A Training-Free Mixture-of-Agents Framework for Multi-Document Summarization using LLMs and Knowledge Graphs

Cuong Vuong Tuan[1], Trang Mai Xuan[1*], Tien-Cuong Nguyen[2], Vu-Duc Ngo[3], Thien Van Luong[4]

[1]Faculty of Artificial Intelligence and Data Science, Phenikaa University, Duong Noi, Ha Noi, 12116, Viet Nam.
[2]VNPT AI, VNPT Group, Ha Noi, Viet Nam.
[3]MobiFone Research and Development Center, MobiFone Corporation, Ha Noi, Viet Nam.
[4]Business AI Lab, Faculty of Data Science and Artificial Intelligence, National Economics University, College of Technology, Ha Noi, Viet Nam.

*Corresponding author(s). E-mail(s): trang.maixuan@phenikaa-uni.edu.vn;
Contributing authors: cuong.vuongtuan@phenikaa-uni.edu.vn; nguyentiencuong@vnpt.vn; duc.ngo@mobifone.vn; thienlv@neu.edu.vn;

Abstract

Multi-Document Summarization (MDS) play a critical role in distilling essential information from collections of textual data. Existing approaches often struggle to capture complex inter-document relationships, rely heavily on large amounts of labeled data for supervised training, or exhibit limited generalization across domains and languages. To address these limitations, we present a training-free mixture-of-agents framework for MDS that leverages the complementary strengths of large language models (LLMs) and knowledge graphs. Our approach decomposes summarization into specialized agent tasks: extractive selection, knowledge-aware abstraction, and iterative refinement, each operating without task-specific fine-tuning. We unify their outputs using a multi-perspective consistency mechanism guided by LLMs. Experiments across four datasets in English and Vietnamese demonstrate state-of-the-art or competitive performance, validating the effectiveness and adaptability of our modular design.

Keywords: Multi-Document Summarization, Large Language Models, Mixture of Agents, Knowledge Graphs

1 Introduction

Rapidly expanding digital information necessitates MDS for synthesizing knowledge from diverse sources, making it a critical task in natural language processing [1, 2]. Early extractive methods [3, 4] and rule-based abstractive systems [5] often produce outputs that lack semantic coherence and fluency. The advent of deep learning, especially sequence-to-sequence architectures [6] and subsequently transformer-based LLMs [7, 8], has significantly improved the quality of abstractive summarization [9, 10]. However, leading supervised models require large task-specific labeled datasets [11, 12], limiting their applicability in low-resource domains or under-resourced languages such as Vietnamese [13].

Recent advances in LLMs have opened new opportunities for MDS through strong zero-shot and few-shot capabilities [14, 15]. However, context length limitations remain a significant barrier. Moreover, The et al. [16] conducted specific fine-tuning on Vietnamese datasets, achieving promising results. Nevertheless, these approaches face challenges including limited testing on diverse English datasets and requirements for extensive data collection and labeling. Furthermore, current training-free methods that use pre-trained LLMs encounter specific difficulties in learning complex inter-document relationships and handling large semantic contexts [17]. Therefore, specialized frameworks are needed that can use existing LLM knowledge without requiring task-specific data collection or fine-tuning, while addressing coordination challenges across multiple documents.

This paper introduces the **Mixture of Agents (MoA)** framework, a novel architectural approach for MDS that leverages pre-trained LLMs within a collaborative multi-agent system **without task-specific fine-tuning**. The MoA architecture decomposes the complex MDS task into complementary sub-problems managed by specialized agents as follows:

(1) **Extractor agent**: Identifies salient factual sentences using a robust scoring model.
(2) **KGSum agent**: Our most sophisticated agent, which constructs a knowledge graph (KG) to model entities and their relationships explicitly. It then generates a KG-based summary by identifying thematic communities and detecting contrastive information within the graph, offering a deep, structured perspective.
(3) **Abstractor agent**: Produces a fluent, coherent abstractive summary directly from the source documents.

The outputs from these parallel agents are orchestrated by our novel **Adaptive Multi-Perspective Fusion (AMF)** mechanism. AMF assesses agent-generated metadata to dynamically select the optimal strategy for synthesizing a final summary, robustly fusing extractive, KG-based, and abstractive perspectives.

Our primary contributions are as follows:

- We propose MoA, a novel architectural framework for training-free multi-document summarization that coordinates three specialized agents: Extractor, KGSum and Abstractor. The main innovation is the KGSum agent that constructs knowledge graphs to explicitly model complex inter-document relationships and contradictions, enabling systematic learning of intricate connections between multiple documents.
- We introduce the Adaptive Multi-Perspective Fusion (AMF) mechanism that uses agent-generated metadata to dynamically determine optimal fusion strategies, allowing MoA to adaptively exploit agent-specific strengths based on document characteristics.
- We demonstrate MoA's effectiveness through comprehensive evaluations using various LLMs on benchmark datasets across English and Vietnamese, achieving state-of-the-art performance and validating our architectural approach for learning complex inter-document relationships.

The remainder of the paper is organized as follows: Section 2 discusses related work in MDS. Section 3 details the MoA framework. Section 4 presents the experimental setup, results, and analysis. Section 5 concludes and suggests future directions.

2 Related Work

2.1 Training-free Approaches for MDS

Early efforts in this paradigm were dominated by traditional unsupervised extractive techniques. For example, maximal marginal relevance, which selects relevant yet diverse sentences, was proposed in [18], while centroid-based approaches that identify sentences central to a document cluster's theme were introduced in [19]. Additionally, graph-based algorithms such as LexRank [3] and TextRank [4] rely on inter-sentence relationships to rank sentences by centrality. Note that these unsupervised learning methods often rely on surface-level features.

More recent unsupervised approaches have incorporated deep learning representations without task-specific fine-tuning. Examples include using pre-trained model embeddings within graph-based frameworks [20] or leveraging topic modeling with transformer representations, as seen in GLIMMER-TTR [21]. Unsupervised graph-based methods were also explored for languages such as Vietnamese [22].

The advent of LLMs has further expanded training-free MDS possibilities, primarily through zero-shot or few-shot prompting [17]. Li et al. [23] focused on strategies to compress or select information before presenting concatenated documents to an LLM due to context window limitations. While these LLM-driven prompting approaches offer flexibility and reduce the need for specific MDS training datasets, directly prompting a single LLM for complex MDS can struggle with information reconciliation and lack coherence across numerous documents.

2.2 Full-training approaches for MDS

Contrasting with training-free methods, full training approaches involve extensive supervised learning on large-scale, task-specific MDS datasets. The advent of deep learning, particularly sequence-to-sequence models [6], significantly advanced abstractive summarization. For example, Pointer-Generator Networks [24] was proposed to improve the factual consistency.

However, the paradigm shifted decisively with PLMs fine-tuned specifically for MDS. Architectures designed for long sequences (e.g., Longformer [25]) were adapted. End-to-end supervised training on large MDS datasets, such as Multi-News [11] and Multi-XScience [12] for English, or VN-MDS and ViMs [13] for Vietnamese, led to powerful models, such as PEGASUS [9] (gap-sentence objective), PRIMERA [10] (entity-masking), and Centrum [26] (discourse coherence). Some research works also explored fine-tuning general-purpose LLMs on MDS tasks, which falls under the full training paradigm due to their reliance on task-specific labeled data.

These models typically achieve SOTA ROUGE scores [27] on in-domain data but are fundamentally dependent on such labeled datasets. This restricts their applicability in low-resource scenarios, novel domains, or for languages whose large specific MDS corpora are unavailable. MoA, as a training-free approach, aims to circumvent this data dependency entirely.

2.3 Agent-Based Systems

The emergence of knowledge-enhanced retrieval systems has shown promise for various AI tasks. Retrieval-Augmented Generation (RAG) [28] represents a major advance in knowledge-intensive tasks, showing strong results in question-answering by combining retrieval with text generation. GraphRAG [29] extends this approach using graph structures to capture relationships and enable better reasoning. However, these query-focused methods may not suit the multi-perspective analysis needed for comprehensive document summarization.

Recently, multi-agent systems powered by LLMs have gained significant attention for complex problem-solving and collaboration [30, 31]. AutoGen [32] enables multiple agents to converse and complete tasks through flexible interaction patterns. Similarly, ChatEval [33] uses multi-agent debates where LLMs collaborate as evaluators to enhance task performance. Moreover, the Mixture-of-Agents approach [34] constructs layered architectures where each agent leverages outputs from previous layer agents as auxiliary information to enhance response quality. While these frameworks excel in their respective domains, MDS presents unique challenges that benefit from parallel processing of diverse summarization perspectives rather than iterative debate refinement.

To the best of our knowledge, this work presents the first attempt at developing a structured multi-agent MoA architecture specifically for MDS. Our key contribution lies in integrating AMF mechanisms within the MoA paradigm for training-free MDS, differentiating our approach from monolithic LLM prompting and traditional MDS techniques.

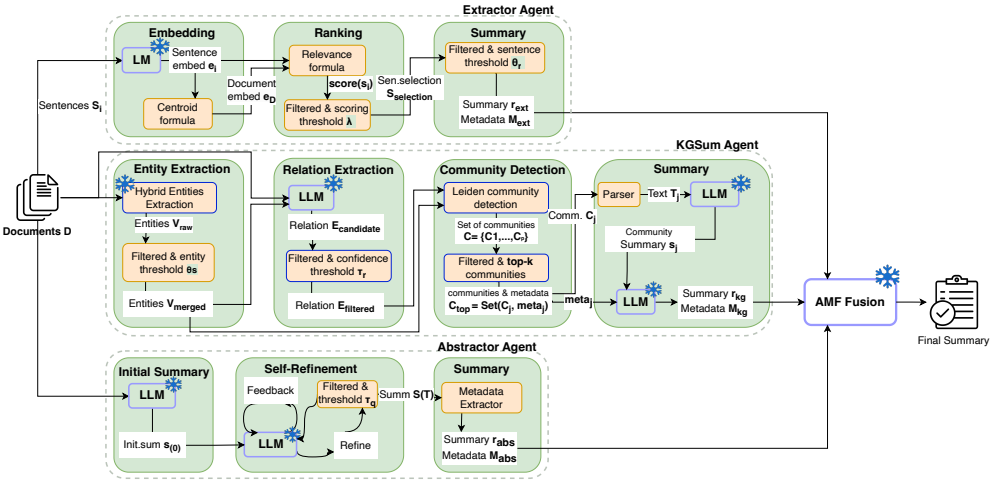

Fig. 1: Overall architecture of our proposed MoA framework. The three agents (Extractor, KGSum, Abstractor) process the input documents D through parallel specialized pipelines, generating their respective summaries (r_{ext}, r_{kg}, r_{abs}) and metadata ($\mathcal{M}_{\text{ext}}$, $\mathcal{M}_{\text{kg}}$, $\mathcal{M}_{\text{abs}}$). The AMF module then synthesizes these outputs into the final summary S_{MoA}. "LM" denotes Language Model (BERT-based), while "LLM" denotes Large Language Model.

3 Methodology

Figure 1 shows the overall framework architecture of MoA framework.

3.1 Extractor Agent

The Extractor Agent systematically identifies and extracts the most semantically significant sentences from the input document set D to generate an extractive summary r_{ext}. Building upon established centroid-based approaches [35], our innovation integrates positional awareness with semantic centrality through a hybrid scoring mechanism that leverages pre-trained contextual embeddings. As illustrated in the Extractor Agent section of Figure 1, this process operates through three sequential stages: embedding generation, relevance scoring, and summary assembly.

In the first stage, neural sentence embedding transforms each sentence s_i from document set D into contextual vector representations using a pre-trained Language Model (LM), specifically Longformer [25]. This encoding process produces sentence embeddings $\mathbf{e}_i$ that capture semantic content within the document context. Subsequently, the document-level semantic representation $\mathbf{e}_D$ is computed as the centroid of all sentence embeddings:

$$\mathbf{e}_D = \frac{1}{m} \sum_{i=1}^{m} \mathbf{e}_i, \tag{1}$$

where m denotes the total number of sentences across all documents in D.

The second stage implements sentence ranking through a hybrid relevance scoring mechanism that balances semantic similarity with positional bias. For each sentence s_i, the relevance score integrates cosine similarity between sentence and document embeddings with positional preference:

$$\text{score}(s_i) = \lambda \cos(\mathbf{e}_i, \mathbf{e}_D) + (1 - \lambda) \left(1 - \frac{\text{pos}(s_i)}{L_d}\right), \tag{2}$$

where $\text{pos}(s_i)$ represents the normalized position of sentence s_i within the concatenated document set of length L_d, and λ controls the semantic-positional balance, as shown in Table D5. Sentences are subsequently ranked by these scores, with the top-α percent forming the candidate set S_{cand}.

