CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Investigating Text-only and Multimodal Retrieval Augmented Generation frameworks for Visual Question Answering

A study on the impact of modality and parameter optimization

Master's thesis in Applied Data Science

Marta Bortkiewicz & Cecilia Rundberg

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2024

Master's thesis 2024

Investigating Text-only and Multimodal Retrieval Augmented Generation frameworks for Visual Question Answering

A study on the impact of modality and parameter optimization

Marta Bortkiewicz & Cecilia Rundberg

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2024

Investigating Text-only and Multimodal Retrieval Augmented Generation frameworks for Visual Question Answering
A study on the impact of modality and parameter optimization
Marta Bortkiewicz & Cecilia Rundberg

© Marta Bortkiewicz & Cecilia Rundberg, 2024.

Supervisor: Ashkan Panahi, Department of Computer Science and Engineering
Advisor: Caroline Bükk, Wiretronic AB
Advisor: Isak Ernstig, Wiretronic AB
Examiner: Simon Olsson, Department of Computer Science and Engineering

Master's Thesis 2024
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2024

Investigating Text-only and Multimodal Retrieval Augmented Generation frameworks for Visual Question Answering
A study on the impact of modality and parameter optimization
Marta Bortkiewicz & Cecilia Rundberg
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Verifying the correctness of product assembly processes in a manufacturing setting is a crucial part of ensuring the quality of products. Automating this procedure can help improve both the security and the efficiency of the routines. Utilizing a machine learning model for automating these kinds of procedures would require fine-tuning and a lot of computational resources for the algorithm to adapt to the specific domain. An alternative approach that has been shown to enhance the performance to the same extent as fine-tuning while not requiring additional computational resources, is utilizing Retrieval-Augmented Generation (RAG) frameworks.

In this project, two RAG frameworks – a Text-only and a Multimodal RAG framework, are developed. The main goal of the frameworks is to accurately answer user queries about products where the ground-truth answer is located in a PDF user manual. The frameworks are developed by integrating a retrieval component and a generative component, either LLaMA2-7B or LLaVA-7B. The retrieval component retrieves relevant to the user query context from manuals, which the generative component uses to base the response on.

In addition to exploring how the performances are affected by the modalities of the frameworks, parameter tuning is explored. Evaluating how different values of chunk size and top-k parameters affect the performances allows for optimizing the RAG frameworks. The evaluation is performed by using BERTScore metrics and LangSmith metrics that provide complementary human-like judgment. The most crucial metrics are B-recall and contextual accuracy, which both evaluate how well the generated response captures the information embedded in the ground-truth answer.

The results show that the Text-only RAG framework is more stable across changes in the parameters than the Multimodal RAG framework, leading to generating more coherent and rational responses. However, finding the most optimal parameters for the Multimodal RAG framework could lead to it outperforming the Text-only RAG framework. Overall, moderate chunk sizes 128 and 256 and top-k values 4 or 6 led to the best-produced performances for both of the RAG frameworks.

Keywords: Data science, Retrieval-Augmented Generation, RAG, Multimodality, project, thesis.

Acknowledgements

We would like to say a big thank you to Wiretronic AB for providing us with the opportunity to collaborate on this Master's thesis project. We are particularly grateful to Caroline and Isak for supporting us and brainstorming with us during challenging times. We would also like to express our appreciation to our academic supervisor Ashkan, who has given us valuable feedback during the entire process.

Marta Bortkiewicz & Cecilia Rundberg, Gothenburg, 2024-06-23

Contents

List of Figures **xv**

List of Tables **xvii**

1 Introduction **1**
1.1 Aim of the project . . . 3
1.2 Research questions . . . 3
1.3 Motivations . . . 4
1.4 Challenges . . . 5

2 Background & Related work **7**
2.1 Background . . . 7
2.1.1 The RAG framework . . . 7
2.1.1.1 Text-only RAG . . . 8
2.1.1.2 Multimodal RAG . . . 9
2.1.2 RAG vs Fine-tuning . . . 10
2.1.3 Optimizing parameters . . . 10
2.1.3.1 Effects of chunk size . . . 11
2.1.3.2 Effects of top-k values . . . 11
2.1.4 Generative Large Language and Vision models . . . 12
2.1.4.1 LLaMA . . . 13
2.1.4.2 LLaVA . . . 13
2.1.5 Hallucination tendencies . . . 14
2.1.6 Evaluation . . . 15
2.1.6.1 BERTScore . . . 15
2.1.6.2 LangSmith . . . 17
2.2 Related work . . . 18
2.2.1 Manufacturing setting . . . 18
2.2.2 RAG . . . 19

3 Methodology **21**
3.1 Data . . . 21
3.2 Data extraction . . . 22
3.3 Chunking . . . 22
3.4 Text-only RAG . . . 23
3.4.1 Generating summaries . . . 23
3.4.2 Embedding and Retrieval . . . 24
3.4.3 Integrating LLaMA2-7B . . . 25
3.5 Multimodal RAG . . . 26
3.5.1 Embedding and Retrieval . . . 26
3.5.2 Integrating LLaVA-7B . . . 26
3.6 Evaluation . . . 27

4 Experiments **29**
4.1 Quantitative analysis . . . 30
4.2 Qualitative analysis . . . 30

5 Results **33**
5.1 Quantitative analysis . . . 33
5.1.1 Evaluating chunk sizes . . . 33
5.1.1.1 BERTScore for Text-only RAG . . . 33
5.1.1.2 BERTScore for Multimodal RAG . . . 34
5.1.1.3 LangSmith for Text-only RAG . . . 35
5.1.1.4 LangSmith for Multimodal RAG . . . 36
5.1.2 Evaluating top-k values . . . 38
5.1.2.1 BERTScore for Text-only RAG . . . 38
5.1.2.2 BERTScore for Multimodal RAG . . . 39
5.1.2.3 LangSmith for Text-only RAG . . . 40
5.1.2.4 LangSmith for Multimodal RAG . . . 41
5.2 Qualitative analysis . . . 42
5.2.1 Manual 1 . . . 42
5.2.2 Manual 2 . . . 43
5.2.3 Manual 3 . . . 45

6 Discussion **47**
6.1 Developing domain-specific RAG frameworks . . . 47
6.2 Parameter and modality impact on RAG framework performance . . . 47
6.2.1 Key evaluation metrics for the domain-specific RAG framework . . . 48
6.2.2 Effects of modality . . . 49

6.2.3 Effects of parameters 50
6.2.3.1 Chunk sizes 51
6.2.3.2 Top-k values 52
6.2.3.3 General observations 52
6.3 Qualitative analysis overview 53
6.4 Future work 54
6.5 Risk analysis and ethical considerations 56

7 Conclusion **57**

Bibliography **59**

A Appendix 1 **I**

List of Figures

1.1 General structure of RAG integrated with an LMM. 2

2.1 Text-only RAG structure. 8
2.2 Multimodal RAG structure. 9

3.1 Simplified chunking process schema. 23

5.1 Plot of BERTScore for Text-only RAG for different chunk sizes. . . 34
5.2 Plot of BERTScore for Multimodal RAG for different chunk sizes. . 35
5.3 Plot of LangSmith scores for Text-only RAG for different chunk sizes. 36
5.4 Plot of LangSmith scores for Multimodal RAG for different chunk sizes. 37
5.5 Plot of BERTScores for Text-Only RAG for different top-k values. . 38
5.6 Plot of BERTScores for Multimodal RAG for different top-k values. 39
5.7 Plot of LangSmith scores for Text-only RAG for different top-k values. 40
5.8 Plot of LangSmith scores for Multimodal RAG for different top-k values. 41

A.1 Generated responses by LLaVA-7B, the Text-only RAG and the Multimodal RAG framework for questions about the Dell manual. . II
A.2 Generated responses by LLaVA-7B, the Text-only RAG and the Multimodal RAG framework for questions about the Samsung manual. III
A.3 Generated responses by LLaVA-7B, the Text-only RAG and the Multimodal RAG framework for questions about the Sony manual. IV
A.4 An example page from one of the manuals chosen for evaluation. . . V

List of Tables

5.1 Table of BERTScore for Text-only RAG for different chunk sizes. . 33
5.2 Table of BERTScore for Multimodal RAG for different chunk sizes. 34
5.3 Table of LangSmith scores for Text-only RAG for different chunk sizes. 35
5.4 Table of LangSmith scores for Multimodal RAG for different chunk sizes. 37
5.5 Table of BERTScores for Text-only RAG for different top-k values. 38
5.6 Table of BERTScores for Multimodal RAG for different top-k values. 39
5.7 Table of LangSmith scores for Text-only RAG for different top-k values. 40
5.8 Table of LangSmith scores for Multimodal RAG for different top-k values. 41

6.1 Comparison of the scores achieved by the best-performing configurations of the Text-only and Multimodal RAG frameworks. 49

List of Abbreviations

QA	Question Answering
RAG	Retrieval-Augmented Generation
LLM	Large Language Model
LMM	Large Multimodal Model
AI	Artificial Intelligence
LLaVA	Large Language-and-Vision Assistant
LLaMA	Large Language Model Meta AI
NLP	Natural Language Processing
CLIP	Contrastive Language-Image Pre-Training
BERT	Bidirectional Encoder Representations from Transformers
OCR	Optical Character Recognition

1

Introduction

In manufacturing, the successful execution of product assembly processes relies on standardized instructions that guide line workers through the procedures. Verification of assembly processes is usually purely manual, which is considered to be time-consuming and prone to errors, calling for more efficient verification methods. Despite the critical role of assembly in industrial settings, utilizing machine learning algorithms to answer questions related to assembly verification remains under-explored compared to open-domain context-based Question Answering (QA) [1]. One reason behind this is the lack of standard benchmark datasets for assembly verification QA. Despite limited progress, the interest in automating human performance in everyday tasks such as verifying assembly processes is rapidly growing. As such, the ability to derive meaningful insights from multimodal data, i.e. a combination of visual and textual information, is a crucial skill for machines to achieve it [2], [3].

Recognizing the need for automated assembly verification methods, an emerging method called Retrieval-Augmented Generation (RAG) can address the challenge. It combines information retrieval with a generative component, either a general-purpose Large Language Model (LLM) or a Large Multimodal Model (LMM). The retriever component extracts information from an external data source, while the generative model produces a text response based on the retrieved context. A simplified structure of a RAG framework is presented in Figure 1.1.

In this project, the aim is to build and explore RAG frameworks capable of analyzing the content of different modalities within electronics assembly manuals. Further, the aim is to integrate such a framework with a generative model to address worker queries effectively. Our project is a part of the long-term research at *Wiretronic AB* for designing a VQA technology where the workers can interact with a machine to receive guided instructions, instead of manually referring to the written instructions. The company has previously been doing research on technologies for extracting visual information using segmentation networks. Building upon this foundation, our investigation aims to extend the modality of data by

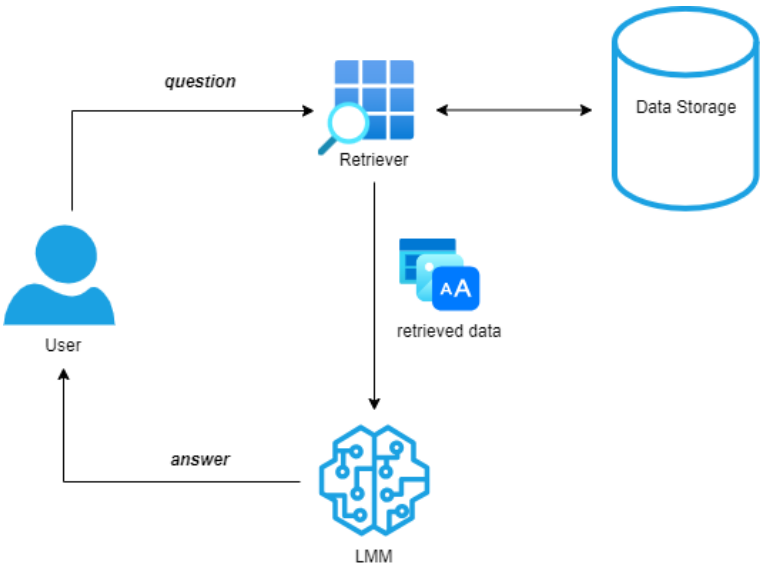


Figure 1.1: General structure of RAG integrated with an LMM.

incorporating text. In particular, it is intended to explore if using context from manuals, retrieved by a RAG framework as an input prompt to an open-source LLM or LMM, can generate accurate answers to workers' queries. By leveraging LLMs and LMMs, the aim is to explore new possibilities for the interaction of line workers with machines for future assembly processes to be verified and guided with high precision and minimal human supervision.

Driven by the growing interest in multimodality, which consists of image and text data, the VQA field emerged recently within the Artificial Intelligence (AI) community. Its goal is, by providing a natural language question about an image, to generate an accurate natural language answer to the image [4]. Currently, among the open-source models developed in this field, Large Language-and-Vision Assistant (LLaVA) [5] stands out as one of the most successful models [6]. Existing VQA tasks are designed to answer questions from web pages or infographics [7], which is not suitable for user manual VQA. They mainly consider single-page documents while product user manuals are composed of multiple pages that should be processed altogether. This research gap needs to be filled [8].

In this project, information retrieval plays a pivotal role as the framework aims to reason over multi-page documents. The framework aims to understand the text, layout, and visual elements of each page to extract regions relevant to a query that would serve as a prompt to an LLM or an LMM. The application of prompting with extracted information remains under-explored for LLMs and LMMs [9], which underscores the importance of exploring RAG frameworks to

enhance the accuracy of these models in processing different data modalities in this domain-specific application.

1.1 Aim of the project

This project aims to build and explore two RAG frameworks, with capabilities of interpreting different modalities, and integrating them with either Large Language Model Meta AI (LLaMA) [10] or LLaVA. The first RAG framework, called the Text-only RAG framework, can interpret and embed text data. It has an additional generative component that transforms all image data into text descriptions. The other RAG framework, called the Multimodal RAG framework, can interpret and embed both text and image data. Hence, the two RAG frameworks are built according to different approaches on how to process both text and image data from user assembly manuals.

Further, this project aims to evaluate how accurately the two frameworks can generate answers based on the retrieved information from the assembly manuals. This goal requires finding the most optimal strategy to generate answers to a worker's query about the assembly process of wire harnesses. It is achieved by a strategic optimization of the frameworks through testing of different parameters. Two parameters are explored, the chunk size of the retrieved text data and the top-k number of retrieved documents. Evaluating different values of these parameters allows for optimizing the performances of the RAG frameworks.

The project outline consists of the following steps:

- Build and explore different modalities of retrieval models that extract texts and images relevant to the query from PDF files to prompt LLaMA2-7B or LLaVA-7B.
- Optimize the RAG framework to find the most efficient strategy to generate answers to users' queries about assembly manuals.
- Evaluate and compare different configurations of the frameworks.

1.2 Research questions

Given the aim of this project, the following research questions are formulated:

1. Is it possible to integrate a pre-trained LLM or LMM with a retrieval component in a RAG framework to generate responses to domain-specific questions?

2. How do the modality and the parameters of a RAG framework affect the performance of generated responses?

Building upon the foundation of prior research in the field of VQA, this project aims to extend the exploration of the effectiveness of VQA models in the specific context of electronics assembly processes. We use insights drawn from the aforementioned studies to incorporate textual instructions, visual information, and user queries, to develop two retrieval mechanisms and integrate them with an LLM or an LMM into RAG frameworks.

1.3 Motivations

While there exists a lot of research for the application of VQA models in various domains such as biomedical imaging, the adaptation of these models in the field of manufacturing, specifically assembly processes, remains under-explored. Developing a RAG framework for this specific domain and integrating it with an LLM or LMM will open opportunities to apply the same kind of structure to various types of domains. Working towards automation is crucial since it will minimize human error, increase safety, and support the decision-making process. Automation reduces manufacturing costs and shortens the production cycle.

Furthermore, RAG frameworks have been shown to produce the same level of enhanced accuracy of VQA answers as fine-tuning the LLM into the specific domain. Since fine-tuning comes with high computational costs, utilizing a RAG framework is a more efficient and flexible solution for domain-specific tasks. By using domain-specific data, the framework adapts to the domain and extracts the relevant information for the user manuals with less required computational resources [11].

The insights gained from this project can extend beyond the electronics manufacturing domain, influencing the broader landscape of Natural Language Processing (NLP) and Computer Vision, by proposing more interactive and adaptable systems.

From Wiretronic's point of view, the motivation is to gain more knowledge about how multimodal models can be used as a tool for manufacturing processes. The long-term goal, which is beyond the scope of this project, is to be able to implement an automated model that can be used as an aid tool for manufacturing workers to verify the assembly process. This project is the starting point of this long-term goal, with expectations to gain insights that can be further elaborated on in the future.

1.4 Challenges

Several challenges were encountered during the development of the RAG frameworks. Due to limited access to manuals that naturally consisted of both text and image data, the implementation of extracting these data required additional effort. Every page of each manual was in image format, which included all texts and images being in image format. Instead of extracting text paragraphs from the manuals as text data, they first had to be converted to a natural text format. This resulted in extra steps in the implementation of the extraction stage.

Another encountered challenge was the running time of the entire procedure, from extracting the text and image data from the manuals to the final evaluation. Due to the long running times, not all manuals could be included. The most optimal option would be to use all 209 manuals present in the dataset to get the most accurate evaluation scores. However, the time limit of the project only allowed for using a subset of 10 manuals for the evaluation.

2

Background & Related work

2.1 Background

Due to the amount of computational resources required to fine-tune an LLM or an LMM, it becomes harder to adapt and utilize it for specific domains. RAG frameworks work as a solution to this issue, making it possible to retrieve, use, and incorporate data from external documents in the process of utilizing a generative model for domain-specific tasks. The RAG framework does not rely on the QA model being fine-tuned into the domain nor requires that it is trained on up-to-date data [12].

2.1.1 The RAG framework

The very first stage before any execution of the RAG framework can take place, is preprocessing the input documents properly. The documents need to be split into representations of the different data types, usually images, texts and tables. The extraction stage consists of specifying which data can later be retrieved by the model and how it should be retrieved. Important factors to consider when deciding how the data should be extracted is the size of the chunks that the text should be split into. The size can impact to what extent the retriever is capable of capturing context. To avoid information loss, the chunks should have a sufficient size. A second factor to consider is how tables should be represented in the extracted data, and if they should be a part of the text chunks or be represented separately from the text data. This should be decided given the domain and the context, and if table data are crucial for the specific purpose. Another aspect that needs to be carefully considered, is how images and specifically text in image format should be represented. For some tasks, keeping text in image format is a better choice, which for example can be an image of a chat. In other cases, the different text messages in the image of the chat are necessary for capturing the context and need to be extracted as text data [13].

After the data has been extracted from the documents as separate texts, alongside table and image representations, the embedding takes place. The data is encoded and mapped into vector representations in a single embedding space. The user's query inputted to the framework is embedded with the same method, and a similarity search between the query and the embedded data is performed. The chunks that get the highest similarity score are then retrieved. Here, an important factor that needs to be decided is the number of documents that are going to be retrieved, called the top-k value. It can affect the generation process by providing either adequate, not sufficient, or noisy context to the prompt. The retrieved parts are augmented with the user query into the prompt. The augmentation is performed to enhance the context of the input that is later prompted to the LLM or LMM. Optimizing the input will improve the models' capability to generate an accurate final response [13].

