero-shot, multi-agent framework for Scientific Query-Focused Summarization. We identified key challenges in Sci-QFS, including the need for high factual accuracy and the lack of specialized training data. Our proposed system addresses these with a novel iterative refinement loop, where agents collaborate to generate, critique, and progressively improve a summary until it is verified by an internal Judge agent. Our experiments suggest this self-correcting approach is effective at generating query-adaptive summaries and efficiently converging on a faithful output.

Limitations. Despite its promising results, our work has several limitations. First, our evaluation, while quantitative, did not include a human study to assess the qualitative aspects of the final summaries. Second, the iterative process is **computationally expensive**, with each iteration incurring additional API costs and latency. Finally, the 10% of runs that failed to converge within the 15-iteration limit typically involved source papers with highly complex, dense tables or queries that were inherently ambiguous, highlighting the system's current struggles with certain content types and ill-posed questions.

Ethical Considerations. The primary ethical risk is generating plausible yet factually inaccurate summaries that could mislead researchers. While our Judge agent is designed to enforce faithfulness, it cannot guarantee perfect accuracy. Therefore, the system should be used as an assistant to augment—not replace—a researcher's own critical verification. The second risk is our reliance on a commercial LLM, propagating the biases and wrong facts the LLM might have, into our system.

Future Work. The efficiency of the system could be significantly improved by implementing the parallelization potential of the architecture (every agent can work on the next iteration once finished with the current iteration, as it doesn't depend on the other agents until the next iteration). A larger-scale evaluation is also a critical next step. Finally, the quality of the entire process is highly dependent on the initial diagnostic questions; exploring methods to enhance the Question Generator agent could yield substantial improvements.

Code and Data Availability

The code and Data for our system and the evaluation scripts are publicly available on GitHub.

- **MAIS System:** `https://github.com/avrymi-asraf/Query-Focused-Summarization`
- **Evaluation:(Code and data)** `https://github.com/avrymi-asraf/QFS-evaluation`

References

Juntai Cao, Xiang Zhang, Raymond Li, Chuyuan Li, Chenyu You, Shafiq Joty, and Giuseppe Carenini. 2025. Multi2: Multi-agent test-time scalable framework for multi-document processing.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yixin Liu, Alexander Fabbri, Yilun Zhao, Pengfei Liu, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023. Towards interpretable and efficient automatic reference-based summarization evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16360–16368, Singapore. Association for Computational Linguistics.

Ruochen Xu, Song Wang, Yang Liu, Shuohang Wang, Yichong Xu, Dan Iter, Pengcheng He, Chenguang Zhu, and Michael Zeng. 2023. LMGQS: A large-scale dataset for query-focused summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14764–14776, Singapore. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models.

Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. 2025. Jasper and stella: distillation of sota embedding models.

Weijia Zhang, Jia-Hong Huang, Svitlana Vakulenko, Yumo Xu, Thilina Rajapakse, and Evangelos Kanoulas. 2024. Beyond relevant documents: A knowledge-intensive approach for query-focused summarization using large language models.

A Qualitative Case Study Summaries

Here we present the full text of the three summaries generated by MAIS for a chosen paper, demonstrating the system's ability to adapt to different user queries. One can see that the summarizations are similar, but each one is different and focuses on the relevant query.

Chosen paper - Jasper and Stella: distillation of SOTA embedding models (Zhang et al., 2025)

Query 1: The progressive constraint strategy in knowledge distillation.

1. SUMMARY: This paper introduces the Jasper embedding model, a 2 billion parameter model that achieves state-of-the-art performance, comparable to 7 billion parameter models, through a multi-stage knowledge distillation framework. The core of this framework lies in its carefully designed loss functions that progressively transfer knowledge from multiple teacher models to the student. Initially, a cosine loss (L_cosine) aligns absolute text representations, but its limitations in optimization direction lead to overfitting. To address this, a similarity loss (L_sim) is introduced to measure semantic discrepancies from a text-pair perspective, ensuring relative consistency. Further enhancing robustness, a relative similarity distillation loss (L_resim) is employed to train the student to mimic the teacher's ranking preferences across text pairs within a batch. The overall loss is a weighted sum of these three functions, with specific hyperparameters (lambda_1=10, lambda_2=200, lambda_3=20). The training progresses through four stages: initial distillation (training FC1), further distillation (training FC1 and last three encoder layers), dimension reduction using Matryoshka Representation Learning (MRL) with L_sim and L_resim, and finally, multimodal alignment.

