

The Cost of Decomposition: Trade-Offs in Long-Context Aspect-Based Scientific Summarization

Pierre Achkar ^{1,2} Elena Senger ^{1,3} Tim Gollub ⁴
Martin Potthast ^{2,5,6} Yuri Campbell ¹

¹Fraunhofer ISI, Germany ²University of Kassel, Germany ⁴Bauhaus-Universität Weimar, Germany

³MaiNLP, Center for Information and Language Processing, LMU Munich, Germany

⁵hessian.AI, Germany ⁶ScaDS.AI, Germany

pierre.achkar@uni-kassel.de elena.senger@isi.fraunhofer.de tim.gollub@uni-weimar.de

martin.potthast@uni-kassel.de yuri.cassio.campbell.borges@isi.fraunhofer.de

Abstract

Aspect-based scientific summarization produces a focused summary of a long document for a given aspect (e.g., *Approach*, *Methods*). Prior work decomposes the task, assuming a model can neither attend over the full document nor resolve aspect complexity in one pass. We revisit both assumptions now that compact yet capable language models handle long contexts natively. Across three datasets and two primary open long-context models, decomposition provides no reliable quality advantage over direct full-document prompting. With the main 9B model, zero-shot is strongest on aggregate lexical, semantic, and claim-recall/F1 metrics; under the 7B robustness model, simple decomposed variants occasionally edge zero-shot, but gains are small and inconsistent. On these primary models, the multi-role agentic pipeline is never best in aggregate and incurs substantially higher token cost, whereas RAG uses roughly one seventh of zero-shot’s tokens but delivers lower, less consistent quality. We characterize where added complexity loses, quantify the cost–quality trade-off, and distill practical deployment guidance. Code is available on GitHub.¹

1 Introduction

Scientific document summarization condenses research papers into concise summaries that preserve salient contributions (Cohan et al., 2018). Because papers are long, structured, and technically dense (Yu et al., 2020), users often need summaries focused on a specific aspect, such as the *Outcome* (Takeshita et al., 2024) or *Method* (Meng et al., 2021), rather than a generic overview. Aspect-based scientific summarization addresses this problem: given a full document and a target aspect, produce a summary covering exactly the aspect-relevant content.

¹<https://github.com/webis-de/emnlp26-aspect-based-scientific-summarisation>

In practice, the task is further constrained: document confidentiality and API cost at scale require locally deployable models, predictable cost, and minimal reliance on labeled data or external APIs.

A common approach in long-context summarization decomposes the task into intermediate steps: evidence retrieval (Feng et al., 2025), extract-then-abstract generation (Takeshita et al., 2024; Zhong and Litman, 2025), iterative self-refinement (Wadhwa et al., 2024; Zhang et al., 2023), or agentic verification (Wang et al., 2025b). These designs share an implicit premise: the model cannot process the whole document in one pass, so the document or task must be broken apart and recombined.

This premise is increasingly questionable for two converging reasons. First, recent evidence on agentic systems shows decomposition is not a free improvement: added coordination introduces information fragmentation, error propagation, and inter-agent misalignment, making its value task- and architecture-dependent rather than automatic (Kim et al., 2025; Cemri et al., 2026).

Second, increasingly capable compact models deployable under a modest GPU budget now offer long-context variants that can ingest a full paper in a single pass (Qwen Team, 2026; Gemma Team et al., 2025). If the model can hold the whole document in a unified context, it is no longer obvious that breaking the input or task into steps improves aspect coverage enough to offset the orchestration cost and the new opportunities for evidence loss.

We therefore ask whether task decomposition pays off for long-context aspect-based scientific summarization with local LLMs. We compare six training-free pipelines under identical decoding and length-control conditions across three datasets (ACLSum, FacetSum, and our PMC-SA benchmark) and two primary open long-context models, with a supplementary 2B scale analysis. Our finding on the primary models is consistent: decomposition does not reliably improve quality. A single-

pass zero-shot generation is the strongest quality default on lexical overlap, semantic similarity, and claim-level factual overlap to reference summaries. Among full-document pipelines, it also provides the most favorable quality–token trade-off; RAG uses about one seventh as many tokens but yields lower and less consistent quality. Moreover, the two most heavily-fragmenting pipelines are never best in aggregate on either primary model, although they have isolated per-aspect wins.

Our contributions are:

- We present a controlled comparison of six local, training-free pipelines for long-context aspect-based scientific summarization.
- We evaluate two models across three datasets using complementary metrics, including ROUGE, BERTScore, and claim-level factual overlap.
- We analyze the quality–cost trade-off of decomposition and provide per-aspect and qualitative evidence showing that zero-shot dominates most alternatives, while decomposed pipelines often lose aspect-relevant evidence.

2 Related Work

Early aspect-based scientific summarization used supervised models with limited context windows. To make training feasible, these methods decomposed papers by discourse structure. SUSIE aligned full-text IMRaD sections with matching structured-abstract sections and trained section-specific generation (Gidiotis and Tsoumakas, 2019). Follow-up work scaled the same section-alignment strategy on large PubMed collections and compared unified versus per-section models (Oh et al., 2023). FacetSum introduced a large-scale dataset with four fixed aspects, enabling tagged aspect-controlled generation (Meng et al., 2021). Self-supervised variants further used section headings as pseudo-aspect prompts to improve transfer to unseen aspect names (Soleimani et al., 2022). Overall, this line improved controllability, but it depended on aligned supervision, section-header priors, or predefined aspect schemes.

Alongside trained-model approaches, recent work shifts to inference-time pipeline design. On ACLSum, an expert-annotated benchmark with extractive and abstractive references for three aspects (Challenge, Approach, Outcome), extract-then-abstract (E2A) pipelines are compared with

end-to-end generation (Takeshita et al., 2024): Explicit extraction helps smaller fine-tuned models, but can become a bottleneck when LLMs must extract evidence. ReflectASP applies E2A to open aspects in educational reflections and reports gains from external factuality-guided refinement over zero-shot prompting (Zhong and Litman, 2025). TRACSUM (Chu et al., 2025) introduces a medical aspect-summary dataset with sentence-level evidence links and shows that evidence tracking before summarization improves grounding over summarize-then-track and end-to-end generation. SARESG (Feng et al., 2025) uses aspect-aware retrieval to prune long inputs before generation.

Beyond aspect-specific pipelines, generic methods use iterative rewriting or explicit verification. Self-refinement frameworks revise drafts through critique loops (Wadhwa et al., 2024; Zhang et al., 2023); densification methods iteratively increase information density under fixed length budgets (Adams et al., 2023); and agentic systems decompose summarization into specialized roles with verification, such as quiz-based coverage checks in SUMMQ (Wang et al., 2025b). These designs increase control, but also add orchestration cost and new failure points.

Overall, current systems trade operational complexity for control or quality gains through extraction, retrieval, densification, refinement, or agentic verification. To our knowledge, no prior study compares these design choices under local, training-free, long-context deployment constraints; we examine whether the added complexity pays off.

3 Experimental Setup

This section describes the evaluation setup: benchmark datasets, compared local models, evaluated inference pipelines, and metrics.

3.1 Datasets

We evaluate on three benchmarks with complementary domains, aspects, and source lengths.

ACLSum² Expert-written summaries of 250 NLP papers across three aspects: Challenge, Approach, and Outcome (Takeshita et al., 2024). Since the original release contains only Abstract, Introduction, and Conclusion text, we source the corresponding full papers for long-context evaluation. For evaluation, we use the full test split of 100 papers, yielding 300 aspect–summary pairs.

²<https://github.com/sobamchan/aclsum>

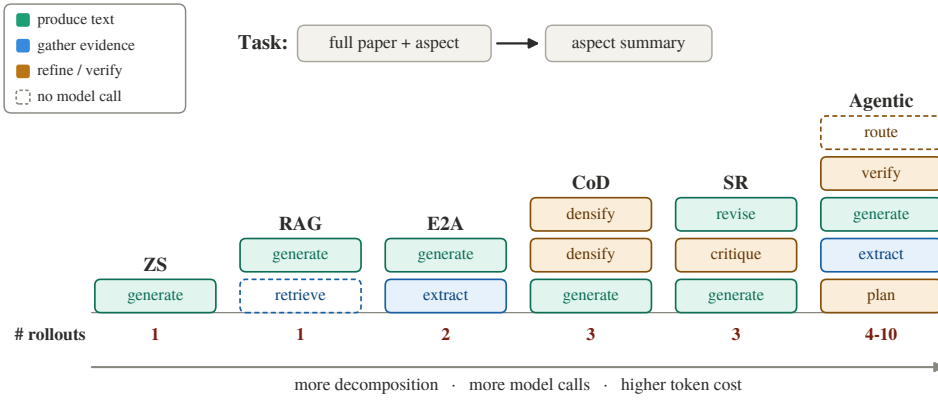


Figure 1: Overview of the compared training-free summarization pipelines used in our experiments.

FacetSum³ Over 60,000 Emerald journal articles covering a diverse range of domains are paired with summaries for four aspects: Purpose, Methods, Findings, and Value (Meng et al., 2021). The full dataset contains 46,289 training, 6,000 development, and 6,000 test documents. For evaluation, we sample 1,000 test documents, yielding 4,000 aspect-summary pairs.

PMC-SA (Structured Abstract) From PubMed Central, we derive a biomedical benchmark of full-text articles with section-aligned structured abstracts. The full filtered dataset contains 122,228 training, 15,278 validation, and 15,280 test documents. It covers Background, Objectives, Methods, Results, and Conclusions. For evaluation, we sample 1,000 test documents, yielding 3,594 aspect-summary pairs as not every document contains all five aspects.

For detailed dataset construction and sampling strategy, see Appendix A. The three benchmarks also differ in where aspect-relevant evidence occurs within the source document: some aspects are strongly localized, while others may draw on sentences distributed throughout the full text. We quantify this with the structural alignment protocol of Meng et al. (2021) and report the resulting position distributions in Appendix A.5.

3.2 Models

We focus on models deployable locally under a modest GPU budget, represented by a single NVIDIA A100 GPU with 40 GB of VRAM in our setup. We choose Qwen/Qwen3.5-9B⁴ for its strong performance within its size family.

To test whether our conclusions depend on a single model family, we additionally evaluate allenai/Olmo-3-7B-Instruct⁵ as a complementary robustness check.

We further use Qwen/Qwen3.5-2B for a supplementary scale analysis across all pipelines and datasets.

Both primary models support long contexts (Qwen3.5-9B: 262,144 tokens; OLMO-3-7B-Instruct: 64K tokens). Our dataset length profiling shows maximum source lengths of 14,874 (ACLSum), 26,701 (FacetSum), and 60,656 (PMC). Since even the largest profiled case remains within the 64K window, we do not apply source truncation for any pipeline in the reported experiments.

Across all experiments, we use the same decoding configuration (*temperature* 0.1, *top-p* 0.95) and set the maximum output length per dataset to match the corresponding reference-summary length scale.

3.3 Pipelines

We compare six training-free pipelines; Figure 1 visualizes their key components:

Zero-Shot (ZS) Direct one-pass generation from the full content of the document.

