BookAsSumQA: An Evaluation Framework for Aspect-Based Book Summarization via Question Answering

Ryuhei Miyazato[1], Ting-Ruen Wei[2], Xuyang Wu[2], Hsin-Tai Wu[3], Kei Harada[1],
[1]The University of Electro-Communications,
[2]Santa Clara University, [3]DOCOMO Innovations, Inc.,
Correspondence: miyazato@uec.ac.jp, harada@uec.ac.jp

Abstract

Aspect-based summarization aims to generate summaries that highlight specific aspects of a text, enabling more personalized and targeted summaries. However, its application to books remains unexplored due to the difficulty of constructing reference summaries for long text. To address this challenge, we propose BookAsSumQA, a QA-based evaluation framework for aspect-based book summarization. BookAsSumQA automatically constructs a narrative knowledge graph and synthesizes aspect-specific QA pairs to evaluate summaries based on their ability to answer these questions. Our experiments on BookAsSumQA revealed that while LLM-based approaches showed higher accuracy on shorter texts, RAG-based methods become more effective as document length increases, making them more efficient and practical for aspect-based book summarization[1].

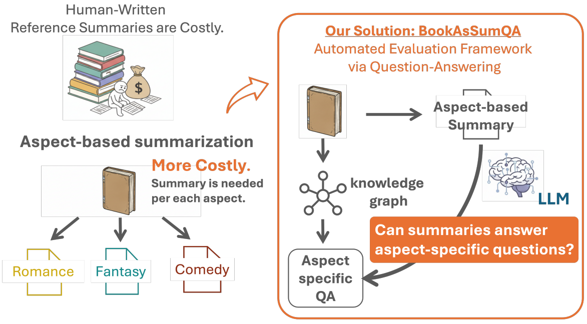

Human-Written
Reference Summaries are Costly.
Aspect-based summarization
More Costly.
Summary is needed per each aspect.
Romance
Fantasy
Comedy
Our Solution: BookAsSumQA
Automated Evaluation Framework via Question-Answering
Aspect-based Summary
knowledge graph
LLM
Can summaries answer aspect-specific questions?
Aspect specific QA

Figure 1: In BookAsSumQA, we generate aspect-specific QA pairs from a knowledge graph and evaluate summaries by testing whether they can answer these questions, thereby assessing aspect coverage without costly human-written references.

1 Introduction

Automatic summarization condenses long texts into concise and informative representations, allowing readers to grasp key information efficiently. Book summarization applies this to novels, which are often lengthy and complex. The progress of automatic book summarization has been accelerated by the release of the BookSum dataset (Kryscinski et al., 2022), which contains novels paired with human-written summaries. With the growing volume of books, there is increasing interest in aspect-based summarization (ABS), which produces summaries tailored to specific aspects, such as themes or genres. Although ABS helps readers quickly access desired information and has been more actively explored in domains such as reviews (Xu et al., 2023) and lectures (Kolagar and Zarcone, 2024), its application to books remains relatively understudied. This is mainly because summarization research relies on manually created reference summaries, and building evaluation datasets for long documents is a labor-intensive and costly process. The longer the original document and the greater the number of aspects, the higher the human and financial costs become.

To address this challenge, we propose BookAsSumQA, a QA-based evaluation framework for aspect-based book summarization that enables evaluation without manually created reference summaries. We synthesize aspect-specific QA pairs from the narrative through a knowledge graph, and evaluate aspect-based summaries by testing whether an LLM can answer these questions using the generated summary as reference. This allows us to measure how well the summary captures information about the aspects of the narrative. In this study, we define aspects as literary genres in novels (example: Figure 1).

First, we construct a knowledge graph that represents relationships among entities in the narrative. Using an LLM, we extract relationships between entities (e.g., characters) with a textual description, keywords, and an importance score, and incre-

[1] https://github.com/ryuhei-miyazato/bookassumqa

Proceedings of the IJCNLP-AACL 2025 Student Research Workshop, pages 123–133
December 20-24, 2025 ©2025 Association for Computational Linguistics

mentally upsert them into the graph to capture the global relationships within the narrative. Next, we construct aspect-specific QA pairs from the knowledge graph. To do so, we first identify edges that are relevant to a target aspect by calculating the cosine similarity between the text embeddings of the aspect term and the edge keywords, and then generate aspect-specific QA pairs based on the descriptions of those edges. Finally, we evaluate ABS methods using the generated QA pairs by assessing whether each generated summary can correctly answer the questions. We then compare the generated answers against the ground-truth using ROUGE-1, METEOR, and BERTScore. By comparing the accuracy, we investigate which method is most suitable for aspect-based book summarization.

2 Related Work

In the field of book summarization, as the BookSum dataset (Kryscinski et al., 2022) provides pairs of public domain novels and generic summaries, obtaining the summaries is well studied (Wu et al., 2021; Xiong et al., 2023; Liu et al., 2023). In this study, we focus on ABS, which generates summaries centered on specific aspects of a text. Unlike Query-Focused Summarization (QFS), which generates summaries in response to specific user queries (e.g., SQuALITY (Wang et al., 2022)), ABS instead focuses on predefined aspects such as genres or themes.

ABS has been actively studied in domains such as news (Zhang et al., 2024), reviews (Xu et al., 2023), lecture materials (Kolagar and Zarcone, 2024), and multi-domain documents (Hayashi et al., 2021), where reference summaries are often manually created or readily available. However, for long documents like books, creating such references is labor-intensive and costly, limiting the application of ABS in this domain.

To overcome this difficulty, we propose a framework that evaluates aspect-based summaries of novels without manual reference summaries. While several studies have proposed reference-free evaluation metrics for summarization that assess summary quality without relying on gold reference summaries (Chen et al., 2021; Liu et al., 2022; Gigant et al., 2024), we introduce a QA-based framework that evaluates summaries without manual references by generating QA pairs from the source text, measuring how much information from the source text is captured in the summary (Hirao et al., 2001; Scialom et al., 2019; Pu et al., 2024). In this work, we further extend this approach by generating aspect-specific QA pairs to evaluate how well each aspect-based summary captures information related to its corresponding aspect in the original text.