2. KEY HIGHLIGHTS: * The distillation framework utilizes three progressively broader loss functions: cosine loss (L_cosine), similarity loss (L_sim), and relative similarity distillation loss (L_resim). * L_cosine aligns absolute representations but can lead to overfitting; L_sim addresses this by focusing on text-pair semantic consistency; L_resim captures relative ranking preferences. * The training process involves four stages: initial distillation, deeper distillation, dimension reduction (using MRL and omitting L_cosine), and multimodal alignment. * Hyperparameter values for the weighted loss are lambda_1=10, lambda_2=200, and lambda_3=20, with a margin of 0.015 for L_resim. * Stage 3, dimension reduction, utilizes L_sim and L_resim with additional FC layers to compress vector dimensionality, while Stage 1 and 2 use all three losses.

Query 2: The paper's dual approach to the engineering problem of embedding dimensionality.

1. SUMMARY: This paper introduces Jasper, a 2 billion parameter embedding model that achieves state-of-the-art performance for its size, comparable to 7 billion parameter models, by employing a novel multi-stage distillation framework. The framework effectively distills knowledge from multiple large teacher embedding models, specifically NV-Embed-v2 (4096 dimensions) and Stella_en_1.5B_v5 (8192 dimensions), using three carefully designed loss functions: cosine loss (Lcosine), similarity loss (Lsim), and relative similarity distillation loss (Lresim). Stages 1 and 2 focus on distilling knowledge from these teachers, with Stage 1 training only a fully connected layer (FC1) and Stage 2 training FC1 and the last three encoder layers. Stage 3 addresses the challenge of high vector dimensionality (12288 dimensions post-concatenation) by utilizing Matryoshka Representation Learning (MRL) with additional fully connected layers (FC2, FC3, FC4) to generate lower-dimensional vectors. In this stage, Lcosine is omitted for the reduced dimension FC layers due to misaligned dimensions, while Lsim and Lresim are used. Stage 4 aims to unlock multimodal potential by aligning image and text embeddings using self-distillation on image-caption pairs, though its current implementation has limitations. Jasper's performance on the MTEB leaderboard, achieving an average score of 71.54 across 56 datasets, validates this dual approach to embedding dimensionality.

2. KEY HIGHLIGHTS: * Jasper employs a multi-stage distillation process using NV-Embed-v2 (4096D) and Stella_en_1.5B_v5 (8192D) as teacher models. * Distillation leverages cosine loss,

similarity loss, and relative similarity distillation loss, with specific parameter training in Stages 1 and 2. * Stage 3 utilizes Matryoshka Representation Learning (MRL) and fully connected layers for dimension reduction, omitting cosine loss for the reduced dimension outputs due to dimension misalignment. * Jasper achieves comparable performance to 7B parameter models on the MTEB leaderboard, demonstrating the effectiveness of its distillation and dimension reduction strategies. * Stage 4 introduces multimodal capabilities through self-distillation but acknowledges limitations in the current implementation.

Query 3: Methodology for fusing knowledge from heterogeneous teacher models.

1. SUMMARY: This paper details a novel multi-stage knowledge distillation framework, named Jasper, designed to fuse knowledge from heterogeneous, state-of-the-art (SOTA) teacher embedding models (specifically NV-Embed-v2 and Stella_en_1.5B_v5) into a smaller student model. The framework employs three carefully designed losses: cosine loss, similarity loss, and relative similarity distillation loss, progressing from specific to broader semantic alignment. Initially, in Stages 1 and 2, the student model's vectors are mapped and trained to align with concatenated teacher vectors (4096 + 8192 = 12288 dimensions). Stage 3 addresses the impractical dimensionality of these concatenated vectors by using Matryoshka Representation Learning (MRL) with additional fully connected layers to reduce them to manageable sizes (e.g., 512 dimensions), while still utilizing similarity-based losses. Stage 4 introduces self-distillation for preliminary multimodal alignment using image-caption pairs. The resulting Jasper model, with 2 billion parameters, achieves a high average score (71.54) on the MTEB leaderboard, comparable to larger 7B parameter models, and significantly outperforms other models in its parameter class. The differences in parameter training between Stage 1 (only FC layer) and Stage 2 (FC layer and last three encoder layers) are crucial for progressive knowledge transfer.

2. KEY HIGHLIGHTS: * Jasper utilizes a multi-stage distillation framework with three losses (cosine, similarity, relative similarity) to transfer knowledge from heterogeneous teacher models (NV-Embed-v2, Stella_en_1.5B_v5). * The student model's vector dimension is initially concatenated from teachers (12288) and then reduced using MRL in Stage 3 to more practical sizes (e.g., 512). * Jasper, a 2B parameter model, achieves performance comparable to 7B models on MTEB, outperforming smaller models. * Stage 1 trains only the initial FC layer, while Stage 2 trains the FC layer and the last three encoder layers for progressive knowledge absorption. * Stage 4 employs self-distillation for initial multimodal alignment, aligning image embeddings with text embeddings.