Extract-to-Abstract (E2A) Two-stage pipeline in which the LLM first extracts aspect-relevant evidence from the full document and then generates the final summary from that evidence (Zhong and Litman, 2025; Takeshita et al., 2024).

Retrieval-Augmented Generation (RAG) Aspect-aware retrieval selects relevant document chunks without an LLM extraction step before summary generation over the retrieved context (Feng et al., 2025).

³<https://huggingface.co/datasets/memray/FacetSum>

⁴<https://huggingface.co/Qwen/Qwen3.5-9B>

⁵<https://huggingface.co/allenai/Olmo-3-7B-Instruct>

Chain of Density (CoD) Iterative densification rewrites an initial draft to increase density under fixed length limits (Adams et al., 2023).

Self-Refine (SR) Three-step self-improvement pipeline that generates, critiques, and refines summaries through iterative self-feedback (Zhong and Litman, 2025; Wadhwa et al., 2024).

Agentic multi-role pipeline (Agentic) Multi-role decomposition into planner, extractor, writer, verifier, and router roles, with bounded revision loops for controlled evidence selection, drafting, and verification.

Implementation details for each pipeline are provided in Appendix B.

3.4 Evaluation Metrics

We measure established ROUGE metrics (Lin, 2004): ROUGE-1 (R-1), ROUGE-2 (R-2), ROUGE-L (R-L), and ROUGE-LSum (R-L-S). ROUGE-LSum is the summary-level variant of ROUGE-L, computed over sentence-level longest common subsequences. We also report BERTScore (Zhang et al., 2020) (BS).

In addition, we report claim-based factual overlap scores, reproducing the TRACSUM evaluation setup (Chu et al., 2025) and its metric design grounded in atomic-claim evaluation (Min et al., 2023). In this setup, both reference and generated summaries are decomposed into atomic claims using mistralai/Mistral-Large-Instruct-2407⁶ and then checked for bidirectional entailment with TRUE as the NLI evaluator (Honovich et al., 2022). For each pair (*generated*, *reference*), we compute Claim Recall (C-R), the fraction of reference claims entailed by the generated summary; Claim Precision (C-P), the fraction of generated claims entailed by the reference summary; and Claim F1 (C-F1), the per-pair harmonic mean of C-P and C-R. Final scores are macro averages over all pairs. This setup captures factual alignment to the reference summary at the claim level rather than direct source-grounded faithfulness.

As a supplementary source-grounded evaluation, we use G-Eval (Liu et al., 2023) with GPT-4.1-mini to rate aspect consistency and relevance against the full source document on 100 documents per dataset. We treat these LLM judgments strictly as an automatic proxy, not as a substitute for human evaluation; full details are in Appendix E.

⁶<https://huggingface.co/mistralai/Mistral-Large-Instruct-2407>

4 Results

This section presents the main quantitative results for the compared pipelines. We first report ROUGE and BERTScore results for Qwen3.5-9B, then summarize claim-based scores and the OLMo-3-7B-Instruct robustness results.

For the Qwen3.5-9B model, Table 1 reports ROUGE and BERTScore results across all three datasets. ZS obtains the best score on every reported metric for ACLSum, PMC-SA, and FacetSum. This ranking holds across both lexical overlap metrics and BERTScore. RAG is the closest competitor on ACLSum. E2A is the highest-scoring decomposed baseline on FacetSum, but still lags behind ZS. The agentic pipeline is never best and only reaches second place on a subset of PMC metrics.

The document-level clustered bootstrap tests in Appendix C report the same comparisons; ZS is higher in all 75 dataset-metric comparisons, with 71 significant at the 95% level. The four non-significant cases are the ZS-RAG comparisons on ACLSum for ROUGE-2, ROUGE-L, ROUGE-Lsum, and BERTScore.

Under claim-based evaluation, Table 2 shows the same trend. ZS achieves the highest claim recall and claim F1 on all three datasets. Claim precision is the only claim metric where decomposed methods rank first in individual cases, such as RAG on ACLSum and SR on FacetSum.

The supplementary source-grounded G-Eval results agree with this ranking: ZS obtains the highest aspect-consistency and aspect-relevance scores on all three datasets, and no decomposition pipeline surpasses it (Appendix E).

We additionally vary RAG retrieval depth and chunk size and increase the E2A evidence budget on a fixed 100-document subset of each dataset. Larger RAG contexts narrow the gap, but ZS remains significantly ahead; the strongest RAG variant trails ZS by 0.054 Claim F1. Increasing the E2A budget yields no significant change. Appendix D provides the full analysis.

The OLMo-3-7B-Instruct results in Appendix G show lower absolute scores and a less uniform ranking than Qwen, but the overall pattern remains similar: decomposed pipelines do not establish a reliable quality advantage. Even when quality is comparable, ZS remains preferable to higher-cost full-context pipelines, while RAG trades lower token use for the risk of missing relevant evidence.

Exp	ACLSum					PMC-SA					FacetSum				
	R-1	R-2	R-L	R-L-S	BS	R-1	R-2	R-L	R-L-S	BS	R-1	R-2	R-L	R-L-S	BS
ZS	0.355	0.129	0.262	0.262	0.889	0.369	0.115	0.218	0.310	0.871	0.343	0.105	0.226	0.279	0.879
E2A	0.314	0.098	0.237	0.237	0.886	0.340	0.095	0.202	0.286	<u>0.869</u>	<u>0.298</u>	<u>0.081</u>	<u>0.197</u>	<u>0.239</u>	0.874
RAG	<u>0.335</u>	<u>0.120</u> [†]	<u>0.250</u> [†]	<u>0.250</u> [†]	<u>0.888</u> [†]	0.294	0.068	0.169	0.248	0.855	0.278	0.063	0.179	0.224	0.868
COD	0.251	0.073	0.190	0.190	0.870	<u>0.353</u>	<u>0.106</u>	0.208	<u>0.297</u>	0.868	0.293	0.077	0.191	0.237	0.867
SR	0.301	0.097	0.231	0.231	0.885	0.328	0.087	0.196	0.279	0.868	0.282	0.072	0.191	0.232	<u>0.875</u>
Agentic	0.260	0.070	0.189	0.189	0.871	0.346	0.103	<u>0.212</u>	0.293	<u>0.869</u>	0.293	0.079	0.195	0.235	0.871

Table 1: Automatic summarization evaluation results on ACLSum, PMC-SA, and FacetSum using ROUGE and BERTScore metrics. R-1, R-2, R-L, and R-L-S denote ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-LSum, respectively, while BS denotes BERTScore. Best results are shown in bold and second-best results are underlined. The four non-statistically significant differences (ZS—pipeline) under document-level clustered bootstrap tests are marked with †; for detailed statistical test analysis, see Appendix C.

Exp	ACLSum			PMC-SA			FacetSum		
	C-P	C-R	C-F1	C-P	C-R	C-F1	C-P	C-R	C-F1
ZS	0.179	0.315	0.150	0.208	0.409	0.198	<u>0.166</u>	0.338	0.154
E2A	<u>0.184</u>	0.212	0.117	0.200	0.317	0.162	0.147	0.229	0.110
RAG	0.195	<u>0.246</u>	<u>0.144</u>	0.096	0.159	0.069	0.098	0.132	0.062
COD	0.094	<u>0.192</u>	0.074	0.179	<u>0.405</u>	<u>0.176</u>	0.100	<u>0.290</u>	0.095
SR	0.180	0.180	0.111	0.203	0.297	0.159	0.171	0.203	0.112
Agentic	0.093	0.203	0.075	<u>0.204</u>	0.327	0.173	0.145	0.241	<u>0.113</u>

Table 2: Claim-based evaluation results. C-P, C-R, and C-F1 denote claim precision, claim recall, and claim F1, respectively. Best results are in bold and second-best results are underlined.

The supplementary 2B scale analysis provides a boundary condition: decomposition earns more local wins, but no pipeline dominates across datasets, and structured multi-stage execution becomes less reliable. Agentic fails to produce a valid output for 11.7–30.9% of cases across datasets (Appendix F).

5 Discussion

This section discusses pipeline complexity, aspect-level behavior, and local deployment.

5.1 Cost-Quality Analysis

The cost–quality trade-off in Figure 2 relates summary quality, measured by ROUGE-LSum, to mean token usage. In our setup, ZS is the strongest quality default and provides the most favorable quality–token trade-off among full-document pipelines. It matches or exceeds the quality of more complex full-context pipelines while remaining simpler to run. RAG occupies a distinct lower-cost point: it uses about one seventh as many tokens as ZS and is competitive on ACLSum, but provides lower and less consistent quality overall, with larger drops on PMC-SA and FacetSum.

Token usage is only an approximation of deployment cost. Multi-step pipelines can also increase latency and add failure points through repeated con-

text construction, additional generation steps, and coordination logic. RAG may therefore be preferable when minimizing token use outweighs its quality reduction, while retrieval quality depends on retaining the relevant evidence. Higher-cost pipelines can be useful when their intermediate outputs are needed for inspection or control.

5.2 Per-aspect Scores

Across individual aspects, Table 3 shows the same pattern. ZS obtains the best BERTScore on eight of eleven aspect categories. The three exceptions are narrow and do not generalize to neighboring aspects: RAG is best on ACLSum *Challenge*, E2A on PMC-SA *Conclusions*, and Agentic on PMC-SA *Background*. All three are aspects whose evidence is positionally concentrated (Figure 4), whereas ZS is best on every aspect whose aligned evidence is broadly distributed. Selective pipelines thus appear to recover their cost only where the relevant content is localized. Appendix H reports the same breakdown for ROUGE variants.

No decomposed pipeline consistently dominates on abstract or difficult aspects. E2A is often competitive but rarely best, suggesting sensitivity to extraction recall; RAG helps in isolated cases but is inconsistent when evidence is dispersed; CoD

Aspect	ZS	E2A	RAG	COD	SR	Agentic
Challenge	0.8785	0.8781	0.8802	0.8655	0.8773	0.8672
Approach	0.8960	0.8944	0.8957	0.8760	0.8900	0.8751
Outcome	0.8937	0.8858	0.8895	0.8671	0.8874	0.8709
Purpose	0.8959	0.8894	0.8845	0.8804	0.8906	0.8927
Methods	0.8701	0.8647	0.8572	0.8641	0.8663	0.8605
Findings	0.8786	0.8731	0.8622	0.8644	0.8740	0.8654
Value	0.8652	0.8631	0.8633	0.8546	0.8605	0.8581
Background	0.8754	0.8741	0.8639	0.8724	0.8690	0.8761
Conclusions	0.8759	0.8789	0.8630	0.8718	0.8781	0.8763
Results	0.8669	0.8608	0.8379	0.8643	0.8630	0.8648
Objectives	0.9057	0.8982	0.8893	0.8941	0.8993	0.9021

Table 3: Aspect-level BERTScore results.