3 BookAsSumQA

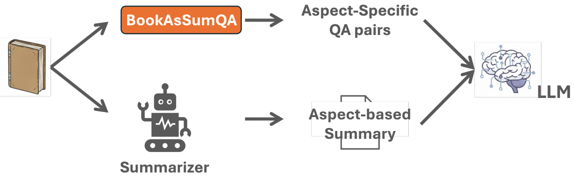


Figure 2: BookAsSumQA: Evaluation framework for aspect-based book summarization.

3.1 ABS Evaluation with BookAsSumQA

In BookAsSumQA (Figure 2), we shift the evaluation of aspect-based summaries into a Question-Answering task. QA pairs are automatically synthesized through a knowledge graph of the narrative, where nodes are enriched with keywords and description to generate comprehensive aspect-specific questions. The quality of a summary is then assessed by measuring how well the generated aspect-based summary enables an LLM to answer these aspect-specific QA, indicating how much information about the target aspect the summary truly captures.

3.2 QA Generation Process

An overview of the QA generation process is illustrated in Figure 3. The process consists of three stages: (1) splitting the text into chunks and extracting entities and relations, (2) inserting the extracted entities and relations into a knowledge graph as nodes and edges, and (3) synthesizing aspect-specific QA pairs from the completed graph.

(1). Chunking and Extraction Each book is split into chunks of 1,200 characters with an overlap of 100 characters, following the parameters of GraphRAG (Edge et al., 2024). From each chunk, entities (e.g., characters, events, concepts) are extracted using an LLM with a specifically designed prompt (2-shot, Appendix C, Figure 6). For each extracted relation, the prompt instructs the LLM to output a textual description, representative keywords, and an importance score ranging from 1 to

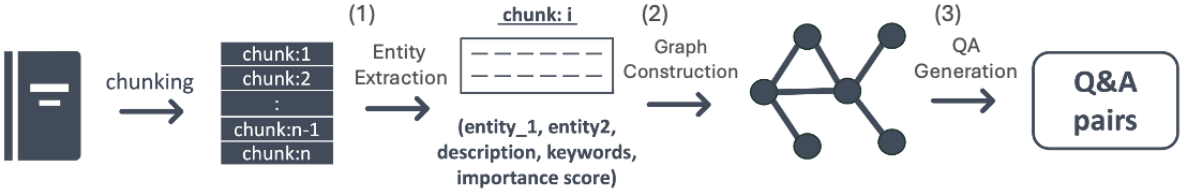


Figure 3: QA Generation Process. (1) splitting the text into chunks and extracting entities and relations, (2) inserting the extracted entities and relations into a knowledge graph as nodes and edges, and (3) synthesizing aspect-specific QA pairs from the completed graph.

10, reflecting the importance of the relationship within the local context.

(2). Knowledge Graph Construction The extracted entities and relations are incrementally inserted into a knowledge graph, where each edge is labeled with keywords, a textual description, and an importance score. If an entity already exists, its information is updated and summarized as needed, with keywords regenerated accordingly. In addition, importance score is accumulated by adding the newly assigned value to reflect repeated or strengthened relationships across chunks.

(3). QA Generation Once the knowledge graph is constructed, we generate aspect-specific QA pairs. We first filter edges to keep only those with an importance score of 10 or higher, considering relationships above this threshold to be important. An importance score of 10 indicates either a salient relationship or one that appears multiple times in the narrative, making it a stronger candidate for generating aspect-specific QA. From these, a maximum of 100 edges were selected. QA pairs are then generated from the description of each edge using a dedicated prompt (1-shot, Appendix C, Figure 7), with keywords from the edge also included in the generated QA. For each aspect, aspect-specific QA pairs were selected by calculating the cosine similarity between the text embeddings of the aspect and those of the QA keywords, and the top five most relevant QA were retained. Examples of aspect-specific QA pairs are also provided in Appendix D.

We utilized GPT-4o-mini [2] for both entity extraction and QA generation and used sentence-transformers/paraphrase-MiniLM-L6-v2 (Reimers and Gurevych, 2019) for text embedding. For implementation of graph-generation, we referred to the code of LightRAG (Guo et al., 2024).

4 Experimental Settings

4.1 Models

Since no existing ABS method specifically targets books, we compare various approaches, including LLMs and RAGs. Detail information about the models is in Appendix B.

LLMs Following the strategy of BooookScore (Chang et al., 2024), we adopt two workflows for summarizing book-length documents that exceed the model's context window: (1) Hierarchical Merging (Hier), which recursively merges summaries of individual chunks into higher-level summaries, and (2) Incremental Updating (Inc), which incrementally updates a single global summary as each new chunk is processed. Detailed descriptions are provided in Appendix B.

For experiments, we use both an open-source model, meta-llama/Llama-3.1-8B-Instruct [3], and a closed-source model, GPT-4o-mini.

RAGs RAG retrieves information relevant to a query from external sources and generates an answer. In this study, we adopt NaiveRAG (Gao et al., 2023), as well as GraphRAG (Edge et al., 2024) and LightRAG (Guo et al., 2024), which employ graph structures to organize external information.

4.2 Setup

The original texts used in this experiment are taken from BookSum (Kryscinski et al., 2022), which sources books from the Project Gutenberg public domain book repository with expired copyrights. We selected texts with varying lengths: over 200,000 words (large), between 90,000 and

[2] https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[3] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

110,000 words (middle), and less than 20,000 words (small), comprising 12, 9, and 9 books respectively, for a total of 30. In this paper, we define fourteen "aspects" as the literary genre of a novel with reference to Wikipedia's List of writing genres[4](see Table 1).