The third stage assembles the final extractive summary through redundancy filtering and metadata generation. An optional redundancy removal step refines S_{cand} into S_{sel}:

$$S_{\text{sel}} = \{s_i \in S_{\text{cand}} : \forall s_j \in S_{\text{sel}}, \cos(\mathbf{e}_i, \mathbf{e}_j) \leq \theta_r\}, \tag{3}$$

where $\mathbf{e}_i$ and $\mathbf{e}_j$ are the embeddings of sentences s_i and s_j, respectively. And θ_r denotes the redundancy threshold (Table D5). Finally, the final summary r_{ext} concatenates selected sentences in their original document order, while the metadata package $\mathcal{M}_{\text{ext}}$ encapsulates sentence scores, selection indices, and processing parameters for subsequent AMF integration.

3.2 KGSum Agent

The KGSum Agent dynamically constructs a knowledge graph from documents D to generate knowledge-enhanced summaries r_{kg}. Building upon established KG construction principles [36], our innovation extends traditional approaches with hybrid entity extraction, contrastive relation modeling, and meta-aware summarization, with detailed novelty analysis provided in Appendix B. The dynamic construction ensures that each new document set or revised data generates a fresh, tailored knowledge graph optimized for the specific input. Figure 1 shows the overall architecture, while Figure 2 demonstrates the processing pipeline. The agent operates through four sequential stages, each building upon the previous stage's outputs.

3.2.1 Entity Extraction from Documents

From the document collection D, the first stage employs hybrid entity extraction to identify key information units, as illustrated in the entity extraction phase of Figure 2. We combine three complementary methods: spaCy NER for standard entities, rule-based patterns for numerical data, and LLM extraction with prompt Π_e (Figure A1) for contextual understanding. The raw entities are aggregated as:

$$V_{\text{raw}} = V_{\text{spacy}} \cup V_{\text{rules}} \cup V_{\text{llm}}. \tag{4}$$

Subsequently, entity consolidation eliminates redundancy through similarity-based merging, as demonstrated in the entity refinement step of Figure 2. For entities

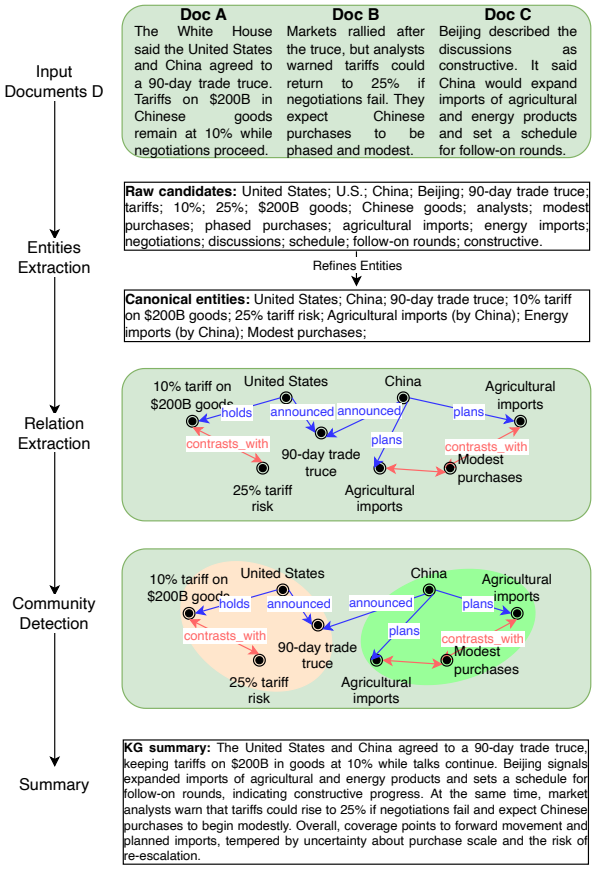

Fig. 2: Process of KGSum Agent for dynamic knowledge graph construction and knowledge-enhanced summary generation. Blue relation edges represent standard semantic relations, while red edges denote contrastive relations capturing conflicting information. The knowledge graph is dynamically constructed for each input document set, ensuring tailored representation of the specific content.

$v_i, v_j \in V_{\text{raw}}$, the merging process follows:

$$V_{\text{merged}} = \begin{cases} extMerge(v_i, v_j) & \text{if } \text{sim}(v_i, v_j) \geq \theta_s \\ \{v_i, v_j\} & \text{otherwise} \end{cases}, \tag{5}$$

where θ_s represents the similarity threshold (Table D5). This process yields the consolidated entity set V_{merged} for subsequent relation extraction.

3.2.2 Relation Extraction from Documents

Leveraging the consolidated entities V_{merged}, the second stage identifies semantic connections to construct graph edges, as shown in the relation extraction phase of

Figure 2. Our approach focuses on extracting both standard semantic relations and contrastive relations that capture conflicting information across documents. Using LLM with specialized prompt Π_{rel} (Figure A2), the relation extraction process is formulated as:

$$E_{\text{candidate}} = \text{LLM}(\Pi_{\text{rel}}, V_{\text{merged}}, D). \tag{6}$$

The prompt systematically guides extraction of traditional semantic relations alongside our novel contrastive relations, with empirical validation of contrastive relation effectiveness detailed in Figure A2. To ensure high-quality graph construction, confidence-based filtering is applied to each candidate relation $r \in E_{\text{candidate}}$:

$$E_{\text{filtered}} = \begin{cases} r \text{ if confidence}(r) \geq \tau_r \\ \emptyset \text{ otherwise} \end{cases}, \tag{7}$$

where τ_r represents the confidence threshold (Table D5). This yields the refined relation set E_{filtered} that forms the knowledge graph edges.

3.2.3 Community Detection in Knowledge Graph

With the knowledge graph $G = (V_{\text{merged}}, E_{\text{filtered}})$ constructed, the third stage partitions it into thematic communities for structured analysis, as illustrated in the community detection phase of Figure 2. We employ the Leiden algorithm [37] for community detection based on modularity optimization. The partitioning process produces:

$$\mathcal{C} = \{C_1, C_2, \ldots, C_p\} = \text{Leiden}(G). \tag{8}$$

Subsequently, metadata extraction generates descriptive information for each community C_j:

$$\text{meta}_j = \{\text{centrality}, \text{contrast\_flags}, \text{key\_entities}, \text{fact\_density}\}. \tag{9}$$

This metadata encompasses centrality scores, contrastive relation indicators, key entities, and content density measures. Finally, importance-based ranking selects the top-k communities (Table D5) to form $\mathcal{C}_{\text{top}}$ for focused summarization.

3.2.4 KG-Based Summary Generation

Building upon the selected communities $\mathcal{C}_{\text{top}}$, the final stage generates the knowledge-enhanced summary r_{kg} through meta-aware fusion, as demonstrated in the summary generation phase of Figure 2. Each community $C_j \in \mathcal{C}_{\text{top}}$ undergoes linearization and targeted summarization:

$$T_j = \text{Linearize}(C_j), \quad s_j = \text{LLM}(\Pi_s, T_j), \tag{10}$$

where T_j converts the graph structure to text, and Π_s (Figure A3) generates community-specific summaries. Subsequently, meta-aware fusion integrates individual

summaries while leveraging their associated metadata:

$$r_{\text{kg}} = \begin{cases} extLLM(\Pi_f, \{(s_j, \text{meta}_j) : C_j \in \mathcal{C}_{\text{top}}\}) & \text{if } |\mathcal{C}_{\text{top}}| > 1 \\ s_1 & \text{if } |\mathcal{C}_{\text{top}}| = 1 \end{cases}, \tag{11}$$

The fusion prompt Π_f (Figure A4) utilizes metadata to weight communities by centrality, highlight conflicts, and ensure comprehensive coverage. This process yields the final knowledge-enhanced summary r_{kg} along with metadata package $\mathcal{M}_{\text{kg}}$ for subsequent AMF integration.

3.3 Abstractor Agent

The Abstractor Agent generates fluent, abstractive summaries by synthesizing information from documents D to produce r_{abs}. Building upon established LLM summarization techniques [38] and self-improvement concepts [39], our innovation introduces an iterative self-refinement loop specifically designed for multi-document summarization. As illustrated in the Abstractor Agent section of Figure 1, this process operates through three sequential stages: initial summary generation, self-refinement, and final output production.

In the first stage, initial summary generation transforms the document collection D into an abstractive summary using an LLM with specialized prompt Π_{init} (Figure A5). This process generates the initial summary:

$$\tilde{s}^{(0)} = \text{LLM}(\Pi_{\text{init}}, D), \tag{12}$$

where $\tilde{s}^{(0)}$ captures the main themes from D and serves as the foundation for subsequent refinement.

The second stage implements iterative self-refinement to enhance summary quality through feedback-driven improvement. For each iteration t, the system evaluates the current summary $\tilde{s}^{(t)}$ using a multi-dimensional feedback function $\mathcal{F}_{\text{MDS}}$ that assesses factual accuracy, coverage, coherence, and specificity. The refinement process is formulated as:

$$\tilde{s}^{(t+1)} = \text{LLM}(\Pi_{\text{ref}}, \tilde{s}^{(t)}, \mathcal{F}_{\text{MDS}}(\tilde{s}^{(t)}, D)), \tag{13}$$

where Π_{ref} represents the refinement prompt (Figure A6) that guides quality improvement. This iterative cycle continues for a maximum of T iterations or until the quality score exceeds threshold τ_q (Table D5).

The third stage produces the final abstractive summary and comprehensive metadata. The refined summary becomes the final output $r_{\text{abs}} = \tilde{s}^{(T)}$, while the metadata package $\mathcal{M}_{\text{abs}}$ encapsulates quality scores, iteration count, and processing parameters. Both r_{abs} and $\mathcal{M}_{\text{abs}}$ are subsequently passed to the AMF module for multi-perspective integration.

3.4 Mixture of Agents Orchestration: The Adaptive Multi-Perspective Fusion (AMF) Mechanism

The three agents produce specialized summaries that must be intelligently synthesized into the final output S_{MoA}. The Mixture of Agents paradigm [34] combines multiple perspectives through iterative refinement. However, existing MoA approaches apply uniform weighting to all agents, which ignores varying quality across inputs. We propose Adaptive Multi-Perspective Fusion (AMF), which extends MoA with adaptive prioritization based on quality signals and domain strengths. As shown in Figure 1, AMF processes summaries r_{ext}, r_{kg}, r_{abs} and metadata $\mathcal{M}_{\text{ext}}$, $\mathcal{M}_{\text{kg}}$, $\mathcal{M}_{\text{abs}}$ through four stages (Figure 3). This coordination mechanism distinguishes our architectural contribution from standard prompting approaches.

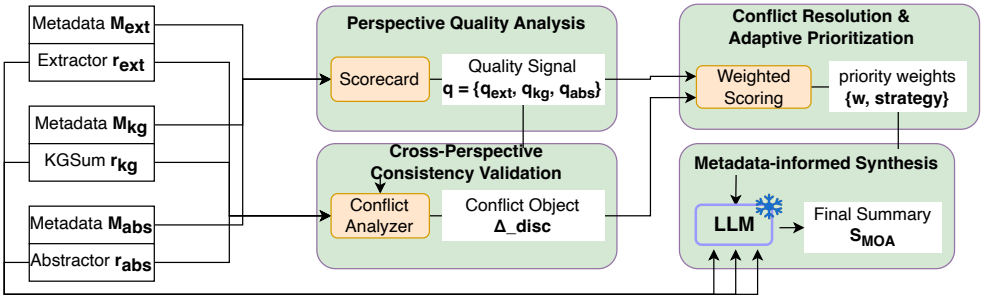

Fig. 3: The four-stage AMF pipeline for adaptive multi-perspective fusion. Stage 1 analyzes quality signals from each agent's metadata. Stage 2 validates cross-perspective consistency and detects conflicts. Stage 3 resolves conflicts and calculates adaptive priority weights. Stage 4 synthesizes the final summary using metadata-informed prompting. This architecture distinguishes our contribution from standard uniform-weighting MoA approaches.

3.4.1 Stage 1: Perspective Quality Analysis

Stage 1 extracts quality signals from agent metadata, as shown in Figure 3. Each agent response $R_{\text{agent}} = \{r_{\text{agent}}, \mathcal{M}_{\text{agent}}\}$ already contains quality metrics computed during summary generation. We filter relevant indicators from these metadata packages without additional computation. For Extractor Agent, we extract from $\mathcal{M}_{\text{ext}}$:

$$\mathbf{q}_{\text{ext}} = \{q_{\text{salience}}, q_{\text{density}}, q_{\text{coverage}}\}, \tag{14}$$

where q_{salience} reflects sentence importance, q_{density} measures factual content, and q_{coverage} assesses entity representation.

Similarly, for KGSum Agent, we extract from $\mathcal{M}_{\text{kg}}$:

$$\mathbf{q}_{\text{kg}} = \{q_{\text{relation}}, q_{\text{contrast}}, q_{\text{entity}}\}, \tag{15}$$

where q_{relation} quantifies relation confidence, q_{contrast} captures contrastive flags, and q_{entity} measures coverage.

For Abstractor Agent, we extract from $\mathcal{M}_{\text{abs}}$:

$$\mathbf{q}_{\text{abs}} = \{q_{\text{fluency}}, q_{\text{coherence}}, q_{\text{synthesis}}\}, \tag{16}$$

where q_{fluency} evaluates linguistic quality, $q_{\text{coherence}}$ measures narrative flow, and $q_{\text{synthesis}}$ assesses integration. These filtered signals enable Stage 2 consistency validation.