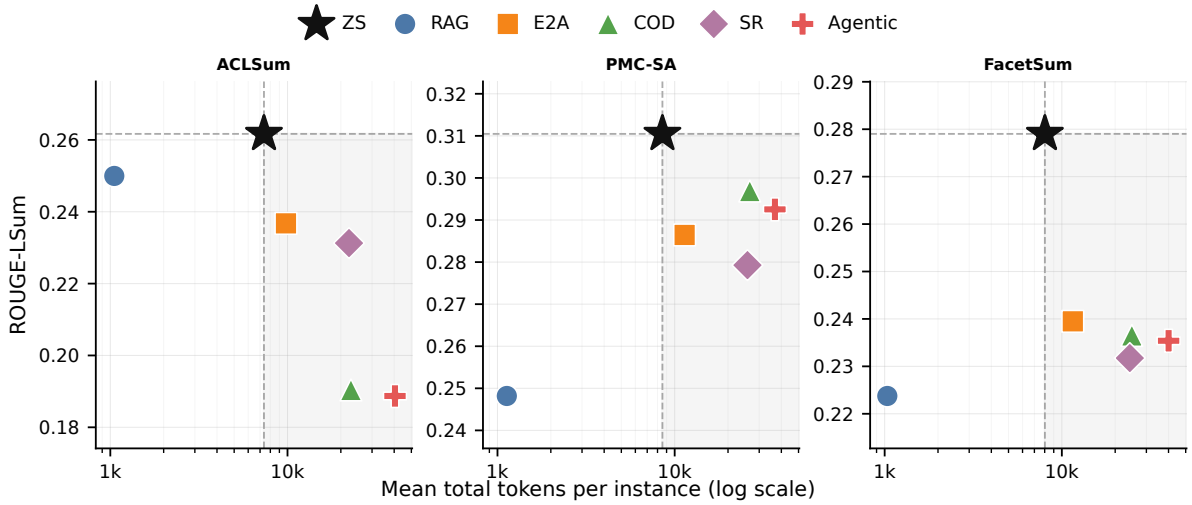


Figure 2: Quality–cost trade-off across datasets. Each point represents one summarization pipeline, with mean total token usage (input plus output tokens) on the x-axis and ROUGE-LSum on the y-axis. Dashed lines indicate the zero-shot pipeline (ZS). The shaded area marks configurations dominated by ZS, i.e., methods that consume more tokens while achieving lower summarization quality.

is generally weaker, suggesting that densification removes aspect-specific details; and Agentic is occasionally competitive but unstable, which is a concern for deployments that need predictable behavior rather than isolated wins.

5.3 Effect of Document Length

A natural objection to single-pass reading is that it should degrade as documents grow longer, and that decomposition should become more useful precisely where the full document is hardest to attend over. Figure 3 tests this by relating ROUGE-LSum to source length; other metrics follow similar patterns (Appendix H.1). On PMC-SA and FacetSum the aggregate ranking is stable across the length range these benchmarks cover: ZS is highest in every bin, and its margin over the decomposed pipelines does not narrow as documents grow longer. RAG is the only pipeline that degrades visi-

bly with length in two datasets. On ACLSum, the smallest set and the one with the widest per-bin errors, RAG is often competitive with or better than ZS on shorter documents.

5.4 Implications for Deployment

The results do not imply that decomposed pipelines are never useful. They show that decomposition should not be treated as a default way to improve quality when long-context local models can read the full document. Additional stages can still be useful when the application needs retrieved evidence, extracted rationales, verification traces, or human-review checkpoints. Thus, pipeline choice should depend on the system requirement: use direct prompting as the quality baseline, and add decomposition when its intermediate outputs provide inspection or control.

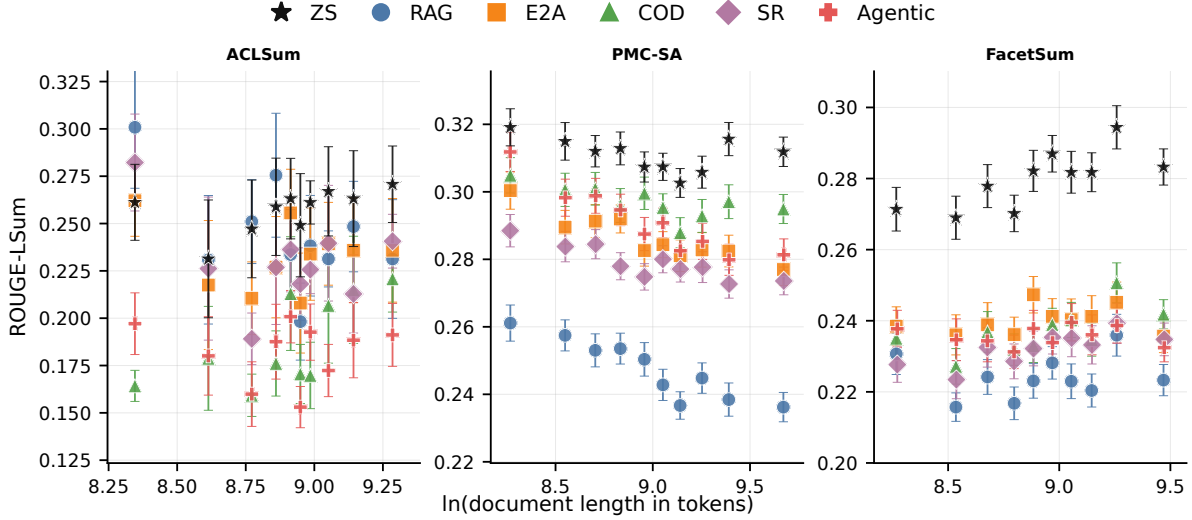


Figure 3: ROUGE-LSum as a function of source document length for Qwen3.5-9B. Aspect scores are averaged within each document before binning, so every document contributes equally regardless of how many aspects it has. Documents are grouped into 10 equal-frequency bins per dataset; each marker is placed at the mean log-length of its bin and shows the mean score over the documents in that bin, with error bars giving the standard error of that mean. Length is measured in Qwen3.5-9B tokenizer tokens.

System	R-L	BS	Summary excerpt and interpretation
Reference	—	—	Requires a dictionary and raw text; context knowledge can be updated as language changes.
ZS	.2667	.8933	Captures dictionary, unannotated text, dependency relations, and semantic coherence.
Agentic	.1176	.8771	Captures an unannotated-text knowledge base but drops the dictionary and update mechanism.
CoD	.0889	.8533	Shifts toward implementation details and misses broader resource assumptions.
E2A	.0930	.8836	Becomes too generic, omitting dictionary and update details.
RAG	.1176	.8621	Captures the motivation rather than the concrete procedure.

Table 4: Qualitative ACLSum case study for the *Approach* aspect. R-L denotes ROUGE-L and BS denotes BERTScore.

5.5 Qualitative Analysis

To illustrate the failure modes, we use one representative ACLSum instance from a paper on unsupervised word sense disambiguation (aspect: *Approach*; Table 4). The target summary should explain how the method works. The method avoids manually annotated resources by using only a dictionary and raw text. From this raw text, it builds context knowledge that can be updated as language changes. The case was selected because ZS is close to its mean ROUGE-L performance, while all decomposed pipelines are below their own means, showcasing their failure modes.

The key reference information is the resource-

light and updateable nature of the approach. ZS captures most of this mechanism. The decomposed methods are plausible but less complete: CoD shifts to implementation details, E2A becomes too generic, RAG summarizes the motivation rather than the procedure, and Agentic omits some key resource assumptions. Full outputs are provided in Appendix I. This case mirrors the aggregate results: additional orchestration can produce fluent but incomplete summaries.

6 Conclusion

We compared six local, training-free pipelines for long-context aspect-based scientific summarization across three datasets and two primary open models under matched decoding and output-length limits, supplemented by a 2B scale analysis. Explicit decomposition does not reliably improve summary quality in this setting. Zero-shot full-document prompting is the strongest quality default across lexical, semantic, and claim-based metrics and provides the most favorable quality–token trade-off among full-document pipelines. RAG is a lower-cost alternative, using about one seventh as many tokens as zero-shot, but its quality is reduced and less consistent across datasets. More complex pipelines may still be useful when intermediate outputs, as evidence traces or human-review checkpoints, justify their added cost and complexity.

7 Limitations

Our study is limited to local, training-free inference pipelines. This design matches deployment settings where labeled data, external APIs, or model training are costly or unavailable. It also means that our conclusions may not hold for proprietary long-context models, fine-tuned summarizers, or systems with task-specific retrievers and extractors. Such systems may benefit more from decomposition than the pipelines evaluated here.

The model scope is also limited. We evaluate two primary open long-context models, with Qwen3.5-9B as the main model and OLMo-3-7B-Instruct as a robustness check, plus a supplementary Qwen3.5-2B scale analysis. These models cover different open-model families and sizes, but they do not represent all model sizes or context lengths. Variation in model family and size does not isolate the effect of instruction tuning, so robustness across instruction-tuning recipes remains untested. Larger models, stronger small models, or models trained specifically for scientific text may change the relative ranking of the pipelines.

The dataset scope is restricted to English scientific documents. ACLSum covers NLP papers, FacetSum covers Emerald journal articles, and PMC-SA covers biomedical articles. These datasets differ in domain and aspect coverage, but they are still scholarly and reference-based, with predefined, relatively coarse aspects. Whether the findings transfer to user-defined, fine-grained, or cross-sectional aspects remains an open question. The results may also not transfer to multilingual settings or non-scientific documents. For FacetSum and PMC-SA, we evaluate sampled test subsets for long-context inference, so small effects may depend on the sample.

The evaluated pipelines are representative but not exhaustive. We compare direct prompting, retrieval, extraction, densification, self-refinement, and agentic orchestration under matched decoding and length constraints. Other implementations could use different prompts, retrieval models, chunking strategies, stopping rules, or verification criteria. We therefore interpret the results as evidence about common decomposition patterns under a controlled setup, not as a proof that all possible decomposed systems are inferior.

Our evaluation is automatic. ROUGE and BERTScore measure overlap with reference sum-

maries. The claim-based metric measures entailment overlap between generated and reference claims. It does not directly measure source-grounded faithfulness. Our source-grounded G-Eval analysis is a supplementary LLM-based proxy and may reflect evaluator bias; it complements but does not replace human evaluation. Errors in claim extraction or entailment classification can also affect scores. The qualitative case study illustrates failure modes but is not a systematic human evaluation.

Finally, mean input-plus-output token usage does not capture latency, memory pressure, batching, engineering complexity, or human review time. Future work should evaluate these costs directly, particularly when analysts use intermediate outputs.