Fantasy	Romance	Comedy
Paranormal	Young Adult	Horror
History	Action	Science Fiction
Mystery	Adventure	Crime
Thriller	Poetry	

Table 1: List of Aspects used in this study.

For each method, aspect-based summaries were generated for the aspects listed in Table 1, with each summary limited to 300 tokens. The generated summaries were evaluated based on their ability to answer the corresponding QA pairs with referring the generated summary. The prompts used for this QA-answering process are provided in the Appendix C (Figure 8). The accuracy of the answers was evaluated using ROUGE-1 (Lin, 2004), METEOR (Banerjee and Lavie, 2005) and BERTScore (Zhang et al., 2020) metrics, measuring the alignment between the generated answers and the ground-truth.

RAG-based methods index the original text once and reuse it to generate summaries for different aspects, whereas LLM-based methods generate a new summary every time for each aspect.

5 Results

5.1 Question Answering Using Aspect-Based Summaries

Type	method	ROUGE-1	METEOR	BERTScore
LLM	Llama + Hier	22.43	19.23	85.66
	GPT + Hier	**22.49**	**19.49**	**85.82**
	Llama + Inc	21.91	18.23	85.48
	GPT + Inc	21.90	18.76	85.47
RAG	NaiveRAG	21.43	18.66	85.44
	GraphRAG	14.66	13.56	84.50
	LightRAG	20.61	18.41	85.51

Table 2: Results of aspect-based summarization using different methods. LLM-based methods include Llama-3.1-8B-Instruct (Llama) and GPT-4o-mini (GPT).

Table 2 shows the accuracy for aspect QA with generated aspect-based summaries. Each value represents the average result across all aspects.

Overall, the method that applies Hierarchical Merging with GPT-4o-mini achieved the highest scores. Among LLM-based methods, Hierarchical Merging was better than Incremental Updating, and LLM-based methods overall surpass RAG-based methods. For RAG, NaiveRAG achieves the best results, while GraphRAG shows considerably lower scores compared to the other methods.

One possible reason for the superior performance of LLM-based methods is that LLM-based methods extract aspect-specific information from finer-grained chunks. Although incremental updating incorporates previous context, using both the prior summary and the current chunk may make it harder to extract targeted information. In GraphRAG, summaries are generated for each community in the graph and used to answer QA, making it less effective at capturing aspect-related stories. According to the results in the Appendix A.1 (Table 4), GraphRAG achieves the highest accuracy in conventional summarization, suggesting that improving the construction of the graph and the summarization process could lead to better scores in the future.

5.2 Comparison by Original Text Length

Size	Method	ROUGE-1	METEOR	BERTScore
Small	GPT + Hier	**25.66**	**21.91**	**86.54**
	GPT + Inc	24.81	20.84	86.14
	NaiveRAG	22.09	19.24	85.58
Middle	GPT + Hier	**21.95**	**19.52**	85.56
	GPT + Inc	21.68	18.68	85.35
	NaiveRAG	**21.95**	19.45	**85.62**
Large	GPT + Hier	20.50	**17.65**	**85.48**
	GPT + Inc	19.88	17.27	85.06
	NaiveRAG	**20.55**	17.64	85.21

Table 3: Comparison by Original Text Length
(Small: <20k words, Middle: 90k–110k, Large: >200k)

We conducted an experiment to compare summarization performance across different lengths of the original text. In this experiment, we used the best-performing models from the LLM-based and RAG-based approaches identified in Section 5.1.

As shown in Table 3, the performance tends to decline as the length of the original text increases. Although NaiveRAG performs worse than the LLM-based method in the small group, its performance becomes comparable to that of the LLM-based approach in the middle and large groups.

Considering that RAG-based methods can generate aspect-based summaries for different queries with a single indexing of the original text, RAG-

[4] https://en.wikipedia.org/wiki/List_of_writing_genres#Fiction_genres

based approaches may be more suitable for aspect-based summarization of longer documents.

6 Conclusion

In this study, we proposed BookAsSumQA, a QA-based evaluation framework for aspect-based book summarization. Constructing knowledge graphs and automatically generating aspect-specific QA enable evaluation of ABS quality without human-annotated reference summaries. In our experiments with BookAsSumQA, while LLM-based approaches performed better on shorter texts, RAG-based methods achieved comparable performance on longer documents. These results suggest that RAG-based methods are more practical and scalable choice for aspect-based book summarization. Future work will explore specialized indexing and retrieval techniques.

Limitations

This study has several limitations. First, we used gpt-4o-mini to generate QA pairs for summary evaluation; the choice of model may affect the evaluation results. In future work, we plan to investigate the impact of different models for QA generation. Second, both QA generation and answering relied on LLMs, which may incorporate external knowledge beyond the original text or summaries. To address this, we plan to explore methods for restricting the model's context strictly to the given text and summaries, ensuring fairer evaluation. Third, we have not yet compared our framework with other reference-free evaluation metrics or with human judgments. Such comparisons would help clarify how BookAsSumQA aligns with human evaluation and how it complements existing automatic metrics in terms of reliability and interpretability.

Acknowledgments

We would like to thank Dr. Shunsuke Kitada for his valuable advice and insightful feedback on the writing of this paper.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. Booookscore: A systematic exploration of book-length summarization in the era of LLMs. In *The Twelfth International Conference on Learning Representations*.

Wang Chen, Piji Li, and Irwin King. 2021. A training-free and reference-free summarization evaluation metric via centrality-weighted relevance and self-referenced redundancy. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 404–414, Online. Association for Computational Linguistics.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2:1.

Théo Gigant, Camille Guinaudeau, Marc Decombas, and Frederic Dufaux. 2024. Mitigating the impact of reference quality on evaluation of summarization systems with reference-free metrics. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19355–19368, Miami, Florida, USA. Association for Computational Linguistics.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*.

Hiroaki Hayashi, Prashant Budania, Peng Wang, Chris Ackerson, Raj Neervannan, and Graham Neubig. 2021. WikiAsp: A dataset for multi-domain aspect-based summarization. *Transactions of the Association for Computational Linguistics*, 9:211–225.