There are several approaches on how to handle multimodal data in documents that will be retrieved by the RAG framework. Depending on the approach, the structure of the RAG framework will differ between text-only and multimodal.

2.1.1.1 Text-only RAG

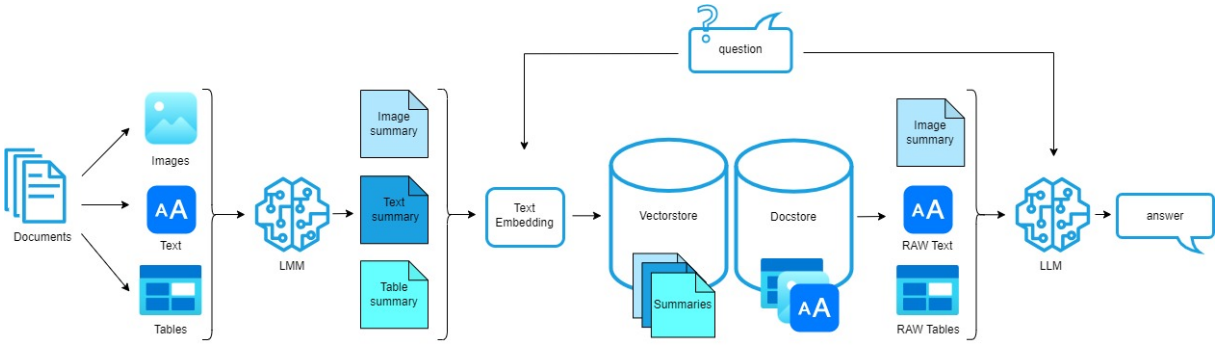


Figure 2.1: Text-only RAG structure.

One approach is based on narrowing down the multimodal data into representations of a text-only format. First, the extraction is performed where text, tables, and images are saved separately. Before the embedding is performed, the images and tables are passed into an LMM. The LMM prompted with an image and a question regarding the image, generates a response to the question in text format only. Extracted tables and images are prompted into the LMM together with a prompt that tells the model to make explicit summaries of the content. After summarizing the content of images and tables, all of the information from the original document is represented in text format and the embedding is performed. The textual data is embedded into vectors. Due to the data being transformed

into a single modality, a text embedding model can be used to create text feature vectors for the whole document and an LLM can be utilized to produce the final The data flow structure in the Text-only RAG framework is illustrated in Figure 2.1.

2.1.1.2 Multimodal RAG

Another approach on how to treat multimodal data in documents that later will be retrieved by the RAG framework is to extract and keep the raw images and tables. In this case, a multimodal embedding model is used which makes it possible to embed text, image, and table data into the same vector space. Due to the extracted data having different modalities, an LMM has to be utilized to produce the final answer. This process is illustrated in Figure 2.2.

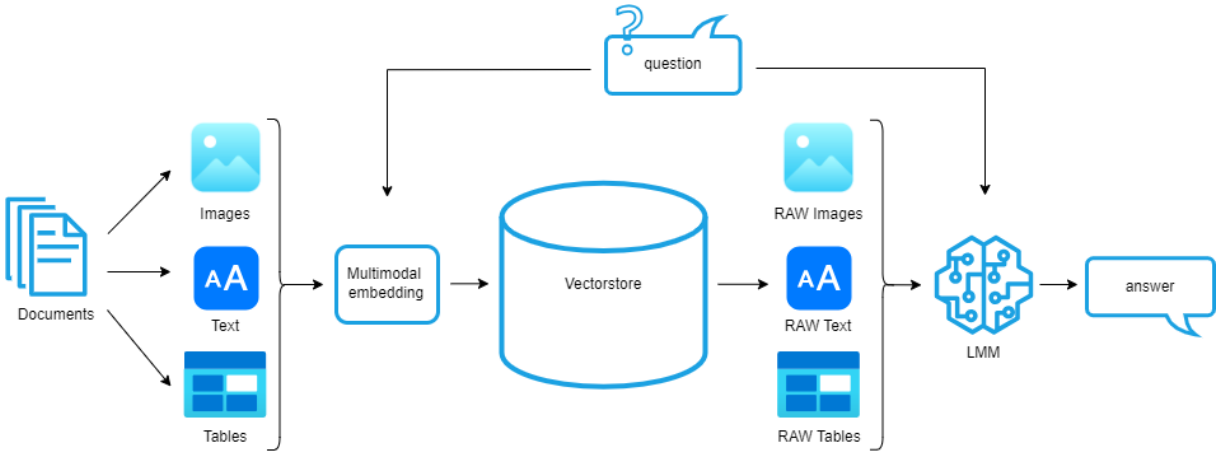


Figure 2.2: Multimodal RAG structure.

The two approaches have certain advantages and drawbacks. The Text-only RAG framework that uses a text embedding model risks losing context that can be crucial for generating an accurate and precise response. This drawback can appear because it retrieves the summary of the information incorporated in the images and not the raw images themselves. The Multimodal RAG framework, that uses a multimodal embedding model instead, allows for capturing all the raw content from the document and minimizes the risk of losing crucial context. However, the multimodal approach comes with challenges as the complexity grows when the modalities are many. Extracting and embedding different types of modalities accurately is more complicated [14].

2.1.2 RAG vs Fine-tuning

Fine-tuning an LLM or LMM comes with a lot of advantages and can make the model adapt to a new domain and give a state-of-the-art performance in generating answers to domain-specific questions. In addition to the advantage of very accurate responses, the required size of input and output tokens remains the same and does not require more computational resources. LLMs are trained on huge amounts of data and typically contain billions of parameters. Fine-tuning includes extending the training of the model by adding more data of the desired domain, allowing the model to update its parameters accordingly. This process involves traversing the LLM through all of its parameters. Due to the huge size and parameter amount of many recently developed LLMs and LMMs, this process can become increasingly computationally expensive and time-consuming. Depending on the task and goal of adapting an LLM or LMM into a specific domain, the advantages and disadvantages should be carefully considered [11], [12].

Due to the limitations of fine-tuning a QA model, the developments of RAG frameworks have become more of interest. The main advantage of using RAG is the small amount of needed computational resources in comparison to fine-tuning. As the LLM or LMM is treated as a black box when using RAG, its parameters are not defined or updated. This leads to a drastically decreased initial cost, as instead of fine-tuning the corresponding process is to create the embeddings, making the process more flexible [12]. The flexibility of using a RAG framework also lies in being able to decide, change, and add data that should be included without making the process more complex. Further, both of the methods have been shown to be able to produce similar improvements in the overall performance. RAG frameworks are also known to be effective in tasks where the data is contextually relevant [11].

2.1.3 Optimizing parameters

One of the most important parts of the RAG framework is the construction of the retrieval vector store and the retriever itself [15]. In the RAG framework, the components of the retriever significantly impact the overall performance by defining how effectively the framework retrieves and utilizes relevant information from the vector store. The key parameters of the retriever are chunk size and top-k value. Although chunking itself is part of the data extraction phase, while top-k chunks retrieval happens at the end of the retrieval phase, they both influence the behavior and the performance of the retriever component directly. The impact of these two parameters is described in this section.

2.1.3.1 Effects of chunk size

RAG frameworks are sensitive to the chunking method chosen to split data into smaller units, which are stored in a vector store. Chunking means breaking down large inputted documents into smaller segments of fixed length called chunk size. Each chunk should keep specific information essential for addressing user queries. A good chunking strategy is crucial to ensure high relevance and accuracy of the retrieved context [15], [16]. Hence, chunking tries to retrieve the context with minimal noise while maintaining semantic relevance [17].

The size of the chunks determines the breadth of the context retrieved by the retriever, which makes it a critical parameter in the retrieval phase. When the chunk size is not adapted to the task, there is a risk that too little or too much information is going to be included in the context for the given query. While smaller chunk sizes might speed up the retrieval phase, they can include incomplete context that is unable to provide enough necessary information for the generation process. For example, chunk sizes like 128 or 256, can capture finer semantic details but can miss some critical information. On the other hand, too big chunk sizes, like the ones containing 512 or 1024 tokens, can preserve more extensive context, but it comes with a risk of containing too much irrelevant, not precise information that can slow down and confuse the generation process [17]. The ideal chunk size should maintain the high accuracy of the generated answer while preserving all the necessary information in the context. To find it, it is recommended to run empirical experiments with various chunk sizes. Besides, the optimal chunk size should also be chosen based on the nature of the documents stored in the vector store, the length of the user query, and specific requirements of the task [16]. In this project, the documents that are used are electronics manuals, which are long, but contain a lot of concise information. The challenge with this data type is finding a balance between granularity and comprehensiveness. Ideally, the optimal chunk size would capture the essence of each step described in the manual while also maintaining an overview of how it contributes to the larger assembly process.

2.1.3.2 Effects of top-k values

The top-k value is another critical parameter of the retrieval phase in the RAG framework. It is the number of text chunks retrieved for each query, which is why it determines the capacity and quality of the retrieved context. The retriever finds the top-k vectors most similar to the query vector in the vector store and uses these to retrieve the respective text chunks. These text chunks are then used to prompt the LLM or LMM, hence, the size of the chunks matters and may affect the final generated result [15].

The amount of information that the model receives in a prompt is dependent on the size of k. The ultimate goal is to retrieve information that is both comprehensive and relevant. When the top-k value is too small, it can lead to the problem of information scarcity. Then, the essential data from the vector store will not be included in the prompt, causing generation of incomplete or inaccurate answers. On the other hand, when the top-k value is too high, it becomes harder to recognize the relevant chunks and can lead to less accurate or incoherent responses [15]. A too-high top-k value also creates a risk of retrieving irrelevant chunks, which may introduce noise and lower the quality of generated responses. In addition, it is usually more computationally expensive and time-consuming to process a larger number of chunks. Hence, finding the optimal top-k value is very important to build an efficient and accurate RAG framework. There is a need to find a balance between providing all the needed context without causing information overload, as it exists for choosing the optimal chunk size. This can be achieved through empirical testing.

Finding top-k documents can be even further optimized by incorporating a re-ranker or a dynamic top-k retrieval. Most vector stores use vector similarity search criteria to search through vectors. However, computing this similarity score between document chunks and the prompt does not always return relevant contexts [18], [19]. In that case, a re-ranker model is a beneficial addition, as it reevaluates the top-k chunks based on criteria other than vector similarity — for example keyword search [18], or models such as cross-encoders. Integrating a re-ranker into RAG frameworks makes the retrieved context smaller and more relevant to the query. On the other hand, a RAG system enhanced with a re-ranker uses more computational resources than a basic vector-similarity-based RAG system [19].

Moreover, in several cases, depending on the complexity of the question, a different number of top-k chunks should be retrieved. In this scenario, dynamic top-k retrieval is used in contrast to static top-k retrieval. It adapts the number of retrieved chunks to the complexity of each query. This can be done by training a cross-encoder to predict the most suitable top-k value for each retrieval task. Dynamic top-k retrieval ensures high relevance and an optimal amount of retrieved information, simultaneously reducing computational costs by omitting the processing of unnecessary information [20]. However, such an approach is only suitable for tasks where the questions have significantly different levels of complexity.

2.1.4 Generative Large Language and Vision models

The integration of an LLM or an LMM within the RAG framework serves as the final step of the framework, delivering the textual output. These models rely on transformer architecture and self-supervised learning to generate human-like text.

They are pre-trained on extensive text corpora and have a deep understanding of natural language, text coherence, and contextual relevance [21]. However, they encounter challenges when the situation requires an understanding of specific information from the external data source. When handling domain-specific or highly specialized queries [22], it is common that they can generate incorrect information, referred to as hallucinations [23]. These limitations emphasize that LLMs or LLMs should not be implemented as solutions in real-world manufacturing environments without additional safeguards [13]. They also cannot learn and retain new information without undergoing a retraining process, which is computationally and time intensive. Therefore, integrating them into the RAG framework to produce accurate and relevant responses is valued. This integration combines the comprehensive internal knowledge from language models with external data retrieval. It can also enhance the models' ability to provide accurate and precise responses.

LLaVA and LLaMA are one of the question-answering models that can be utilized as generative models in the RAG framework, depending on the modality of the information retrieved by the retriever. LLaMA is leveraged to produce summaries of text paragraphs, while LLaVA is used to provide text summaries of images present in the documents. When the output of the retriever is only in text format, LLaMA is used to produce the final answer to the user query. In cases where the output is multimodal, LLaVA is used instead. LLaVA and LLaMA can be run locally with Ollama, a local inference framework client. The local execution that this framework provides, ensures data privacy, as the information is not externally shared anywhere.

2.1.4.1 LLaMA

LLaMA is a foundational large language model that works only with text modality by taking a sequence of words as input and recursively generating text. It is based on a transformer architecture with implementations of optimizers and Causal Multi-Head attention to improve performance. LLaMA models are available in several sizes between 7-65B parameters and they can reach similar or even better results on several benchmarks as ground-breaking larger models when enough data is used for training. Smaller LLaMA models, trained on more tokens, are also easier to adapt to specific use cases [10].

2.1.4.2 LLaVA

LLaVA is built on the foundation of LLaMA, combined with an image encoder and a text decoder, allowing it to integrate a visual and a textual embedding space [5]. The Contrastive Language-Image Pre-Training (CLIP) model [24], based on contrastive language-image pre-training, is used as the image encoder that

converts the image into the same vectors of number matrices as text. It does so by connecting the visual features from input images to language embeddings through a trainable projection matrix. These visual tokens, possessing the same dimensionality of the word embedding space as the language tokens, are integrated with the user text prompt. Then, the LLM component of LLaVA generates the final text response. LLaVA has shown to be able to adapt and produce state-of-the-art performance within various domains, such as the challenging Science QA benchmark [25].

2.1.5 Hallucination tendencies

As for many LLMs and LMMs, LLaMA and LLaVA tend to hallucinate when generating a response. A hallucination is a made-up answer to a question, that typically comes off as being true. The response is written as factual information, which can make it hard to detect a hallucination. Because of this, it is important to be cautious when interacting with an LLM or LMM about subjects that are outside the user's expertise [26].

Several factors have been shown to trigger hallucinations for LLaMA. In the study *LLM Lies: Hallucinations are not Bugs, but Features as Adversarial Examples* by Yao et al., it is shown that the format of the prompt can have an impact on the extent to how often a hallucination is triggered or not. Two kinds of modifications of a typical prompt were tested to see how they would affect the outcome. The first modification kept the semantic context of the prompt but had a few tokens changed to random tokens. The second modification included randomizing the initial tokens of the prompt, leading to a non-specified semantic context. The two different formats showed that hallucinations were triggered by a rate of 54% and 31% respectively. These results indicate that careful prompt engineering is an important factor to avoid hallucinations [26].

For LLaVA, which uses the multimodal embedding model CLIP, there are other challenges to consider to avoid hallucinations. Since CLIP is able to encode both textual and visual data, there is a risk that there becomes an information gap between the two types of data. The gap can lead to an increased risk of triggering hallucinations. Hence, the part of the model that aligns the two data modalities needs to keep a high level to minimize the potential impact of this issue. Another aspect that has been shown to trigger hallucinations for LLaVA is the resolution of the images that are passed to and encoded in CLIP. A lower image resolution has been proven to be a factor that triggers hallucinations. This issue is probably caused by lacking visual information in low-resolution images [27].

2.1.6 Evaluation

The evaluation of generative tasks in machine learning poses specific challenges, which are different from what is known for traditional classification or regression tasks [28]. During the evaluation of a RAG framework two key stages – retrieval and generation phase, should be assessed separately. When evaluating the retrieval quality, the relevance of the retrieved documents to the user query is calculated. The generator's assessment tests how coherent and relevant is the answer produced from the retrieved context. By assessing these stages separately, the quality of the retrieved context and the accuracy of produced content are both examined.

The issue with traditional quantitative metrics like BLEU or METEOR is that they often fall short when capturing the domain-specific effectiveness of RAG models [29]. These N-gram metrics do not account for word order or semantic variations. One metric that addresses these shortcomings is BERTScore. It is an evaluation metric based on Bidirectional Encoder Representations from Transformers (BERT) embeddings. It measures how similar the generated response is to the ground truth answer. However, since there are various ways to sufficiently answer a query in written language, the performance measurement still relies on subjective judgment. Therefore, to supplement the BERTScore output with more 'human-like' judgment, LLMs or LMMs can be utilized to assess the generated answers. Researchers have named the approach of utilizing an LLM to evaluate responses of LLM-based RAG framework as the "LLM-As-A-Judge" [30] approach. In the case of Multimodal RAG, a judging model capable of considering both textual and visual context is required, therefore an LMM needs to be utilized.

Retrieval quality can be evaluated most fundamentally by calculating page-level and paragraph-level accuracy [16]. It involves comparing the manually selected ground truth section from the pages of the document with the chunks returned by the retrieval algorithm. When the reference and the retrieved context are located on the same page or paragraph, the page-level or paragraph-level accuracy will be high. Since there are no manually selected ground truth regions of text in the dataset used in this project, only ground truth answers, it is needed to employ the LLM-As-A-Judge approach.

2.1.6.1 BERTScore

To evaluate the semantic similarity between the generated response and the ground-truth answer, BERTScore is used. It employs pre-trained BERT contextual embeddings for both the generated and reference answers. BERT contextual embeddings, unlike regular ones, can produce different vector representations for a given word in different sentences, depending on the surrounding words that establish

the context of the target word [31]. Each word's representation is calculated using a Transformer encoder, which iteratively employs self-attention and nonlinear transformations. Then, the pairwise cosine similarity between each token x_i in the reference sentence and each token $\hat{x_j}$ in the candidate sentence is calculated. The cosine similarity of these two non-null vectors is calculated as:

$$\frac{{x_i}^T \hat{x_j}}{||x_i||||\hat{x_j}||} \tag{2.1}$$

Since pre-normalized vectors are used, the similarity is reduced to the dot product:

$${x_i}^T \hat{x_j} \tag{2.2}$$

The complete BERTScore consists of precision, recall, and F1 metrics. Calculating recall involves matching each token in reference x with a token in candidate $\hat{x}$ while calculating precision involves matching each token in $\hat{x}$ with each token in x. Then, greedy matching is used to maximize the similarity score. The F1 score is calculated by combining precision and recall [31]. Below, the equations for calculating recall, precision, and F1 score are shown:

$$R_{BERT} = \frac{1}{|\hat{x}|} \sum_{x_i \in x} max_{\hat{x_j} \in \hat{x}} {x_i}^T \hat{x_j} \tag{2.3}$$

$$P_{BERT} = \frac{1}{|\hat{x}|} \sum_{\hat{x_j} \in \hat{x}} max_{x_i \in x} {x_i}^T \hat{x_j} \tag{2.4}$$

$$F_{BERT} = 2\frac{P_{BERT} \cdot R_{BERT}}{P_{BERT} + R_{BERT}} \tag{2.5}$$

The final step of calculating the BERTScore involves re-scaling the output values to make them more human-readable. Since the cosine similarity values lie in a very limited range between [-1, 1] interval, BERTScore is re-scaled linearly, as follows:

$$\hat{R}_{BERT} = \frac{R_{BERT} - b}{1 - b} \tag{2.6}$$

After re-scaling, $\hat{R}_{BERT}$ falls typically between 0 and 1, and the same procedure is applied for P_{BERT} and F_{BERT}. The constant b is derived from averaging BERTScores calculated on randomly paired candidate-reference sentences from Common Crawl monolingual datasets.