References

- Griffin Adams, Alex Fabbri, Faisal Ladhak, Eric Lehman, and Noémie Elhadad. 2023. [From sparse to dense: GPT-4 summarization with chain of density prompting](#). In *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 68–74, Singapore. Association for Computational Linguistics.
- Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, and 1 others. 2026. Why do multi-agent llm systems fail? *Advances in Neural Information Processing Systems*, 38.
- Bohao Chu, Meijie Li, Sameh Frihat, Chengyu Gu, Georg Lodde, Elisabeth Livingstone, and Norbert Fuhr. 2025. [TracSum: A new benchmark for aspect-based summarization with sentence-level traceability in medical domain](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 844–864, Suzhou, China. Association for Computational Linguistics.
- Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. [A discourse-aware attention model for abstractive summarization of long documents](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 615–621. Association for Computational Linguistics.
- Bradley Efron and R. J. Tibshirani. 1994. [An Introduction to the Bootstrap](#). Chapman and Hall/CRC.
- Yichao Feng, Shuai Zhao, Yueqiu Li, Luwei Xiao, Xiaobao Wu, and Anh Tuan Luu. 2025. [Aspect-based summarization with self-aspect retrieval enhanced](#)

- generation. In *International Joint Conference on Neural Networks, IJCNN 2025, Rome, Italy, June 30 - July 5, 2025*, pages 1–8. IEEE.
- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. [Gemma 3 Technical Report](#). Preprint, arXiv:2503.19786.
- Alexios Gidiotis and Grigorios Tsoumakas. 2019. [Structured summarization of academic publications](#). In *Machine Learning and Knowledge Discovery in Databases - International Workshops of ECML PKDD 2019, Würzburg, Germany, September 16-20, 2019, Proceedings, Part II*, volume 1168 of *Communications in Computer and Information Science*, pages 636–645. Springer.
- Or Honovich, Roei Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. [TRUE: Re-evaluating factual consistency evaluation](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.
- Yubin Kim, Ken Gu, Chanwoo Park, Chunjong Park, Samuel Schmidgall, A Ali Heydari, Yao Yan, Zhihan Zhang, Yuchen Zhuang, Yun Liu, and 1 others. 2025. Towards a science of scaling agent systems. *arXiv preprint arXiv:2512.08296*.
- Philipp Koehn. 2004. [Statistical significance tests for machine translation evaluation](#). In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. [G-eval: NLG evaluation using gpt-4 with better human alignment](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.
- Rui Meng, Khushboo Thaker, Lei Zhang, Yue Dong, Xingdi Yuan, Tong Wang, and Daqing He. 2021. [Bringing structure into summaries: a faceted summarization dataset for long scientific documents](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*, pages 1080–1089. Association for Computational Linguistics.
- Sewon Min, Kalpesh Krishna, Xinxu Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. [FActScore: Fine-grained atomic evaluation of factual precision in long form text generation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.
- Hanseok Oh, Seojin Nam, and Yongjun Zhu. 2023. [Structured abstract summarization of scientific articles: Summarization using full-text section information](#). *J. Assoc. Inf. Sci. Technol.*, 74(2):234–248.
- Qwen Team. 2026. [Qwen3.5-omni technical report](#). Preprint, arXiv:2604.15804.
- Amir Soleimani, Vassilina Nikoulina, Benoît Favre, and Salah Ait-Mokhtar. 2022. [Zero-shot aspect-based scientific document summarization using self-supervised pre-training](#). In *Proceedings of the 21st Workshop on Biomedical Language Processing, BioNLP@ACL 2022, Dublin, Ireland, May 26, 2022*, pages 49–62. Association for Computational Linguistics.
- Sotaro Takeshita, Tommaso Green, Ines Reinig, Kai Eckert, and Simone Paolo Ponzetto. 2024. [Aclsum: A new dataset for aspect-based summarization of scientific publications](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 6660–6675. Association for Computational Linguistics.
- Manya Wadhwa, Xinyu Zhao, Junyi Jessy Li, and Greg Durrett. 2024. [Learning to refine with fine-grained natural language feedback](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12281–12308, Miami, Florida, USA. Association for Computational Linguistics.
- Tairan Wang, Xiuying Chen, Qingqing Zhu, Taicheng Guo, Shen Gao, Zhiyong Lu, Xin Gao, and Xiangliang Zhang. 2025a. [New paradigm for evaluating scholar summaries: A facet-aware metric and a meta-evaluation benchmark](#). *ACM Trans. Inf. Syst.*, 43(4).
- Weixuan Wang, Minghao Wu, Barry Haddow, and Alexandra Birch. 2025b. [Learning to summarize by learning to quiz: Adversarial agentic collaboration for long document summarization](#). *CoRR*, abs/2509.20900.
- Tiezheng Yu, Dan Su, Wenliang Dai, and Pascale Fung. 2020. [Dimsum @LaySumm 20](#). In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 303–309, Online. Association for Computational Linguistics.
- Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023. [SummIt: Iterative text summarization via ChatGPT](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10644–10657, Singapore. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with bert](#). *Preprint*, arXiv:1904.09675.

Yang Zhong and Diane J. Litman. 2025. [From information to insight: Leveraging llms for open aspect-based educational summarization](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2025, Vienna, Austria, July 27 - August 1, 2025, pages 1914–1947. Association for Computational Linguistics.

A Data Construction and Preprocessing

A.1 ACLSum

ACLSum (Takeshita et al., 2024) provides expert-written one-sentence summaries on the aspects Challenge, Approach, and Outcome. The original release contains Abstract, Introduction, and Conclusion text, but not full-paper text. To support long-context evaluation, we retrieve the corresponding paper PDFs and parse their text with PaddleOCR-VL⁷. We preserve the original aspect labels and reference summaries. We use the full test split without additional sampling.

A.2 FacetSum

For FacetSum (Meng et al., 2021), we keep the original train, development, and test splits. We use the full-text setup and retain all examples. Our experiments use a sample of 1,000 documents from the test split and keep all four aspect summaries for each sampled document.

A.3 PMC-SA

A.3.1 Source and Acquisition

We construct the PMC-SA dataset from the PubMed Central Open Access Subset. We follow the section-aligned data construction approach introduced by Oh et al. (Oh et al., 2023), processing the PMC bulk XML archives directly to ensure scalability and reproducibility. From each article, we extract pairs of *structured abstracts* and corresponding *full-text body sections*, retaining only documents whose abstracts are explicitly segmented into labeled sections.

A.3.2 Section Normalization

Scientific articles exhibit substantial variation in section header naming (e.g., “*Study Design*”, “*Materials and Methods*”). To obtain a consistent aspect space, we normalize all abstract and full-text

headers into five canonical sections: **Background**, **Objectives**, **Methods**, **Results**, and **Conclusions**.

Section headers are normalized using the National Library of Medicine (NLM) Category Mapping, which resolves over 3,000 known header variants into these canonical categories. Within a document, multiple sections mapped to the same canonical label are concatenated to form a single section representation.

A.3.3 Quality Filtering and Alignment

To ensure high-quality aspect–summary pairs, we apply strict filtering criteria:

- **Structural validity:** abstracts must contain at least three canonical sections and include a terminal section (**Results** or **Conclusions**);
- **Length constraints:** abstracts are restricted to 10–2,500 words and full texts to 150–80,000 words;
- **Compression ratio:** documents are discarded if the full-text-to-abstract word ratio lies outside $[5, 500]$, removing trivial extractive cases and extreme compression outliers.

We enforce a **strict one-to-one section alignment policy**. For each normalized abstract section, we require the existence of a corresponding full-text section with the same canonical label. Documents are discarded if any abstract section lacks a matched full-text counterpart.

This strict alignment ensures that each training instance pairs an aspect-specific abstract section with explicitly grounded source evidence.

A.4 Sampling Strategy

For the larger FacetSum and PMC-SA Datasets, we sample 1,000 documents per dataset from their respective test splits. To ensure the evaluation remains representative, we apply stratified sampling across the target aspects. This approach follows established LLM benchmarking precedents (Wang et al., 2025a) by balancing statistical robustness with the computational sustainability required for long-context inference.

A.5 Evidence Localization

Alignment protocol. To characterize where aspect-relevant content occurs in the source, we reproduce the structural alignment of Meng et al. (2021) (their Section 2.2). Let $S = (s_1, \dots, s_{|S|})$

⁷<https://github.com/PaddlePaddle/PaddleOCR>

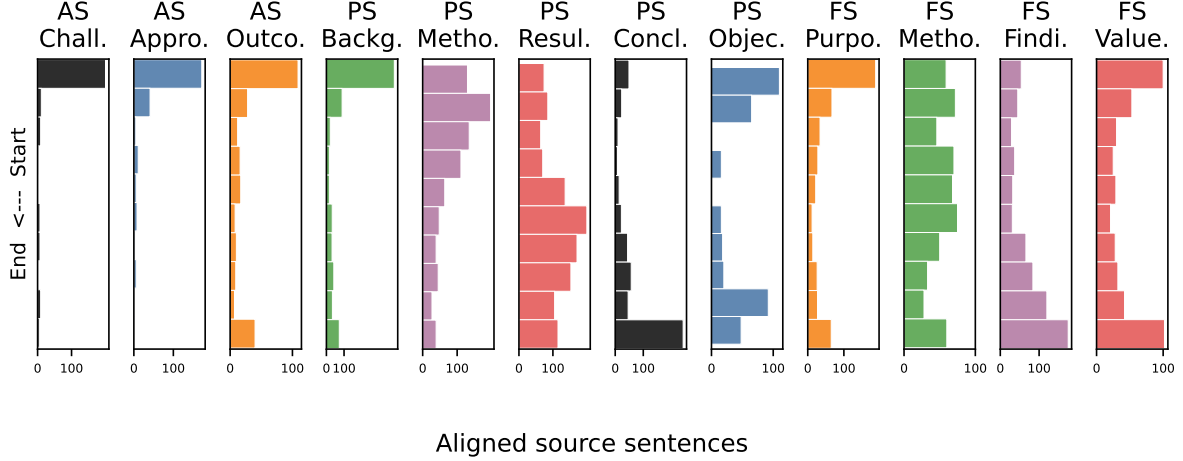


Figure 4: Oracle evidence localization for the twelve dataset–aspect combinations. AS, PS, and FS denote ACLSum, PMC-SA, and FacetSum. Each panel is a histogram of the normalized positions of source sentences aligned to the reference aspect summary under Equation 1, over ten equal-width position bins running from the start of the document (top) to the end (bottom). Bars give the number of aligned source sentences; note that panel totals differ, because the number of aligned sentences per instance depends on the reference summary length, and that ACLSum references are single sentences so each ACLSum instance contributes exactly one position.

denote the sentences of the full source document and $A = (a_1, \dots, a_{|A|})$ the sentences of the corresponding reference aspect summary. Each summary sentence is aligned to its single best-matching source sentence under ROUGE-1 F_1 :

$$S_A = \left\{ \arg \max_{s_i \in S} \text{ROUGE-1}(s_i, a_j) : a_j \in A \right\}. \quad (1)$$

S_A is a set, so several summary sentences selecting the same source sentence contribute a single element. For each aligned index i we record the normalized position $(i + 0.5)/|S|$, where 0 is the start and 1 the end of the document. Pooling these positions over documents gives the oracle evidence distribution of an aspect: it describes where a system would have to read in order to recover the reference content, and it is defined without reference to any model output.

Configuration. For each of the twelve dataset–aspect combinations, we draw 256 instances from the corresponding evaluation split with a fixed seed; if the requested sample is larger than the population, we use the whole population. Figure 4 bins the pooled positions into ten equal-width intervals per panel.