Tsutomu Hirao, Yutaka Sasaki, and Hideki Isozaki. 2001. An extrinsic evaluation for question-biased text summarization on qa tasks. In *Proceedings of the NAACL 2001 Workshop on Automatic Summarization*, pages 61–68.

Zahra Kolagar and Alessandra Zarcone. 2024. HumSum: A personalized lecture summarization tool for humanities students using LLMs. In *Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024)*, pages 36–70, St. Julians, Malta. Association for Computational Linguistics.

Wojciech Kryscinski, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2022. BOOKSUM: A collection of datasets for long-form narrative summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6536–6558, Abu Dhabi, United Arab Emirates. 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.

Dongqi Liu, Yifan Wang, and Vera Demberg. 2023. Incorporating distributions of discourse structure for long document abstractive summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5574–5590, Toronto, Canada. Association for Computational Linguistics.

Yizhu Liu, Qi Jia, and Kenny Zhu. 2022. Reference-free summarization evaluation via semantic correlation and compression ratio. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2109–2115, Seattle, United States. Association for Computational Linguistics.

Xiao Pu, Mingqi Gao, and Xiaojun Wan. 2024. Is summary useful or not? an extrinsic human evaluation of text summaries on downstream tasks. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9389–9404, Torino, Italia. ELRA and ICCL.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2019. Answers unite! unsupervised metrics for reinforced summarization models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3246–3256, Hong Kong, China. Association for Computational Linguistics.

Alex Wang, Richard Yuanzhe Pang, Angelica Chen, Jason Phang, and Samuel R. Bowman. 2022. SQuALITY: Building a long-document summarization dataset the hard way. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1139–1156, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jeff Wu, Long Ouyang, Daniel M. Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul F. Christiano. 2021. Recursively summarizing books with human feedback. *CoRR*, abs/2109.10862.

Wenhan Xiong, Anchit Gupta, Shubham Toshniwal, Yashar Mehdad, and Scott Yih. 2023. Adapting pretrained text-to-text models for long text sequences. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5566–5578, Singapore. Association for Computational Linguistics.

Hongyan Xu, Hongtao Liu, Zhepeng Lv, Qing Yang, and Wenjun Wang. 2023. Pre-trained personalized review summarization with effective salience estimation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10743–10754, Toronto, Canada. Association for Computational Linguistics.

Lemei Zhang, Peng Liu, Marcus Tiedemann Oekland Henriksboe, Even W. Lauvrak, Jon Atle Gulla, and Heri Ramampiaro. 2024. Personalsum: A user-subjective guided personalized summarization dataset for large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. An extrinsic evaluation for question-biased text summarization on qa tasks. In *In 8th International Conference on Learning Representations*.

A Experiment with Generic Summaries

A.1 Comparison Results between Reference Summaries and Standard Summaries

Type	method	ROUGE1	METEOR	BERTScore
LLM	GPT + Hier	20.64	9.87	82.89
	GPT + Inc	21.64	10.29	82.49
	Llama + Hier	23.96	11.28	**83.10**
	Llama + Inc	24.03	11.21	82.60
RAG	NaiveRAG	20.13	9.58	81.94
	GraphRAG	**25.37**	**14.78**	80.29
	LightRAG	20.66	10.00	81.87

Table 4: Comparison Results between Reference Summaries and Standard Summaries.

We conducted an experiment comparing the generic summaries generated by each model with the reference summaries in BookSum to evaluate the models' capabilities for generic summarization. The results are shown in Table 4.

In BookAsSumQA, the performance of GraphRAG was considerably worse than other methods. However, for standard summarization, it achieved the highest scores on two metrics based on character overlap. In contrast, it obtained the lowest score on BERTScore, which compares semantic similarity.

A.2 Results of BookAsSumQA with Generic Summaries

Type	method	ROUGE	METEOR	BERT_Score
LLM	GPT + Hier	20.65	18.45	85.35
	GPT + Inc	20.63	17.51	85.23
	Llama + Hier	19.86	16.45	85.23
	Llama + Inc	20.72	17.27	85.41
RAG	NaiveRAG	19.76	17.28	85.05
	GraphRAG	15.12	14.37	84.81
	LightRAG	20.29	17.79	85.48

Table 5: The results of BookAsSumQA with generic summaries.

We conducted an experiment comparing the accuracy of answering QA pairs generated by BookAsSumQA, using standard summaries produced by each model employed in our experiments in Section 4. The results are shown in Table 5.

Compared to the results in Table 2, aspect-based summaries achieved higher accuracy in answering aspect-specific QA. Additionally, while there were notable differences among methods when using aspect-based summaries, the results for generic summaries were more similar across methods. These findings indicate that BookAsSumQA serves as an evaluation framework for aspect-based summarization.

B Detail Information of Summarizer

LLMs

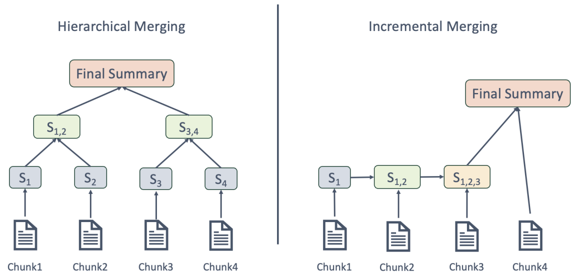


Figure 4: (1) Hierarchical Merging and (2) Incremental Updating.

For LLM-based summarization, we adopt two prompting workflows for summarizing book-length documents that exceed the model's context window (Figure 4): (1) Hierarchical Merging (Hier) and (2) Incremental Updating (Inc), following BooookScore (Chang et al., 2024).

In both workflows, the input document is first divided into smaller chunks (e.g., a chunk size of 2048 tokens). In the hierarchical merging strategy, each chunk is summarized separately, and the resulting summaries are merged using additional prompts. In the incremental updating strategy, a global summary is updated and compressed step-by-step as the model processes each chunk.