BERTScore allows for the evaluation to be more precise than evaluation metrics that make use of N-gram methods, such as BLUE and METEOR. As discussed, N-gram-based metrics come with several drawbacks. For instance, the BLEU score [32] simply only assesses the N-gram overlap between the candidate and the reference. One drawback of such an approach is the inability to capture dependencies that may be located far from each other in a text. In contrast, BERTScore utilizes the aforementioned pre-trained contextual embeddings, which capture the context of words and can recognize the order and distant dependencies in the text. Also, N-gram methods do not usually perform well on texts that are rewritten with the use of synonyms or summaries of an original text. BERTScore can detect paraphrases in comparison to N-gram-based metrics which assign low scores to semantically correct sentences that rather deviate from original sentences, only assigning high scores to similar tokens. In BERTScore, on the other hand, computing the sum of the cosine similarities between token embeddings allows for detecting paraphrases. By overcoming the shortcomings of N-gram-based metrics, BERTScore has shown to be a more reliable evaluation metric. It has been also proven to correlate with human judgments, which is an important indicator when evaluating text generation tasks [31].

2.1.6.2 LangSmith

LangSmith is a platform provided by LangChain, that allows users to track, evaluate, and monitor ongoing processes that are powered by LLMs and LMMs. It provides real-time monitoring of the models epochs and uses traces to log almost every aspect of each run. It is possible to view and get statistics of these results with the available logging and visualization components.

Additionally, the LangSmith API offers several built-in metrics that follow the LLM-As-A-Judge method for in-depth evaluation. They are a valuable tool to support traditional evaluation methods when dealing with generated content that has complex language nuances and requires contextual understating. One downside is that they return a binary score for each data point. Therefore, to accurately measure differences in prompt or model performance, it is most effective to aggregate results across a larger dataset [33].

In this project, the chosen metrics for evaluating the Text-only and Multimodal RAG frameworks are contextual accuracy with chain of thought (COT) reasoning, coherence, and relevance. Contextual accuracy is a standard metric that measures the correctness of a generated response to a user query. Coherence and relevance are both part of the Labeled Criteria metrics from LangSmith, which ask an LLM in the prompt to provide reasoning behind assigning a label for a prompted criteria.

The following metrics specifically operate by:

- **Contextual accuracy** works by instructing an LLM to grade a response as "correct" or "incorrect" based on the ground truth answer. It is enhanced by the chain of thought reasoning providing examples of a logical progression of thoughts before determining a final verdict. This approach helps to better align the responses with human judgment.
- **Coherence** tests how well the response is structured sequentially and logically. The criteria prompted to an LLM along with the generated and ground truth answer in this evaluator is:

  "Is the submission coherent, well-structured, and organized?"

  The answer that is labeled 1 should be organized and easy to read. It should consist of text that addresses the topic at hand.
- **Relevance** measures how well the generated response matches the question. The criteria prompted to an LLM along with the generated and ground truth answer in this evaluator is:

  "Is the submission referring to a real quote from the text?"

  The answer that is labeled 1 should be genuinely relevant to the user query posed, once compared with the query.

2.2 Related work

In this section, previous studies with findings relevant for the goal of this project are presented. The studies are divided into sections related to the manufacturing setting of the project and RAG frameworks.

2.2.1 Manufacturing setting

Previous research in the field has shown that it is possible to develop a VQA model that increases the quality and production efficiency of human-technology manufacturing processes. In the study *Digital twin improved via VQA for vision-language interactive mode in human-machine collaboration* by Wang et al. (2021), a VQA model is developed to give responses to different kinds of questions regarding the manufacturing process. The core of this model is based on Computer Vision and NLP and can generate a response to either open-ended questions or multiple-choice questions. A Convolution Neural Network (CNN) is used as a first step in the model and contributes to understanding the visual input. The second

step of the implementation is a Long Short-Term Memory (LSTM) network which contributes to text processing and language understanding. Finally, a fusion between the visual and textual features takes place to prepare for the decoding of the generated answers. The authors of the paper concluded that the VQA model manages to answer both open-ended and multiple-choice questions which results in it being able to identify certain problems and challenges during the manufacturing process [34].

Zhang et al. [7] have recently developed a framework called Multimodal Product Manual Question Answering (MPMQA), which interacts with product manuals to retrieve a relevant part as an answer to a user's query. Unlike most of the existing models that leverage only textual information [1], MPQMA requires the model to comprehend both the visual and the textual contents. Given a textual question and a multipage digital user manual, MPMQA provides a multimodal answer for a given question. To support this task, a large-scale, diverse dataset called PM209 with human annotations was created. It consists of 22,021 QA pairs from user manuals of electronic brands. MPMQA addresses two stages – page retrieval and multimodal QA. The model employed for this task is the Unified Retrieval and Question Answering (URA) model, which consists of a URA Encoder, URA Decoder, and Region Selector. In the page retrieval stage, the model encodes questions and pages separately and calculates their relevant scores with token-level interaction. In the multimodal QA stage, the model encodes questions and pages jointly and produces the textual parts and visual parts of the multimodal answer through the Decoder and Region Selector. Finally, URA is optimized in a multitask learning manner. It achieves competitive results compared to multiple task-specific models and proves successful in both information retrieval and multimodal QA tasks.

2.2.2 RAG

As of the very first development of the RAG framework, the core idea was to bridge the field of generative AI with retrieval-based systems to enhance overall performance. Since then, a lot of prominent developments have been made in the field, contributing to technical improvements and a broader range of application areas [35].

As one of the recent developments in the field of RAG, Tang et al. presents the benchmark dataset *MultiHop-RAG* which contains queries and ground-truth answers that are located across multiple documents. In the report, it is presented how well some of the most prominent embedding models and generative models, such as GPT-4 and LLaMA2-70B, perform on these types of queries. The core idea is to measure the extent to how well the models can retrieve and generate responses

based on information located across multiple documents. What is shown in the study, is that the models do not manage to perform as well as expected on multi-hop queries as on queries whose answers can be found in a single document. When utilizing RAG frameworks for real-world applications, the assumption that the response to a query can be retrieved from multiple sources should be addressed for optimizing accurate results as well as for ethical reasons. Bringing the MultiHop-RAG benchmark, the authors address where the current RAG framework contains some flaws and that there is room for further development [36].

Another recent framework, developed with the purpose of further advancing the usage of RAG, is *Retrieval-Augmented Planning* (RAP). This framework is based on the core structure of the RAG framework and contains a memory. The memory allows the model to draw on past experiences, which will be retrieved and utilized for generating a response to the current query. Allowing an LLM to base its responses on both provided context and on context from past experiences, it advances the capability of planning and decision-making. These advancements can be utilized to guide the user, based on the query, through a set of substeps to reach the goal. The RAP framework has been shown to produce a state-of-the-art performance when integrated with text-only modality LLMs. Integrating with an LMM, the framework improves the performance slightly, but there is room for improvements [21].

3

Methodology

This chapter describes the methods used to build and evaluate the Text-only and the Multimodal RAG frameworks. First, sections 3.1 to 3.3 give an overview of the data that is used, detailing how they are extracted and processed from raw PDF files into a format suitable for input to the RAG frameworks. Next, the consecutive steps of RAG frameworks' implementations are described separately for the two frameworks. Section 3.4 focuses on the Text-only RAG, while section 3.5 addresses the Multimodal RAG. Both sections consist of subsections describing first the embedding and retrieval processes followed by the integration of an LLM or LMM model. For the Text-only RAG, an additional subsection describes the process of generating summaries, which is unique for this framework. At the end, a brief overview of the evaluation methodology is presented in section 3.6. A detailed explanation of specific experiments is included in the next chapter Experiments.

3.1 Data

The baseline dataset that is used in this project is the **PM209 open-source dataset** containing digital product manuals from well-known consumer electronic brands. The dataset is chosen since the structure, length, and nature of data in the manuals resemble the characteristics of Wiretronic's assembly manuals. The dataset consists of 22,021 QA pairs from 209 product manuals among 27 consumer electronic brands. Each question is in a text format and has a corresponding multimodal answer that includes text and related visual regions from the manuals. The dataset is diverse, with the manuals being 10 to 500 pages long and covering various subjects from more than 90 different product categories. The question-answer pairs were designed to emphasize the multimodal content in product manuals and to support the VQA task.

Due to the computational resources and time limitations, a subset of 10 manuals, that resemble Wiretronic's assembly manuals the most, will be used. An example page of one of the used manuals can be seen in A.4.

3.2 Data extraction

The first step of building the RAG frameworks is to choose the method that extracts raw data from the documents. Given the domain-specific application of this project, all of the data formats including text blocks, images, and tables, are relevant for the context. The extraction phase is similar across both of the RAG frameworks, where different techniques are used for image extraction and text extraction.

To extract images from the documents, the module **partition_pdf** is used. This module belongs to the Unstructured library which specializes in processing raw and unstructured data in documents. Based on the task, the document data can be treated and grouped by different chunking strategies, where in this case it groups the data based on a 'by_title' strategy. All of the images are extracted by this strategy and saved to an output directory. To extract the text data from the documents, the **PyMuPDF** library is used. Since the dataset contains a lot of text in images, it is necessary to extract both text blocks that are in natural text format and text blocks in image format. To achieve this, PyMuPDF is used to extract images for the specific purpose of extracting the texts within them. These images are therefore not saved or used for any other purpose. This procedure is done to make sure that all context is captured to minimize information loss. Meanwhile, the images extracted by partition_pdf are the ones used for further processing. To be able to extract the text in each image, Optical Character Recognition (OCR) is used. **Pytesseract** is an OCR method that uses LSTM to translate the text in images into machine-readable characters. The text data and the text extracted from the image data are stored separately in dictionaries where the key tells if the text originally belonged as "text" or "image" in the document. Tables are extracted by using the **Tabula** library. This library identifies table structures in the documents and stores them as a DataFrame object.

3.3 Chunking

After the data extraction phase, the chunking phase takes place. The procedure involves splitting the extracted text data into varying sizes of blocks and is repeated one time for each chosen chunk size. The chosen chunk sizes are 64, 128, 256, 512, and 1024 respectively. This is done for both of the RAG frameworks. To specify the chunk size, the **TokenTextSplitter** module from the LangChain library is used. The data extraction phase, including chunking, is illustrated in Figure 3.1.

Once the data extraction part and the chunking part are done, the further structures of the two RAG frameworks differ. In the next section, the outline of the

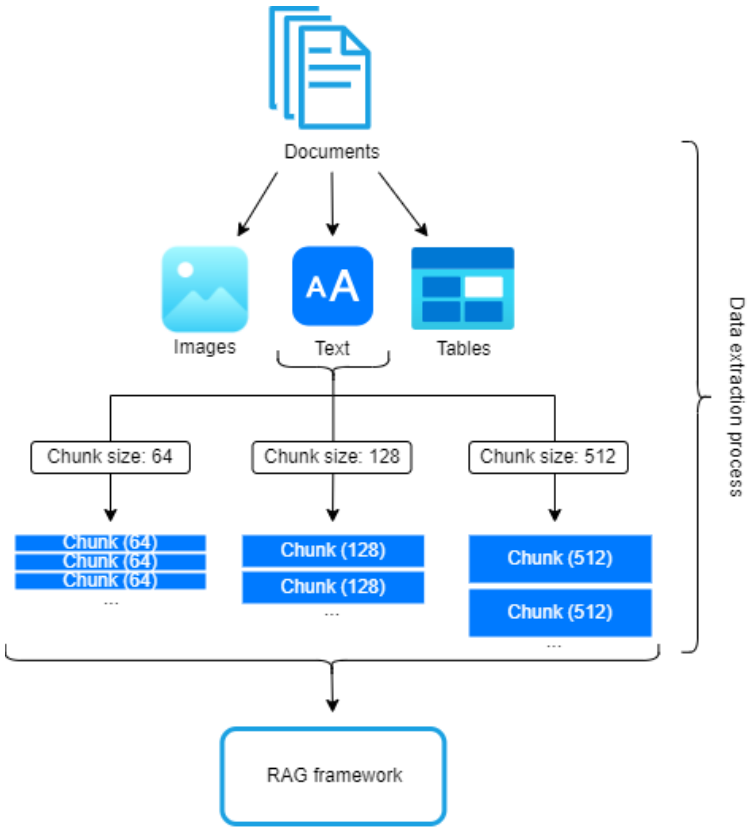


Figure 3.1: Simplified chunking process schema.

Text-only RAG framework model is presented, followed by the outline of the Multimodal RAG framework. All further procedures for both of the RAG frameworks take place five times each, once for the data belonging to each chunk size.

3.4 Text-only RAG

In this section, the different steps of the implementation of the Text-only RAG framework are presented. The first step – generating summaries, takes place right after the data extraction phase. Then, the generating summaries phase is followed by the embedding and retrieval phase and finally, LLaMA2-7B is integrated with the framework.

3.4.1 Generating summaries

When the extraction phase is done, all of the text, image, and table elements are stored in a specified output directory in dictionaries. The next step is to generate

and save summaries of the extracted data. By doing this, the different modalities of the extracted data are flattened and represented as a text-only modality. To summarize the data, either LLM or LMM is employed depending on whether the data is textual or visual. In this project, all of the LLMs and LMMs are run through **ChatOllama**, which allows for running the open-source models locally. This approach is chosen because a potential internal extension of the project could involve sensitive company data. This approach ensures a safe way to process data without it being exposed externally.

To summarize the text chunks and tables, ChatOllama's LLaMA2-7B is used. Each chunk, together with the ChatOllama model as a parameter, is passed into LangChain's summarization chain where the summaries are created based on the map-reduce method. This technique is used for summarizing larger documents, as it splits them into smaller blocks and summarizes them separately in the map step, and then combines these summaries into the final summary in the reduce step. The final summaries are saved into separate text files.

Since LLaMA2-7B is only capable of processing text data, it can not be used for generating image summaries. Instead, ChatOllama's LLaVA-7B is chosen to summarize the extracted images. Here, instead of using LangChain's summarization chain, a specified prompt that tells LLaVA-7B to make detailed summaries of the images is defined. It is formulated as follows:

"You are an assistant whose task is to describe images for developing a Visual Question Answering tool. Provide a comprehensive description of the image, including all relevant details and elements like graphs, charts, diagrams, or textual information. Describe any notable features or patterns observed. Ensure that the description is clear, detailed, and covers all aspects of the image to facilitate understanding it."

The running procedure of LLaVA-7B is defined in a separate script, which together with the prompt is iterated over each extracted image, saving the generated summary into a separate text file. In the end, all of the text files with summaries undergo a cleaning process, where empty lines are deleted.

3.4.2 Embedding and Retrieval

Once all document data is represented as text summaries, the embedding takes place. The first step is to create a storage for the embedded vectors and for the raw data elements, by utilizing a vector store and an in-memory document store. The vector storage is created with **Chroma**, a module from the LangChain library,

which takes the chosen embedding model as a parameter. The chosen embedding model is **Sentence Transformers** from HuggingFaceEmbeddings. This model converts all the text summaries into vector representations, which are then stored in the vector store. The document storage is created with **InMemoryStore** from the LangChain library. The purpose of creating this storage is to keep track of the connection between raw data elements and their corresponding embedded summaries. The raw data elements serve as the parent node in the document store, and the corresponding embedded summaries serve as the child nodes in the vector store.

For the retrieval phase, a **MultiVectorRetriever** is created, which is a module from the LangChain library. The retriever integrates the document store and the vector store by indexing. Each raw data element and its embedded summary is assigned a unique ID, which is a crucial step for the further retrieval stage. During the retrieval, MultiVectorRetriever computes a similarity search between the embeddings stored in the vector store and the embedded user query. Then, documents that have the highest semantic similarities to the query are identified. The retriever's search parameter can be configured manually, which sets the number – top-k value, of the retrieved chunks.

3.4.3 Integrating LLaMA2-7B

Once all of the extracted, summarized and embedded data is stored, an instance of ChatOllama's LLaMA2-7B is integrated with the retriever. At this stage, the ChatOllama model is the central processing unit and is joined with the previous structure and the final QA pipeline is created. The pipeline is constructed as a chain, consisting of a context, a question, a defined prompt, and LLaMA2-7B together with the LangChain_Core modules **RunnablePassthrough**, **StrOutputParser** and **PromptTemplate**. These modules help to construct the pipeline.

Before these parameters are passed into the chain, a prompt template is created using the PromptTemplate module, which enables the interaction within the pipeline. This template is used to construct the prompt that goes into the chain. In the template, the prompted message is formulated as follows:

"Answer the question based only on the following context, which can include text and tables."

The chain also takes a context and a question argument, together with the template transformed into a prompt variable. In the chain, the context is acquired from the retriever and the question is defined as RunnablePassthrough which can be filled

out by the user. When the chain is created, a user query can invoke it and then the different stages of the RAG framework are executed. A response to the question is generated and presented as the final output with the use of StrOutputParser.

3.5 Multimodal RAG

In this section, the different steps of the implementation of the Multimodal RAG framework are presented. The first step – embedding and retrieval, takes place right after the data extraction phase, presented in section 3.2. The embedding and retrieval phase is followed by the stage where LLaVA-7B is integrated with the framework.

3.5.1 Embedding and Retrieval

After extracting all the tables, text, and images from the raw PDF files, the embedding and retrieval process takes place. In the Multimodal RAG framework, images are embedded into the vector store alongside textual data, which results in a unified, multimodal vector store. For this purpose, the Chroma vector store and the **OpenCLIPEmbedding** model is used. This embedding model is an open-source implementation of OpenAI's CLIP published in [24], which has been pre-trained on a variety of image-text pairs. It uses a contrastive learning approach to map images and text to a common embedding space. The Chroma vector store stores these embeddings in memory, organizing them in a structured database with ID keys for each document. Chroma's function 'add_images' stores images as base64 encoded strings so that they can be passed to an LMM like LLaVA-7B. Similarly to the Text-only RAG framework, a document store is created next to the vector store, which stores raw textual data and image metadata and is connected with the vector store by a unique ID.

For the retrieval phase, LangChain's MultiVectorRetriever is initialized to handle multiple vectors – text and image embeddings. Then, the vector store is converted to this retriever instance with a manually specified search parameter, indicating the top-k value of retrieved chunks. As in the Text-only RAG framework, the retriever uses semantic similarity search to match the user's query with stored vector embeddings to retrieve the original context in the end.

3.5.2 Integrating LLaVA-7B

To integrate ChatOllama's LLaVA-7B with the retriever, a prompt function that formats all the retrieved context into a single string is created. An additional message for LLaVA-7B, which is added at the end of the prompt by this function,

is formulated as follows:

"Provide a precise answer to the user question based on the provided context."