3.4.2 Stage 2: Cross-Perspective Consistency Validation

Building on the quality signals from Stage 1, Stage 2 validates consistency across summaries $\{r_{\text{ext}}, r_{\text{kg}}, r_{\text{abs}}\}$, as shown in Figure 3. We extract numerical values and named entities from each summary using regex patterns and SpaCy NER. For numerical consistency, we compute pairwise differences between extracted numbers across agents. For entity consistency, we measure overlap using Jaccard similarity on entity sets. The consistency analysis produces:

$$\Delta_{\text{disc}} = \{\text{contradictions}, \text{unique\_contributions}, \text{consistency\_score}\}, \tag{17}$$

where contradictions lists conflicting facts between agents, unique_contributions identifies perspective-specific insights, and consistency_score aggregates numerical and entity agreement. Specifically, consistency score is computed as:

$$\text{consistency\_score} = \frac{1}{3}\left(\text{num\_agreement} + \text{entity\_overlap} + \text{fact\_overlap}\right), \tag{18}$$

where each component ranges from 0 to 1. This structured output guides Stage 3 prioritization.

3.4.3 Stage 3: Conflict Resolution and Adaptive Prioritization

Stage 3 implements adaptive weighting, the core innovation distinguishing AMF from standard MoA (Figure 3). Using quality signals from Stage 1 and consistency analysis from Stage 2, we calculate agent-specific priority weights. Unlike uniform weighting in existing MoA, our approach adjusts priorities dynamically based on input characteristics. For each agent, the base quality score aggregates filtered signals:

$$\text{score}_{\text{agent}} = \text{mean}(\mathbf{q}_{\text{agent}}), \tag{19}$$

where the mean aggregates quality signal components.

Subsequently, we adjust scores with domain-specific bonuses and consistency-based penalties. For KGSum Agent, we add bonuses when relation confidence exceeds thresholds. For Extractor Agent, bonuses apply when salience scores are high. For Abstractor Agent, bonuses reward fluency. Moreover, agents with low consistency scores receive penalties. The final priority weight is:

$$w_{\text{agent}} = \frac{\text{score}_{\text{agent}} + \text{bonus}_{\text{agent}} - \text{penalty}_{\text{agent}}}{\sum_{\text{all}}(\text{score} + \text{bonus} - \text{penalty})}, \tag{20}$$

where normalization ensures $\sum_{\text{agent}} w_{\text{agent}} = 1$. These weights $\mathbf{w} = \{w_{\text{ext}}, w_{\text{kg}}, w_{\text{abs}}\}$ prioritize high-quality agents for each input.

Furthermore, we select fusion strategy based on weight distribution:

$$\text{strategy} = \begin{cases} \text{KG\_LED} & \text{if } w_{\text{kg}} > 0.5 \text{ and } q_{\text{relation}} > 0.7 \\ \text{EXTRACTOR\_LED} & \text{if } w_{\text{ext}} > 0.5 \text{ and } q_{\text{salience}} > 0.7 \\ \text{ABSTRACTOR\_LED} & \text{if } w_{\text{abs}} > 0.5 \text{ and } q_{\text{fluency}} > 0.7 \\ \text{ADAPTIVE} & \text{otherwise} \end{cases}, \tag{21}$$

where thresholds determine when one agent dominates. This strategy guides Stage 4 synthesis.

[Instruction]
You are an expert summarization orchestrator. Your task is to synthesize a single, high-quality, coherent, and factually accurate summary by intelligently fusing the outputs from three specialized agents:

1. Extractor Agent Summary (r_{ext}): Provides key factual sentences.
2. KGSum Agent Summary (r_{kg}): Offers structured knowledge, entity relationships, and potentially highlighted contrasts.
3. Abstractor Agent Summary (r_{abs}): Delivers a fluent, abstractive overview.

You will also receive an Analysis Metadata ($A_{analysis}$) object containing quality scores for each agent's summary (q), a list of identified discrepancies (Δ_{disc}), priority weights for each agent (w), and a suggested fusion strategy.

[Task]
Fuse summaries from Extractor (r_{ext}), KGSum (r_{kg}), and Abstractor (r_{abs}) agents into one high-quality summary.

[Objective]
Synthesize a final, integrated summary based on agent summaries and $A_{analysis}$.
FUSION STRATEGY GUIDANCE:
1. If KG LED: Use r_{kg} as core, augment with r_{ext} specifics, use r_{abs} for fluency.
2. If EXTRACTOR LED: Start with r_{ext}, enrich with r_{kg} context, use r_{abs} for flow.
3. If BALANCED: Weigh contributions by w, ensure key info from high-weight agents.

[Input]

1. r_{ext}: {text from extractor agent}
2. r_{kg}: {text from kgsum agent}
3. r_{abs}: {text from abstractor agent}
4. Analysis Metadata ($A_{analysis}$): Contains quality signals (q), discrepancies (Δ_{disc}), priority weights (w), and chosen fusion strategy.

[Output]
[INTEGRATED SUMMARY]

Fig. 4: Structure of the enhanced fusion prompt Π_{AMF} used in Stage 4 of AMF. The prompt dynamically incorporates quality signals $\mathbf{q}$, priority weights $\mathbf{w}$, discrepancy information Δ_{disc}, and the selected fusion strategy. This metadata-driven prompting enables the LLM to perform intelligent synthesis by explicitly prioritizing high-quality perspectives and resolving conflicts based on quantitative analysis.

3.4.4 Stage 4: Metadata-Informed Synthesis

Stage 4 synthesizes the final summary using metadata from Stages 1 to 3 (Figure 3). We consolidate all analysis components into:

$$\mathcal{A}_{\text{analysis}} = \{\mathbf{q}, \Delta_{\text{disc}}, \mathbf{w}, \text{strategy}\}, \tag{22}$$

where quality signals, discrepancies, weights, and strategy guide synthesis.

The fusion prompt Π_{AMF} embeds this metadata to direct LLM attention (Figure 4). Unlike standard prompting with equal attention, our prompt explicitly specifies priority weights and conflict resolution. The final summary is:

$$S_{\text{MoA}} = \text{LLM}(\Pi_{\text{AMF}}, \{r_{\text{ext}}, r_{\text{kg}}, r_{\text{abs}}\}, \mathcal{A}_{\text{analysis}}), \tag{23}$$

where the LLM receives summaries with explicit guidance from adaptive prioritization. This architectural coordination mechanism demonstrates that performance gains stem from intelligent multi-agent orchestration rather than simple prompting. AMF produces comprehensive metadata $\mathcal{M}_{\text{AMF}}$ for transparency.

4 Experiments

In this section, we present a comprehensive experimental evaluation of our proposed MoA framework for MDS. This evaluation is conducted on diverse real-world datasets. MoA is designed as a training-free, multi-agent architecture, while both SOTA full-training (supervised) and training-free (unsupervised) baselines are considered. Through rigorous comparisons with SOTA baselines and in-depth analyses, we aim to address the following research questions:

- **RQ1:** How does the choice of LLM (open-source vs. closed-source) influence MoA's performance? Can MoA effectively leverage more advanced LLMs?
- **RQ2:** How does MoA perform on MDS tasks compared to SOTA supervised and unsupervised baselines?
- **RQ3:** What is the contribution of each agent within the MoA framework to the overall summarization quality?

We first investigate the impact of different LLM choices (both closed-source and open-source) on MoA's efficacy to answer RQ1. Based on these findings, we select the optimal LLM. We then establish MoA's performance against SOTA baselines (RQ2) over both English and Vietnamese datasets. Finally, an ablation study is carried out to dissect individual component contributions to answer RQ3.

4.1 Experimental Setup

Evaluation Datasets We evaluate MoA on four diverse MDS datasets spanning multiple domains and languages: Multi-News (news articles, English) [11], Multi-XScience (scientific papers, English) [12], VN-MDS (news articles, Vietnamese)[1], and ViMs (diverse documents, Vietnamese) [13]. Table 1 presents comprehensive statistics for

[1] https://github.com/lupanh/VietnameseMDS

all datasets, including test sample counts, number of reference summaries per cluster, average document lengths, and summary lengths. Our evaluation encompasses substantial scale variation. Document quantities range from 29 samples (VN-MDS) to 5,622 samples (Multi-News), while average document lengths span from 419 words to 2,103 words. The number of reference summaries per cluster varies from 2.79 to 6.56, reflecting diverse multi-document aggregation requirements. Therefore, our evaluation validates architectural effectiveness across realistic scale variations encountered in established MDS benchmarks. (Detailed descriptions of dataset characteristics, preprocessing procedures, and complexity analysis can be found in Appendix C.)

Table 1: Statistics of evaluation datasets

Dataset	Language	Test Samples	# refs	doc. len	summ. len
Multi-News	English	5,622	2.79	2,103.49	263.66
Multi-XScience	English	5,093	4.42	778.08	116.44
VN-MDS	Vietnamese	29	3.14	419.0	172.1
ViMs	Vietnamese	45	6.56	470.2	231.2

Evaluation Metrics Following previous works [9], we use ROUGE F1-scores as standard evaluation metrics: ROUGE-1 (R1) for content overlap, ROUGE-2 (R2) for fluency and local coherence, and ROUGE-L (RL) for structural similarity and overall coherence against reference summaries [27]. Higher scores indicate better summarization quality.

Reporting Conventions All ROUGE scores are reported as percentages. Results for our MoA are reported as mean ± standard deviation over three runs. Bold entries indicate the best score in a column, and underlined entries indicate the second-best. A hyphen "-" denotes that a method was not evaluated on that dataset or the original paper did not report that metric. Baseline scores are taken from the original publications. Abbreviations used in tables: FT = Full Training, TF = Training-Free, Prop. = Proprietary, Open = Open-source.

Implementation Details MoA's architectural framework operates independently of specific LLM choices, demonstrating training-free adaptability across diverse model types. We systematically evaluate MoA with four leading closed-source LLMs (GPT-4.1-mini[2], Grok-3-mini[3], GPT-4o-mini[4], and Gemini-1.5-flash-8b[5] and three representative open-source LLMs (Llama-3.1-8B-Instruct[6], Qwen2.5-7B-Instruct[7], Gemma-2-9b-it[8]). All experiments are conducted on NVIDIA Quadro RTX 8000 GPU. We perform three independent runs for statistical validation, with mean and standard deviation reported. (Complete hyperparameter configurations and detailed parameter selection procedures are provided in Appendix D. API cost analysis for closed-source

[2] https://openai.com/index/gpt-4-1/
[3] https://x.ai/news/grok-3
[4] https://openai.com/index/hello-gpt-4o/
[5] https://blog.google/technology/ai/google-gemini-next-generation-model-february-2024/
[6] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
[7] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct
[8] https://huggingface.co/google/gemma-2-9b-it

models is provided in Appendix E.) For ablation studies, we utilize the few-shot prompt template Π_{few} presented in Appendix A for single LLM baseline comparisons.

Baseline Selection We compare MoA against two types of baselines. Full-training (supervised) baselines include PEGASUS [9], LED [25], PRIMERA [10], and Centrum [26] for English, and LatVisNewshead [16] for Vietnamese. Training-free baselines include SRI+beam [40] and GLIMMER-TTR [21] for English, and Graph combine Abstractive [22] and Bert-VBD [41] for Vietnamese. Training-free methods use pre-trained LLMs without additional fine-tuning on MDS datasets. Therefore, they do not require task-specific data collection or labeling, which is useful when specialized training data is limited or computational resources are constrained.

4.2 Impact of LLM Choice on MoA Performance (RQ1)

Table 2: Performance comparison of our proposed MoA with different LLM backbones on English datasets

		Multi-News			Multi-XScience		
Method	**Type**	**R1**	**R2**	**RL**	**R1**	**R2**	**RL**
GPT-4.1-mini	Prop.	**54.97**	**24.90**	**29.40**	**36.66**	**5.64**	**19.73**
Grok-3-mini	Prop.	53.91	24.68	28.61	35.60	5.42	18.94
GPT-4o-mini	Prop.	52.67	23.72	27.97	31.05	4.72	16.28
Gemini-1.5-flash-8b	Prop.	51.92	23.10	27.51	30.11	4.50	15.82
Llama-3.1-8b-Instruct	Open	52.40	25.98	28.90	23.42	2.60	13.24
Qwen2.5-7B-Instruct	Open	48.17	21.75	26.55	23.64	2.55	12.63
Gemma-2-9b-it	Open	25.08	8.48	14.81	24.05	2.47	12.84

Table 3: Performance comparison of our proposed MoA with different LLM backbones on Vietnamese datasets

		VN-MDS			ViMs		
Method	**Type**	**R1**	**R2**	**RL**	**R1**	**R2**	**RL**
GPT-4.1-mini	Prop.	**80.11**	**52.32**	**48.29**	**80.26**	**55.57**	**49.04**
Grok-3-mini	Prop.	79.05	52.10	47.50	79.20	55.35	48.25
GPT-4o-mini	Prop.	72.90	46.10	40.50	68.80	49.50	39.00
Gemini-1.5-flash-8b	Prop.	72.25	45.60	40.02	68.05	48.80	38.42
Llama-3.1-8b-Instruct	Open	71.80	45.20	39.80	67.50	48.50	38.50
Qwen2.5-7B-Instruct	Open	65.10	40.50	38.80	61.00	41.00	34.00
Gemma-2-9b-it	Open	63.20	38.70	37.10	59.00	39.00	32.00

Tables 2 and 3 present comprehensive performance comparison of MoA with different LLM backbones across all four evaluation datasets. Since MoA operates as a multi-agent system where LLMs serve as the primary reasoning engines for each agent, we systematically investigate how different LLM capabilities affect overall architectural performance. The results reveal clear performance hierarchies and important insights about LLM selection for MDS tasks.