Results. Figure 4 shows that the twelve dataset–aspect combinations fall into clearly different localization regimes. ACLSum Challenge and Approach, PMC-SA Background, and FacetSum Pur-

pose are strongly front-loaded, with most aligned sentences in the first tenth of the document. PMC-SA Conclusions is the mirror image, concentrated at the very end. PMC-SA Objectives and FacetSum Value are bimodal, drawing on both the opening and the closing of the paper. PMC-SA Methods and Results and FacetSum Methods and Findings are distributed far more broadly, with substantial mass across the interior of the document.

Two consequences follow for the pipelines compared in this work. First, no single positional heuristic serves all aspects: a retrieval or extraction stage tuned to recover opening material would systematically miss PMC-SA Conclusions, and one tuned to section boundaries would miss the aspects whose evidence is spread across the body. Second, the aspects with the broadest distributions are precisely those for which a fixed retrieval budget is least likely to cover the reference content, which is consistent with the sensitivity of RAG to retrieval depth reported in Appendix D.

B Detailed Pipeline Implementation

This appendix summarizes the concrete implementation of the six compared inference pipelines.

B.1 Zero-Shot (ZS)

ZS uses a single prompt over the normalized document. The prompt consists of a fixed system instruction and a task template with slots for the as-

pect, source type, length instruction, and source text:

System prompt

You are an expert scientific editor, you are given a scientific text and tasked to write a summary focused on a specific aspect.

Prompt template

```
# CONTEXT
You will be provided with a scientific
text and a specific aspect to focus on.

# TASK
Given the scientific text below and
a focused aspect, which is {aspect},
write a short summary using your own
words.
The summary needs to be a coherent
paragraph and should include the major
points.
Write in free form, avoid bullet points
or numbered lists.
The summary should focus on the provided
aspect only, contain only information
about the aspect, and avoid adding
irrelevant sentences or your own
opinions and suggestions.

# INPUT
The      source      text      is      a
{source_type_pretty}.      This      is
{source_type_desc}.
FOCUSED ASPECT: {aspect}
{summary_line}
SOURCE TEXT:
{source_text}

# OUTPUT
```

B.2 Extract-to-Abstract (E2A)

E2A is implemented as a two-stage structured pipeline. An extractor first returns a JSON evidence set of verbatim spans relevant to the target aspect, and a writer then generates the final summary using only this evidence and the dataset-specific length constraint. Both stages use schema-constrained decoding. The pipeline uses separate system and user prompts for the extractor and writer:

Extractor system prompt

You are an Information Extractor specializing in scientific literature. You excel in identifying and extracting verbatim evidence that is necessary for summarizing specific aspects of research papers. You value precision; You never alter quotes or add interpretation. You produce structured JSON OUTPUT only.

Extractor prompt template

```
### CONTEXT
You are given a scientific document and
a specific aspect.

### TASK
Your task is to extract verbatim
evidence from the document that is
relevant to the specified aspect and
would be useful for writing a summary
about that aspect.

### INPUTS
Target Aspect:
{{ aspect_name }}

Source Document:
{{ document_text }}

### INSTRUCTIONS
1. Find spans of text that contain
claims or data relevant to the aspect.
2. Extract verbatim text spans, do not
paraphrase or summarize.
3. Each extracted span should be
self-contained and informative on its
own.
4. Avoid extracting very short or very
long spans; aim for concise, meaningful
excerpts that capture key points.
5. The spans should be sorted in the
order they appear in the document to
preserve context.
6. Return at most {{ max_evidence_items
}} evidence items.
7. Keep each span short (prefer 1-2
sentences, max about {{ max_span_chars
}} characters).
8. Avoid references, equations, and
low-signal boilerplate.
9. Output valid JSON only.
```

```
### OUTPUT FORMAT
Return a valid JSON object matching the
ExtractorOutput schema:
{
  "evidence_set": [
    "Exact quote 1...",
    "Exact quote 2..."
  ]
}
```

Writer system prompt

You are an expert scientific assistant and editor. Your expertise lies in synthesizing information from multiple pieces of evidence to create concise, accurate summaries focused on specific aspects of scientific research. You use the same wording and tone used in the source text, which is scientific and formal. Write concise, faithful aspect-focused summaries.

Writer prompt template

```

### CONTEXT
You are given a set of verbatim
evidence extracted from a scientific
document, all related to the aspect "{{
aspect_name }}"

```

```

### TASK
Write a short aspect-focused summary
with your own words for the following
aspect: "{{ aspect_name }}"
Use ONLY the provided evidence set.
The summary needs to be a coherent
paragraph and should include the major
points.

```

```

### EVIDENCE SET (ORDERED)
{{ evidence_set }}

```

```

### SUMMARY LENGTH
{{ summary_length }}

```

```

### INSTRUCTIONS
1. The summary should focus on the
provided aspect only.
2. Use ONLY the evidence set above.
3. Write in free form, avoid bullet
points or numbered lists.
4. Avoid adding irrelevant sentences or
your own opinions and suggestions.
5. Respect the summary length
instruction.

```

```

### OUTPUT FORMAT
Return a valid JSON object matching the
WriterOutput schema:
{
  "summary_text": "Your single-paragraph
summary..."
}

```

B.3 Retrieval-Augmented Generation (RAG)

The RAG pipeline follows the implementation in (Feng et al., 2025). Each document is split into sentence-aware chunks of about 256 words. We encode chunks with the dense embedding model NovaSearch/stella_en_1.5B_v5⁸, retrieve the top 10 aspect-relevant chunks, reconstruct them in their original document order, and pass the resulting retrieved context to the writer model, which generates the final summary.

B.4 Chain of Density (CoD)

CoD first generates an initial aspect summary from the full document and then applies two densification steps, following (Adams et al., 2023). At each step, the model identifies missing aspect-relevant entities or findings and rewrites the current summary to make it denser while preserving the dataset-specific sentence and word limits. If a revised

⁸https://huggingface.co/NovaSearch/stella_en_1.5B_v5

summary violates these limits or is empty, it is discarded and the previous valid summary is kept. The pipeline uses separate prompts for the initial draft and each densification step:

Initial system prompt

You are an expert scientific editor, you are given a scientific text and tasked to write a summary focused on a specific aspect. You use the same wording and tone used in the source text, which is scientific and formal. Write concise, faithful aspect-focused summaries.

Initial prompt template

```

# CONTEXT
You will be provided with a scientific
text and a specific aspect to focus on.

```

```

# TASK
Given the scientific text below and
a focused aspect, which is {aspect},
write a short summary using your own
words.
The summary needs to be a coherent
paragraph and should include the major
points.
Write in free form, avoid bullet points
or numbered lists.
The summary should focus on the provided
aspect only, contain only information
about the aspect, and avoid adding
irrelevant sentences or your own
opinions and suggestions.
This is the initial draft, so keep it
relatively high-level and leave room
for adding missing details in later
refinement passes.

```

```

# INPUT
The source text is the full text of a
scientific paper.
FOCUSED ASPECT: {aspect}
SUMMARY TARGET LENGTH: Maximum
{max_sentences} sentences and
{max_words} words.
SOURCE TEXT:
{source_text}

```

```

# OUTPUT
Return only the initial summary
paragraph text.

```

Iteration system prompt

You are an expert scientific editor, you are given a scientific text and tasked to revise a summary focused on a specific aspect. You use the same wording and tone used in the source text, which is scientific and formal. Write concise, faithful aspect-focused summaries.

Iteration prompt template

```
# CONTEXT
You will be provided with a scientific
text, a focused aspect, and a draft
summary for that aspect.

# TASK
Revise the draft summary for aspect
"{aspect}".
Compare the draft summary against
the source text and identify missing
high-value information.
If important information is missing,
add it.
Keep the summary focused on the same
aspect and faithful to the source.
Condense and remove redundancy if needed
to keep the summary concise.

# MISSING ENTITY DEFINITION
A missing entity should be:
- Relevant: central to the focused
aspect
- Specific: concrete name, term, model,
dataset, method, metric, number, or
finding
- Novel: not already present in previous
summary
- Faithful: explicitly supported by
source text

# REVISION METHOD
- Check for missing high-value details:
named entities, methods/models,
datasets, key numbers, and main
findings.
- Integrate only the most important
missing details.
- Rewrite the paragraph for better
clarity, flow, and scientific style.
- If the draft is already complete,
still rewrite for clearer wording and
smoother style.
- Avoid copying the draft verbatim.

# STRICT LENGTH RULES
- Your revised summary must satisfy BOTH
limits:
- at most {max_words} words
- at most {max_sentences} sentences
- Keep approximately the same length
as the previous summary ({prev_words}
words, {prev_sentences} sentences).

# STYLE RULES
- Write in the same tone and style
as sentences in scientific abstracts,
which is concise, logical to follow,
and formal.
- Single coherent paragraph.
- No bullet points and no numbered
lists.
- No unsupported claims.

# INPUT
SOURCE TYPE: full text of a scientific
paper
```

```
FOCUSED ASPECT: {aspect}
ITERATION:      {iteration_idx}      /
{total_iterations}
SUMMARY TARGET LENGTH: Maximum
{max_sentences} sentences and
{max_words} words.
SOURCE TEXT:
{source_text}

PREVIOUS SUMMARY:
{draft_summary}

# OUTPUT
Return exactly this format:
MISSING_ENTITIES: entity1; entity2;
entity3
DENSER_SUMMARY: <single paragraph>
```

B.5 Self-Refine (SR)

SR is implemented as a fixed three-step procedure. The model first generates an initial aspect summary from the full document, then produces 2–3 concrete suggestions for improving that summary in the same chat history, and finally generates a revised summary in a fresh refinement step conditioned on the source text, the initial summary, and the suggestions. The same model is used for all three stages, and the procedure runs for exactly one refinement round. The pipeline uses the following prompts:

Generate system prompt

You are an abstractive summarizer. Your task is to summarize a scientific text focusing on one specific aspect.

Generate prompt template

Write a short summary about the aspect "{aspect}" using only the source text below.

```
Target length:
{summary_target_instruction}.
```

```
Reflections:
{source_text}
```

Summary:

Suggestion prompt template

Give 2–3 concrete suggestions to improve the generated summary so that it becomes more concise and more focused on the aspect "{aspect}".

Base the suggestions on the original source text and the generated summary. Do not give generic advice.

Refine system prompt

You are an abstractive summarizer. Your task is to summarize a scientific text focusing on one specific aspect.

Refine prompt template

Improve the short summary below using the suggestions.

The revised version must stay focused on the aspect "{aspect}" based on the source text below.

Target length:
{summary_target_instruction}.

Reflections:
{source_text}

Original summary:
{initial_summary}

Suggestions for improvement:
{suggestions}

Refined summary:

B.6 Agentic Multi-Role Pipeline (Agentic)

The agentic pipeline is implemented with four LLM modules (planner, extractor, writer, verifier) that use different prompt templates and role-specific output schemas. The router is deterministic.

Role sequence. The pipeline follows planner → extractor → writer → verifier → router.