If there are images in the retrieved context, this function creates a message containing an image URL. Then, the user question and formatted context texts are stored in another text message. In the end, the generated messages are returned to the prompt from the function as a **HumanMessage** LangChain_Core object. As it is in the Text-only RAG framework, LLaVA-7B and the final prompt are constructed as a chain. The chain uses the retriever to get the context from the documents, and the following modules – RunnablePassthrough to input user questions, the aforementioned prompt function to construct the prompt for LLaVA-7B, and a StrOutputParser that outputs the generated answer. When in use, the RAG chain can be invoked with a user's query and then the context data is retrieved and integrated into the prompt which is passed into LLaVA-7B. LLaVA-7B interprets and analyzes the retrieved multimodal information, generating responses to user queries.

3.6 Evaluation

To assess the performance of the Text-only and Multimodal RAG frameworks and optimize their parameters, BERTScore and LangSmith evaluation libraries are employed. Since access to ground truth answers in this project is provided, BERTScore is used as a foundational metric for assessing the semantic accuracy of the generated answers. This metric outputs its own precision, recall and F1 scores. In addition to this, the LangSmith library is utilized, from which coherence, relevance, and contextual accuracy metrics are selected for the project. While BERTScore is a standard metric for evaluating RAG systems, LangSmith's metrics use the LLM-As-A-Judge approach that complements traditional metrics with human-like reasoning for texts with language nuances and deep contextual understanding. Details on how these metrics function are discussed in Section 2.1.6. Additionally, a manual qualitative evaluation of selected answers is performed. The specifics of the experiments performed and the parameters tested are described in the next chapter.

4

Experiments

The performances of the Text-only and Multimodal RAG frameworks are evaluated by quantitative and qualitative analysis. The base version of the RAG frameworks used in the experiments consists of the chunk size 256 and the top-k value 4.

In the quantitative part, one component, being either chunk size or top-k value, is changed at a time, and the RAG framework's performance is evaluated. Since the current state of evaluation of RAG frameworks focuses mainly on the LLM component in the RAG pipeline [15], it is decided that the experiments in this project would focus on evaluating the other crucial part of the framework – the retrieval stage. Therefore, the effect of two key parameters of the retriever, chunk size and top-k value, is evaluated on two different sets of scores – BERTScore and LangSmith metrics. BERTScore is used to evaluate the generation part of the framework, by calculating the similarity of embeddings between the generated and ground truth answers. It consists of three metrics that are named similarly to the conventional classification metrics, namely recall, precision, and F1 score. It is important to address that BERT metrics are different from the standard classification metrics, although they have a somewhat similar interpretation which is explained in section 2.1.6.1. To avoid confusion, from here on they are referred to as B-recall, B-precision, and B-F1, respectively. The LangSmith metrics are used to enhance evaluation by utilizing a LLM-As-A-Judge approach. The calculated metrics from this framework include coherence, relevance, and contextual accuracy. Such scores are chosen to conduct the most comprehensive evaluation of different configurations of the RAG frameworks.

For the qualitative analysis, generated answers by the Text-only and the Multimodal RAG frameworks are compared to each other as well as with generated answers by LLaVA-7B. LLaVA-7B is in this part of the analysis used as a baseline model. The same questions are asked to all three models and are categorized into different complexities. This analysis is made to visualize how the models understand the retrieved context and match the query when the questions need progressively more comprehension of the context. These responses are analyzed

qualitatively to investigate a possible correlation between the resolution of the question and the quality of the response.

4.1 Quantitative analysis

The quantitative evaluation is run on a total of 100 questions per chunk size, combining 10 questions per manual for 10 different manuals. The performances are measured by changing one parameter at a time, either the chunk size or the top-k value. The chunk size evaluation is run for the five different chunk sizes of 64, 128, 256, 512, and 1024 respectively. The top-k value evaluation is run for k values of 2, 4, 6, and 8 respectively. During the chunk size parameter experiments, the default value of top-k is set to 4 and during the top-k experiments, the default value of the chunk size is set to 256. The results are presented separately for the two RAG frameworks and also separately for two evaluators – BERTScore and LangSmith.

4.2 Qualitative analysis

In the qualitative analysis, the output answers obtained by the Text-only RAG and Multimodal RAG are investigated and compared to the baseline model – LLaVA-7B. The analysis is performed on three different manuals from the dataset, and with questions of 4 levels of complexity. The complexity of a user query is determined based on the location and modality of the data needed to answer it. The four levels of complexity in user queries are the following:

1. **Top page**: Question with text answer located at the top of the page.
2. **Middle page**: Question with text answer located in the middle of the page.
3. **Scattered**: Question with text answer split between multiple OCR regions.
4. **Multimodal**: Questions requiring understanding of both text and image data.

For each level of complexity, there are three questions that are fed to the models, one from each manual. The Top page complexity is designed to test the models' abilities to interpret information that is easily accessible and does not require navigating through dense or overlapping data. The second level of complexity, the Middle page, is more focused and requires more precise information extraction abilities. It tests the models' abilities to locate and interpret valuable content from dense paragraphs, located in the middle of the page among surrounding text. Advancing in complexity, Scattered level of complexity challenges models on their

ability to integrate information spread across multiple paragraphs or sections. It tests if the models can maintain integrity and coherence when the important context is scattered. While the described three levels of complexity require processing only text data, the last level of complexity adds another layer to analyze, which is visual content. Multimodal retrieval complexity is designed to evaluate the highest resolution of answer generation, which integrates both textual and visual cues to generate an answer. The three manuals are fed into the Text-only RAG and the Multimodal RAG framework in order for them to generate responses to the questions. LLaVA-7B however, is not capable of taking a document as an input. It is therefore decided that an image of only the region of the page containing the correct answer is used as an input, alongside with the corresponding question.

Each level progressively builds on the previous one to identify patterns and to distinguish the performance of Text-only RAG, Multimodal RAG, and baseline LLaVA-7B on increasingly complex retrieval-augmented generation tasks.

5

Results

In this section, the results of quantitative and qualitative analysis of Text-only and Multimodal RAG frameworks are presented. The setup of these experiments is reported in section 4.

5.1 Quantitative analysis

In the first part of the quantitative analysis, the results of the chunk size experiments are presented, followed by the second part where the results of the top-k experiments are presented.

5.1.1 Evaluating chunk sizes

Below, the values of BERTScore and LangSmith metrics are reported for the Text-only and Multimodal RAG frameworks across the varying chunk sizes 64, 128, 256, 512, and 1024.

5.1.1.1 BERTScore for Text-only RAG

Table 5.1 and figure 5.1 show the results of B-precision, B-recall, and B-F1 scores for the Text-only RAG framework for the different chunk sizes.

Chunk size	B-precision	B-recall	B-F1
64	0.826	0.872	0.848
128	0.825	0.873	0.848
256	0.827	0.874	0.850
512	0.824	0.876	0.849
1024	0.827	0.877	0.851

Table 5.1: Table of BERTScore for Text-only RAG for different chunk sizes.

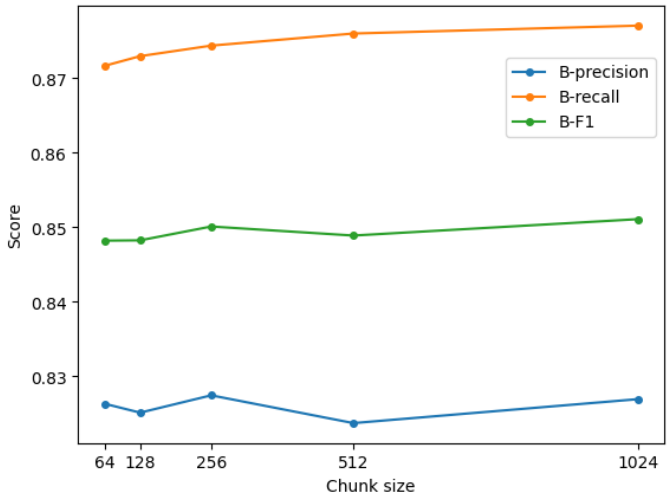


Figure 5.1: Plot of BERTScore for Text-only RAG for different chunk sizes.

As can be seen, the scores for all metrics do not change significantly across the different chunk sizes but remain rather stable with only minor variations. For B-recall, the score tends to slightly increase as the chunk size rises, achieving the best performance at chunk size 1024. On the other hand, B-precision follows no clear trend. It achieves the highest performance at chunk sizes of 256 and 1024. All B-F1 scores are relatively consistent over various chunk sizes with some minor alterations only. Therefore, it can be concluded that all of the three chunk sizes 256, 512 and 1024 are the optimal ones for Text-only RAG as they strike a balance between B-precision and B-recall. These findings also demonstrate that Text-only RAG has a stable performance since it handles different chunk sizes without significant loss in either B-precision or B-recall.

5.1.1.2 BERTScore for Multimodal RAG

Table 5.2 and figure 5.2 show the results of B-precision, B-recall, and B-F1 scores for the Multimodal RAG framework for the different chunk sizes.

Chunk size	B-precision	B-recall	B-F1
64	0.847	0.891	0.868
128	0.850	0.895	0.872
256	0.841	0.887	0.863
512	0.648	0.689	0.667
1024	0.563	0.598	0.580

Table 5.2: Table of BERTScore for Multimodal RAG for different chunk sizes.

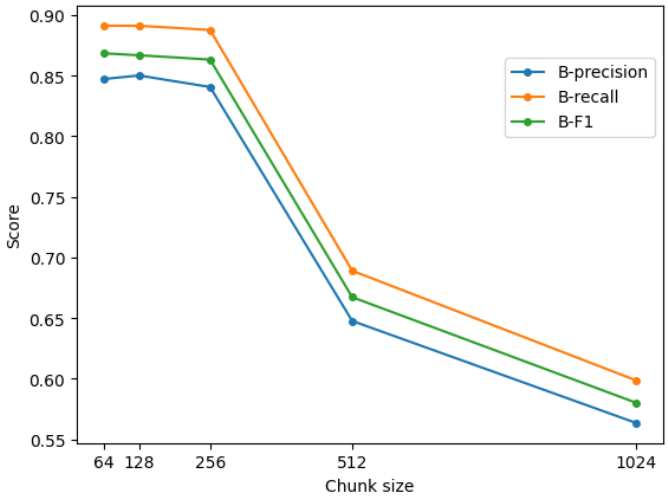


Figure 5.2: Plot of BERTScore for Multimodal RAG for different chunk sizes.

It can be seen that all the BERTScore metrics follow a similar trend. They increase slightly between chunk sizes 64 and 128, peaking in their performance at chunk size 128. For chunk sizes above 128, there is a decrease in all three metrics. Especially, for chunk sizes 512 and 1024, there is a visible drastic drop in B-precision, B-recall, and B-F1. These findings suggest that smaller chunk sizes, where size 128 is the optimal one, are more effective for precise answer generation for the Multimodal RAG. Additionally, since the Multimodal RAGas cannot handle large contexts without an overall loss in performance, it becomes its limitation compared to the Text-only RAG.

5.1.1.3 LangSmith for Text-only RAG

Table 5.3 and figure 5.3 show the scores of coherence, contextual accuracy, and relevance for the Text-only RAG for the different chunk sizes.

Chunk size	Coherence	Contextual Accuracy	Relevance
64	0.820	0.480	0.690
128	0.770	0.550	0.700
256	0.920	0.520	0.740
512	0.850	0.480	0.810
1024	0.840	0.560	0.680

Table 5.3: Table of LangSmith scores for Text-only RAG for different chunk sizes.

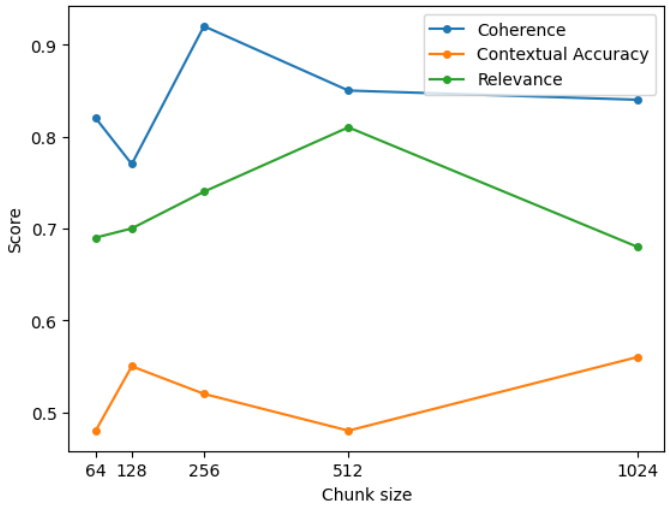


Figure 5.3: Plot of LangSmith scores for Text-only RAG for different chunk sizes.

What can be seen for all the metrics is that the LangSmith scores are inconsistent across the chunk sizes and no clear trend can be noticed. For coherence, there is a relatively significant increase between chunk sizes 128 and 256. For the two chunk sizes greater than 256, the coherence scores seem to get more stable, although it is not known what happens after the chunk size 1024. For the relevance scores, it can be seen that a maximum is reached at chunk size 512. The scores for contextual accuracy are inconsistent across the chunk sizes, where the scores for size 64 and 512 are similarly low around 0.48 and the scores for size 128 and 1024 both are close to 0.56. No trend can be noticed there due to the inconsistent scores. However, it can be seen that contextual accuracy has the opposite behavior of relevance. When one increases between two chunk sizes, the other one decreases. This can imply a trade-off between the inclusion of broader context and direct relevance to specific queries. The optimal chunk size for this case would be mid-range like 256 or 512, since they provide a reasonable balance across all metrics.

5.1.1.4 LangSmith for Multimodal RAG

Table 5.4 and figure 5.4 show the results of coherence, contextual accuracy, and relevance for the Multimodal RAG framework for the different chunk sizes.

It can be noticed that the scores for relevance and coherence seem to follow a similar trend, where a slight decrease between chunk sizes 64 and 128 is followed by a slight increase between chunk sizes 128 and 256. For chunk sizes 512 and 1024, the scores are slightly lower than for 256. For contextual accuracy, the difference between the highest and the lowest score is highly significant and appears between

Chunk size	Coherence	Contextual Accuracy	Relevance
64	0.790	0.360	0.770
128	0.750	0.490	0.690
256	0.870	0.370	0.730
512	0.780	0.070	0.660
1024	0.740	0.230	0.520

Table 5.4: Table of LangSmith scores for Multimodal RAG for different chunk sizes.

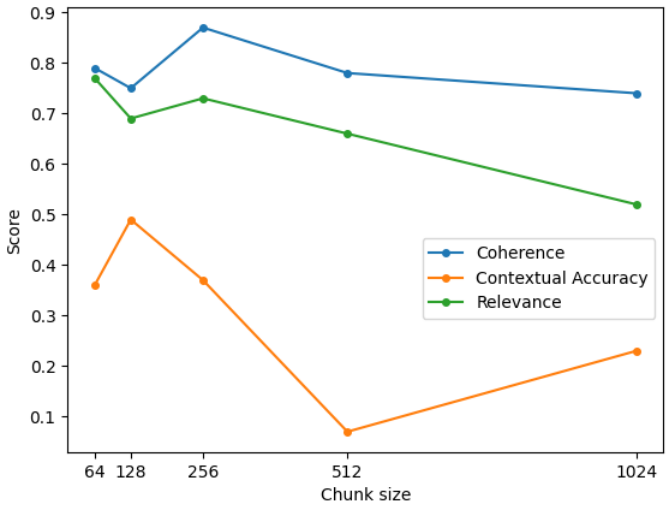


Figure 5.4: Plot of LangSmith scores for Multimodal RAG for different chunk sizes.

chunk sizes 128 and 512 which give the scores 0.49 and 0.07 respectively. Since the score for 512 is close to 0, it makes it the least desirable configuration of the multimodal model. It can be observed that for chunk sizes larger than 256, none of the metrics seem to top off previous scores. Additionally, a similar trend that is observed for LangSmith metrics for the Text-only RAG can be spotted – contextual accuracy increases after dropping at the chunk size 512, while the two other metrics decrease at the end of the chunk sizes' axis. Moreover, what strikes the eye are the notably lower scores for contextual accuracy for the Multimodal RAG compared to the Text-only RAG. In the case of the Multimodal RAG, the optimal chunk size according to LangSmith seems to be either 128 for maximizing contextual accuracy or 256 for maximizing coherence, without the other metrics drastically dropping.

5.1.2 Evaluating top-k values

Below, BERTScore and LangSmith metrics are presented for the Text-only and Multimodal RAG frameworks across the top-k values 2, 4, 6 and 8.

5.1.2.1 BERTScore for Text-only RAG

Table 5.5 and figure 5.5 show the B-precision, B-recall, and B-F1 scores for the Text-only RAG framework across the different top-k values.

Top-k	B-precision	B-recall	B-F1
2	0.825	0.874	0.849
4	0.827	0.874	0.850
6	0.823	0.873	0.847
8	0.819	0.870	0.843

Table 5.5: Table of BERTScores for Text-only RAG for different top-k values.

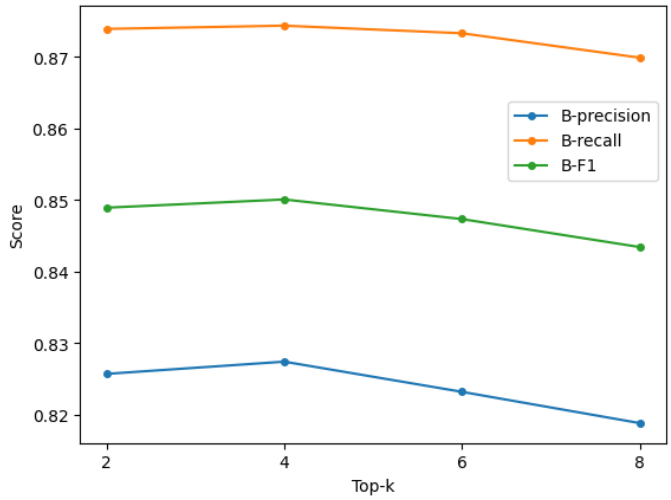


Figure 5.5: Plot of BERTScores for Text-Only RAG for different top-k values.

It can be noticed that all three metrics, namely B-precision, B-recall, and B-F1, follow a similar pattern. At first, their scores slightly increase, up to top-k value 4. After top-k value 4, however, there is a decrease in performance of all three metrics for top-k values 6 and 8, with a minimum reached at top-k value 8. These findings imply that there is a short initial improvement, as more chunks are retrieved, but when the larger values of top-k are used, it can introduce noise to the retrieved context, which confuses the model in the generation stage. It can be observed that

the optimal top-k value for the Text-only RAG according to BERTScore seems to be 4 since all the metrics reach their maximum performance at this point. However, the changes in the scores for the different top-k values are minimal, which makes it difficult to prove this statement. These minor variations also prove that the Text-only RAG framework is rather stable and can maintain a good performance across different top-k values, as is observed for different chunk sizes.

5.1.2.2 BERTScore for Multimodal RAG

Table 5.6 and figure 5.6 show the B-precision, B-recall, and B-F1 scores for the Multimodal RAG framework across the different top-k values.

Top-k	B-precision	B-recall	B-F1
2	0.812	0.861	0.835
4	0.841	0.887	0.863
6	0.845	0.888	0.866
8	0.714	0.754	0.733

Table 5.6: Table of BERTScores for Multimodal RAG for different top-k values.