RAGs

For RAG-based method, we used several RAG as described below. We used the default settings for indexing and retrieval methods, and built the same database for each aspect-based summarization approach. For each aspect, summaries were generated using query (Figure 5) corresponding to that aspect as queries.

Generate a summary of the ${aspect} in this story.

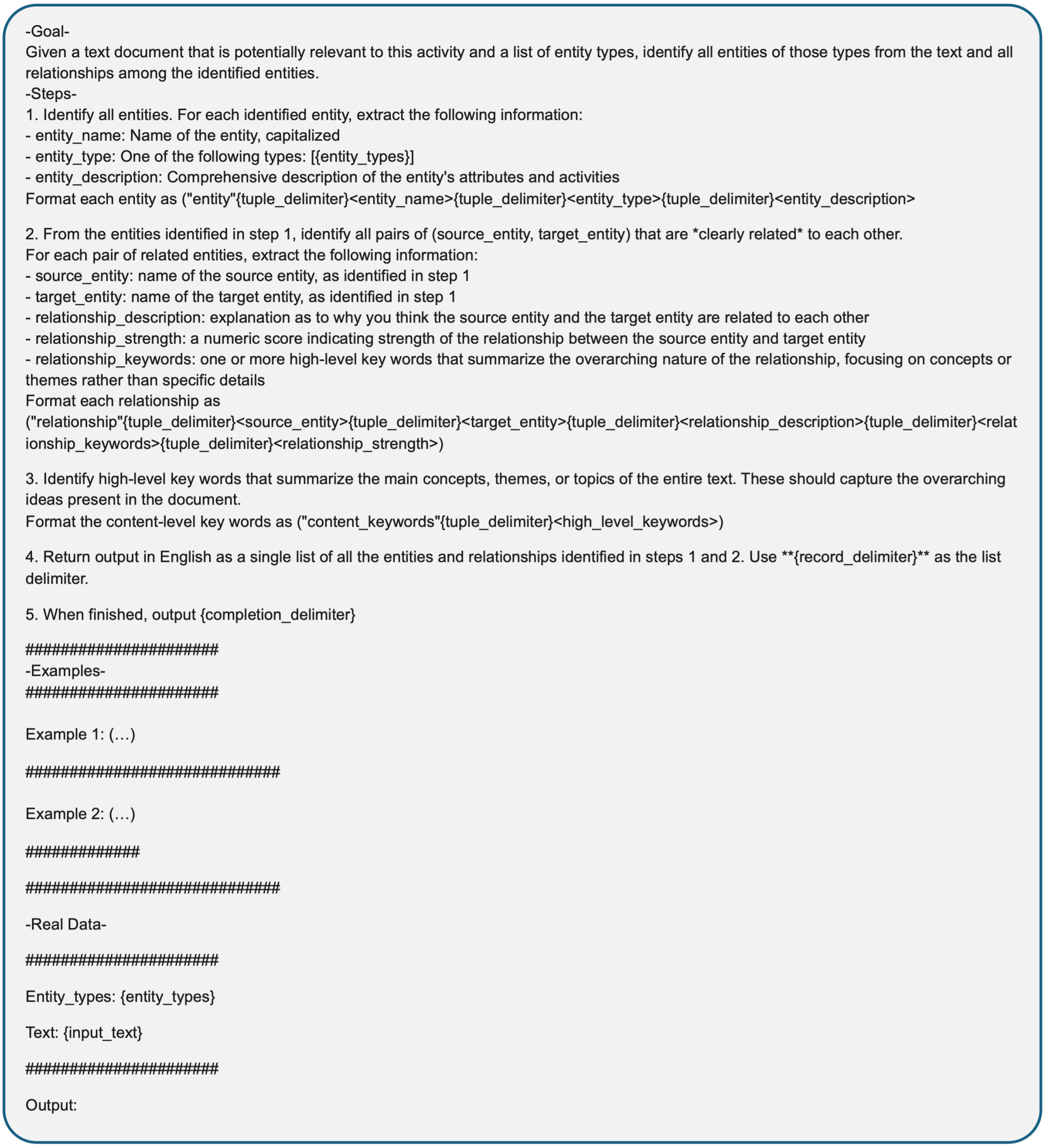
Figure 5: The query used for RAG-ased method.

- **NaiveRAG (Gao et al., 2023)**
NaiveRAG is a standard RAG system. It splits texts into chunks, embeds them, retrieves the most similar ones to a query, and generates an answer.

- **GraphRAG (Edge et al., 2024)**
GraphRAG creates a knowledge graph from the source text, generates community summaries by summarizing subgraphs, and uses them to answer queries.