On English datasets (Table 2), closed-source models demonstrate substantial superiority over open-source alternatives. Specifically, GPT-4.1-mini achieves the highest scores across all metrics: 54.97 R1, 24.90 R2, and 29.40 RL on Multi-News, along with 36.66 R1, 5.64 R2, and 19.73 RL on Multi-XScience. These scores surpass Grok-3-mini by 2.0 percent in R1 on Multi-News and 3.0 percent on Multi-XScience, demonstrating consistent leadership. Moreover, GPT-4o-mini and Gemini-1.5-flash-8b achieve competitive results with 52.67 R1 and 51.92 R1 on Multi-News respectively, validating their strong general capabilities. In particular, among open-source LLMs, Llama-3.1-8b-Instruct achieves the best performance with 52.40 R1 on Multi-News, substantially outperforming other open-source alternatives. However, Gemma-2-9b-it, designed as a multimodal model, shows suboptimal performance with only 25.08 R1 on Multi-News, indicating that specialized capabilities do not always transfer effectively to text-only MDS applications.

On Vietnamese datasets (Table 3), the performance hierarchy remains consistent with English results. GPT-4.1-mini achieves 80.11 R1 on VN-MDS and 80.26 R1 on ViMs, closely followed by Grok-3-mini with 79.05 R1 and 79.20 R1 respectively. The minimal performance gap (1.3 percent on VN-MDS, 1.3 percent on ViMs) confirms both models' strong multilingual capabilities. Furthermore, the performance differences between Vietnamese and English datasets primarily reflect dataset complexity characteristics rather than language-specific advantages. Specifically, Vietnamese datasets exhibit significantly higher extractiveness (94.3 percent versus 19.2 percent for English) and content redundancy, inflating ROUGE scores through exact n-gram matching rather than demanding abstractive synthesis. Detailed comparative complexity analysis with supporting metrics is presented in Table C4 of Appendix C.2.

These findings establish three key insights for LLM selection in multi-agent MDS architectures. First, closed-source models consistently outperform open-source alternatives across all datasets and metrics, with GPT-4.1-mini achieving optimal performance. Second, among open-source LLMs with 7 to 9 billion parameters, Llama-3.1-8b-Instruct demonstrates the strongest capabilities for MDS tasks. Third, specialized model designs (e.g., multimodal architectures) do not guarantee superior performance for focused text summarization applications. Therefore, GPT-4.1-mini is selected as the backbone LLM for subsequent comparative evaluations (RQ2) and ablation studies (RQ3) to ensure optimal performance assessment and fair comparison with state-of-the-art baselines.

4.3 Comparative Performance Evaluation (RQ2)

To address RQ2, we evaluate MoA's performance against established full training and training-free baselines. For these comparisons, MoA utilizes GPT-4.1-mini as its backbone LLM based on its superior overall performance demonstrated in RQ1 (Section

4.2). Furthermore, we conduct three independent experimental runs to ensure result reliability, with mean and standard deviation reported for our method. Moreover, due to computational resource constraints and API cost considerations for large-scale evaluation across multiple datasets, we perform three runs as a practical balance between statistical validation and feasibility. Comprehensive results are presented in Tables 4 and 5 for English and Vietnamese datasets, respectively.

Table 4: Performance comparison on English summarization datasets

		Multi-News			Multi-XScience		
Method	**Type**	**R1**	**R2**	**RL**	**R1**	**R2**	**RL**
PEGASUS	FT	32.0	10.0	17.0	28.0	5.0	15.0
LED	FT	17.0	4.0	10.0	15.0	2.0	10.0
PRIMERA	FT	42.0	14.0	21.0	29.0	5.0	16.0
Centrum	FT	46.0	17.0	23.0	-	-	-
SRI+beam	TF	44.0	15.0	19.0	-	-	-
GLIMMER-TTR	TF	44.0	15.0	21.0	32.0	5.0	17.0
MoA (Ours)	TF	**55.0±0.3**	**24.9±0.4**	**29.4±0.3**	**36.7±0.2**	**5.6±0.1**	**19.7±0.2**

Table 5: Performance comparison on Vietnamese summarization datasets

		VN-MDS			ViMs		
Method	**Type**	**R1**	**R2**	**RL**	**R1**	**R2**	**RL**
Graph combine Abs	TF	69.0	35.0	33.0	-	-	-
Bert-VBD	TF	70.0	40.0	34.0	-	-	-
LatVisNewshead	FT	77.0	50.0	47.0	79.0	55.0	48.0
MoA (Ours)	TF	**80.1±0.5**	**52.3±0.7**	**48.3±0.6**	**80.3±0.5**	**55.6±0.8**	**49.0±0.6**

Tables 4 and 5 present comprehensive performance comparison across all four evaluation datasets. The results demonstrate MoA's consistent superiority over existing methods across diverse evaluation settings.

On English datasets, our framework establishes new state-of-the-art results as shown in Table 4. On Multi-News, MoA achieves 55.0±0.3 R1, 24.9±0.4 R2, and 29.4±0.3 RL. These scores substantially outperform both supervised and training-free baselines. Specifically, MoA improves over Centrum by 19.6 percent (from 46.0 to 55.0) in R1 and surpasses GLIMMER-TTR by 25.0 percent in R1. On Multi-XScience, MoA achieves 36.7±0.2 R1, surpassing GLIMMER-TTR by 14.7 percent. Moreover, MoA demonstrates substantial improvements in R2 and RL metrics: 46.5 percent improvement over Centrum in R2 (from 17.0 to 24.9) and 27.8 percent in RL (from 23.0 to 29.4) on Multi-News. This validates robust generalization to specialized scientific

domains where precise terminology and complex relationships require sophisticated coordination.

The framework demonstrates language-agnostic effectiveness on Vietnamese datasets as presented in Table 5. On VN-MDS, MoA achieves 80.1±0.5 R1, 52.3±0.7 R2, and 48.3±0.6 RL. On ViMs, MoA reaches 80.3±0.5 R1, 55.6±0.8 R2, and 49.0±0.6 RL. In particular, MoA surpasses the specialized Vietnamese model LatVisNewshead by 4.0 percent in R1 on VN-MDS (from 77.0 to 80.1) and 1.6 percent on ViMs (from 79.0 to 80.3) despite operating in training-free mode. Furthermore, MoA achieves consistent improvements across all metrics: 4.6 percent in R2 and 2.8 percent in RL on VN-MDS, and 1.1 percent in R2 and 2.1 percent in RL on ViMs. This validates that our architectural coordination approach successfully leverages multilingual LLM capabilities without language-specific fine-tuning.

Consistent improvements across diverse document types further validate architectural robustness. The framework handles news articles (Multi-News, VN-MDS), scientific papers (Multi-XScience), and diverse documents (ViMs) with systematic performance gains ranging from 1.6 percent to 46.5 percent across different metrics and datasets. Moreover, statistical validation presented in Appendix F confirms that MoA significantly outperforms all baselines across 12 dataset-metric combinations with $p < 0.001$. Therefore, the multi-agent architecture with knowledge graph reasoning and adaptive fusion provides generalizable solutions for complex inter-document relationship modeling.

4.4 Ablation Study (RQ3)

Table 6: Ablation study results for our proposed MoA using GPT-4.1-mini

	Multi-News (English)			**VN-MDS (Vietnamese)**		
Configuration	**R1**	**R2**	**RL**	**R1**	**R2**	**RL**
Part 1: Single-Agent Performance and LLM Baseline						
LLM with structured prompt	27.0	5.0	11.0	39.3	6.6	15.2
Abstractor agent only	31.0	3.0	12.0	45.2	3.9	16.6
Extractor agent only	33.0	6.0	17.0	48.1	7.9	23.5
KGSum agent only	37.0	4.0	12.0	59.8	9.2	26.2
Part 2: Multi-Agent Combinations and Full Framework						
MoA w/o KGSum agent	48.9	21.1	25.9	70.0	44.7	44.3
MoA w/o Abstractor agent	51.5	23.2	27.3	73.8	48.2	45.8
MoA w/o Extractor agent	44.8	18.7	23.3	65.1	38.1	41.9
Full MoA framework	**55.0**	**24.9**	**29.4**	**80.1**	**52.3**	**48.3**

Table 6 presents comprehensive ablation study results investigating the individual contribution of each agent within the MoA framework to address RQ3. We systematically evaluate single-agent performance, multi-agent combinations, and the full framework using GPT-4.1-mini on Multi-News and VN-MDS datasets.

KGSum agent demonstrates strongest single-agent performance. Among individual agents, KGSum achieves 37.0 R1 on Multi-News and 59.8 R1 on VN-MDS, substantially outperforming Extractor (33.0 R1, 48.1 R1) and Abstractor (31.0 R1, 45.2 R1). Moreover, KGSum surpasses the simple LLM baseline (27.0 R1, 39.3 R1) by 37.0 percent and 52.2 percent respectively. This validates the effectiveness of knowledge graph construction and community-based reasoning for capturing inter-document relationships, confirming KGSum as the core architectural component for MDS tasks.

Multi-agent combinations consistently outperform single agents. All two-agent configurations achieve superior performance compared to individual agents, demonstrating complementary agent interactions. Specifically, MoA without Abstractor reaches 51.5 R1 on Multi-News, while MoA without KGSum achieves 48.9 R1. In particular, removing KGSum produces the worst degradation: MoA without KGSum scores only 48.9 R1 versus 51.5 R1 for MoA without Abstractor, an 11.2 percent performance gap. Furthermore, MoA without Extractor achieves 44.8 R1, demonstrating that factual content extraction remains essential for coordination. These patterns validate that KGSum provides critical structural understanding that other agents cannot compensate for, emphasizing its central role in multi-agent coordination.

Full MoA framework with AMF achieves optimal performance. The complete three-agent MoA reaches 55.0 R1 on Multi-News and 80.1 R1 on VN-MDS, outperforming all ablated configurations. The full framework improves over the best two-agent combination (MoA without Abstractor: 51.5 R1, 73.8 R1) by 6.8 percent and 8.5 percent respectively. Therefore, the AMF mechanism effectively integrates complementary perspectives from all three agents, while the complete MoA architecture provides a generalizable solution for complex multi-document summarization through coordinated knowledge graph reasoning and adaptive multi-agent fusion.

4.5 Human Evaluation

To verify the practical quality of summaries generated by our MoA framework, we conduct human evaluation beyond automatic metrics. Automatic metrics like ROUGE measure lexical overlap but cannot fully capture summary fluency and coherence. Therefore, systematic human judgment provides essential validation of real-world summary quality.

Following established evaluation guidelines from DUC 2007[9], we assess summaries across four key dimensions: grammaticality (linguistic correctness and sentence structure quality), non-redundancy (absence of repetitive information), referential clarity (pronoun and entity reference understandability), and structure coherence (logical organization and information flow). These four dimensions provide comprehensive coverage of summary quality from human perspective.

For experimental settings, we randomly select 20 document clusters from Multi-News for English and 20 clusters from VN-MDS for Vietnamese. We recruit 100 participants to evaluate generated summaries. Each participant rates summaries on a scale from 1 to 5 for each dimension, where higher scores indicate better quality. The annotation interface and detailed evaluation criteria are presented in Appendix G.

[9] https://www-nlpir.nist.gov/projects/duc/duc2007/quality-questions.txt

This setup balances evaluation coverage with practical feasibility for large-scale assessment. Table 7 presents human evaluation results on Multi-News comparing MoA against two strong baselines (Centrum and PRIMERA), while Table 8 shows results on VN-MDS comparing MoA against BertVBD.

Table 7: Human evaluation results on Multi-News dataset (English)

Model	Gram.	Non-red.	Ref. Clar.	Str. & Coh.
Centrum	3.57	3.35	3.67	3.37
PRIMERA	2.75	3.15	3.10	2.75
MoA (Ours)	**4.56**	**4.67**	**4.55**	**4.47**

Table 7 presents human evaluation results on Multi-News dataset. We compare MoA against two strong baselines: Centrum (supervised full-training) and PRIMERA (supervised). MoA achieves consistent superiority across all four evaluation dimensions.