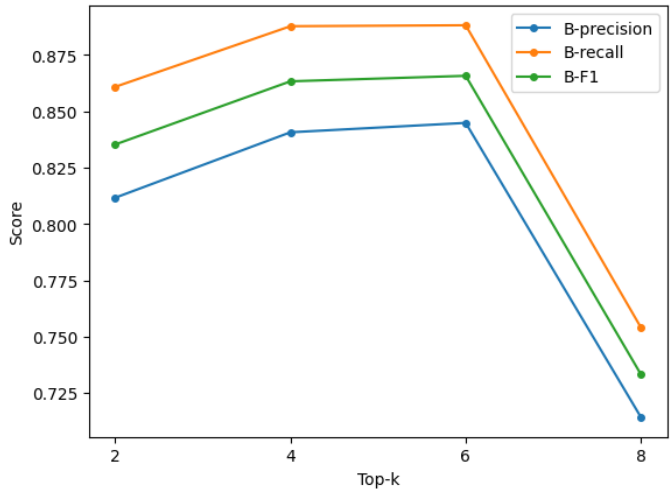


Figure 5.6: Plot of BERTScores for Multimodal RAG for different top-k values.

As can be seen, the scores follow a similar trend as the Text-only RAG above. At first, there is a slight increase in the scores as the top-k value rises from 2 to 6, and for the top-k value 8, there is a drastic drop in performance. Similarly to Text-only RAG, the reason behind this is probably the introduction of irrelevant information in the context leading to confusion in the model. However, for Multimodal RAG

there is a more drastic decrease in the performance for the largest top-k value. The most optimal top-k value in this case seems to be 6, although its performance is only slightly better than at top-k value 4. On the other hand, what can be observed is that the threshold at which additional chunks introduce noise is higher for Multimodal RAG, occurring at top-k value 6, while also the performance drop is more drastic than compared to the Text-only RAG.

5.1.2.3 LangSmith for Text-only RAG

Table 5.7 and figure 5.7 show LangSmith scores for the Text-only RAG framework across the different top-k values.

Top-k	Coherence	Contextual Accuracy	Relevance
2	0.960	0.520	0.720
4	0.920	0.520	0.740
6	0.810	0.560	0.760
8	0.850	0.440	0.680

Table 5.7: Table of LangSmith scores for Text-only RAG for different top-k values.

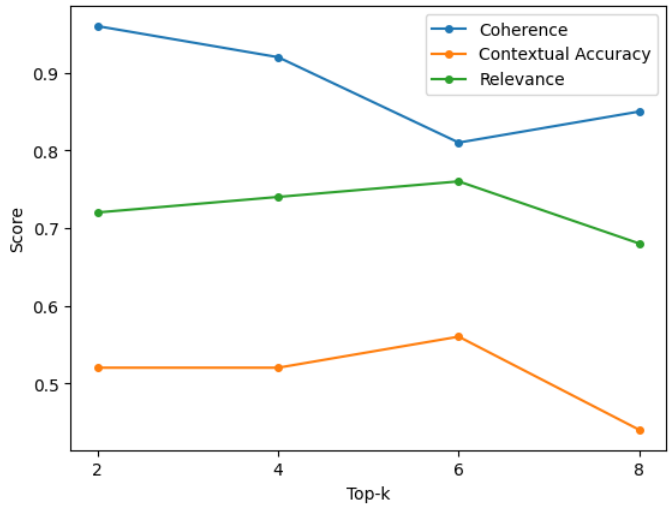


Figure 5.7: Plot of LangSmith scores for Text-only RAG for different top-k values.

What can be seen is that relevance and contextual accuracy follow a similar pattern, slightly increasing their scores up to top-k value 6 and then decreasing at top-k value 8. They both peak in their performances at top-k value 6. On the other hand, coherence exhibits an almost exactly reverse pattern to contextual accuracy. It peaks at the beginning at top-k value 2, then drops at top-k value 6 and slightly

recovers at top-k value 8. These results suggest that the optimal top-k value for the Text-only RAG according to LangSmith is between 4 and 6. All of the three metrics achieve relatively high scores for these k values. It can be also observed, based on the increasing trend in relevance and contextual accuracy up to top-k value 6, that the model benefits from additional chunks up to this point. However, coherence acts in the opposite way. While relevance and contextual accuracy benefit from more context up to a point, coherence is best maintained with fewer chunks.

5.1.2.4 LangSmith for Multimodal RAG

Table 5.8 and figure 5.8 show LangSmith scores for the Multimodal RAG framework across the different top-k values.

Top-k	Coherence	Contextual Accuracy	Relevance
2	0.900	0.340	0.720
4	0.870	0.370	0.730
6	0.880	0.330	0.680
8	0.850	0.190	0.690

Table 5.8: Table of LangSmith scores for Multimodal RAG for different top-k values.

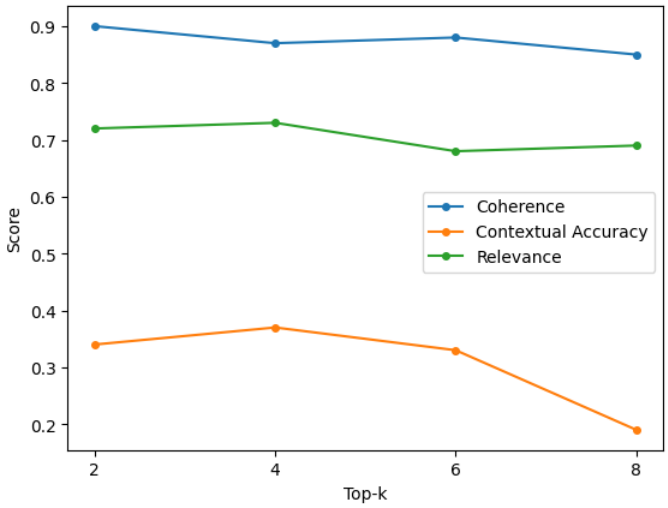


Figure 5.8: Plot of LangSmith scores for Multimodal RAG for different top-k values.

It can be observed that the coherence scores are pretty stable across different top-k values. Coherence peaks at the beginning where the top-k value is 2, similar

to the Text-only RAG. Also, the scores for relevance are relatively stable. There is only a slightly distinguishable peak at top-k value 4. However, for contextual accuracy, it is easier to distinguish the peak and minimum. Similar to relevance, contextual accuracy exhibits a slight maximum at top-k value 4, while for top-k value 8 – a minimum is reached. It can be also noticed that the scores for the Multimodal RAG are generally lower at almost every point compared to the Text-only RAG according to the LangSmith metrics, with contextual accuracy being most notably lower. Moreover, adding more chunks with higher top-k values improves contextual accuracy and relevance at the very beginning, up to top-k value 4. For top-k value 8, contextual accuracy significantly decreases, while other metrics remain somewhat stable but low across all values.

5.2 Qualitative analysis

In this section, questions of varying complexity and the ground truth answers are presented together with the generated responses of the baseline model LLaVA-7B, the Text-only RAG, and the Multimodal RAG. This qualitative assessment is done to test how the RAG frameworks understand the retrieved context and match the query when the questions need progressively more comprehension of the context.

Three manuals for different products are chosen for this analysis — a Dell cellphone manual, a Samsung vacuum cleaner manual, and a Sony laptop manual. From here on, these manuals are going to be labeled Manual 1, Manual 2, and Manual 3. The manuals are chosen because of the resemblance in structure to Wiretronic's product manuals. One question from each complexity category – Top page, Middle page, Scattered, and Multimodal – is chosen for each manual.

For each manual, the four questions of different complexity are asked to LLaVA-7B, the Text-only RAG, and the Multimodal RAG. The responses from the three models are presented for each manual in the Appendix.

5.2.1 Manual 1

Table A.1 shows the generated responses of LLaVA-7B, the Text-only RAG, and the Multimodal RAG framework to the four questions of different complexity for the Dell cellphone manual.

What can be observed, is that generally, LLaVA-7B did not perform well on the questions about Manual 1. All generated responses consist of hallucinations and information that seems to be assumptions. Since LLaVA-7B is trained on a big corpus of data, these kinds of guesses are probably generalizations of data from

other contexts than the one in the manual. This can particularly be seen in the responses to the questions in the categories Top page and Scattered. The response to the question in the category Middle page, however, seems to contain somewhat correct information. The correct information, however, is not as detailed as in the manual, so it is hard to distinguish if the response is based on retrieved context from the manual or if it is also based on general training data. For the question in the category Multimodal, the response is a complete hallucination. None of the information in the response can be found in the manual.

For the Text-only RAG framework, the responses to the different questions are varied in their correctness. The generated responses to the questions in categories Top page and Middle page are both somewhat true. They are general and do not contain a lot of details, which makes it hard to distinguish if their correctness is due to previous training knowledge or if the regions with the ground-truth answers are actually retrieved. For the question in the category Scattered, the response is a general description and does not contain any specific details from the manual. This implies that in this case, the Text-only RAG framework did not manage to retrieve any information from any region of the manual. The same trend can be seen for the response to the question in the category Multimodal. The response is too general and not based on actual retrieved information from the manual.

The Multimodal RAG framework performs slightly better and gives more accurate answers than the Text-only RAG framework and baseline LLaVA-7B. Even though the response to the question in the category Top page is incorrect, it is based on actual information in the manual that is located in another region than the ground truth. For the question in the Middle page category, the response does not seem incorrect. However, it is not based on actual information from the manual. It seems like the response is based on data that has been used to train LLaVA-7B. Even so, the response is not considered a hallucination since it is not incorrect. The response to the question in the category Scattered is correct and contains additional relevant information to the question. The same pattern is seen for the Multimodal question – the response is accurate, since it captures a lot of relevant information from the correct region, plus is supported with additional information from the manual.

5.2.2 Manual 2

Table A.2 shows the generated responses of LLaVA-7B, the Text-only RAG, and the Multimodal RAG framework to the four questions of different complexity for the Samsung vacuum cleaner manual.

The quality of the generated responses by LLaVA-7B varies across the question

complexities. For the question in the category Top page, it manages to generate a very accurate response with no unnecessary additional information. For the question in the category Middle page, it generates a somewhat accurate response. Some parts of the response align with the ground truth, while others seem to not be retrieved from the manual. There is also additional information from the manual included in this response, but it is not directly associated with the question and hence not needed. The responses to the questions in the Scattered and the Multimodal categories seem to be complete hallucinations. The given information can not be found in the manual which also indicates that the information could have been retrieved from the general data used to train LLaVA-7B.

The Text-only RAG framework performs well in generating accurate responses that are similar to the ground-truth answers for this manual. For the Top page category, the response is completely aligned with the ground truth. The response also contains additional safety information that is associated with the correct answer, which makes this response even more valuable than the ground truth. For the Middle page category, the generated response contains somewhat correct information along with other information that is not relevant to the question. The irrelevant parts seem to be completely hallucinated. The response to the question in the category Scattered seems to follow a similar trend. The first sentence of this response is correct but is followed by a long instruction that is not related to the question but seems to be retrieved from elsewhere in the manual. For the last question, belonging to the category of Multimodal, the generated response contains hallucinated information. As it refers to specific names of parts that the question concerns it may seem true, but when reading the manual it becomes clear that the information is false. Some instructions in this response, however, are relevant.

The Multimodal RAG framework seems to not perform as well as the Text-only RAG framework on questions for this manual. For the Top page category question, the generated response is completely incorrect. However, the response is retrieved from the same region as where the ground truth answer is located. This implies that the context of the response is somewhat similar to the ground truth answer even if the details are incorrect. The response for the question in the category Middle page is also incorrect. Here again, the information in the response is not a hallucination, but actual information from elsewhere in the manual. The information from the answer can be found on another page, which implies that the retriever part of this framework failed to capture the correct context. For the question in the category Scattered, the generated response seems to be incorrect too. It is not clear whether the response contains information from another region in the manual or if it is a hallucinated response. Since the response is inconsistently

written, it implies that it could be a completely hallucinated answer. Finally, for the question in the category Multimodal, the generated response seems to be similar to the response generated by the Text-only RAG framework. It does not provide a precise answer but gives somewhat accurate instructions that contain relevant details.

5.2.3 Manual 3

Table A.3 shows the generated responses of LLaVA-7B, the Text-only RAG, and the Multimodal RAG framework to the four questions of different complexity for the Sony laptop manual.

For LLaVA-7B, it can be seen that overall it manages to capture somewhat correct information for the questions of different complexity. However, it can be observed that a big part of each generated answer contains information that is in the manual but is not related to the question. There also occur some hallucinations. For example, the answer to the question from the category Top page contains actual words from the manual, but the sequence of instructions is false. Next, the answer to the question in the Middle page category is somewhat true but contains a lot of information that is not relevant to the question and could be a hallucination. For the question in the Scattered category, the answer is not wrong but does not contain any relevant details and is unnecessarily long. It is a general description of how to use the product, which makes it hard to distinguish if the content is retrieved from the manual or if the response is based on other general data. For the question in the category Multimodal, it is clear that the response is not based on content in the manual. This response is probably based on data that has been used to train LLaVA-7B, which is a problem that also appeared when evaluating LLaVA-7B's responses to the other two manuals.

The responses generated by the Text-only RAG framework have varying quality for questions of different complexity. In general, the responses are not very accurate to the ground-truth answers for this manual. For the question in the Top page category, the response includes relevant words, but the instruction sequence is not correct and can not be found in the manual. It is most likely a hallucinated instruction. For the question in the Middle page category, the generated response is not correct. However, the content in the response is actual information retrieved from the manual, but not related to the question. The same pattern can be seen in the generated response to the question in the Scattered category. The information in the response seems to exist in the manual but is not related to the ground truth answer. This implies that the model manages to retrieve actual information from the manual, but it fails to locate the correct regions. For the Multimodal question, the generated response seems to be a general troubleshooting for how to handle

issues with computers. The information does not seem to be retrieved from the manual, but rather from the data used to train LLaMA2-7B.

For the Multimodal RAG framework, it can be observed that the responses generated for the questions of various complexity are mostly accurate. For the question in the Middle page category, the response is accurate and includes content from other pages of the manual as well. Looking at the response to the Scattered question, it can be seen that it manages to capture accurate information from the correct region. However, the response is too general and not as detailed as the ground truth which makes it a bit irrelevant to the question. For the Multimodal category, the framework manages to capture the correct answer in the very first sentence of the response. However, the response is long and the following sentences are not as relevant and would not have to be included for the response to be accurate. What is noticed for the Multimodal RAG framework in this case, is that all responses across the different questions contain similar information about warnings and what to be cautious about. It is hard to tell whether there is a flaw in the framework that makes it repeatedly retrieve the same information. On the other hand, using a RAG framework for question-answering about products, it would probably be preferred to receive too much rather than too little information about what to be cautious about. In general, the responses generated by the Multimodal RAG framework are somewhat accurate but contain unnecessary information. It shows only little tendencies of hallucinations which is a good indicator of the framework. Generally, the Multimodal RAG framework performs well on scattered questions where parts of the ground truth are located in different regions.

6

Discussion

In this chapter, the Text-only and the Multimodal RAG frameworks will be compared. First, the key evaluation metrics chosen for this discussion are explained, followed by an evaluation of the effects of chunk sizes, top-k values, and the impact of the modality of the frameworks. The most optimal configurations of the two frameworks are compared and discussed. Then, key conclusions drawn from qualitative analysis are presented. This chapter ends with presenting suggestions for future work and discussing risk and ethical considerations.

6.1 Developing domain-specific RAG frameworks

To answer the first research question posed in this project – *Is it possible to integrate a pre-trained LLM or LMM with a retrieval component in a RAG framework to generate responses to domain-specific questions?* – it is proved to not only be feasible but effective in the conducted experiments. Two RAG frameworks, Text-only and Multimodal RAG, are built and their different behavior is observed when generating answers to questions obtained from electronics-related manuals. The integration of a similarity search-based retriever with either LLaMA2-7B or LLaVA-7B proved that it is feasible to leverage pre-existing knowledge by retrieving specific information to navigate the answer generation phase of LLMs and LMMs in this specific domain. The effectiveness of the built frameworks is examined on a set of evaluation metrics, namely BERTScore and LangSmith metrics.

6.2 Parameter and modality impact on RAG framework performance

This section aims to answer the second research question – *How do the modality and the parameters of a RAG framework affect the performance of generated responses?*. First, the thought process behind the selection of the most relevant metrics to base

the comparison on is explained. Then, the effects of modality are discussed in a comparison between the Text-only and Multimodal RAG frameworks. Next, the specific effects of parameters, being chunk sizes and top-k values, are discussed.

6.2.1 Key evaluation metrics for the domain-specific RAG framework

Given the objective of this project, which is to build a framework that most accurately answers users' queries about electronics manuals, it seems like the most important metric from BERTScore to base our assessment on is B-recall. It matches each token in the reference sentence to the most similar token in the generated sentence, while the opposite is done for B-precision. That is to say, a high B-precision score means that the information included in the generated answer is similar to the ground truth answer. However, it does not mean that all relevant information is included. A high B-recall score, on the other hand, means that all of the relevant information from the reference is covered in the generated answer, which is crucial for our goal. Missing out on some critical information from the reference answer could lead to incomplete or inaccurate assembly instructions which can be dangerous for workers. All necessary details should be covered in the generated answer to not cause any damage in the assembly processes. Because of this, the focus is mainly on the B-recall metric in order to find the most optimal framework. As for the LangSmith metrics, they are used as complementary metrics to BERTScore and will provide additional context to help capture aspects of the generated answers that might be more aligned with human judgment. It needs to be addressed that these metrics are based on an LLM assessment and its imperfections can affect the results. Also, these metrics are returning binary scores for each data point, which makes them non-optimal as some aspects of the data can be lost. However, among the LangSmith metrics, contextual accuracy is critical since it assesses if the generated answers accurately capture the essence of the ground truth answer. Relevance is also important since it checks if the generated answers directly address the worker's query. Coherence could be the least important out of the three, but is also interesting to evaluate as it assesses if the answer is clear and well-structured. The main focus will therefore be placed on B-recall and contextual accuracy while discussing the results. However, they will not be of the greatest importance if a high score among these metrics indicates a significant drop in other metrics. Also, more emphasis will be placed on BERTScore metrics while finding the optimal configurations of the frameworks as they are objectively and quantifiably calculated, while LangSmith involves some level of subjectivity.

6.2.2 Effects of modality

The first observation across the performances of the Text-only and the Multimodal RAG frameworks is that the Multimodal RAG framework shows greater sensitivity to changes in the chunk size. While the performance of the Multimodal RAG framework is more unstable, the Text-only RAG only shows small changes in performance across all chunk sizes. Both of the frameworks perform best with moderate chunk sizes – 128 being the optimal one for Multimodal and 256 or 512 for Text-only RAG. However, the Multimodal RAG framework seems to be more affected by larger contexts which is a clear limitation compared to the more stable Text-only RAG framework.