- **Planner (LLM):** Input: aspect name, source document, target summary length. Output: structured plan with aspect definition, extraction cues, and acceptance criteria. Planner inference runs with thinking enabled.
- **Extractor (LLM):** Input: aspect name, source document, planner output, and optional router correction instruction. Output: ordered verbatim evidence spans.
- **Writer (LLM):** Input: evidence spans, aspect name, target summary length, and optional router correction instruction. Output: one summary paragraph.
- **Verifier (LLM):** Input: writer summary, evidence spans, and planner output. Output: one structured decision in {accept, revise_summary, revise_evidence} and a correction instruction. Verifier inference runs with thinking enabled.

- **Router (deterministic):** Input: verifier decision and correction instruction. Output: next action (stop, send correction to writer, or send correction to extractor).

Deterministic routing and bounded revision.

Routing is deterministic. If the verifier returns accept, the current iteration ends and the sample is finalized. If it returns revise_summary, the router sends the correction instruction back to the writer with the same evidence. If it returns revise_evidence, the router sends the correction instruction back to the extractor to update evidence before writing again. An iteration ends at either acceptance or when the retry budget is exhausted; the implementation uses at most two retries.

Full role-specific prompt templates are provided in the codebase.

Development variants. Before fixing the reported configuration, we piloted an extractor–writer pipeline, a planner–extractor–writer pipeline without verification, and the full planner–extractor–writer–verifier loop. We also varied role compositions and iterated on prompts through qualitative inspection. None of these variants outperformed ZS, and the full loop reported in the main experiments was the strongest Agentic configuration tested. The recurring failure mode was planner error propagation: vague or incorrect aspect definitions and acceptance criteria guided the extractor toward the wrong evidence, which the writer then summarized faithfully. Because the verifier checks the summary against the selected evidence and plan rather than re-deriving the plan, it could not reliably correct this upstream error. These development runs were not retained as a controlled benchmark, so we report no numerical ablation and draw no component-level causal conclusion from them.

C Statistical Significance of Zero-Shot vs. Pipeline Differences

Test strategy. To assess whether the aggregate quality differences between direct zero-shot prompting (ZS) and the decomposed pipelines reflect more than sampling noise, we run paired bootstrap tests (Efron and Tibshirani, 1994; Koehn, 2004). For each dataset–pipeline pair we compute, for every evaluated aspect–summary instance i , the score difference $d_i = s_i^{\text{ZS}} - s_i^{\text{pipeline}}$ on the same input, so that document- and aspect-level difficulty is controlled for by construction.

Document-level clustering. The evaluated instances are not mutually independent. Each source document contributes one instance per aspect—three for ACLSum, four for FacetSum, and up to five for PMC-SA—and these instances are derived from the same source text, so a document that is hard to summarize for one aspect tends to be hard for its remaining aspects as well. Resampling instances independently would treat the n aspect-summary pairs as n independent draws and thereby understate the sampling variance of Δ . We therefore resample at the document level: each bootstrap replicate draws N documents with replacement from the N documents of the evaluation set ($N = 100$ for ACLSum, $N = 1,000$ for FacetSum and PMC-SA) and retains all aspect instances of every drawn document, so that within-document correlation is preserved in the resampling distribution.

Procedure. We use $B = 10,000$ replicates, compute the mean difference $\bar{d}^{(b)}$ over the pooled instances of replicate b , and report the observed mean difference $\Delta = \bar{d}$ together with the percentile 95% confidence interval given by the 2.5th and 97.5th percentiles of $\{\bar{d}^{(b)}\}_{b=1}^B$. A difference is marked significant when the interval excludes zero, corresponding to a two-sided test at $\alpha = 0.05$. The choice of resampling unit affects only the width of the interval and not Δ itself. Resampling uses a fixed random seed (13) for reproducibility. The procedure is applied independently to each of the five automatic metrics (ROUGE-1/2/L/Lsum F_1 and BERTScore F_1), each of the three datasets (ACLSum, FacetSum, PMC-SA), and each of the five pipelines, yielding the $5 \times 3 \times 5 = 75$ comparisons in Table 5. We deliberately report effect sizes with confidence intervals rather than p -values alone, since with large evaluation sets even negligible differences can reach significance; the intervals make the magnitude of each gap directly visible.

Results. Table 5 reports $\Delta = \text{score}(\text{ZS}) - \text{score}(\text{pipeline})$ for the main model under the document-level clustered bootstrap. All 75 observed differences are positive, i.e., zero-shot prompting matches or exceeds every pipeline on every metric and dataset, and 71 of 75 remain statistically significant at the 95% level once within-document correlation is accounted for. The only comparisons whose intervals cross zero are ZS vs. RAG on ACLSum for ROUGE-2, ROUGE-L, ROUGE-Lsum, and BERTScore. ACLSum

provides the fewest resampling units of the three evaluation sets (100 documents), and RAG is the pipeline closest to ZS in aggregate quality, so the remaining gap there cannot be distinguished from noise. Clustering is a conservative rather than a decisive correction in this setting: relative to instance-level resampling it widens the intervals by at most 5% on average per dataset and by at most 15% for any single comparison, and it does not change which comparisons are significant. This is consistent with aspect summaries of the same paper drawing on largely disjoint source content, which limits how strongly their per-instance score differences co-vary. None of the pipelines significantly outperforms zero-shot prompting in any dataset-metric combination, which supports the central claim that task decomposition provides no reliable quality advantage over direct full-document prompting.

Dataset	Pipeline	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	BERTScore
ACLSum	RAG	2.02 [0.21, 3.95]	0.90 [†] [-0.48, 2.34]	1.17 [†] [-0.32, 2.72]	1.17 [†] [-0.34, 2.70]	0.09 [†] [-0.18, 0.37]
	E2A	4.10 [2.77, 5.44]	3.16 [2.02, 4.34]	2.49 [1.29, 3.73]	2.49 [1.26, 3.72]	0.33 [0.12, 0.53]
	COD	10.36 [8.89, 11.80]	5.61 [4.54, 6.70]	7.13 [5.95, 8.30]	7.13 [5.97, 8.31]	1.99 [1.75, 2.23]
	Agentic	9.52 [8.08, 11.01]	5.92 [4.81, 7.05]	7.30 [6.03, 8.62]	7.29 [6.02, 8.62]	1.83 [1.61, 2.06]
	SR	5.42 [3.96, 6.93]	3.22 [1.98, 4.50]	3.04 [1.80, 4.36]	3.04 [1.80, 4.34]	0.45 [0.23, 0.68]
FacetSum	RAG	6.49 [6.12, 6.86]	4.17 [3.90, 4.44]	4.64 [4.34, 4.93]	5.52 [5.18, 5.86]	1.09 [1.03, 1.16]
	E2A	4.54 [4.25, 4.82]	2.35 [2.14, 2.55]	2.83 [2.60, 3.06]	3.95 [3.69, 4.22]	0.46 [0.41, 0.51]
	COD	5.04 [4.78, 5.30]	2.75 [2.56, 2.95]	3.47 [3.27, 3.68]	4.25 [4.02, 4.49]	1.22 [1.17, 1.27]
	Agentic	5.01 [4.73, 5.31]	2.54 [2.32, 2.76]	3.00 [2.77, 3.24]	4.37 [4.10, 4.64]	0.85 [0.79, 0.90]
	SR	6.09 [5.81, 6.37]	3.31 [3.10, 3.53]	3.47 [3.24, 3.71]	4.73 [4.47, 4.99]	0.44 [0.39, 0.48]
PMC-SA	RAG	7.58 [7.28, 7.89]	4.65 [4.43, 4.87]	4.89 [4.66, 5.13]	6.23 [5.96, 6.51]	1.65 [1.58, 1.71]
	E2A	2.93 [2.70, 3.17]	1.96 [1.78, 2.14]	1.53 [1.33, 1.72]	2.40 [2.18, 2.63]	0.23 [0.18, 0.28]
	COD	1.61 [1.42, 1.81]	0.83 [0.69, 0.97]	0.91 [0.75, 1.06]	1.36 [1.18, 1.54]	0.33 [0.29, 0.38]
	Agentic	2.34 [2.08, 2.60]	1.17 [0.98, 1.36]	0.52 [0.32, 0.72]	1.80 [1.56, 2.03]	0.19 [0.14, 0.24]
	SR	4.17 [3.94, 4.41]	2.80 [2.62, 2.97]	2.18 [1.99, 2.37]	3.12 [2.90, 3.34]	0.31 [0.26, 0.35]

Table 5: Document-level clustered bootstrap significance tests for $\Delta = \text{score}(\text{ZS}) - \text{score}(\text{pipeline})$, in points ($\times 100$), with percentile 95% confidence intervals in brackets ($B = 10,000$ resamples over source documents, retaining all aspect instances of each drawn document). Positive values favor zero-shot. Differences whose interval excludes zero are significant at $\alpha = 0.05$; the four non-significant differences are marked with [†]. Pipeline abbreviations follow Section 3.

D RAG and E2A Sensitivity Analysis

We test whether the main results depend on the evidence budgets used by RAG and E2A. For RAG, we increase retrieval depth from 10 to 20 chunks, chunk size from 256 to 512 words, and both jointly. For E2A, we increase the maximum number of evidence items from 50 to 75 and the extractor generation budget from 4,000 to 6,000 tokens. All other components, prompts, and decoding settings remain fixed. We evaluate Qwen3.5-9B on a fixed subset of 100 documents per dataset, including all available aspects, and pool the paired examples across datasets. Confidence intervals use paired bootstrap resampling with $B = 10,000$.

All three larger-context RAG variants significantly improve over the original configuration on the displayed metrics, with the joint 512-word, top-20 setting producing the largest overall gain. Nevertheless, ZS remains significantly ahead of every variant; under the strongest RAG setting, the Claim-F1 gap narrows from 0.075 to 0.054. In contrast, none of the differences between the original and expanded-budget E2A configurations is significant, including Claim F1 ($\Delta_{\text{orig-expanded}} = -0.002$, 95% CI $[-0.011, 0.008]$). These results indicate that retrieval budget is consequential but does not account for the full RAG-ZS gap, whereas the tested extraction cap does not explain E2A’s result. A comparable oracle-evidence experiment across all datasets is not possible because only ACLSum provides gold evidence spans; an ACLSum-only

oracle would not test whether the conclusion generalizes across the three benchmarks.

Pipeline	Configuration	Δ R-Lsum	Δ BS	Δ C-F1
RAG	256 words, top-10	.044 [.038, .051]	.010 [.009, .011]	.075 [.061, .088]
RAG	256 words, top-20	.032 [.026, .039]	.006 [.005, .008]	.059 [.045, .072]
RAG	512 words, top-10	.033 [.026, .039]	.007 [.006, .009]	.063 [.049, .077]
RAG	512 words, top-20	.032 [.025, .038]	.006 [.005, .007]	.054 [.041, .068]
E2A	50 items, 4k tokens	.031 [.025, .036]	.004 [.003, .005]	.039 [.027, .051]
E2A	75 items, 6k tokens	.030 [.024, .035]	.003 [.002, .004]	.038 [.025, .050]

Table 6: Sensitivity results pooled across the three dataset subsets. Values are $\Delta = \text{score}(\text{ZS}) - \text{score}(\text{configuration})$ with percentile 95% confidence intervals; positive values favor ZS. All displayed differences are significant at $\alpha = 0.05$. Bold marks the strongest RAG configuration. R-Lsum, BS, and C-F1 denote ROUGE-LSum, BERTScore, and Claim F1.