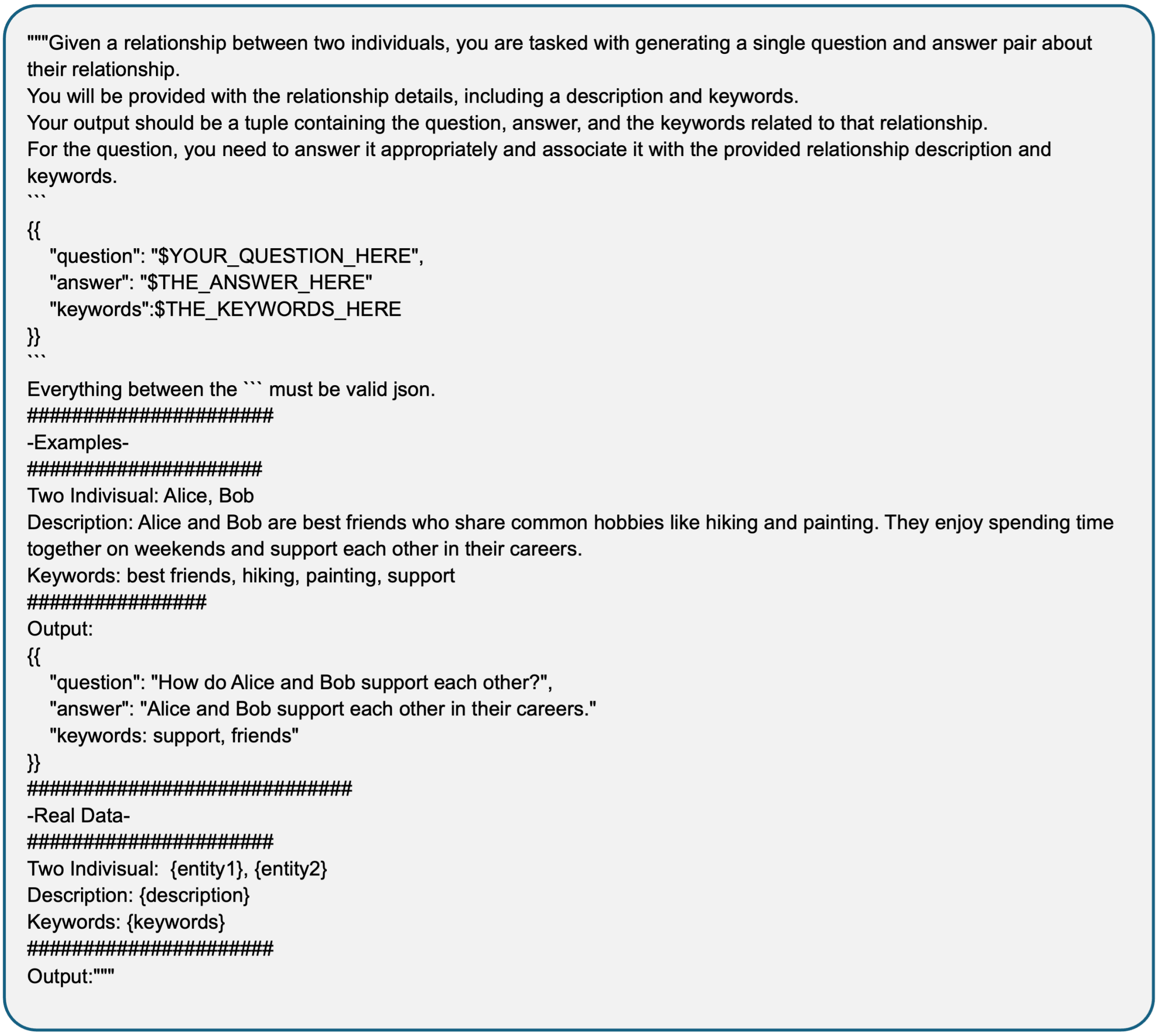
- **LightRAG (Guo et al., 2024)**
LightRAG builds a knowledge graph from the source text, retrieves relevant parts via the graph based on query keywords, and generates an answer.

C Prompt



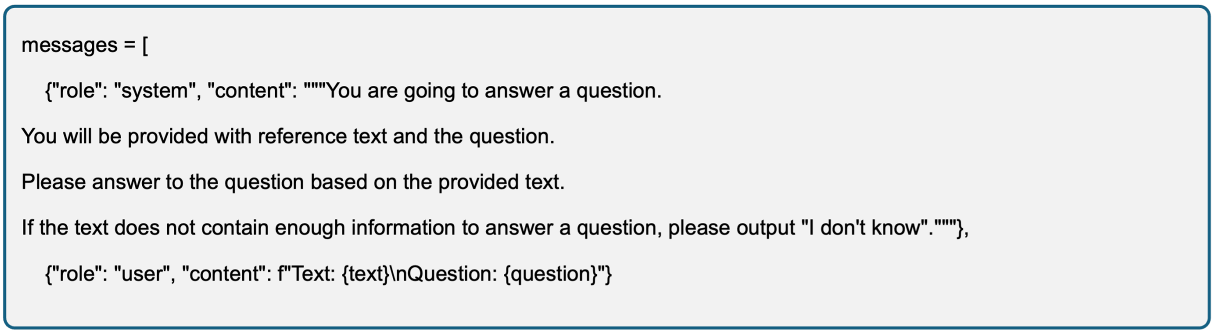
```
-Goal-
Given a text document that is potentially relevant to this activity and a list of entity types, identify all entities of those types from the text and all relationships among the identified entities.
-Steps-
1. Identify all entities. For each identified entity, extract the following information:
- entity_name: Name of the entity, capitalized
- entity_type: One of the following types: [{entity_types}]
- entity_description: Comprehensive description of the entity's attributes and activities
Format each entity as ("entity"{tuple_delimiter}<entity_name>{tuple_delimiter}<entity_type>{tuple_delimiter}<entity_description>

2. From the entities identified in step 1, identify all pairs of (source_entity, target_entity) that are *clearly related* to each other.
For each pair of related entities, extract the following information:
- source_entity: name of the source entity, as identified in step 1
- target_entity: name of the target entity, as identified in step 1
- relationship_description: explanation as to why you think the source entity and the target entity are related to each other
- relationship_strength: a numeric score indicating strength of the relationship between the source entity and target entity
- relationship_keywords: one or more high-level key words that summarize the overarching nature of the relationship, focusing on concepts or themes rather than specific details
Format each relationship as
("relationship"{tuple_delimiter}<source_entity>{tuple_delimiter}<target_entity>{tuple_delimiter}<relationship_description>{tuple_delimiter}<relationship_keywords>{tuple_delimiter}<relationship_strength>)

3. Identify high-level key words that summarize the main concepts, themes, or topics of the entire text. These should capture the overarching ideas present in the document.
Format the content-level key words as ("content_keywords"{tuple_delimiter}<high_level_keywords>)

4. Return output in English as a single list of all the entities and relationships identified in steps 1 and 2. Use **{record_delimiter}** as the list delimiter.

5. When finished, output {completion_delimiter}

######################
-Examples-
######################

Example 1: (…)

#############################

Example 2: (…)

#############

#############################

-Real Data-

######################

Entity_types: {entity_types}

Text: {input_text}

######################

Output:
```