Secondly, the top-k evaluation proves that the Text-only RAG framework is more stable. Both frameworks perform best with medium top-k values, being either 4 or 6. However, the results of the BERTScore evaluation show that the decrease in the performance for the largest top-k value is more drastic for Multimodal RAG. In addition to this, the LangSmith metrics showed that the Text-only RAG framework significantly outperforms the Multimodal RAG in contextual accuracy.

Framework	**B-precision**	**B-recall**	**B-F1**	**Coherence**	**Contextual Accuracy**	**Relevance**
Text-only RAG (chunk size = 256, top-k = 4)	0.827	0.874	0.850	0.920	0.520	0.740
Multimodal RAG (chunk size = 128, top-k = 4)	0.850	0.895	0.872	0.750	0.490	0.690

Table 6.1: Comparison of the scores achieved by the best-performing configurations of the Text-only and Multimodal RAG frameworks.

Next, to get a general overview and to compare the individual scores achieved by the best-performing configurations of the Text-only and the Multimodal RAG frameworks, Table 6.1 is created. From the previously discussed optimal configurations, the ones with the highest number of top scores, with relatively high B-recall and contextual accuracy, are chosen and visualized in the table. These configurations are the Text-only RAG framework with a chunk size of 256 and top-k value of 4 and the Multimodal RAG framework with a chunk size of 128 and top-k value of 4. It can be deducted from the table that when the most optimal chunk size and top-k value are found for the Multimodal RAG framework, it does outperform the Text-only RAG framework for the three BERTScore metrics. Looking at B-recall, the best-performing configuration of the Multimodal RAG framework scores

0.895, while the best-performing configuration of the Text-only framework scores 0.874. However, the LangSmith metrics show otherwise. All of the LangSmith metrics are lower for the Multimodal RAG framework compared to the Text-only RAG. The score of the Text-only RAG framework for contextual accuracy is 0.520 compared to 0.490 for the Multimodal RAG framework. Coherence should be also highlighted, as it is significantly higher for the best-performing Text-only RAG configuration, with a score of 0.920 versus 0.750 for the best-performing Multimodal RAG framework.

Observing all experiments and not only the best-performing configurations, the most notable difference can be seen in the contextual accuracy scores. These scores are significantly lower for the Multimodal RAG framework, in both the chunk size and in the top-k experiments, compared to the Text-only RAG framework. However, it needs to be addressed that the contextual accuracy metric from LangSmith is based on an LLM judgment and its imperfections can affect the results.

In conclusion, while the best configuration of the Multimodal RAG framework shows a better performance in B-precision and B-recall according to BERTScore, the Text-only RAG framework has a superior performance in coherence and contextual accuracy according to LangSmith. Lower LangSmith scores can suggest that the Multimodal RAG framework struggles more with producing coherent answers that have a logical flow and organized structure. It can also imply that it does not manage to produce answers that align with the question content or the human judgment on how to answer it. One reason behind this can be that it struggles with integrating the multimodal information into a structured and contextually accurate way, leading to confusion. The Text-only RAG framework offers more reliable performance, with more robustness to changes in the chunk size and the top-k value. This implies that without further refinement and optimization of the Multimodal RAG framework, it should not be favored over the more reliable Text-only RAG framework for the purpose of this project.

6.2.3 Effects of parameters

As proven in the experiments, the two key parameters of the retrieval stage – chunk size and top-k value, do influence the performance of RAG frameworks. These two parameters both determine how much context is presented to the LLM or LMM, which affects the quality of the generated answers [15]–[17]. As demonstrated in this project, optimizing chunk size and top-k value is crucial to generate accurate and coherent responses. What distinguishes their roles is that chunk size optimizes the granularity of information in the context, while top-k values optimize the quantity of context. Even if they have different objectives, they influence the same stage of the framework but in different places, therefore also affecting each

other. To achieve the highest results, it would be best to find the most optimal combination of the two. One parameter with a high value could be balanced out by the other one with a low value. However, in this project, the goal was to test the individual effect of each parameter, therefore only one parameter at a time was changed.

When discussing the effects of parameters, the individual impact of chunk sizes and top-k values separately on the Multimodal and Text-only RAG frameworks is first described. Then, collective observations are drawn on how these parameters influence RAG frameworks in general, supported by the literature.

6.2.3.1 Chunk sizes

Observing the Text-only RAG framework, the BERTScore metrics show a relatively stable performance across the different chunk sizes. These results imply that the framework has a potential advantage in robustness against changes. Therefore, it is difficult to distinguish an optimal chunk size, since all the scores in the BERTScore evaluation are similar. Nevertheless, the best-performing configuration of the Text-only RAG framework could have either chunk size 128, 256, or 512. The scores for these chunk sizes do not vary significantly and offer different advantages when it comes to other metrics. Therefore, it is difficult to distinguish a single optimal chunk size due to the stability of the Text-only RAG framework. There are even higher contextual accuracy and B-recall scores observed with chunk size 1024, however, it is noticed that it has an opposite behavior to relevance. A too-high chunk size can not be chosen as the most optimal in this case, since it indicates a potential drop in the B-precision and the relevance scores. Chunk sizes 256 and 512 have a good balance between the B-precision and the B-recall scores, while chunk size 256 showcases the highest coherence score, and chunk size 512 showcases the highest relevance score.

The performance of the Multimodal RAG framework is negatively affected by larger chunk sizes, which is observed in the BERTScore evaluation for all three metrics. This is a clear limitation of this framework, as it can not handle large contexts without loss in performance. The Multimodal RAG framework performs best with smaller chunk sizes, with 128 being the most optimal one. The configuration of the Multimodal RAG framework with this chunk size gives the highest scores for B-recall, B-precision, B-F1 and contextual accuracy. The LangSmith scores for this framework showcased a similar pattern to the Text-only RAG framework, with the contextual accuracy increasing as the chunk size grows larger meanwhile the coherence and the relevance scores decrease. However, it is observed that the changes in the contextual accuracy scores throughout different chunk sizes generally are more significant for the Multimodal RAG framework than for the Text-only RAG

framework. This further proves that the Multimodal RAG framework is more unstable and sensitive to changes in the chunk size parameter.

6.2.3.2 Top-k values

The most optimal top-k value for the Text-only RAG framework according to the BERTScore evaluation seems to be 4 across all three metrics. The scores are similar for all of the BERTScore metrics across all top-k values. These results imply that to be able to distinguish an actual pattern, more values would have to be investigated. However, looking at the B-recall scores, it is seen that top-k values 2 and 4 both reach the highest score of 0.874. This indicates that the Text-only RAG framework seems to be stable and manages to perform well despite parameter changes. The LangSmith evaluation shows more unstable scores across the top-k values for the Text-only RAG framework. For the contextual accuracy metric, the highest score of 0.560 is reached for the top-k value of 6. This implies that the generated responses of the framework manage to best capture the context of the ground-truth answers when the top-k value grows larger. This is slightly unexpected, as the larger the value of k, the harder it gets to locate and retrieve correct information.

The BERTScore evaluation for the Multimodal RAG framework indicates that larger values of k negatively affect the performance. The B-precision, B-recall, and B-F1 metrics all show a relatively stable performance for top-k values 1, 4, and 6 but drop drastically for the top-k value of 8. These results indicate that a larger k is not beneficial for the Multimodal RAG framework and that it doesn't manage to capture the context across a higher amount of regions. The LangSmith evaluation shows more varying results which makes it difficult to distinguish any trend. The contextual accuracy scores, however, slightly confirm the trend observed in the BERTScore evaluation. The performance is somewhat stable for the top-k values 2, 4, and 6 but drops for the top-k value 8.

6.2.3.3 General observations

As supported by the literature, smaller chunk sizes provide more specific and focused information [17], which increases the precision. Larger chunk sizes could be beneficial for broader questions since they include wider contexts, but they could also add more confusing or irrelevant information, which is difficult to distinguish in the next stages of a RAG framework [17]. This is confirmed in this project, as the coherence and relevance scores are the lowest for the largest chunk size. Larger chunk sizes, particularly 512 and 1024, result in significant performance drops for the Multimodal RAG framework as it gets harder to connect longer contexts with visual cues. The Text-only RAG framework remains relatively stable for different

chunk sizes. The top-k value determines the volume of retrieved information. Low top-k values may not provide enough relevant context, which could generate incomplete or incoherent answers, while high top-k values may overwhelm the model and make the relevant chunks harder to recognize, producing less accurate responses [15]. This is proven in the experiments where the middle top-k values (4 or 6) provide the best performance, ensuring sufficient context without including noise. Including unnecessary context also increases the computational cost by increasing the number of processed input tokens, which makes a too large chunk size or top-k value inadvisable.

It can be deducted from both the chunk size and top-k values evaluation, that when evaluated on BERTScore, B-recall is consistently higher than B-precision in every experiment. This could imply that false positives slightly outnumber false negatives and that RAG frameworks have a tendency to include more tokens to avoid missing relevant ones, although the price is that it may bring in more irrelevant tokens. This is an advantage for this application since the goal is focused more on covering all necessary information (high recall) than providing relevant information (high precision).

The scores for the LangSmith metrics coherence and relevance are significantly higher than for the contextual accuracy metric. Coherence being the highest scored metric across all experiments suggests that even though the RAG frameworks produce organized and structured responses, they could not have a high alignment with the ground truth answers. For our application, it is crucial to provide correct responses which makes logically structured responses less prioritized than providing misleading or incomplete information.

In conclusion, moderate chunk sizes – 128 to 256 – and top-k values – 4 to 6 – generally return the best performance across both the RAG frameworks. Optimizing these parameters is crucial because the optimal chunk size will provide a good balance between the level of detail and the breadth of context, while the optimal top-k value will provide enough context without adding noise. For the future extension of the project, it is believed that it would be most beneficial to compensate a small chunk size (such as 128) with a larger top-k value (such as 6) since processing larger chunks is more computationally extensive than larger top-k values.

6.3 Qualitative analysis overview

What can be seen in the qualitative results is that all generated responses by LLaVA-7B, the Text-only RAG framework, and the Multimodal RAG framework

are long and nested compared to the ground-truth answers. The ground-truth answers, however, are manually documented from the manuals which means that the length and included details have been subjectively chosen. Because of this, the correctness of the information rather than the semantic similarity is what should be compared between the generated responses and the ground truth answers. The results show that the quality of the generated responses varies a lot across the three models. No significant trend can be distinguished in any of the performances. However, the results show that the Text-only RAG framework and the Multimodal RAG framework manage to retrieve the correct answer more often than LLaVA-7B. This observation is expected due to LLaVA-7B being used as a baseline model without any fine-tuning or retrieval component to the specific context. This observation also implies that the two RAG frameworks are built and implemented successfully and serve their purpose of enhancing domain-specific performance.

Looking at the performances of the Text-only and the Multimodal RAG frameworks, the quality varies a lot across the questions of the different complexities. What can be noticed for both of the RAG frameworks is that they tend to hallucinate more for the questions in the Scattered and the Multimodal category. Most likely, to be able to distinguish any further pattern of how the performance varies across question complexity, more manuals would have to be analyzed. By doing so, it would be expected for the Multimodal RAG framework to perform better on multimodal questions. Even if the Text-only RAG framework shows to perform well on multimodal questions, keeping the image data intact would ensure that no context is lost in the process of summarizing the images into text. The expected performance for the Top page and Middle page questions would probably not differ a lot between the RAG frameworks. A contributing factor to these performances would rather be the chunk size and the top-k value. Making a qualitative analysis to test how the RAG frameworks perform on the different question complexities with different values of chunk size and top-k would be a natural extension of this project. Also, the performances of the questions in the Scattered category would probably depend on the chunk size and the top-k value.

6.4 Future work

There are several areas to consider in a potential future extension of this project. Firstly, the amount of manuals used should be increased, since this project was limited to analyzing only 10 manuals. Performing evaluation on more manuals would probably give clearer, more distinguishable results where the scores of the different configurations would stand out more. Similarly, the range of parameters tested should be extended in the future. It could make the findings more generalizable

and could help to further optimize the Multimodal RAG framework, which has the potential to outperform the Text-only RAG framework. Due to the limitations of the computational resources used and the time frame of the project, investigating a wider range of parameters and incorporating more manuals was not possible, but should be considered in a future extension.

Secondly, the qualitative analysis clearly shows that the generated answers by the two RAG frameworks most of the time are long and unnecessarily extensive. A natural continuation would be to investigate prompting techniques to optimize the format of the responses. This type of analysis would be qualitative and dependent on the user's requirements. In certain cases, extensive responses may be preferred while in other cases they would not be. Investigating different prompting techniques could also be used as a method to minimize hallucinations. As prior research shows that the format of the prompt can have an effect on how much the frameworks hallucinate, this would be a natural extension of the project.

To improve the performance and efficiency of the parameter tuning experiments, it would be valuable to integrate dynamic chunk size and top-k retrieval mechanisms. These techniques could adjust the parameter value based on the complexity of the query. For more complex queries, more context would be retrieved but the parameter value would not be static, ensuring lower computational costs [20]. Additionally, re-ranking techniques that make sure that the best matching chunks are retrieved could be explored to even further improve the accuracy of the generated responses. However, it would come with additional computational costs [19]. These two methods could further optimize the RAG frameworks built in this project.

Also, this project focuses on tuning parameters that influence the retriever component of a RAG framework. However, since the main components of this framework are both the retriever and the generative component, it would be beneficial to explore how different LLMs or LMMs affect performance. In this project, LLaMA2-7B and LLaVA-7B are implemented, but it would be valuable to try some models that have more parameters or are generally newer, as they may offer different advantages.

Another natural extension of this project would be to include a more thorough and separate evaluation of the retriever. Even though the parameters of the retrieval component are being optimized, only the final generated answers are evaluated. Evaluating the retrievers performance independently by printing and evaluating the retrieved chunks would provide more insights into the retrieval phase's actual effectiveness. One approach to do that would be to manually select regions of manuals where the ground truth is located since the dataset used in this project

does not provide them, and use them to calculate the page-level or paragraph-level accuracy. Another solution would be to use an LLM-As-A-Judge approach to ask the LLM to reason about the relevancy of the retrieved chunks to a user's query.

Further on, different retrievers could be compared that employ other techniques than the vector-based similarity search used in this project. Overall, even if the generative component of the RAG framework manages to produce accurate responses based on the information that is retrieved, it is crucial to evaluate how accurate the retrieved information is.

Another potential future research area could be to further investigate the efficiency of the two RAG frameworks. Evaluating the frameworks based on a cost-efficiency approach could give valuable insights. Measuring the running times for different tasks for the two frameworks would allow for taking it into account in the overall evaluation. If the frameworks show very similar performances but the running times drastically differ, prioritizing which framework is the most suitable for certain tasks would become easier. As the running times for the extraction, summarization and evaluation are long, another future area of interest would be to implement parallel programming. Splitting the runs across several cores would most likely minimize the running time, enhancing the overall efficiency of the frameworks.

6.5 Risk analysis and ethical considerations

While industrial automation comes with a lot of advantages, certain risks need to be considered. An LLM may have tendencies to hallucinate, which will make it generate false responses. Providing false information about the production may result in security risks and danger for the workers as well as quality issues with the products. A careful evaluation of the final model and human supervision is therefore crucial before real-world usage and implementation.

Another ethical aspect to consider is how employees will be affected by this kind of automated tool. Regular workflows and the need for human labor may be affected which are aspects that come with both advantages and disadvantages which should be weighed against each other. Other ethical aspects to consider are that the company data should be treated carefully and according to agreement. The open-source data that is used follows the GDPR.

7

Conclusion

In this project, a Text-only and a Multimodal RAG framework are developed to enhance the performance of LLMs and LMMs by integrating knowledge from an external database created from electronics user manuals. By doing so, the goal of answering the two research questions posed in the beginning is met, namely:

- Is it possible to integrate a pre-trained LLM or LMM with a retrieval model in a RAG framework to generate responses to domain-specific questions?
- How do the modality and parameters of the RAG framework affect the performance of the generated responses?

For the first research question, it is proved that it is feasible and effective to connect a similarity search-based retriever with either LLaMA2-7B or LLaVA-7B. This integration is achieved with the help of the LangChain library and the Chroma vector store. The final architectures of the two frameworks differ. The Text-only RAG framework first employs LLaMA2-7B to generate summaries of text and tables extracted from the raw PDFs and uses LLaVA-7B to summarize features of extracted images. Then, a Multi-Vector Retriever, that retrieves image summaries and raw text and tables, is used to prompt LLaMA2-7B to generate a final response. The Multimodal RAG framework uses CLIP embeddings to create a unified vector space for text and image data and then connects it by a Multi-Vector Retriever to LLaVA-7B. Both frameworks generate answers at a satisfying level of B-recall, although the behaviors differ in various scenarios. Through qualitative analysis, it is proved that both frameworks manage to retrieve the correct answer more often than the baseline model LLaVA-7B, which implies that they serve their purpose of enhancing the domain-specific performance.

The second research question is addressed by evaluating two key parameters of the retriever – chunk size and top-k value – as well as investigating how the modality of the frameworks affects the performance. The findings prove that the performance of the Multimodal RAG framework is more sensitive to changes in both chunk size and top-k value, while the Text-only RAG framework is more stable. Generally,

moderate chunk sizes – 128 or 256 – and top-k values – 4 or 6 – returned the best scores across both frameworks. When the optimal parameters for the Multimodal RAG framework are found, it slightly outperforms the Text-only RAG framework according to the BERTScore metrics. However, the Text-only RAG framework shows superior performance in coherence and contextual accuracy according to the LangSmith metrics when testing the different parameter configurations. This implies that the Text-only RAG framework is more reliable because it generates more coherent and rational responses. Ultimately, it is the preferred framework to use to achieve the goal of this project as stability and reliability are crucial aspects of assembly processes. Even though the Multimodal RAG framework shows its potential with the right optimization, it would require additional tuning to improve its stability and to possibly outperform the Text-only RAG framework.

The final product of the project is the optimization of the two RAG frameworks to a significant extent. The aforementioned findings of this project can be later used as a foundation for further research and development in the field of automated assembly verification and RAG-based VQA. The future work for enhancing the VQA system proposed in this project should aim to refine the Multimodal RAG framework in order to boost its stability and performance. Furthermore, there is a need to elaborate on ethical aspects. Especially since the trustworthiness of the responses is crucial in this domain, to not lead to any misinformation and resulting errors.

Bibliography

[1] A. Nandy, S. Sharma, S. Maddhashiya, K. Sachdeva, P. Goyal, and N. Ganguly, "Question answering over electronic devices: A new benchmark dataset and a multi-task learning based qa framework," *arXiv preprint arXiv:2109.05897*, 2021.

[2] Y. Qin, "Application of virtual process verification in production preparation of instrument panel assembly line," in *Proceedings of China SAE Congress 2021: Selected Papers*, Springer, 2022, pp. 932–940.

[3] S. K. Sampat, Y. Yang, and C. Baral, "Visuo-linguistic question answering (vlqa) challenge," *arXiv preprint arXiv:2005.00330*, 2020.

[4] S. Antol, A. Agrawal, J. Lu, *et al.*, "Vqa: Visual question answering," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2425–2433.

[5] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Visual instruction tuning," *arXiv preprint arXiv:2304.08485*, 2023.

[6] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," *arXiv preprint arXiv:2310.03744*, 2023.