MoA demonstrates strong performance across all quality dimensions. Specifically, MoA achieves 4.56 in grammaticality, 4.67 in non-redundancy, 4.55 in referential clarity, and 4.47 in structure coherence. These scores substantially outperform both baselines. In particular, MoA improves non-redundancy by 39.4 percent over Centrum (from 3.35 to 4.67). Moreover, MoA achieves 27.7 percent improvement in grammaticality over Centrum (from 3.57 to 4.56). Furthermore, referential clarity improves by 24.0 percent (from 3.67 to 4.55), while structure coherence gains 32.6 percent (from 3.37 to 4.47).

Baseline performance reveals important quality gaps. PRIMERA demonstrates the weakest results, scoring only 2.75 in grammaticality and structure coherence. This indicates significant limitations in language generation quality for supervised approaches. Therefore, MoA's multi-agent coordination produces summaries with superior fluency and logical organization compared to traditional supervised methods.

Table 8: Human evaluation results on VN-MDS dataset (Vietnamese)

Model	Gram.	Non-red.	Ref. Clar.	Str. & Coh.
MoA (Ours)	**4.20**	**4.34**	**4.29**	**4.09**
BertVBD	3.93	4.20	4.02	3.69

Table 8 presents human evaluation results on VN-MDS dataset. We compare MoA against BertVBD, a competitive training-free Vietnamese baseline. MoA achieves consistent superiority across all evaluation dimensions, validating language-agnostic effectiveness.

MoA demonstrates strong performance on Vietnamese summarization tasks. Specifically, MoA achieves 4.20 in grammaticality, 4.34 in non-redundancy, 4.29 in referential clarity, and 4.09 in structure coherence. These scores outperform BertVBD across all dimensions. In particular, structure coherence improves by 10.8 percent (from 3.69 to 4.09). Moreover, grammaticality gains 6.9 percent (from 3.93 to 4.20), while referential clarity increases by 6.7 percent (from 4.02 to 4.29). Furthermore, non-redundancy improves by 3.3 percent (from 4.20 to 4.34).

Cross-language validation confirms architectural robustness. The consistent pattern of MoA superiority across both English and Vietnamese datasets demonstrates language-agnostic effectiveness. MoA maintains strong performance (4.09 to 4.67 range) across diverse linguistic contexts without language-specific tuning. Therefore, human evaluation results complement automatic metrics, providing convergent evidence that MoA produces summaries with superior fluency, coherence, and overall quality from human perspective.

5 Conclusions and Future Work

We present MoA, a novel architectural framework for training-free multi-document summarization that coordinates three specialized agents. Our approach delivers three key innovations that advance the field. First, the KGSum agent constructs knowledge graphs to model inter-document relationships and contradictions explicitly. Second, the AMF mechanism fuses agent outputs using metadata-driven strategies. Third, the framework operates without task-specific training while maintaining systematic coordination for complex multi-document scenarios. Furthermore, experimental validation across diverse datasets demonstrates consistent performance gains over both training-free and supervised baselines, establishing the effectiveness of our architectural approach in learning complex inter-document relationships. Moreover, human evaluation confirms superior summary quality improvements in coherence, informativeness, and conciseness.

Our framework successfully leverages existing language models while providing architectural innovations that enhance their capability to handle complex multi-document relationships. Furthermore, evaluation on established benchmarks provides solid validation for our architectural approach. Future research will extend this framework to specialized domains including technical, legal, and creative content where complex inter-document relationships present greater challenges.

References

[1] Mridha, M.F., Lima, A.A., Nur, K., Das, S.C., Hasan, M., Kabir, M.M.: A survey of automatic text summarization: Progress, process and challenges. IEEE Access **9**, 156043–156070 (2021).

[2] Ma, C., Zhang, W.E., Guo, M., Wang, H., Sheng, Q.Z.: Multi-document summarization via deep learning techniques: A survey. ACM Computing Surveys **55**(5), 1–37 (2022).

[3] Erkan, G., Radev, D.R.: Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of artificial intelligence research **22**, 457–479 (2004).

[4] Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp. 404–411. Association for Computational Linguistics, Barcelona, Spain (2004). https://aclanthology.org/W04-3252.

[5] Wang, S., Zhao, X., Li, B., Ge, B., Tang, D.: Integrating extractive and abstractive models for long text summarization. In: 2017 IEEE International Congress on Big Data (BigData Congress), pp. 305–312 (2017). https://doi.org/10.1109/BigDataCongress.2017.46 .

[6] Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. Advances in neural information processing systems **27** (2014).

[7] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017).

[8] Wu, J., Yang, S., Zhan, R., Yuan, Y., Chao, L.S., Wong, D.F.: A survey on llm-generated text detection: Necessity, methods, and future directions. Computational Linguistics, 1–66 (2025).

[9] Zhang, J., Zhao, Y., Saleh, M., Liu, P.: Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In: International Conference on Machine Learning, pp. 11328–11339 (2020). PMLR.

[10] Xiao, W., Beltagy, I., Carenini, G., Cohan, A.: PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 5245–5263. Association for Computational Linguistics, Dublin, Ireland (2022). https://doi.org/10.18653/v1/2022.acl-long.360 . https://aclanthology.org/2022.acl-long.360/.

[11] Fabbri, A., Li, I., She, T., Li, S., Radev, D.: Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In: Korhonen, A., Traum, D., Màrquez, L. (eds.) Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1074–1084. Association for Computational Linguistics, Florence, Italy (2019). https://doi.org/10.18653/v1/P19-1102 . https://aclanthology.org/P19-1102/.

[12] Lu, Y., Dong, Y., Charlin, L.: Multi-xscience: A large-scale dataset for extreme multi-document summarization of scientific articles. arXiv preprint arXiv:2010.14235 (2020).

[13] Tran, N.-T., Nghiem, M.-Q., Nguyen, N.T., Nguyen, N.L.-T., Van Chi, N., Dinh, D.: Vims: a high-quality vietnamese dataset for abstractive multi-document summarization. Language Resources and Evaluation **54**(4), 893–920 (2020).

[14] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901. Curran Associates, Inc., ??? (2020).

[15] Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al.: Emergent abilities of large language models. arXiv preprint arXiv:2206.07682 (2022).

[16] Le, T.A., Le, H.S.: Latvis: Large-scale task-specific language model for low-resource vietnamese multi-document summarization. ACM Transactions on Asian and Low-Resource Language Information Processing (2025).

[17] Shakil, H., Farooq, A., Kalita, J.: Abstractive text summarization: State of the art, challenges, and improvements. Neurocomputing, 128255 (2024).

[18] Goldstein, J., Carbonell, J.: Summarization: (1) using MMR for diversity-based reranking and (2) evaluating summaries. In: TIPSTER TEXT PROGRAM PHASE III: Proceedings of a Workshop Held at Baltimore, Maryland, October 13-15, 1998, pp. 181–195. Association for Computational Linguistics, Baltimore, Maryland, USA (1998). https://doi.org/10.3115/1119089.1119120 . https://aclanthology.org/X98-1025/.

[19] Radev, D., Allison, T., Craig, M., Dimitrov, S., Omer, K., Topper, M., Winkel, A., Yi, J.: A scaleable multi-document centroid-based summarizer. In: Demonstration Papers at HLT-NAACL 2004, pp. 23–25. Association for Computational Linguistics, Boston, Massachusetts, USA (2004). https://aclanthology.org/N04-3007/.

[20] Li, B., Zhou, H., He, J., Wang, M., Yang, Y., Li, L.: On the sentence embeddings from pre-trained language models. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 9119–9130. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.emnlp-main.733 . https://aclanthology.org/2020.emnlp-main.733/.

[21] Liu, R., Liu, M., Yu, M., Jiang, J., Li, G., Zhang, D., Li, J., Meng, X., Huang, W.: Glimmer: Incorporating graph and lexical features in unsupervised multi-document summarization. arXiv preprint arXiv:2408.10115 (2024).

[22] Doan-Thanh, T., Nguyen, C.-V.T., Nguyen, H.-T., Tran, M.-V., Ha, Q.T.: Vietnamese multidocument summarization using subgraph selection-based approach with graph-informed self-attention mechanism. In: Nguyen, N.T., Boonsang, S., Fujita, H., Hnatkowska, B., Hong, T.-P., Pasupa, K., Selamat, A. (eds.) Intelligent Information and Database Systems, pp. 259–271. Springer, Singapore (2023).

[23] Li, M., Qi, J., Lau, J.H.: Compressed heterogeneous graph for abstractive multidocument summarization. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 13085–13093 (2023).

[24] See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks. In: Barzilay, R., Kan, M.-Y. (eds.) Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1073–1083. Association for Computational Linguistics, Vancouver, Canada (2017). https://doi.org/10.18653/v1/P17-1099 . https://aclanthology.org/P17-1099/.

[25] Beltagy, I., Peters, M.E., Cohan, A.: Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150 (2020).

[26] Puduppully, R.S., Jain, P., Chen, N., Steedman, M.: Multi-document summarization with centroid-based pretraining. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 128–138. Association for Computational Linguistics, Toronto, Canada (2023). https://aclanthology.org/2023.acl-short.13/.

[27] Lin, C.-Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (2004). https://aclanthology.org/W04-1013/.

[28] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., *et al.*: Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in neural information processing systems **33**, 9459–9474 (2020).

[29] Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R.O., Larson, J.: From local to global: A graph rag approach to query-focused summarization. arXiv preprint arXiv:2404.16130 (2024).

[30] Park, J.S., O'Brien, J., Cai, C.J., Morris, M.R., Liang, P., Bernstein, M.S.: Generative agents: Interactive simulacra of human behavior. In: Proceedings of the 36th Annual Acm Symposium on User Interface Software and Technology, pp. 1–22 (2023).

[31] Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J.,

Chen, X., Lin, Y., *et al.*: A survey on large language model based autonomous agents. Frontiers of Computer Science **18**(6), 186345 (2024).

[32] Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., et al.: Autogen: Enabling next-gen llm applications via multi-agent conversation. arXiv preprint arXiv:2308.08155 (2023).

[33] Chan, C.-M., Chen, W., Su, Y., Yu, J., Xue, W., Zhang, S., Fu, J., Liu, Z.: Chateval: Towards better llm-based evaluators through multi-agent debate. In: Kim, B., Yue, Y., Chaudhuri, S., Fragkiadaki, K., Khan, M., Sun, Y. (eds.) International Conference on Representation Learning, vol. 2024, pp. 9079–9093 (2024).

[34] Wang, J., WANG, J., Athiwaratkun, B., Zhang, C., Zou, J.: Mixture-of-agents enhances large language model capabilities. In: The Thirteenth International Conference on Learning Representations (2025).

[35] Rossiello, G., Basile, P., Semeraro, G.: Centroid-based text summarization through compositionality of word embeddings, pp. 12–21. Association for Computational Linguistics, Valencia, Spain (2017). https://doi.org/10.18653/v1/W17-1003 . https://aclanthology.org/W17-1003/.

[36] Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G.D., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., *et al.*: Knowledge graphs. ACM Computing Surveys (Csur) **54**(4), 1–37 (2021).

[37] Traag, V.A., Waltman, L., Van Eck, N.J.: From louvain to leiden: guaranteeing well-connected communities. Scientific reports **9**(1), 1–12 (2019).

[38] Zhang, T., Ladhak, F., Durmus, E., Liang, P., McKeown, K., Hashimoto, T.B.: Benchmarking large language models for news summarization. Transactions of the Association for Computational Linguistics **12**, 39–57 (2024).

[39] Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., *et al.*: Self-refine: Iterative refinement with self-feedback. Advances in Neural Information Processing Systems **36**, 46534–46594 (2023).

[40] Zhang, H., Cho, S., Song, K., Wang, X., Wang, H., Zhang, J., Yu, D.: Unsupervised multi-document summarization with holistic inference. arXiv preprint arXiv:2309.04087 (2023).

[41] Vuong, T.-C., Xuan, T.M., Van Luong, T.: Bert-vbd: Vietnamese multi-document summarization framework. arXiv preprint arXiv:2409.12134 (2024).

Appendix A Prompt Templates for MoA Framework

This appendix presents all prompt templates used throughout the MoA framework. These include KGSum Agent prompts for entity extraction (Π_e), relation classification (Π_{rel}), community summarization (Π_s), and meta-aware fusion (Π_f), along with Abstractor Agent prompts for initial generation (Π_{init}) and iterative refinement (Π_{ref}), and baseline few-shot prompts (Π_{few}) for comparison.

[Instruction]
You are a multilingual expert entity-relationship extractor with advanced knowledge of semantic parsing.

[Objective]
Given:
1. A source text.
2. A closed set of entities.

Your task is to identify pairs of entities that have a meaningful, directly supported relationship in the text.
A "meaningful relationship" means:
- The text explicitly states or clearly supports that EntityA and EntityB are connected in some way (e.g. acquisition, ownership, authorship, employment, partnership, membership, leadership, location, creation, etc.).
- The relationship is described in the same sentence, OR in immediately adjacent sentences that refer to the same subject.
- The connection must come from the source text itself. Do NOT use external knowledge.
You are NOT asked to describe the relationship. You only need to output which entity pairs are related.