Figure 6: Entity extraction prompt (Vanity Fair).



~~~
"""Given a relationship between two individuals, you are tasked with generating a single question and answer pair about their relationship.
You will be provided with the relationship details, including a description and keywords.
Your output should be a tuple containing the question, answer, and the keywords related to that relationship.
For the question, you need to answer it appropriately and associate it with the provided relationship description and keywords.
```
{{
    "question": "$YOUR_QUESTION_HERE",
    "answer": "$THE_ANSWER_HERE"
    "keywords":$THE_KEYWORDS_HERE
}}
```
Everything between the ``` must be valid json.
#######################
-Examples-
######################
Two Indivisual: Alice, Bob
Description: Alice and Bob are best friends who share common hobbies like hiking and painting. They enjoy spending time together on weekends and support each other in their careers.
Keywords: best friends, hiking, painting, support
#################
Output:
{{
    "question": "How do Alice and Bob support each other?",
    "answer": "Alice and Bob support each other in their careers."
    "keywords: support, friends"
}}
##############################
-Real Data-
#######################
Two Indivisual:  {entity1}, {entity2}
Description: {description}
Keywords: {keywords}
#######################
Output:"""
~~~

Figure 7: QA generation prompt (Vanity Fair).



```
messages = [

    {"role": "system", "content": """You are going to answer a question.

You will be provided with reference text and the question.

Please answer to the question based on the provided text.

If the text does not contain enough information to answer a question, please output "I don't know"."""},

    {"role": "user", "content": f"Text: {text}\nQuestion: {question}"}
```

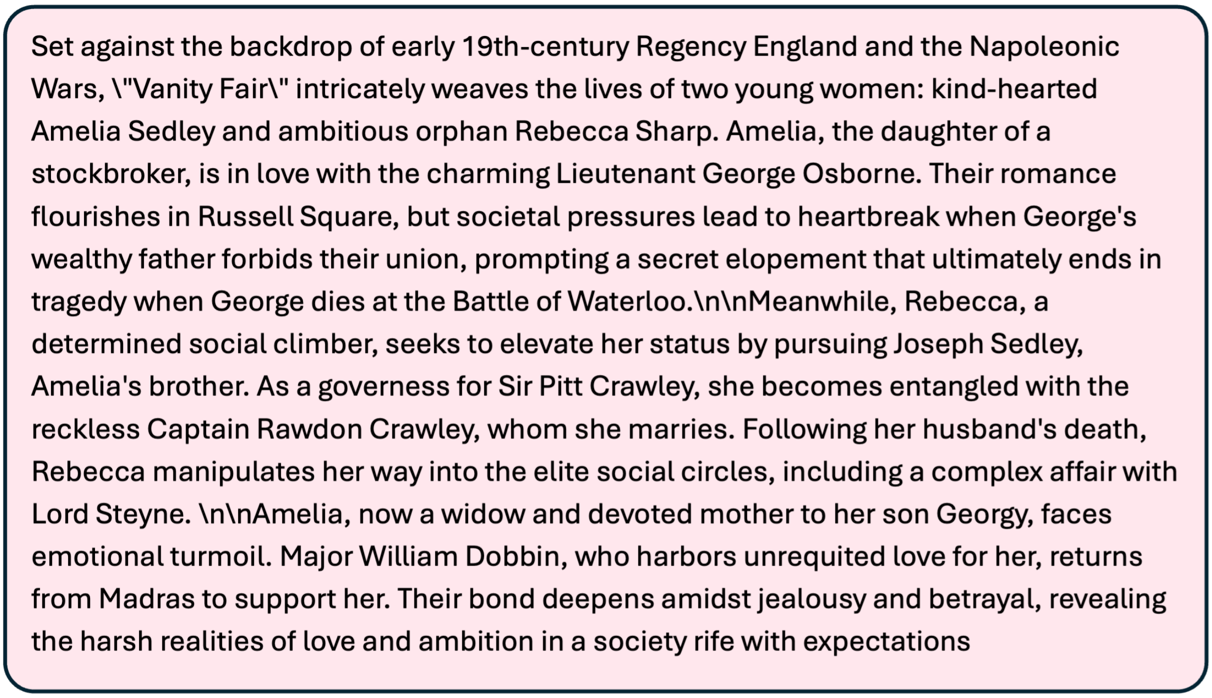
Figure 8: QA answering prompt (Vanity Fair).

D Example of Generated QA

Aspect	Question	Answer	Keywords
Romance	What indicates Joseph Sedley's romantic interest in Rebecca?	Joseph Sedley's romantic interest in Rebecca is indicated by his efforts to impress her, his sensitivity about his vanity, and his dependency on her during his illnesses, all of which suggest a growing intimacy and admiration for her.	romantic interest, intimacy, admiration, dependency
Action	What is the nature of the relationship between Miss Sharp and Miss Pinkerton?	The relationship between Miss Sharp and Miss Pinkerton is characterized by conflict and personal animosity, with Miss Sharp openly defying Miss Pinkerton's authority and expressing hatred towards her.	conflict, authority, antagonism, rebellion, defiance
Fantasy	What complexities characterize the friendship between George and Rawdon?	Their friendship is characterized by playful interactions, rivalry in romantic interests, elements of manipulation, and shared gambling habits, which create both camaraderie and challenges.	friendship, rivalry, manipulation, gambling, camaraderie
Young Adult	What is the nature of the relationship between Rebecca and Lord Steyne?	The relationship between Rebecca and Lord Steyne is multifaceted, characterized by mentorship, ambition, and social dynamics, with Rebecca leveraging Lord Steyne's favoritism for her family's benefit while also being dependent on his financial support.	mentor-mentee, social dynamics, influence, ambition