[7] L. Zhang, A. Hu, J. Zhang, S. Hu, and Q. Jin, "Mpmqa: Multimodal question answering on product manuals," *arXiv preprint arXiv:2304.09660*, 2023.

[8] R. Tito, D. Karatzas, and E. Valveny, "Hierarchical multimodal transformers for multipage docvqa," *Pattern Recognition*, vol. 144, p. 109 834, 2023.

[9] R. Awal, L. Zhang, and A. Agrawal, "Investigating prompting techniques for zero-and few-shot visual question answering," *arXiv preprint arXiv:2306.09996*, 2023.

[10] H. Touvron, T. Lavril, G. Izacard, *et al.*, "Llama: Open and efficient foundation language models.," *arXiv preprint arXiv:2302.13971.*, 2023.

[11] O. Oded, M. Brief, M. Moshik, and E. Oren, "Fine-tuning or retrieval? comparing knowledge injection in llms," *arXiv preprint arXiv:2312.05934.*, 2023.

[12] A. Balaguer, V. Benara, R. Cunha, *et al.*, "Rag vs fine-tuning: Pipelines, tradeoffs, and a case study on agriculture.," *arXiv e-prints, arXiv-2401.*, 2024.

[13] Y. Gao, Y. Xiong, X. Gao, *et al.*, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, 2023.

[14] *Multi-vector retriever for rag on tables, text, and images*, Accessed: 24-03-2024. [Online]. Available: https://blog.langchain.dev/semi-structured-multi-modal-rag/.

[15] Y. Lyu, Z. Li, S. Niu, *et al.*, "Crud-rag: A comprehensive chinese benchmark for retrieval-augmented generation of large language models," *arXiv preprint arXiv:2401.17043*, 2024.

[16] S. Setty, K. Jijo, E. Chung, and N. Vidra, "Improving retrieval for rag based question answering models on financial documents," *arXiv preprint arXiv:2404.07221*, 2024.

[17] R. Schwaber Cohen, *Chunking strategies for llm applications*, Accessed 15-04-2024, Jun. 2023. [Online]. Available: https://www.pinecone.io/learn/chunking-strategies/.

[18] R. T. Ashish Abraham Mór Kapronczay, *Optimizing RAG with Hybrid Search & Reranking | VectorHub by Superlinked — superlinked.com*, Accessed 21-05-2024, 2024. [Online]. Available: %5Curl%7Bhttps://superlinked.com/vectorhub/articles/optimizing-rag-with-hybrid-search-reranking%7D.

[19] J. Chen, *Optimizing RAG with Rerankers: The Role and Trade-offs - Zilliz blog — zilliz.com*, Accessed 21-05-2024, 2024. [Online]. Available: %5Curl%7Bhttps://zilliz.com/learn/optimize-rag-with-rerankers-the-role-and-tradeoffs#How-Does-a-Reranker-Enhance-Your-RAG-Apps%7D.

[20] S. Joshi, *Optimizing Retrieval Augmentation with Dynamic Top-K Tuning for Efficient Question Answering — sauravjoshi23*, Accessed 21-05-2024, 2023. [Online]. Available: %5Curl%7Bhttps://medium.com/@sauravjoshi23/optimizing-retrieval-augmentation-with-dynamic-top-k-tuning-for-efficient-question-answering-11961503d4ae%7D.

[21] T. Kagaya, T. J. Yuan, Y. Lou, *et al.*, "Rap: Retrieval-augmented planning with contextual memory for multimodal llm agents," *arXiv preprint arXiv:2402.03610.*, 2024.

[22] N. Kandpal, H. Deng, A. Roberts, E. Wallace, and C. Raffel, "Large language models struggle to learn long-tail knowledge," in *International Conference on Machine Learning*, PMLR, 2023, pp. 15 696–15 707.

[23] Y. Zhang, Y. Li, L. Cui, *et al.*, "Siren's song in the ai ocean: A survey on hallucination in large language models," *arXiv preprint arXiv:2309.01219*, 2023.

[24] A. Radford, J. W. Kim, C. Hallacy, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, PMLR, 2021, pp. 8748–8763.

[25] C. Li, C. Wong, S. Zhang, *et al.*, "Llava-med: Training a large language-and-vision assistant for biomedicine in one day," *arXiv preprint arXiv:2306.00890*, 2023.

[26] J.-Y. Yao, K.-P. Ning, Z.-H. Liu, M.-N. Ning, and L. Yuan, "Llm lies: Hallucinations are not bugs, but features as adversarial examples," *arXiv preprint arXiv:2310.01469*, 2023.

[27] H. Liu, W. Xue, Y. Chen, *et al.*, "A survey on hallucination in large vision-language models," *arXiv preprint arXiv:2402.00253*, 2024.

[28] U. Lee, M. Jeon, Y. Lee, *et al.*, "Llava-docent: Instruction tuning with multimodal large language model to support art appreciation education," *arXiv preprint arXiv:2402.06264*, 2024.

[29] P. Liang, R. Bommasani, T. Lee, *et al.*, "Holistic evaluation of language models," *arXiv preprint arXiv:2211.09110*, 2022.

[30] L. Zheng, W.-L. Chiang, Y. Sheng, *et al.*, "Judging llm-as-a-judge with mt-bench and chatbot arena," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[31] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.

[32] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[33] *How to use off-the-shelf evaluators*, Accessed 18-03-2024, 2024. [Online]. Available: `https://docs.smith.langchain.com/old/evaluation/faq/evaluator-implementations`.

[34] T. Wang, J. Li, Z. Kong, X. Liu, H. Snoussi, and H. Lv, "Digital twin improved via visual question answering for vision-language interactive mode in human–machine collaboration," *Journal of Manufacturing Systems*, vol. 58, pp. 261–269, 2021.

[35] D. Cain, *Retrieval augmented generation and the evolution of large language models.* Accessed: 2024-03-24. [Online]. Available: `https://www.linkedin.com/pulse/retrieval-augmented-generation-evolution-large-language-david-cain/`.

[36] Y. Tang and Y. Yang, "Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries," *arXiv preprint arXiv:2401.15391.*, 2024.

A

Appendix 1

A. Appendix 1

Question complexity	Question	LLaVA-7B	Text-only RAG	Multimodal RAG	Ground-truth
Top page	For the Dell cellphone, what does a WARNING indicate?	A warning indicator on a product, such as a Dell cellphone, typically signifies the presence of a potential hazard or a condition that requires attention. In this case, the warning says "CAUTION: DO NOT CLOSE THE BATTERY COMPARTMENT WITH DATA AND/OR MEDIA INSIDE." This indicates that there is important information or media inside the battery compartment of the cellphone, and closing it without removing any data or media could result in damage to these items. It is essential for users to follow this warning to avoid potential loss of data or damage to their device.	Based on the given context, a WARNING indicator on the Dell cellphone likely indicates that there is a potential risk or danger associated with the device or its usage. The arrow pointing to the location of the WARNING suggests that it is located nearby and worth paying attention to.	For the Dell cellphone, a WARNING indicator typically means there is an important message or notification related to the phone's functionality or operation. It can be a reminder, an alert for low battery or signal strength, or a message from the carrier provider. To access the warning indicator, users should tap on the Messaging icon on the Home screen and then select the "More" option to change the message service settings. For more information on other options for this task, see the User's Manual on the Dell Support website at supportapj.dell.com.	It shows potential for property damage, personal injury, or death.
Middle page	For the Dell cellphone, what is the function of the Camera button?	The image displays an instructional graphic explaining the features of a Dell smartphone. In this image, the Camera button is shown at the top left corner of the phone's user interface. The purpose of the Camera button is to quickly access the camera application on the smartphone. When pressed, it will open the camera app, allowing the user to take photos or record videos. This feature provides a convenient one-touch access to the camera function without having to navigate through multiple menus or press additional buttons on the phone's screen.	Based on the provided context, the function of the Camera button on the Dell cellphone can be inferred as follows: 1. The image shows two buttons located at the bottom part of the device: one closer to the left side and another near the center-right area. Since the button closer to the left side is labeled "Camera," it suggests that this button might control the phone's camera functionality. 2. The presence of a keypad on the cellphone suggests that it may be an older model, possibly from the early 2000s or late 1990s. In those times, the keypad was a common design feature for mobile phones, and the Camera button could have been used to activate the camera function. 3. The image also shows several wires connected to various components on the phone, indicating that charging or data transfer may be possible through these connections. However, given the age of the device, it is more likely that the wires are for charging purposes only. 4. Finally, the tiny person visible in the image may represent someone holding the device, suggesting that the Camera button could be used to take pictures or videos with the phone's built-in camera. Based on these observations, it is probable that the Camera button on the Dell cellphone is used to activate and control the phone's built-in camera function for taking photos or videos.	The function of the Camera button on a Dell cellphone is to allow users to quickly and easily access the camera application, enabling them to take photos or record videos. This can be particularly useful for capturing spontaneous moments or quickly sharing experiences with others through social media or messaging apps. By providing a dedicated camera button, the phone's user interface makes it easier for users to navigate to the camera application and start using its features without having to search through multiple menus.	You can press it to activate the camera and you can take photos or capture videos.
Scattered	For the Dell cellphone, what should I do before using the smartphone?	The image you've shared seems to be a page from a manual or guide on setting up a Dell cellphone. It provides instructions for the following steps: 1. **Set up the phone**: This likely involves turning on the phone, connecting it to a Wi-Fi network, and following the setup prompts that will guide you through creating an account and customizing your phone's settings. 2. **Activate phone service**: Before you can use your smartphone for calls and data, you need to activate a cellular plan with a mobile service provider (such as Verizon, AT&T, T-Mobile, etc.). This typically involves inserting a SIM card into the phone if it's not already done. 3. **Charge the battery**: Ensuring that your Dell smartphone is fully charged before using it for the first time can help prevent any interruptions or issues during setup. 4. **Use a USB cable to connect to a computer**: This step suggests connecting the phone to a computer if you want to transfer data, such as contacts and media files, from your old phone to the new Dell cellphone. 5. **Register your device**: If there is an activation code provided by the service provider when you purchased your smartphone, this step involves entering that code into the phone's settings menu to complete the activation process. 6. **Sign in to activate the phone**: This implies that you need to sign in with your account credentials after setting up the phone to access all itsfeatures. 7. **Charge your battery to 50% or more**: This is a safety measure, as it's important not to let the battery level drop below 50% while setting up a new device. 8. **Update the software**: The last step suggests updating the phone's operating system and any apps that might have been preinstalled on the device. Updating the software can improve the performance of your smartphone and ensure it has the latest features and security patches.	Before using your Dell Mini 3i smartphone, you should follow the instructions provided in the "Setting Up Your Mini 3i Smartphone" document. This will help you properly set up and configure your device for optimal performance and safety. Specifically, you should: 1. Read the user manual carefully to understand the features and capabilities of your smartphone. 2. Ensure that your smartphone is compatible with your service provider's network. 3. Activate your smartphone by following the instructions provided by your service provider. 4. Set up your smartphone's security features, such as a password or fingerprint recognition, to protect your device and personal data. 5. Familiarize yourself with the smartphone's physical layout and button functions to avoid accidental calls or text messages. 6. Review the safety precautions and cautions outlined in the "Notes, Cautions, and Warnings" section of the document to avoid potential damage to your device or personal injury. By following these steps, you can ensure a smooth and safe transition into using your Dell Mini 3i smartphone.	Before using the Dell Mini 3i smartphone, you need to set it up by installing the battery, SIM card, and charging the battery. Additionally, you must activate the phone by signing up for a service plan with a service carrier in your area and registering the phone with the network. Once activated, you can use the smartphone to send and receive messages, connect to Bluetooth devices, access the internet browser, and make calls using the dialer icon on the Home screen.	You have to install the battery, and SIM card, and charge the battery.
Multimodal	For the Dell cellphone, what is the function of the icons?	In the image you provided, there are several icons on the screen of a Dell cellphone. These icons typically represent different functions and features that can be accessed on the device. From top to bottom: 1. **Top left icon**: This usually represents the home screen or the app launcher. It allows users to view their installed apps, widgets, and shortcuts. 2. **Top center icon**: This is often the status bar, which typically shows information about the phone's connection status (such as Wi-Fi, mobile data), time, battery level, and other notifications or alerts. 3. **Top right icon**: This is commonly a settings or profile icon that allows users to access system settings, user account settings, or profile management. 4. **Bottom left icon**: These are likely quick launch icons for frequently used apps or functions. The exact function of each icon would depend on the specific layout and features of the device's operating system. 5. **Bottom center icon**: This is typically an app drawer or application launcher that allows users to browse and select from all installed spps. 6. **Bottom right icon**: These are usually shortcuts to frequently used functions or settings, such as a flashlight feature or a quick access panel for system functions like toggling Wi-Fi or Bluetooth on or off. The specific functions of these icons would be determined by the device's operating system and the user interface design. In this image, it appears to be a screenshot from an older model of a Dell cellphone running a version of the Android operating system, which is known for its customizable home screen and app launcher.	Based on the context provided, it seems that the image depicts a smartphone with various options and settings visible on its screen. The image includes several labeled numbers, which may indicate measurements or specifications related to the phone's design or dimensions. Additionally, there are two smaller figures visible at the top-left corner of the image, which might represent more options or settings related to the app being used or accessed. As for the function of the icons on the screen, it is likely that they are part of the interface or illustrations that accompany the user experience. The icons may represent different features or functions of the phone, such as a camera, internet browser, or music player. Alternatively, the icons could be used to indicate different settings or preferences related to the app being used. Without more information or context, it is difficult to provide a definitive answer to this question.	The icons on the Dell smartphone represent various functions and status indicators. For example, there are icons for incoming calls, missed calls, battery charge levels, signal strength, Bluetooth connectivity, and more. To learn more about these icons and the features they indicate, you can refer to the user's manual on the Dell Support website at supportapj.dell.com.	It indicates the new message, upcoming calendar event, alarm, or other information important to you.

Figure A.1: Generated responses by LLaVA-7B, the Text-only RAG and the Multimodal RAG framework for questions about the Dell manual.