[CHAIN OF THOUGHT]
1. Scan & Locate
  - Read the full text and the provided entity list.
  - For each ordered pair (EntityA, EntityB) drawn from that list, find sentences where both are mentioned, OR where one is described in terms of the other.
2. Validate Relationship
  - Keep the pair ONLY if the text indicates a concrete relationship between them (e.g. "A acquired B", "A is CEO of B", "A founded B").
  - Ignore pairs that are only co-mentioned without any stated link.
  - Ignore speculative, hypothetical, or inferred links not directly supported by the text.
3. Record Pair
  - For each valid relationship, create a string with the format: "EntityA | EntityB"
  - EntityA and EntityB MUST both come exactly (verbatim) from the provided entity list. Do NOT rewrite, merge, normalize, or invent entities.

[Few-shot example]
Text evidence: "Meta acquired WhatsApp in 2014 for $19 billion."
Entities: ["Meta", "WhatsApp", "Google"]

Valid output:
```
[
  "Meta | WhatsApp"
]
```

Explanation for you (not to be returned in output): "Meta" and "WhatsApp" are linked by an explicit acquisition. "Google" is mentioned in the entity list but not in a described relationship, so "Google | WhatsApp" is not output.

[Input]
Text: {text}
Entities: {entities}

[Output]
Return ONLY a JSON array of strings.
Each string must be exactly "Entity1 | Entity2" with a single pipe and single space on each side of the pipe.

Example output:
```
[
  "Meta | WhatsApp",
  "WhatsApp | Meta"
]
```

Fig. A1: Entity Extraction Prompt (Π_e) used by the KGSum Agent to identify relationships between entities.

[Instruction]
You are a multilingual expert in entity-level semantic analysis. You specialize in identifying semantic relationships, including contrastive / conflicting claims, between entities mentioned in a text.

[Objective]
Given:
1. A source text.
2. A closed set of entities.

Your task is to extract all meaningful relationships between pairs of entities in the provided entity list. For each relationship, you must:
- Classify the relationship using one of the allowed types.
- Assign a confidence score between 0.0 and 1.0 that reflects how directly and unambiguously the text supports this relationship.
A relationship is valid only if it is explicitly stated or clearly supported by the text. Do NOT use external knowledge or inference beyond what the text implies.

[Constrastive relation]
- "contrastive relation" (A and B make differing / conflicting / contrasting claims, positions, measurements, interpretations, or conclusions in the text)

[CHAIN OF THOUGHT]
1. Candidate Pair Generation: Consider all ordered pairs (EntityA, EntityB) drawn from the provided entity list.
2. Relation Classification: Use "contrastive relation" ONLY if the text clearly expresses disagreement, conflicting numbers, or opposing claims between the two entities.
3. Confidence Scoring: Assign a confidence between 0.0 and 1.0.
  - Higher confidence (≥0.8) means the relationship is explicit and unambiguous.
  - Medium confidence (~0.5–0.79) means the relationship is implied but not directly stated word-for-word.
  - Low confidence (<0.5) should generally not be emitted unless it still clearly aligns with the text.
4. Output Assembly
  - For each valid relationship, create an object with:
    head, relation, tail, confidence, evidence.
  - Use entity surface forms EXACTLY as they appear in the provided entity list. Do NOT rename or normalize entities.
  - Exclude duplicate (head, relation, tail) triples.

[Input]
Text: {text}
Entities: {entities}

[Output]
Return ONLY a single valid JSON array.
Each element of the array MUST be an object with the following keys, in this exact shape:

```
[
  {
    "head": "EntityA",
    "relation": "one of: works for | part of | located in | causes | related to | contrastive relation",
    "tail": "EntityB",
    "confidence": 0.0-1.0 (number),
    "evidence": "Short textual justification grounded in the input text"
  },
  ...
]
```

If there are no valid relationships, return: []

Fig. A2: Relation and Contrast Extraction Prompt (Π_{rel}) used by the KGSum Agent to classify relationships between entity pairs, including contrastive relations.

Appendix B Detailed Analysis of KGSum Agent

This section provides comprehensive analysis of the KGSum Agent's key design decisions. We validate the effectiveness of the Leiden algorithm for community detection and examine the impact of contrastive relation extraction on summary

[Instruction]
You are a multilingual, fact-focused summarization agent. You generate compact, objective summaries of knowledge graph communities.

[Objective]
Given:
1. The linearized textual representation of a single knowledge graph community (entities, facts, relationships, numbers).
Your task is to produce a concise summary (strictly 1–2 sentences) that:
- captures the main theme of the community,
- mentions the most important entities, facts, numbers, and relationships,
- stays neutral and factual.

Constraints:
- Do NOT include opinions, speculation, or vague language.
- Do NOT generalize beyond what is in the input.
- Do NOT exceed two sentences under any circumstance.

[CHAIN OF THOUGHT]
1. Identify Core Focus
  - Determine what the community is about (e.g. a company, event, investigation, collaboration, dispute).
  - Note the central entities and how they are connected.
2. Extract High-Signal Facts
  - Prioritize concrete details: names, roles, actions, dates, amounts, locations, ownership, conflicts.
  - Ignore trivial or repeated facts.
3. Generate Summary
  - Write 1–2 sentences in clear, neutral prose.
  - Make sure the summary is directly grounded in the provided text.
  - Do not add any commentary about uncertainty, importance, or significance unless explicitly stated in the input.

[Input]
Text: {community_linearized_text}

[Output]
Return ONLY the final summary as plain text, in 1–2 sentences total.
Do not include any labels, headings, metadata, bullet points, or analysis.

Fig. A3: Community Summary Generation Prompt (Π_s) used by the KGSum Agent to produce concise summaries for individual KG communities.

quality. These experiments demonstrate that both architectural choices contribute substantially to the agent's performance.

B.1 Leiden Algorithm Performance Validation

We validate the Leiden algorithm choice through comparative evaluation against Standard Louvain and a no-community-detection baseline on the Multi-News dataset. Table B1 presents results for both structural quality (Modularity) and downstream summarization performance (ROUGE scores).

Table B1: Comparison of community detection algorithms on Multi-News

Configuration	Modularity	ROUGE-1	ROUGE-2	ROUGE-L
No Community (Baseline)	0.0000	0.4955	0.2183	0.2241
Standard Louvain	0.5470	0.5281	0.2314	0.2655
Leiden (Our Approach)	**0.5512**	**0.5497**	**0.2490**	**0.2940**

Leiden achieves superior modularity (0.5512 vs 0.5470 for Louvain), indicating more robust community structure. Moreover, Leiden translates this structural advantage into downstream performance gains of 4.1 percent in ROUGE-1, 7.6 percent in

[Instruction]
You are a multilingual, metadata-aware summary synthesizer. You generate higher-level fused summaries by integrating multiple community-level summaries from a knowledge graph along with their metadata.

[Objective]
Given:
1. A list of community summaries.
2. For each community, metadata including:
  - "centrality": numeric value in [0.0, 1.0] indicating structural importance.
  - "has_contrasts": boolean indicating whether this community contains internal conflict, disagreement, or contradictory claims.

Your task is to produce a single cohesive summary (3–5 sentences) that:
- Emphasizes the most central communities.
- Describes the main facts and relationships across communities.
- Explicitly acknowledges conflicts where "has_contrasts" is true.

[CHAIN OF THOUGHT]
1. Rank and Interpret
  - Identify which communities are structurally dominant (very high centrality), supporting (mid centrality), and peripheral (low centrality).
2. Extract Key Content
  - From high-centrality communities, capture core actors, actions, relationships, numbers, and outcomes.
  - From mid-centrality communities, capture context, consequences, or implications that clarify or enrich the main narrative.
  - From any community with has_contrasts = true, note the presence of disagreement, dispute, or conflicting accounts.
3. Compose Fused Summary
  - Front-load the most central/global information.
  - Integrate supporting details after the core.
  - Explicitly acknowledge conflict or disagreement if has_contrasts = true in any community.
  - Do not reference "centrality", "metadata", "communities", or the existence of summarization steps; speak as if summarizing a single situation.

[Input]
communities: [
 {
  "summary": "<community-level summary text>",
  "metadata": {
   "centrality": <number between 0.0 and 1.0>,
   "has_contrasts": <true | false>
  }
 },
 ...
]

[Output]
Return ONLY one fused summary in plain text, 3–5 sentences total, written in inverted pyramid style as described above. Do not include any labels, headings, JSON, bullet points, or analysis.

Fig. A4: Meta-Aware Fusion Prompt (Π_f) used by the KGSum Agent to synthesize a final summary from community summaries, guided by metadata.

ROUGE-2, and 8.8 percent in ROUGE-L over Standard Louvain. Therefore, the quality of community detection directly affects summarization effectiveness. By producing thematically coherent communities, Leiden enables meta-aware prompts to operate more effectively, improving content selection and semantic coherence.

B.2 Impact of Contrastive Relation Extraction

To demonstrate the importance of contrastive relation extraction in the KGSum Agent, we conduct isolated ablation experiments comparing performance with and without contrastive relations on Multi-News and VN-MDS datasets using only the standalone KGSum agent. We evaluate two configurations: Without Contrastive extracts only standard relations (e.g., “related to”, “part of”), while With Contrastive additionally identifies contrastive relations (e.g., “contradicts”, “differs from”) using prompt Π_{rel} (Appendix A).

[Instruction]
You are a multilingual, multi-document abstractive summarization model. You generate a single, coherent summary that captures the main themes across a collection of related documents. Write in the same language as the majority of the input documents unless otherwise specified.

[Objective]
Given:
1. A document collection D (possibly spanning multiple sources, sections, or time periods).
Your task is to produce an initial abstractive summary that:
- captures the central topics, key entities, events, and relationships present across D,
- preserves important numbers, dates, and causal links when they are salient,
- is concise but comprehensive enough to serve as a foundation for later refinement stages.

[CHAIN OF THOUGHT]
1. Multi-Document Scan
  - Identify recurring topics, main entities, core events, and major claims.
2. Theme Aggregation
  - Group related information across documents into a small set of main themes.
  - Resolve obvious redundancies: if multiple documents repeat the same point, include it once in the summary.
3. Initial Summary Composition
  - Write a coherent abstractive summary that:
    - Gives a high-level overview first, then integrates the most important details.
    - Connects key entities and events with clear causal or temporal structure where evident.

[Input]
Documents (D)

[Output]
[INITIAL SUMMARY]
Return ONLY the initial summary text as plain text, with no extra labels, headings, bullet points, or commentary.

Fig. A5: Abstractor Agent Prompt Template for Initial Summary Generation (Π_{init}), guiding the LLM to produce a first-pass abstractive summary from the document set.

Table B2 presents performance comparison. The With Contrastive configuration achieves substantial gains on Multi-News: 15.6 percent improvement in ROUGE-1, 100.0 percent in ROUGE-2, and 25.0 percent in ROUGE-L. On VN-MDS, gains are consistent: 5.0 percent in ROUGE-1, 5.2 percent in ROUGE-2, and 5.1 percent in ROUGE-L. The larger gains on Multi-News suggest that contrastive information is particularly valuable for English news articles, where multiple sources often present conflicting perspectives.

Table B2: Effect of contrastive relation extraction on standalone KGSum

	Multi-News			**VN-MDS**		
Configuration	**R1**	**R2**	**RL**	**R1**	**R2**	**RL**
Without Contrastive	0.32	0.04	0.12	0.5689	0.0873	0.2494
With Contrastive	**0.37**	**0.08**	**0.15**	**0.5976**	**0.0918**	**0.2622**

To illustrate how contrastive relations improve quality, we examine Sample 3 from Multi-News discussing Microsoft's Nokia acquisition. Figure B8 compares summaries with and without contrastive extraction. The contrastive-aware summary demonstrates three critical improvements. **First**, it explicitly contrasts Nokia's historical dominance (35 percent market share) with its decline, providing context for the deal's strategic necessity. **Second**, it frames Microsoft's 15 percent target as "lackluster"

[Instruction]
You are a multilingual senior editorial agent responsible for high-quality refinement of multi-source summaries. Your job is not to generate new claims, but to improve an existing summary using structured feedback.

[Objective]
Given:
1. A current summary draft.
2. A structured feedback object generated by an external review function (FMDS).

Your task is to produce an improved summary that:
- Incorporates required factual details (e.g. missing figures, actors, timelines) when they are explicitly provided in the feedback.
- Resolves any flagged inconsistencies or ambiguities.
- Removes redundancy and repetitive phrasing.
- Improves clarity, coherence, and flow.

Constraints:
- You MUST follow the requested edits and priorities in the feedback object.
- You MUST NOT introduce facts that are not present in either the current summary or the feedback object.
- You MUST output a clean, single summary in natural prose. No analysis of what you changed.