E Source-Grounded G-Eval

We supplement the reference-based metrics with G-Eval (Liu et al., 2023) using GPT-4.1-mini. For each generated summary, the evaluator receives the target aspect and full source document and assigns two scores from 1 to 5: aspect consistency measures whether the summary’s claims are supported by the source, and aspect relevance measures coverage of important aspect-specific information while penalizing irrelevant or redundant content. We evaluate all six pipelines with Qwen3.5-9B on a fixed sample of 100 documents per dataset, including all available aspects. Decoding uses temperature 0 and the final score is computed from the model’s output-token probabilities.

ZS ranks first on both dimensions for all three datasets, and no decomposition pipeline surpasses it. This provides complementary source-grounded evidence consistent with the reference-based results. However, G-Eval remains an automatic LLM-based proxy with possible evaluator bias and does not replace human judgments of factuality, relevance, or summary quality.

Pipeline	ACLSum		FacetSum		PMC-SA	
	Cons.	Rel.	Cons.	Rel.	Cons.	Rel.
ZS	4.097	4.151	4.365	4.497	4.465	4.662
RAG	3.967	3.910	3.964	3.986	3.768	3.772
E2A	4.004	4.016	4.155	4.245	4.310	4.479
CoD	4.000	3.988	4.276	4.369	4.449	4.625
SR	3.972	3.957	4.160	4.203	4.277	4.421
Agentic	3.975	3.970	4.205	4.262	4.274	4.443

Table 7: Source-grounded G-Eval scores using GPT-4.1-mini on 100 documents per dataset. Cons. and Rel. denote aspect consistency and aspect relevance on a 1–5 scale. Best results are bold.

F Model-Scale Analysis

We evaluate all six pipelines with Qwen3.5-2B to probe whether decomposition becomes more useful with a weaker backbone. Table 8 reports three representative quality metrics and structured-output failure rates. Missing outputs caused by parser failures are assigned zero, so the scores represent end-to-end deployed pipeline performance rather than quality conditional on successful execution.

Decomposition produces more local wins at 2B than at 9B: CoD narrowly leads on all three displayed FacetSum metrics and PMC-SA ROUGE-1, while Agentic leads ACLSum Claim F1. However, no method wins consistently across datasets, and the margins among successful non-agentic pipelines are small. Structured execution is also less reliable: Agentic fails on 23.3% of ACLSum, 30.9% of FacetSum, and 11.7% of PMC-SA cases; E2A failure rates are 0.0%, 0.8%, and 0.4%, respectively. Thus, weaker models can benefit locally from decomposition while being less able to execute its intermediate stages reliably.

Dataset	Pipeline	R-1	R-Lsum	C-F1	Failure
ACLSum	ZS	.298	.227	.064	0.0%
	RAG	.279	.202	.065	0.0%
	E2A	.290	.214	.066	0.0%
	CoD	.291	.216	.063	0.0%
	SR	.293	.211	.058	0.0%
	Agentic	.243	.178	.070	23.3%
FacetSum	ZS	.319	.262	.094	0.0%
	RAG	.276	.220	.050	0.0%
	E2A	.254	.199	.053	0.8%
	CoD	.321	.263	.098	0.0%
	SR	.319	.257	.089	0.0%
	Agentic	.212	.169	.064	30.9%
PMC-SA	ZS	.358	.306	.153	0.0%
	RAG	.301	.253	.062	0.0%
	E2A	.318	.267	.100	0.4%
	CoD	.358	.305	.151	0.0%
	SR	.350	.294	.138	0.0%
	Agentic	.316	.268	.126	11.7%

Table 8: Failure-adjusted Qwen3.5-2B results. Parser failures are scored as zero. R-1, R-Lsum, and C-F1 denote ROUGE-1, ROUGE-Lsum, and Claim F1; best results are bold. Failure is the percentage of aspect-summary cases without a valid final output.

G OLMo Results

On OLMo-3-7B-Instruct, a smaller model included for its fully documented training recipe and therefore end-to-end auditability, the same qualitative pattern holds, with the added noise expected from a weaker backbone. The most decomposed pipelines remain weakest: the agentic pipeline is again never best on any metric or dataset, and extract-then-abstract is likewise never best. Zero-shot is the single best method on PMC-SA across all ROUGE variants and on claim recall and F1, and remains in the leading cluster elsewhere. Where it is edged, chain-of-density on ACLSum and FacetSum and RAG on ACLSum, the gaps are small (within a few thousandths of BERTScore and ≤ 0.015 ROUGE). Chain-of-density is a zero-shot full-document draft followed by densification, rather than document fragmentation. RAG is the lone genuinely fragmenting method to lead, and only on ACLSum, where it drops on PMC-SA and FacetSum. Thus no pipeline that fragments the document or splits the task improves reliably over single-pass reading on either model; the only methods that occasionally surpass zero-shot are minor variations on it.

Tables 9 and 10 report the corresponding results.

H Per Aspect Results

This section provides the remaining per-aspect Qwen3.5-9B results for the ROUGE metrics. Tables 11, 12, 13, and 14 report the corresponding per-aspect results for ROUGE-1, ROUGE-2, ROUGE-

L, and ROUGE-Lsum. The BERTScore breakdown is reported in the main paper.

H.1 Effect of Document Length

Figures 5 and 6 complete the analysis presented in Section 5.3. Both show similar trends, with ZS outperforming all other pipelines in all ten bins for FacetSum and PMC-SA. On ACLSum, RAG is again better than or competitive with ZS for shorter documents but degrades over longer contexts. Figure 7 presents BERTScore as a function of source document length for the three datasets. Again, ZS is outperformed only by RAG in the short-document setting.

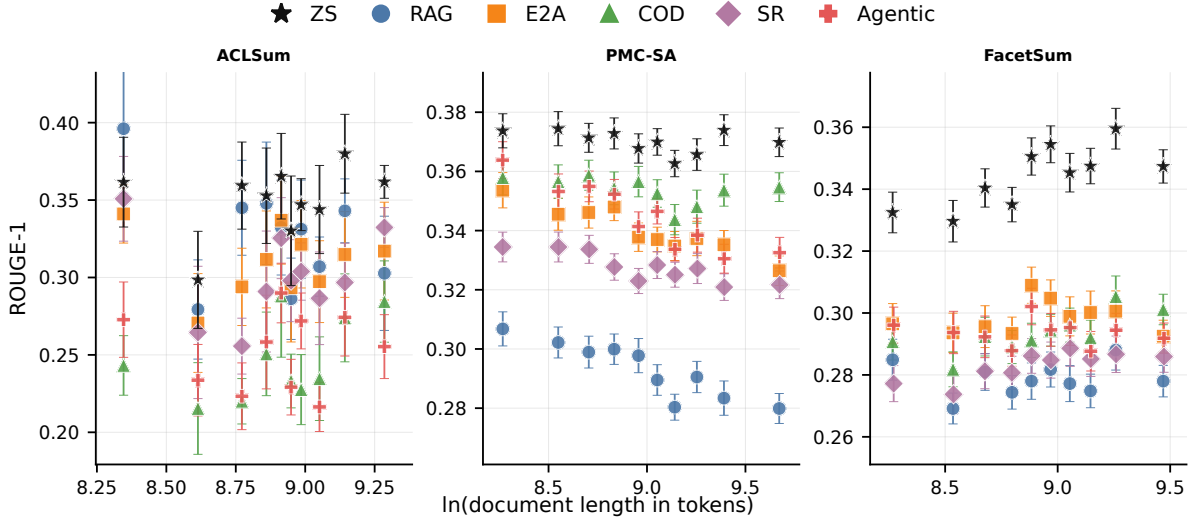


Figure 5: ROUGE-1 as a function of source document length for Qwen3.5-9B. Aspect scores are averaged within each document before binning, so every document contributes equally regardless of how many aspects it has. Documents are grouped into 10 equal-frequency bins per dataset; each marker is placed at the mean log-length of its bin and shows the mean score over the documents in that bin, with error bars giving the standard error of that mean. Length is measured in Qwen3.5-9B tokenizer tokens.

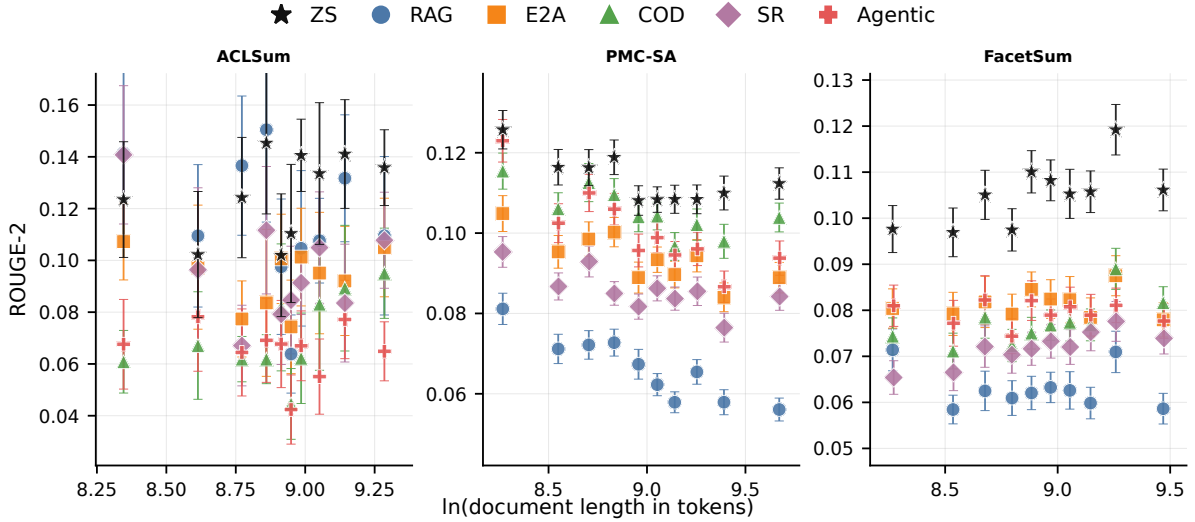


Figure 6: ROUGE-2 as a function of source document length for Qwen3.5-9B. Aspect scores are averaged within each document before binning, so every document contributes equally regardless of how many aspects it has. Documents are grouped into 10 equal-frequency bins per dataset; each marker is placed at the mean log-length of its bin and shows the mean score over the documents in that bin, with error bars giving the standard error of that mean. Length is measured in Qwen3.5-9B tokenizer tokens.