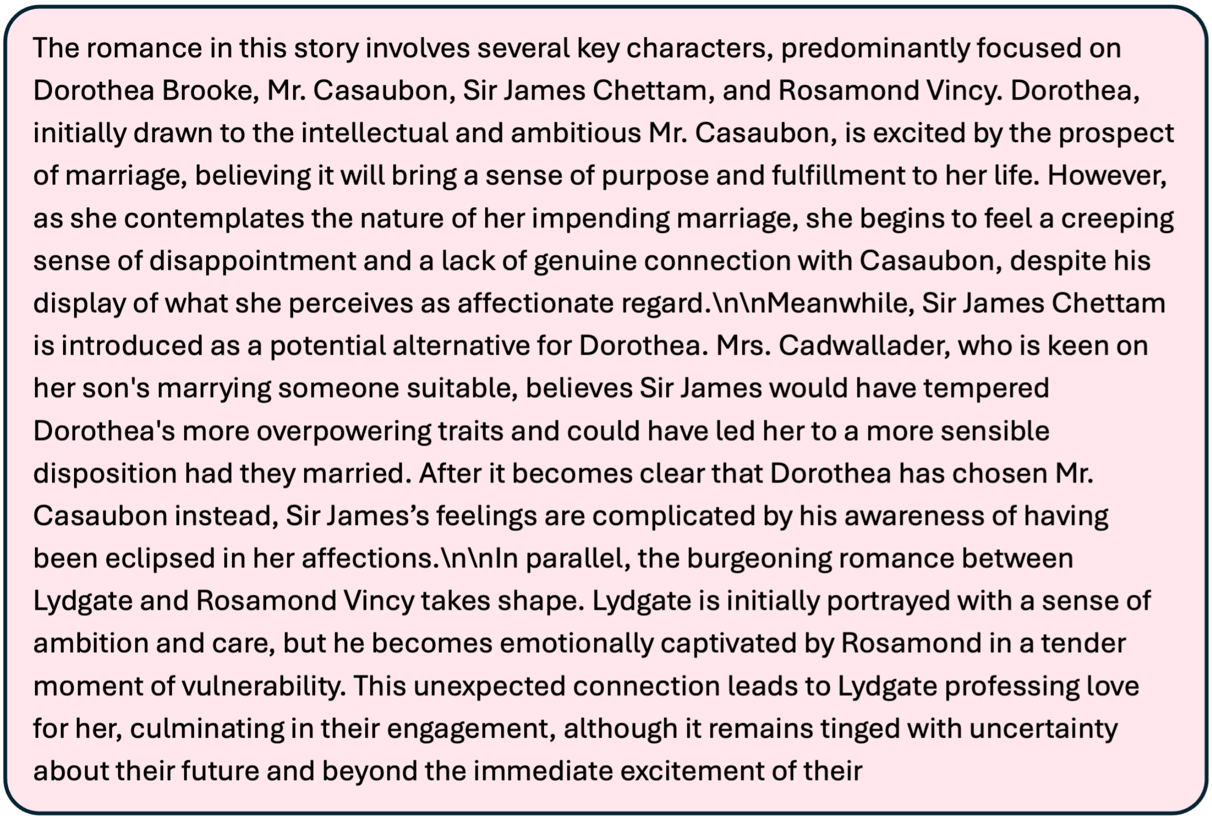
Table 6: Aspect-based QA Examples (Vanity Fair).

E The Example of Aspect-Based Summary (Romance)



Set against the backdrop of early 19th-century Regency England and the Napoleonic Wars, \"Vanity Fair\" intricately weaves the lives of two young women: kind-hearted Amelia Sedley and ambitious orphan Rebecca Sharp. Amelia, the daughter of a stockbroker, is in love with the charming Lieutenant George Osborne. Their romance flourishes in Russell Square, but societal pressures lead to heartbreak when George's wealthy father forbids their union, prompting a secret elopement that ultimately ends in tragedy when George dies at the Battle of Waterloo.\n\nMeanwhile, Rebecca, a determined social climber, seeks to elevate her status by pursuing Joseph Sedley, Amelia's brother. As a governess for Sir Pitt Crawley, she becomes entangled with the reckless Captain Rawdon Crawley, whom she marries. Following her husband's death, Rebecca manipulates her way into the elite social circles, including a complex affair with Lord Steyne. \n\nAmelia, now a widow and devoted mother to her son Georgy, faces emotional turmoil. Major William Dobbin, who harbors unrequited love for her, returns from Madras to support her. Their bond deepens amidst jealousy and betrayal, revealing the harsh realities of love and ambition in a society rife with expectations

Figure 9: Romance-based summary by GPT + Hier (Vanity Fair).



The romance in this story involves several key characters, predominantly focused on Dorothea Brooke, Mr. Casaubon, Sir James Chettam, and Rosamond Vincy. Dorothea, initially drawn to the intellectual and ambitious Mr. Casaubon, is excited by the prospect of marriage, believing it will bring a sense of purpose and fulfillment to her life. However, as she contemplates the nature of her impending marriage, she begins to feel a creeping sense of disappointment and a lack of genuine connection with Casaubon, despite his display of what she perceives as affectionate regard.\n\nMeanwhile, Sir James Chettam is introduced as a potential alternative for Dorothea. Mrs. Cadwallader, who is keen on her son's marrying someone suitable, believes Sir James would have tempered Dorothea's more overpowering traits and could have led her to a more sensible disposition had they married. After it becomes clear that Dorothea has chosen Mr. Casaubon instead, Sir James's feelings are complicated by his awareness of having been eclipsed in her affections.\n\nIn parallel, the burgeoning romance between Lydgate and Rosamond Vincy takes shape. Lydgate is initially portrayed with a sense of ambition and care, but he becomes emotionally captivated by Rosamond in a tender moment of vulnerability. This unexpected connection leads to Lydgate professing love for her, culminating in their engagement, although it remains tinged with uncertainty about their future and beyond the immediate excitement of their

Figure 10: Romance-based summary by NaiveRAG (Vanity Fair).