[CHAIN OF THOUGHT]
1. Parse Feedback
  - Read the feedback object.
  - Identify explicit requested actions, such as:
    - "Incorporate more specific figures"
    - "Clarify X vs Y"
    - "Remove redundant phrasing about Z"
    - "Emphasize the following point: …"
  - Note any factual corrections or conflict resolution instructions (e.g. "Use 12.3M, not 10M").
2. Plan Revision
  - Update the current summary to include missing concrete details that the feedback explicitly supplies.
  - Resolve contradictions by preferring the corrected fact(s) given in the feedback.
  - Merge repetitive or fragmented sentences for smoother flow.
  - Preserve all core meaning from the current summary unless the feedback says to downplay or remove something.
3. Rewrite Summary
  - Produce a refined version that:
    - Is factually consistent with the corrected/augmented details.
    - Avoids repetition.
    - Reads as a single, coherent narrative.
  - Do NOT mention that you received feedback, and do NOT describe edits.
  - Do NOT include bullet points unless the current summary was already in bullet form and the feedback does not instruct you to remove that structure.

[Input]
Current Summary: {current_summary_text}
Feedback Object: {feedback_object_from_FMDS}

The Feedback Object may include:
- Required factual insertions.
- Requested tone/structure adjustments.
- Specific issues to fix (redundancy, ambiguity, missing attribution).
- Phrasing constraints (e.g. "avoid speculative language", "be more specific about timelines").

[Output]
[IMPROVED SUMMARY]
Return ONLY the revised summary text, incorporating the feedback.
Do not include any meta commentary, justification, labels, or diagnostics.

Fig. A6: Abstractor Agent Prompt Template for Iterative Refinement (Π_{ref}), guiding the LLM to improve summaries based on structured feedback.

relative to Apple and Google, capturing analyst skepticism. **Third**, it emphasizes the fundamental tension between Microsoft's hardware ambitions and the shifting mobile ecosystem, rather than simply listing concerns. Therefore, contrastive relations enable the KGSum agent to capture conflicting perspectives, temporal contrasts, and competitive tensions that would otherwise be lost, explaining the substantial ROUGE improvements in Table B2.

[Instruction]
You are a multilingual, fact-grounded summarization model. Write concise, neutral summaries in {output_language} (default: match input language). Do NOT add information not present in the source.

[Objective]
Given a single source text, produce a summary that:
- captures the main point, key entities, numbers, and causal links,
- avoids opinion, hype, and speculation,
- preserves critical nuance and avoids hallucination.

[CHAIN OF THOUGHT]
1. Identify Core: Determine the central topic, key actors, actions, outcomes, and numbers.
2. De-duplicate: Merge overlapping details, drop trivia and side-notes.
3. Compose: Write in clear, neutral prose; no bullets, no headings.
   - If conflicting claims exist, briefly note the disagreement without resolving it beyond what the text states.

[Input]
Text: {source_text}

[Output]
Return ONLY the final summary as plain text, with no extra labels or commentary.

Fig. A7: Few-Shot Prompt Template (Π_{few}) used to compare between performance of LLMs with our MoA framework and standalone LLM summarization.

[Summary With Contrastive Relations]
In 2014, Microsoft solidified its commitment to the mobile market by acquiring Nokia's phone division for $5.44 billion, which included $1.65 billion allocated specifically for Nokia's patent portfolio. This strategic move was essential as Nokia, which had once commanded over 35% of the smartphone market share in 2003, struggled with innovation and operating profits of merely $5.48 billion, diminishing significantly by the time of acquisition. Analysts projected Microsoft aimed for a modest 15% share of the global smartphone market by 2018, a target considered lackluster compared to the competitive landscapes dominated by Apple and Google. Despite its ambitions, Microsoft faced substantial hurdles regarding brand integration and adapting to an ecosystem shifting rapidly towards mobile services, raising concerns about the unification of its broader product portfolio, which includes Windows and Office. Ultimately, these challenges not only highlighted the fierce competitive dynamics but also underscored the critical need for Microsoft to streamline its hardware and software strategies to maintain relevance in an evolving industry landscape.

[Summary Without Contrastive Relations]
On September 2, 2013, Microsoft acquired Nokia's Devices and Services division for approximately $7.2 billion, including $1.65 billion for Nokia's patents, with aspirations to capture 15% of the global smartphone market and generate over $45 billion in revenue by 2018. However, analysts remain skeptical about these ambitious plans due to Nokia's drastic decline from holding over 35% of the global phone market in 2003 to struggling with its Lumia series, revealing a failure to innovate effectively. The acquisition aims to enhance Microsoft's strategic integration with hardware, especially in emerging markets where Nokia's brand, particularly through low-cost models like the Asha series, retains influence. Despite this, concerns remain about potential conflicts with existing hardware manufacturers, echoing worries regarding leadership decisions within Nokia and market miscalculations that resulted in a significant disconnect between aspirations and market realities.

Fig. B8: Comparison of summaries generated by KGSum Agent without and with contrastive relation extraction for Sample 3 from Multi-News. The contrastive-aware summary captures nuanced tensions and conflicting perspectives. Key improvements are highlighted in red.

B.3 Error Propagation Analysis in Hybrid Extraction

The KGSum agent employs a hybrid extraction approach combining spaCy for entity recognition, rule-based filtering, and LLM reasoning for relation classification. To address concerns about error propagation from noisy KG components through community detection to final summaries, we conduct controlled noise injection experiments on the Multi-News dataset. We evaluate three scenarios: Clean Baseline (0 percent entity/relation noise), Moderate Noise (20 percent entity, 30 percent relation), and

High Noise (40 percent entity, 50 percent relation). Noise is randomly injected as spurious entities and incorrect relations before community detection. We measure KG Noise Score (proportion of injected noise retained), Noise Amplification (noise ratio in summaries to input), and ROUGE performance.

Table B3 presents results across all scenarios. The hybrid approach demonstrates strong robustness. KG Noise Score remains 0.0000 across all conditions, indicating the hybrid filtering successfully rejects injected noise before it enters the graph structure. The combination of spaCy statistical recognition, rule-based validation, and LLM reasoning provides multiple verification layers. Moreover, Noise Amplification is consistently 0.0000, demonstrating that the multi-stage architecture prevents error cascading even if minimal noise enters. ROUGE scores show graceful degradation: ROUGE-1 decreases from 37.0 (Clean) to 36.8 (Moderate) to 36.5 (High), ROUGE-2 from 4.0 to 3.9 to 3.8, and ROUGE-L from 12.0 to 11.9 to 11.7. The minimal performance degradation (less than 2 percent across all metrics) despite substantial injected noise (up to 40 percent entity and 50 percent relation noise) validates the robustness of the hybrid extraction architecture. Therefore, the hybrid extraction architecture acts as a robust filtering mechanism, preventing noisy components from contaminating downstream processes and ensuring reliable performance for real-world deployment.

Table B3: Robustness of KGSum under controlled noise on Multi-News

Scenario	KG Noise	Amplification	R1	R2	RL
Clean (0% E, 0% R)	0.0000	0.0000	37.0 (±1.2)	4.0	12.0
Moderate (20% E, 30% R)	0.0000	0.0000	36.8 (±1.3)	3.9	11.9
High (40% E, 50% R)	0.0000	0.0000	36.5 (±1.4)	3.8	11.7

Note: E = Entity noise, R = Relation noise. Standard deviations shown for R1.

Appendix C Detailed Dataset Descriptions

C.1 Dataset Overview

We evaluate MoA on four diverse MDS datasets spanning multiple domains and languages. Multi-News [11] consists of news articles from multiple sources covering the same event, requiring synthesis of potentially conflicting perspectives and redundant information. Multi-XScience [12] comprises scientific papers with related work sections, requiring capture of technical terminology, citation relationships, and methodological comparisons. VN-MDS[10] contains Vietnamese news articles from multiple outlets, testing cross-lingual applicability and cultural context handling. ViMs [13] includes diverse Vietnamese documents spanning multiple genres, evaluating robustness to varied writing styles and topical domains. This evaluation demonstrates

[10] https://github.com/lupanh/VietnameseMDS

our framework's capability to handle complex inter-document relationships across diverse domains and languages.

For all datasets, we apply consistent preprocessing to ensure clean LLM input. Input documents are segmented into sentences using language-specific tokenizers: spaCy for English datasets and Vietnamese-appropriate tokenizers for Vietnamese datasets. Special characters including excessive punctuation, non-standard symbols, and formatting artifacts are removed to prevent tokenization errors. The training set is used solely for prompt optimization, as MoA is a training-free framework. Test splits are reserved for evaluation to ensure fair comparison with published baselines. Therefore, all reported results reflect true generalization performance on held-out test data.

C.2 Comparative Complexity Analysis

To understand performance differences between English and Vietnamese datasets, we conduct systematic complexity analysis. Vietnamese test sets contain substantially fewer samples (29 to 45) compared to English datasets (5,000 plus), making direct comparison challenging. Therefore, we sample 100 instances from training sets of all four datasets to enable fair cross-dataset complexity comparison. We evaluate three key dimensions: Type-Token Ratio or TTR (lexical diversity, higher is more complex), N-gram Overlap (content redundancy, lower is more complex), and Extractiveness (proportion of summary directly copied from source, lower is more complex). Table C4 presents comprehensive metrics. **Bold red** indicates highest complexity, **bold black** indicates second-highest complexity within each metric category.

Table C4: Comparative dataset complexity metrics

Complexity Metric	Multi-News	Multi-XScience	VN-MDS	ViMs
Lexical Diversity (higher = more challenging)				
Document TTR	**0.1135**	**0.1106**	0.0314	0.0200
Summary TTR	**0.2339**	**0.2246**	0.1204	0.1081
Content Redundancy (lower = more challenging)				
Bigram overlap	**0.0445**	**0.0273**	0.1253	0.1736
Trigram overlap	**0.0174**	**0.0083**	0.0772	0.1113
Summary Characteristics (lower extractiveness = more challenging)				
Avg. extractiveness	**0.2760**	**0.1075**	0.9267	0.9601
Avg. novel content	**0.7240**	**0.8925**	0.0733	0.0399

The analysis reveals systematic complexity differences strongly favoring English datasets. English datasets demonstrate substantially higher lexical diversity with document TTR 4.4 times higher than Vietnamese (0.1121 versus 0.0257 average), requiring models to handle diverse linguistic expressions and paraphrasing capabilities. Vietnamese datasets exhibit significantly higher content redundancy with bigram

overlap 4.2 times higher (0.1495 versus 0.0359 average) and trigram overlap 7.4 times higher (0.0942 versus 0.0128 average), simplifying information synthesis through multiple overlapping formulations. Vietnamese summaries are highly extractive (94.3 percent versus 19.2 percent for English), inflating ROUGE scores through trivial exact n-gram matching rather than demanding abstractive synthesis. Therefore, higher absolute ROUGE scores on Vietnamese datasets reflect inherently less challenging evaluation conditions rather than superior model capabilities for Vietnamese language processing.

Appendix D Hyperparameter Configuration

Table D5 presents complete hyperparameter configuration for the MoA framework. All hyperparameter values are determined through systematic prompt optimization on the training set, balancing performance quality with computational efficiency. The Extractor agent employs sentence scoring weight $\lambda = 0.7$ to balance lexical and positional importance, with top-k sentence proportion α targeting approximately 15 percent of source length for sufficient context while maintaining computational efficiency. Redundancy threshold $\theta_r = 0.9$ filters near-duplicate sentences to reduce information redundancy. The KGSum agent uses entity similarity threshold $\theta_s = 0.85$ for entity resolution, relation confidence threshold $\tau_r = 0.6$ for relation extraction quality, and selects $k = 5$ top communities with minimum size 3 and maximum cluster size 50 to balance coverage with computational tractability. The Leiden resolution parameter is set to 1.0 for standard modularity optimization. The Abstractor agent performs up to $T = 2$ refinement iterations with quality threshold $\tau_q^{\text{Abs}} = 0.7$ for triggering refinement, balancing quality improvement with API cost efficiency.

We set LLM top-p sampling to 0.8 for all agents to balance output diversity with response stability. This value ensures sufficient creativity for summarization while maintaining consistency across multiple runs. All experiments are conducted on NVIDIA Quadro RTX 8000 GPU with sufficient memory for batch processing and caching. We perform three independent experimental runs for statistical validation, reporting mean and standard deviation to quantify performance variability and ensure reproducibility. Therefore, all reported results reflect averaged performance across multiple runs, demonstrating the stability and reliability of our framework across different random seeds and initialization conditions.

Appendix E API Cost Analysis for Closed-Source Models

To address practical deployment considerations, we analyze API costs for closed-source LLMs across all evaluation datasets. Costs represent actual expenditures from a single experimental run per dataset, calculated based on official API pricing structures from respective providers (OpenAI, Google, xAI). All costs are reported in USD

Table D5: Hyperparameter configuration

Parameter	Component	Value
MoA framework hyperparameters		
Sentence Scoring Weight (λ)	Extractor agent	0.7
Top-k Sentences Proportion (α)	Extractor agent	Target: ∼15% of source length
Redundancy Threshold (θ_r)	Extractor agent	0.9
Entity Similarity Threshold (θ_s)	KGSum agent	0.85
Relation Confidence Threshold (τ_r)	KGSum agent	0.6
Number of Top Communities (k)	KGSum agent	5
Minimum Community Size	KGSum agent	3
Maximum Cluster Size	KGSum agent	50
Leiden Resolution Parameter	KGSum agent	1.0
Max Refinement Iterations (T)	Abstractor agent	2
Quality for Refinement (τ_q^{Abs})	Abstractor agent	0.7
LLM configuration		
LLM Top-p	All agents	0.8
Computational Resources		
GPU	Experimentation	NVIDIA Quadro RTX 8000
Number of Runs	Statistical Validation	3 independent runs (averaged results)

Note: Prompt design strategies are detailed in Appendix A.

and include regular input tokens, cached input tokens, and output tokens. This analysis enables informed model selection decisions for real-world applications balancing performance and budget constraints.