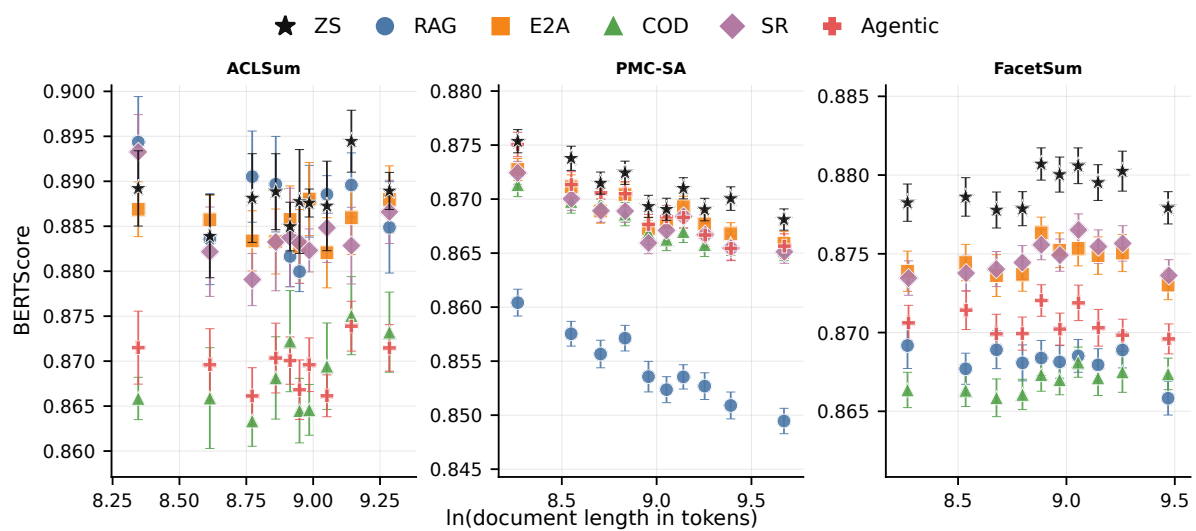


Figure 7: BERTScore as a function of source document length for Qwen3.5-9B. Aspect scores are averaged within each document before binning, so every document contributes equally regardless of how many aspects it has. Documents are grouped into 10 equal-frequency bins per dataset; each marker is placed at the mean log-length of its bin and shows the mean score over the documents in that bin, with error bars giving the standard error of that mean. Length is measured in Qwen3.5-9B tokenizer tokens.

I Qualitative Case Outputs

For reproducibility of the qualitative analysis in Table 4, we provide the exact reference and model-generated summaries for the selected ACLSum datapoint `aclsum_sampled_test_22_approach_long` (aspect: *Approach*). Table 15 reports the corresponding outputs.

Exp	ACLSum					PMC-SA					FacetSum				
	R-1	R-2	R-L	R-L-S	BS	R-1	R-2	R-L	R-L-S	BS	R-1	R-2	R-L	R-L-S	BS
ZS	0.292	0.085	0.200	0.205	0.875	0.363	0.115	0.207	0.310	0.868	0.336	0.105	0.211	0.275	0.870
E2A	0.257	0.064	0.184	0.184	0.873	0.313	0.075	0.176	0.260	0.858	0.259	0.051	0.165	0.196	0.862
RAG	<u>0.300</u>	0.082	0.213	0.213	0.880	0.315	0.074	0.181	0.262	0.859	0.300	0.069	0.195	0.229	0.871
COD	0.308	0.088	<u>0.210</u>	<u>0.211</u>	0.877	<u>0.361</u>	<u>0.105</u>	<u>0.206</u>	<u>0.295</u>	0.868	0.349	0.104	0.222	<u>0.267</u>	0.876
SR	0.290	0.075	0.203	0.204	<u>0.878</u>	0.356	0.099	0.205	0.293	0.869	0.319	0.085	0.205	0.247	<u>0.874</u>
Agentic	0.284	0.068	0.195	0.195	0.874	0.349	0.095	0.201	0.289	0.867	0.286	0.069	0.186	0.216	0.868

Table 9: Automatic summarization evaluation results with OLMo-3-7B-Instruct on ACLSum, PMC-SA, and FacetSum using ROUGE and BERTScore metrics. R-1, R-2, R-L, and R-L-S denote ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-LSum, respectively, while BS denotes BERTScore. Best results are shown in bold and second-best results are underlined.

Exp	ACLSum			PMC-SA			FacetSum		
	C-P	C-R	C-F1	C-P	C-R	C-F1	C-P	C-R	C-F1
ZS	0.074	<u>0.270</u>	<u>0.074</u>	0.190	0.404	0.192	0.125	0.387	0.130
E2A	0.069	0.124	0.049	0.124	0.197	0.091	0.056	0.090	0.036
RAG	0.083	0.178	0.067	0.137	0.146	0.078	0.106	0.124	0.059
COD	0.082	0.305	0.090	<u>0.193</u>	<u>0.300</u>	0.149	0.143	<u>0.306</u>	<u>0.125</u>
SR	0.121	0.178	<u>0.074</u>	0.216	0.281	<u>0.155</u>	<u>0.133</u>	0.224	0.097
Agentic	<u>0.092</u>	0.179	<u>0.074</u>	0.176	0.272	0.141	0.098	0.161	0.068

Table 10: Claim-based evaluation results with OLMo-3-7B-Instruct. C-P, C-R, and C-F1 denote claim precision, claim recall, and claim F1, respectively. Best results are in bold and second-best results are underlined.

Aspect	ZS	E2A	RAG	COD	SR	Agentic
Challenge	0.2713	0.2383	<u>0.2637</u>	0.1890	0.2469	0.2005
Approach	0.4081	0.3831	0.3921	0.3077	0.3370	0.2940
Outcome	0.3858	0.3208	0.3487	0.2578	0.3187	0.2858
Purpose	0.4260	0.3736	0.3617	0.3591	0.3582	<u>0.3960</u>
Methods	0.3679	0.3251	0.2897	<u>0.3433</u>	0.3193	0.3161
Findings	0.3367	<u>0.2808</u>	0.2474	0.2796	0.2697	0.2734
Value	0.2551	0.2226	<u>0.2367</u>	0.2134	0.1945	0.2055
Background	0.3699	0.3503	0.3239	0.3566	0.3195	<u>0.3634</u>
Conclusions	0.3373	<u>0.3355</u>	0.2777	0.3204	0.3262	0.3316
Results	0.3966	0.3343	0.2640	<u>0.3751</u>	0.3345	0.3505
Objectives	0.4521	<u>0.4004</u>	0.3563	0.3825	0.3990	0.4312

Table 11: Aspect-level ROUGE-1 results.

Aspect	ZS	E2A	RAG	COD	SR	Agentic
Challenge	<u>0.1895</u>	0.1726	0.1942	0.1443	0.1844	0.1451
Approach	0.3029	<u>0.2813</u>	0.2880	0.2343	0.2445	0.2164
Outcome	0.2926	0.2564	<u>0.2678</u>	0.1926	0.2649	0.2045
Purpose	0.2903	0.2549	0.2396	0.2376	0.2462	<u>0.2740</u>
Methods	0.2283	0.2029	0.1776	<u>0.2130</u>	0.2018	0.2010
Findings	0.2206	<u>0.1857</u>	0.1570	<u>0.1827</u>	0.1847	0.1799
Value	0.1631	0.1433	<u>0.1490</u>	0.1372	0.1290	0.1329
Background	<u>0.2171</u>	0.2080	0.1863	0.2104	0.1896	0.2228
Conclusions	<u>0.2002</u>	0.2038	0.1589	0.1889	0.1961	0.2025
Results	0.2271	0.1939	0.1474	<u>0.2179</u>	0.1976	0.2139
Objectives	0.3301	0.2986	0.2390	0.2630	0.2660	<u>0.3249</u>

Table 13: Aspect-level ROUGE-L results.

Aspect	ZS	E2A	RAG	COD	SR	Agentic
Challenge	0.0709	0.0567	0.0727	0.0423	0.0676	0.0420
Approach	0.1548	<u>0.1333</u>	0.1468	0.1005	0.1091	0.0846
Outcome	0.1619	0.1029	<u>0.1411</u>	0.0765	0.1143	0.0827
Purpose	0.1660	0.1251	0.1085	0.1159	0.1145	<u>0.1419</u>
Methods	0.1149	0.0875	0.0678	<u>0.1017</u>	0.0833	0.0852
Findings	0.0957	<u>0.0725</u>	0.0478	0.0681	0.0669	0.0657
Value	0.0484	<u>0.0421</u>	0.0394	0.0362	0.0260	0.0330
Background	0.1153	0.0994	0.0863	<u>0.1095</u>	0.0795	0.1112
Conclusions	0.0971	<u>0.0936</u>	0.0551	<u>0.0874</u>	0.0825	0.0928
Results	0.1269	0.0944	0.0535	<u>0.1167</u>	0.0973	0.1107
Objectives	0.2092	<u>0.1818</u>	0.1283	0.1625	0.1654	0.1986

Table 12: Aspect-level ROUGE-2 results.

Aspect	ZS	E2A	RAG	COD	SR	Agentic
Challenge	<u>0.1895</u>	0.1726	0.1942	0.1443	0.1844	0.1451
Approach	0.3029	<u>0.2813</u>	0.2880	0.2343	0.2445	0.2164
Outcome	0.2926	0.2564	<u>0.2678</u>	0.1926	0.2649	0.2049
Purpose	0.3529	<u>0.3030</u>	0.2911	0.2920	0.2964	0.3217
Methods	0.3013	0.2659	0.2397	<u>0.2822</u>	0.2649	0.2573
Findings	0.2762	<u>0.2291</u>	0.2030	0.2286	0.2251	0.2225
Value	0.2017	0.1750	<u>0.1847</u>	0.1689	0.1565	0.1619
Background	0.3123	0.2962	0.2736	<u>0.3006</u>	0.2738	0.3082
Conclusions	0.2782	<u>0.2766</u>	0.2278	0.2627	0.2722	0.2748
Results	0.3394	0.2879	0.2309	<u>0.3215</u>	0.2906	0.3032
Objectives	0.3802	0.3412	0.3024	0.3374	<u>0.3786</u>	0.3456

Table 14: Aspect-level ROUGE-LSum results.

System	Summary Text
Reference	They propose an unsupervised method which requires only a dictionary and raw texts as context knowledge allowing automatic updates of the state of a language.
ZS	The proposed approach employs a fully unsupervised method utilizing an electronic dictionary and unannotated web text to automatically acquire dependency relations, which are then matched against glosses to determine the most semantically coherent word sense.
Agentic	This fully unsupervised WSD approach constructs a knowledge base from unannotated text and selects senses by maximizing semantic coherence via dependency, backward, and synonym matching strategies.
CoD	The approach uses Minipar parsing and TreeMatching with dependency, synonym, and backward matching to select the most semantically coherent sense.
E2A	The approach employs a fully unsupervised WSD system using semantic coherence matching strategies on unannotated text and WordNet.
RAG	To overcome the knowledge acquisition bottleneck, the approach employs automatic techniques to create and maintain a comprehensive, dynamic context knowledge base inspired by human disambiguation processes.

Table 15: Paraphrased reference and generated summaries for the ACLSum qualitative case discussed in the main paper (aclsum_sampled_test_22_approach_long, aspect *Approach*).