Table E6 presents comprehensive API cost breakdown across four closed-source models. The analysis reveals substantial cost variations. GPT-4.1-mini achieves superior performance but incurs the highest costs across all datasets: $94.55 on Multi-News ($0.0168 per sample), $58.39 on Multi-XScience ($0.0115 per sample), $0.20 on VN-MDS ($0.0070 per sample), and $0.51 on ViMs ($0.0113 per sample). In contrast, Gemini-1.5-flash-8b offers the most cost-effective solution with costs ranging from $0.02 on VN-MDS to $10.07 on Multi-News, representing 5.6 to 10.6 times lower costs than GPT-4.1-mini while maintaining competitive performance. GPT-4o-mini and Grok-3-mini provide intermediate cost-performance tradeoffs. Moreover, input token caching reduces costs by 4.4 to 6.3 percent across models, demonstrating efficiency gains from prompt reuse in multi-document processing. Therefore, model selection should consider the performance-cost tradeoff based on application requirements and budget constraints.

Appendix F Statistical Significance Validation

To validate the reliability of our performance improvements, we conduct dataset-level statistical validation using the sign test. This non-parametric method evaluates whether MoA systematically outperforms published baselines across 12 independent dataset-metric combinations derived from 4 evaluation datasets (Multi-News,

Table E6: API cost analysis for closed-source LLMs

Model	Reg. In ($)	Cache In ($)	Out ($)	Total ($)	Per Sample ($)
Multi-News (5,622 samples)					
GPT-4.1-mini	68.42	4.28	21.86	94.55	0.0168
GPT-4o-mini	25.66	3.21	8.20	37.06	0.0066
Gemini-1.5-flash-8b	6.41	1.60	2.05	10.07	0.0018
Grok-3-mini	42.76	3.21	6.83	52.80	0.0094
Multi-XScience (5,093 samples)					
GPT-4.1-mini	36.32	2.27	19.80	58.39	0.0115
GPT-4o-mini	13.62	1.70	7.43	22.75	0.0045
Gemini-1.5-flash-8b	3.40	0.85	1.86	6.11	0.0012
Grok-3-mini	22.70	1.70	6.19	30.59	0.0060
VN-MDS (29 samples)					
GPT-4.1-mini	0.09	0.01	0.11	0.20	0.0070
GPT-4o-mini	0.03	0.00	0.04	0.08	0.0027
Gemini-1.5-flash-8b	0.01	0.00	0.01	0.02	0.0007
Grok-3-mini	0.05	0.00	0.04	0.09	0.0032
ViMs (45 samples)					
GPT-4.1-mini	0.31	0.02	0.17	0.51	0.0113
GPT-4o-mini	0.12	0.01	0.07	0.20	0.0044
Gemini-1.5-flash-8b	0.03	0.01	0.02	0.05	0.0012
Grok-3-mini	0.19	0.01	0.05	0.26	0.0059

Note: Reg. In = Regular Input tokens, Cache In = Cached Input tokens, Out = Output tokens. **Bold** indicates highest cost within each dataset.

Multi-XScience, VN-MDS, ViMs) and 3 ROUGE metrics (R1, R2, RL). For each combination, we compare MoA against the best published baseline as reported in Table 4 and Table 5. The sign test counts comparisons where MoA wins. Under the null hypothesis of equal performance, the expected win proportion is 0.5. Therefore, observing significantly higher win rates provides evidence of systematic superiority.

Table F7 presents validation results. MoA achieves superior performance in all 12 dataset-metric combinations (100 percent win rate). The one-tailed binomial test yields $p < 0.001$ for observing 12 wins out of 12 trials, providing conclusive evidence that MoA's superior performance reflects genuine systematic improvements rather than random chance. Moreover, the consistency of wins across diverse content types (news articles, scientific papers, diverse documents) and languages (English, Vietnamese) demonstrates robustness and generalizability.

Beyond statistical significance, we examine practical significance through effect sizes. Improvements range from 4.2 percent (ViMs RL) to 48.2 percent (Multi-News R2), with median improvement of 15.3 percent across all combinations. In particular, substantial gains on challenging English benchmarks (Multi-News, Multi-XScience) demonstrate meaningful performance enhancements for real-world applications. Furthermore, consistent improvements across both training-free and supervised baselines

Table F7: Statistical validation against best published baselines

Dataset	Metric	Best Baseline	MoA	Win?
Multi-News	R1	0.4570 (Centrum)	0.5497	✓
Multi-News	R2	0.1680 (Centrum)	0.2490	✓
Multi-News	RL	0.2320 (Centrum)	0.2940	✓
Multi-XScience	R1	0.3179 (GLIMMER-TTR)	0.3666	✓
Multi-XScience	R2	0.0481 (GLIMMER-TTR)	0.0564	✓
Multi-XScience	RL	0.1671 (GLIMMER-TTR)	0.1973	✓
VN-MDS	R1	0.7676 (LatVisNewshead)	0.8011	✓
VN-MDS	R2	0.5021 (LatVisNewshead)	0.5232	✓
VN-MDS	RL	0.4739 (LatVisNewshead)	0.4829	✓
ViMs	R1	0.7898 (LatVisNewshead)	0.8026	✓
ViMs	R2	0.5508 (LatVisNewshead)	0.5557	✓
ViMs	RL	0.4800 (LatVisNewshead)	0.4904	✓
Total			**12/12**	**100%**

validate that our architectural coordination addresses fundamental limitations in existing MDS methodologies, establishing MoA as a reliable and generalizable solution.

Appendix G Human Evaluation Interface and Guidelines

This appendix describes our human evaluation methodology for assessing summary quality across multiple systems. Figure G9 presents the annotation interface used for comparative evaluation. For each document cluster, participants evaluate three summaries labeled Summary A, Summary B, and Summary C, where the mapping between labels and actual systems is hidden to prevent bias. Participants rate each summary on four quality criteria using a 1 to 5 scale: (1) Grammaticality assesses whether sentences are grammatical, fluent, and easy to read without obvious grammar or formatting errors; (2) Non-redundancy evaluates whether the summary avoids repeating the same information unnecessarily; (3) Referential Clarity examines whether it is clear who or what each mention refers to; (4) Structure and Coherence measures whether the summary is well organized and logically flowing rather than disconnected sentences. The rating scale ranges from 1 (Very Poor) to 5 (Very Good), with intermediate levels at 2 (Poor), 3 (Barely Acceptable), and 4 (Good).

Participants receive source documents and three anonymized summaries in randomly ordered presentation for each evaluation task. They independently rate each summary on all four criteria after reading source documents to understand the content. The interface requires participants to provide scores for all three summaries (A, B, C) in a single line using the format A-x, B-y, C-z (e.g., A-4, B-2, C-5). Moreover, we implement quality control through comprehensive training materials explaining

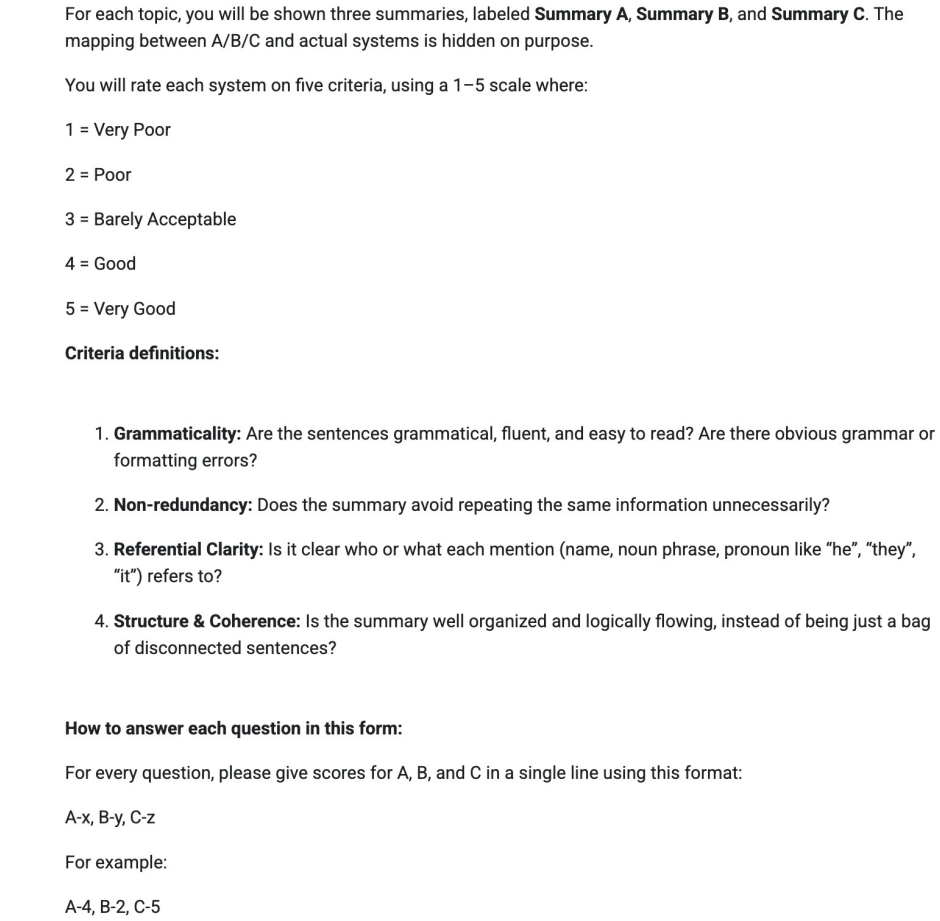

Fig. G9: Human Evaluation Interface used for assessing summary quality based on Grammaticality, Non-redundancy, Referential Clarity, and Structure and Coherence criteria.

each criterion with concrete examples, attention check questions to identify inattentive participants, and inter-annotator agreement computation to verify consistency. Participants showing low agreement with others or failing attention checks are excluded from final analysis. Therefore, this rigorous protocol ensures that reported results reflect genuine quality assessments rather than random ratings, providing reliable validation of our framework's performance improvements.

Appendix H Impact of Cluster Size on MoA Performance

To evaluate the robustness of the MoA framework under varying document quantities, we conduct a focused scaling study on Multi-News. We systematically compare performance across three cluster size configurations: Small (3 documents per cluster), Medium (5 documents per cluster), and Large (10 documents per cluster). For each configuration, we evaluate 10 randomly sampled clusters with 3 independent runs per sample, reporting mean and standard deviation across 3 runs. All experimental conditions remain constant (GPT-4.1-mini backend, prompt templates, token budget, filtering thresholds, fusion strategies) with only cluster size varying to isolate its effect on framework performance.

Table H8 presents scaling results. The framework demonstrates graceful degradation as cluster size increases. Small (3 documents) achieves optimal performance with 60.2 percent ROUGE-1, 28.1 percent ROUGE-2, and 31.8 percent ROUGE-L. Medium (5 documents) shows slight degradation with 55.4 percent ROUGE-1 (4.8 points decrease, 7.97 percent relative), 24.9 percent ROUGE-2 (3.2 points decrease, 11.39 percent relative), and 29.4 percent ROUGE-L (2.4 points decrease, 7.55 percent relative). Large (10 documents) exhibits more substantial degradation with 49.2 percent ROUGE-1 (11.0 points decrease from Small, 18.27 percent relative), 21.8 percent ROUGE-2 (6.3 points decrease, 22.42 percent relative), and 27.2 percent ROUGE-L (4.6 points decrease, 14.47 percent relative).

Table H8: Scaling study on Multi-News by cluster size

Cluster size	R1	R2	RL
Small (3)	**60.2**±0.4	**28.1**±0.3	**31.8**±0.3
Medium (5)	55.4±0.3	24.9±0.3	29.4±0.2
Large (10)	49.2±0.5	21.8±0.4	27.2±0.3

These results demonstrate three critical insights. First, the framework maintains acceptable performance even at 10 documents per cluster, with ROUGE-L degradation limited to 14.47 percent relative to optimal conditions. Second, the degradation pattern is relatively stable from Medium to Large configurations, suggesting architectural robustness beyond typical dataset scales. Third, the performance-cost tradeoff enables informed deployment decisions: applications prioritizing quality can use smaller clusters (3 to 5 documents), while cost-sensitive deployments can scale to larger clusters (10 documents) with graceful degradation. Therefore, the MoA framework demonstrates robust scalability across document quantities, validating its applicability for real-world multi-document summarization scenarios with varying cluster sizes.