Question complexity	Question	LLaVA-7B	Text-only RAG	Multimodal RAG	Ground-truth
Top page	For the Samsung vacuum cleaner, can I plug in my vacuum cleaner directly?	In the image provided, it shows an instructional label indicating that the vacuum cleaner should not be plugged in unless you are familiar with all instructions. If you are unfamiliar with the specific vacuum cleaner or the particular model of the Samsung vacuum cleaner, it is recommended to read and follow all instructions to ensure safe usage.	Based on the context provided, it is not recommended to plug in your vacuum cleaner directly without proper preparation. The instructions provided highlight several important safety precautions that must be taken before using the vacuum cleaner, including: 1. Reading all warnings and instructions before using the vacuum cleaner. 2. Not leaving the vacuum cleaner unattended when it is plugged in. 3. Unplugging the vacuum cleaner from the outlet when not in use and before servicing. 4. Not using the vacuum cleaner on wet surfaces. 5. Not allowing children to operate the vacuum cleaner or use it as a toy. The labels provided in the image also highlight the various parts of the vacuum cleaner, including the dust bin, hose pipe, and extension wand. These labels can help you understand how to properly assemble and use the vacuum cleaner for optimal cleaning results. Therefore, based on the safety instructions and labels provided, it is important to carefully read and follow all instructions before using the Samsung vacuum cleaner to ensure safe and effective operation.	Based on the provided context, you can plug in your Samsung vacuum cleaner directly to an electrical outlet for operation. However, it is important to note that when not in use, you should unplug the power cord and store it securely to avoid accidents. Additionally, you should always follow safety precautions such as avoiding the use of the vacuum cleaner on flammable or combustible materials, picking up hard, sharp objects with the cleaner, and storing the hose securely when not in use.	No, you can't. You should be familiar with all instructions and operating procedures first.
Middle page	For the Samsung vacuum cleaner, what should I do if the Mini Turbo Brush is clogged or debris is wrapped around the brush roll?	The image you've provided is a troubleshooting guide for common issues that can occur with vacuum cleaners. Specifically, the section that addresses the Mini Turbo Brush shows several solutions: 1. **Problem: Vacuum brush roll clogged or debris wrapped around the brush roll** - **Solution: Remove the debris and clean the bristles** - Check if the bristles are properly attached to the brush roll. - If necessary, remove any stuck debris with your hands or a small tool. - If the debris is tightly wrapped around the brush roll, you can try using a pair of scissors or pliers to carefully cut it free, taking care not to damage the roll. - After removing the debris, clean the bristles by gently running them under lukewarm water with some dishwashing liquid. Avoid using hot water, as this could damage the brush roll. - Thoroughly rinse and dry the bristles. Make sure to remove any excess water before reinstalling it back into your vacuum cleaner. Please note that if you are unsure about performing these steps yourself or if the issue persists, it is recommended to contact a service center for professional assistance.	For the Samsung vacuum cleaner, if the Mini Turbo Brush is clogged or debris is wrapped around the brush roll, follow these steps to resolve the issue: 1. Turn off the vacuum cleaner and unplug it from the power source before attempting any cleaning or repair tasks. 2. Locate the Mini Turbo Brush on the vacuum cleaner and identify the clogged area or debris wrapped around the brush roll. Use a soft-bristled brush or cloth to gently remove any visible debris or blockages. 3. If the debris is too tightly wedged or the brush roll is damaged beyond cleaning, proceed with caution to avoid damaging the vacuum cleaner further. Use pliers or tweezers to carefully remove any stubborn debris without applying excessive force. 4. Once the clog is cleared or the debris removed, test the Mini Turbo Brush by rotating it gently and checking if it spins freely without any blockages. If it still doesn't work properly, consider replacing the brush roll or consulting a professional for further assistance. 5. Always refer to the user manual or manufacturer's instructions for proper maintenance and cleaning procedures of your Samsung vacuum cleaner to avoid damaging the device or compromising its performance.	If the Mini Turbo Brush of your Samsung vacuum cleaner is clogged or debris is wrapped around the brush roll, you should follow these steps to fix it: 1. Turn off the vacuum cleaner before attempting any maintenance work. 2. Locate the screws that secure the brush cover in place. They are typically located on the bottom side of the vacuum cleaner. Remove the screws using a screwdriver. 3. Carefully lift the brush cover off the vacuum cleaner to access the brush roll and motor pulley areas. 4. Use a soft-bristled brush or tool to remove any strings, hair, and other debris from the rotating floor brush and motor pulley areas. Be sure to check both areas for any obstructions. 5. Check the drive belt for any signs of wear, cuts, or stress. Replace it if necessary. 6. If the drive belt is in good condition, stretch it over the motor pulley and down into its designated area on the rotating floor brush. The drive belt should center itself on the motor pulley automatically. 7. Once the drive belt is properly installed, replace the brush cover back onto the vacuum cleaner. Make sure to securely fasten the screws in place. 8. Turn the vacuum cleaner back on and test it to make sure it is working correctly. If you encounter any issues during this process or if you need further assistance, please do not hesitate to contact Samsung customer support for help.	Please maintain Mini Turbo Brush, clear clog (see page 8)/
Multimodal	For the Samsung vacuum cleaner, where should I install the Combination Tool?	To install the Combination Tool, follow these steps: 1. Turn off the vacuum cleaner and unplug it from the power source. 2. Locate the lower intake wand on the back of the vacuum cleaner. 3. Slide the wand out completely to expose the inner hose. 4. Unclip the handle of the Combination Tool, if applicable. 5. Align the inner hose of the Combination Tool with the end of the vacuum cleaner's lower intake wand. 6. Push the Combination Tool onto the vacuum cleaner's lower intake wand until it clicks into place. 7. Ensure that the Combination Tool is securely attached to prevent any damage or injury. Remember to follow all safety precautions when attaching and using vacuum cleaner tools, and consult your vacuum cleaner's user manual for specific instructions related to your model.	The Combination Tool for the Samsung vacuum cleaner can be installed in various locations depending on your cleaning needs and preferences. Here are some options to consider: 1. Extension Wand: The Combination Tool can be attached to the extension wand of the vacuum cleaner, providing you with more reach and flexibility when cleaning hard-to-reach areas such as stairs, ceilings, and corners. 2. Dusting Brush: You can install the Combination Tool on the dusting brush of the vacuum cleaner for quick and easy cleaning of upholstery, curtains, and other delicate surfaces. 3. Floor Cleaner Tool: Attach the Combination Tool to the floor cleaner tool of the vacuum cleaner for deeper cleaning of carpets, rugs, and hardwood floors. This tool is particularly useful for removing dirt and debris from tight spaces and crevices. 4. Handheld Vacuum: If you want a more portable cleaning solution, you can detach the Combination Tool from the vacuum cleaner and use it as a handheld vacuum for cleaning small areas, stairs, and upholstery. Remember to always refer to the user manual for specific installation instructions and guidelines for your Samsung vacuum cleaner.	The Combination Tool should be installed on top of the vacuum cleaner's dust bin, by lifting and rotating the handle of the tool to select one of its two types of cleaning: the Brush side for dusting furniture, blinds, books, lamps, shelves, and baseboards or the Upholstery side for curtains, draperies, Dusting Brush 'cushions, fabrics.	You should attach it to the holder on the left side of the vacuum (as it is shown on the pic)*
Scattered	For the Samsung vacuum cleaner, what should I do after removing a screw in brushbar?	The image you've provided shows a Samsung vacuum cleaner with instructions on how to replace the brushbar. After removing a screw from the brushbar, you should: 1. Clean any accumulated dust or debris from the brush bar using a soft-bristled brush or compressed air. This will help remove any pet hair, dust, or other small particles that may have become entangled in the brush bars. 2. Use an appropriate tool (such as a vacuum cleaner itself or a dedicated cleaning tool) to gently remove any remaining dirt or debris from between the brush bars and the vacuum's body. 3. If necessary, you can use a soft-bristled brush to help loosen any stubborn dirt or debris that may be stuck in between the brush bars and the vacuum's body. 4. Once all debris is removed, reattach the screw that was previously holding the brush bar in place, ensuring it is tightened securely. 5. Reconnect any additional parts that were temporarily disconnected during the cleaning process (such as hoses, filters, or other attachments). 6. Ensure that the vacuum cleaner is properly assembled and all parts are securely fastened before using it again. This may include reattaching any brush bars, hoses, filters, or other components. 7. Test the vacuum to ensure that it is functioning correctly and that there is no debris or buildup interfering with its operation. Following these steps will help keep your vacuum clean and functioning properly.	After removing a screw in the brush bar of the Samsung vacuum cleaner, you should carefully remove the brush bar from the vacuum cleaner. This will allow you to access and replace any worn or damaged brushes inside the bar. Here are some general steps you can follow to replace the brushes in your Samsung vacuum cleaner: 1. Unplug the vacuum cleaner from the power outlet before starting any maintenance or repairs. 2. Remove the brush bar from the vacuum cleaner by gently pulling it away from the housing. Be careful not to touch any of the internal components, as they may be damaged or broken. 3. Locate the brushes inside the brush bar. They are usually held in place by small screws or clips. Remove these screws or clips to access the brushes. 4. Gently remove any debris or dust from the brushes using a soft-bristled brush or a clean cloth. Be careful not to damage the brushes themselves. 5. Examine the brushes for any signs of wear or damage. If they are damaged beyond repair, you may need to replace them with new ones. 6. Install the new brushes in the reverse order of how you removed the old ones. Make sure they are securely fastened and will not come loose during use. 7. Reassemble the brush bar and reattach it to the vacuum cleaner housing. Make sure everything is securely tightened before turning on the vacuum cleaner. 8. Test the vacuum cleaner to ensure that all parts are working properly and there are no leaks or other issues. It's important to note that some Samsung vacuum cleaners may have slightly different brush bar designs, so be sure to consult your owner's manual for specific instructions on how to replace the brushes in your particular model.	After removing a screw from your Samsung vacuum cleaner, you should follow the steps provided in the manual to properly maintain and operate the cleaner: 1. Place handle in Low position and turn vacuum, cleaner over so bottom side is up. 2. Remove the six screws as shown with screwdriver, lift the Brush Cover off. 3. Clean strings, hair and other debris from Rotating Floor brush, Motor Pulley and Drive Belt areas. 4. Check Drive Belt carefully for wear, cuts or stressed areas. 5. Clean Air Passageway for obstructions and remove debris from this area. 6. Replace Drive Belt if necessary. 7. Unplug cleaner from electrical outlet before replacing the bottom plate for maintenance. 8. Use only on dry, indoor surfaces, never change the plug in any way or modify the polarized plug to fit a non-polarized outlet or extension cord. 9. Do not touch the brush roller when it's rotating and keep suction away from your face when using nozzles and pipe. 10. Keep the vacuum cleaner away from sharp corners and do not allow it to be trapped. 11. Do not use the Mini Turbo Brush (Select Models) for areas where flammable or combustible materials may be present, and always unplug before connecting turbine nozzle. 12. Store the cord by grasping the polarized plug and looping it around the cord wraps on the back of the vacuum cleaner. Remember to empty the Dust Bin when the dirt reaches the "MAX" ZX WARNING line, and always follow the maintenance steps provided in the manual to ensure maximum cleaning performance.	You can remove the belt guard, then remove the brushbar.

Figure A.2: Generated responses by LLaVA-7B, the Text-only RAG and the Multimodal RAG framework for questions about the Samsung manual.

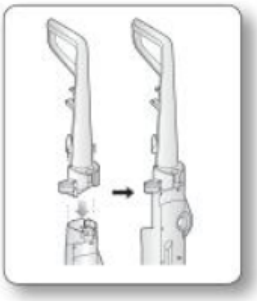
Question complexity	Question	LLaVA-7B	Text-only RAG	Multimodal RAG	Ground-truth
Top page	For the Sony laptop, how can I adjust the 3D settings?	The image you've provided appears to be a screenshot of a 3D settings menu for the Sony laptop, specifically showing how to adjust the 3D settings. To adjust the 3D settings on your Sony laptop, follow these steps: 1. **Click Start/Windows icon**: This will open the start menu or the windows desktop. 2. **Right-click Computer and select Properties**: You can access this by right-clicking the "Computer" or "This PC" icon on the desktop or from the start menu. 3. **Go to Display settings**: In the properties window, click on the "Display settings" link under the "Display" tab. 4. **Click Adjust resolution**: Once you are in the display settings window, look for the "Adjust resolution" option and click on it. 5. **Set 3D effect as desired**: This is where you can adjust the 3D effects for your computer screen. Look for options such as "3D Effects", "3D Display Mode", or similar settings and set them to your preference. Please note that the exact steps might vary depending on the specific model of your Sony laptop and the operating system it is running. If you are unable to find these specific settings or if they do not exist, check your laptop's user manual for more detailed instructions or contact Sony's customer support for further assistance.	To adjust the 3D settings on your Sony laptop, follow these steps: 1. Click on the Start menu and select Settings. 2. In the Settings window, click on the "Display" option. 3. In the Display window, click on the "3D" tab. 4. Under the "3D" tab, you will see options to adjust the 3D settings. You can choose the type of 3D content you want to play, such as Blu-ray 3D discs or 3D photos. 5. Adjust the brightness and contrast of the 3D image to your liking. 6. If you are using a 3D display, you can adjust the 3D depth level to enhance the 3D experience. 7. Once you have made your adjustments, click "Apply" to save your changes. Note: The exact steps may vary depending on the Sony laptop model and its specific features. Refer to the user manual or online support resources for more detailed instructions.	To adjust the 3D settings for your Sony laptop, first click on Start, All Programs, then open VAIO Control Center, click on "3D Effect Settings" in the left panel of the window, and adjust the 3D effect as needed.	You can start the software that su[]prts 3D photo or video playback and start 3D photo or video playback, then turn on the 3D glasses and wear them, and then press the 3D button on your computer. The 3D indicator light turns on during the 3D playback.
Middle page	For the Sony laptop, I like suring the internet on the computer tilting my head. If I do the same thing while viewing 3D images, what would happen?	If you tilt your head while viewing 3D images, depending on the technology used and how it's designed to function, there are a few possible outcomes: 1. **Misalignment**: If the system relies on sensors or tracking devices to determine its position in relation to the viewer, tilting your head could cause misalignment. This might lead to incorrect perspective or distortion in the 3D image. 2. **Image Shift**: Some 3D systems use stereoscopic technology where each eye sees a slightly different image. Tilting your head might make it difficult for both eyes to align properly, which could cause the image to shift or change perspective. 3. **Viewing Comfort**: For some viewers, tilting their head may be uncomfortable and could lead to discomfort or strain on the neck. 4. **Depth Perception**: In some cases, especially for virtual reality systems where you're wearing a headset, tilting your head might affect depth perception, as the sensors are typically positioned on the front of the device and tilted head could interfere with their alignment. To avoid these issues, it's generally recommended to maintain a neutral head position when viewing 3D images, especially if you're using an active tracking system like those used in VR headsets or some 3D monitor setups. However, for passive viewing systems that don't require tracking, tilting your head is unlikely to cause significant issues unless the image content is sensitive to viewpoint changes.	When viewing 3D images on your Sony laptop, it's important to follow the instructions in the manuals that came with your computer to avoid any discomfort or eye strain. Some people may experience eye strain, fatigue, or nausea while watching 3D video images or using stereoscopic 3D-related software. To minimize the risk of discomfort, Sony recommends that all viewers take regular breaks while watching 3D video images or using stereoscopic 3D-related software. The length and frequency of necessary breaks will vary from person to person, so you should decide what works best for you. If you experience any discomfort, stop watching the 3D video images or using stereoscopic 3D-related software until the discomfort ends; consult a doctor if necessary. It's also important to position yourself correctly while viewing 3D content. Sony suggests that you sit at a comfortable distance from the screen, ideally between 1.5 and 2.5 times the screen's diagonal measurement. You should also ensure that the screen is perpendicular to your line of sight and that there are no obstructions between you and the screen. Remember, it's essential to prioritize your eye health when viewing 3D content on your laptop. If you have any concerns or experience discomfort during use, consult a doctor or an optometrist for professional advice.	If you tilt your head while viewing 3D images on your Sony laptop, it may affect the 3D effect or change the color of the images. It is recommended to adjust the angle of the LCD screen so that your eyes are perpendicular to your computer screen and maintain a distance of at least 2 feet (60 cm) from the screen. Additionally, take regular breaks while viewing 3D video images or using stereoscopic 3D-related software to avoid discomfort such as eye strain, fatigue, or nausea.	The 3D effet might be less pronounced or the colour of the images may change.
Scattered	For the Sony laptop, how can I view 3D photos or videos?	To view 3D photos or videos on a Sony laptop, you can use one of the following methods: 1. Use an external 3D display device such as a 3D monitor, 3D TV, or 3D projector that is compatible with your laptop. 2. Install specialized software for viewing 3D content on your laptop's screen, which may require the use of 3D glasses to view the content in 3D. 3. Use a pair of 3D glasses that are compatible with your laptop if it supports stereoscopic display technology. 4. Utilize a virtual reality headset or VR headset that is compatible with your laptop and the 3D software you have installed. 5. If your laptop supports it, use an NVIDIA Vision Workstation (VW) to view and process 3D image data. 6. For 3D photos, you can use 3D photo editing or viewing software that is designed for this purpose. 7. For 3D videos, you can use a media player that supports 3D video playback, such as VLC Media Player or MX Player with the appropriate plugins or codecs installed. 8. If your laptop has a built-in 3D display capability like an OLED panel (which is not common), then you should be able to view 3D content directly on the screen without any additional equipment. Remember that in some cases, you may need specialized software or drivers for your specific laptop model to enable 3D functionality. Always check the compatibility of any hardware or software with your device before proceeding.	To view 3D photos or videos on your Sony laptop, follow these steps: 1. Ensure that your laptop has the latest drivers and software updates installed. 2. Connect your laptop to a 3D-enabled display device (such as a 3D TV or monitor) using an HDMI cable or wireless connection. 3. Open the Photos or Videos application on your laptop. 4. Locate the 3D content you wish to view (e.g., 3D photos or videos). 5. Click on the "Play" button to start playing the content. 6. Adjust the 3D settings in the display device's settings menu according to your preference. Some common options include adjusting the 3D depth, brightness, and contrast. 7. Enjoy your 3D content! Note: If you don't have a 3D-enabled display device, you can also view 3D content on your laptop using built-in stereoscopic 3D capabilities. To enable this feature, go to the Display settings in your laptop's settings menu and select "Stereoscopic 3D" or a similar option.	To view 3D photos or videos on a Sony laptop, you should start the software that supports 3D photo or video playback and turn on the 3D glasses before pressing the 3D button on your computer to activate the 3D indicator light during playback. You can switch between 3D and 2D images by pressing the 3D button, but keep in mind that some people might experience discomfort while viewing 3D content, so it's essential to take regular breaks if needed and consult a doctor if you experience any discomfort. Also, ensure that young children are supervised when watching 3D video images or using stereoscopic 3D-related software.	You can click Start, All Programs, and VAIO Control Center, then click 3D Effect Settings in the left pane of the window to adjust the 3D effect.
Multimodal	For the Sony laptop, why I cannot find the 3D button and the 3D indicator on my computer when I display 3D images on 3D TV?	The image you've shown is a guide on how to connect and configure a Sony 3D TV with a computer for displaying 3D content. It specifically mentions steps related to using HDMI cables and changing settings within the computer's operating system (which appears to be Windows in this case). The reason you might not find a dedicated "3D" button or indicator on your computer when displaying 3D images is because the 3D functionality is integrated into your operating system, and it typically relies on hardware that is compatible with displaying 3D content. This includes having a compatible monitor or TV, as well as ensuring that your graphics card supports 3D rendering. In Windows, for instance, you would typically need to have an Nvidia or AMD graphics card that supports OpenGL and OpenCL, which are required for some 3D applications to run in stereo mode. Additionally, the operating system should be capable of handling 3D content at a software level. If your computer meets these requirements but you're still unable to find the 3D button or indicator, it could be that the application you are using to view or play 3D content does not have native support for displaying 3D images. In this case, you might need to look for third-party software that can handle and convert 2D images into 3D format on the fly. Remember that when connecting a Sony 3D TV to a computer, you must use an HDMI cable compatible with 3D signals, as non-3D HDMI cables will not transmit 3D information from your computer to the TV. The guide in the image provides instructions for this process, including adjusting settings within your operating system and ensuring that the right ports are used on both the computer and the 3D TV.	The 3D function on the Sony laptop is designed to work with compatible 3D content and 3D-capable TVs. If you are not seeing the 3D button or indicator on your computer when displaying 3D images on a 3D TV, there could be several reasons why: 1. Compatibility issues: Make sure that the 3D content you are trying to play is compatible with the Sony laptop's 3D function. Some content may not be supported or may require additional software or hardware configurations. 2. 3D TV compatibility: Ensure that your 3D TV is compatible with the Sony laptop's 3D signal. Check the TV's manual or manufacturer's website for information on its 3D capabilities and connectivity options. 3. HDMI connection: Make sure that the Sony laptop is connected to the 3D TV via an HDMI cable. If the connection is not proper, you may not see the 3D button or indicator on the computer. 4. Display settings: Check your computer's display settings to ensure that it is set to display 3D content correctly. You may need to adjust the display settings in the laptop's settings menu or use the TV's built-in settings menu. 5. Software issues: Ensure that the Sony laptop's software is up-to-date and compatible with the 3D function. Check for any software updates or patches that may be needed to resolve compatibility issues. 6. Hardware issues: In some cases, hardware issues can prevent the 3D function from working correctly. Check the laptop's hardware components, such as the graphics card or display panel, to ensure they are functioning properly. If you have checked all of these potential issues and are still unable to find the 3D button or indicator on your computer, contact Sony support for further assistance.	The 3D button and the 3D indicator on your computer are not available when 3D images are displayed on the 3D TV.	The 3D button and indicator are not available on your computer while displaying 3D images on a 3D TV. If images are not displayed in 3D on a 3D-capable TV, you can follow the steps to display initial settings for NVIDIA 3D Vision on the TV screen and complete the initial settings. To disable the 3D function, you can use VAIO Control Center. The message about NVIDIA 3D Vision may appear on the bottom-right corner of the playback window during playback depending on the instructions in the manuals that came with your computer when viewing 3D images. Always wear the supplied 3D glasses while viewing 3D images and adjust the angle of the LCD screen so that your eyes are perpendicular to your computer screen and keep a distance of 2 feet (60 cm) or more from the screen. The range of the 3D sync transmitter on your computer may vary depending on ambient environment and location where you use the computer.

Figure A.3: Generated responses by LLaVA-7B, the Text-only RAG and the Multimodal RAG framework for questions about the Sony manual.

Assembling the cleaner

ATTACH HANDLE TO VACUUM

1. Stand the vacuum upright and slide the base of the handle firmly into the grooves at the top of the vacuum body.
2. Check to be sure the handle is fitted to the vacuum body. If there is a gap, continue to push firmly until the handle is flush.
3. Insert the screw provided from the screw packet into the existing hole.
   Tighten securely with a screwdriver.



ATTACH DUST BIN TO VACUUM

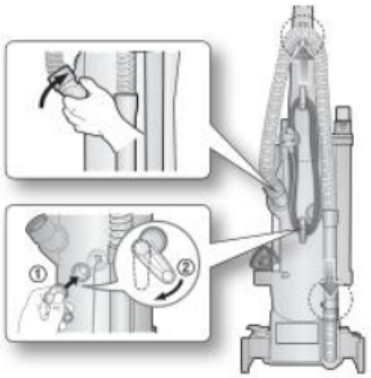
Fit the groove located on the lower front side of the Dust Bin to the hook located on the vacuum body and attach the hook located on the Dust Bin Handle to the vacuum body as shown.

ATTACH VACUUM HOSE AND TOOLS

1. Attach the Twist Snap Hose by lining up the taps on the base with the corresponding notches on the back of the vacuum as shown.
2. Turn the Twist Snap Hose clockwise to lock the hose into place.
3. Grasping the Hose Wand, slide the end of the it onto the Hose Wand Base. Snap the Hose into the Hose Clip.

Note: The vacuum will not operate effectively unless the Hose Wand is firmly attached.

4. Attach the Lower Cord Wrap by lining up the tab on the base with the corresponding notch on the back of the vacuum and turn it clockwise as shown.
5. Attach the Extension Wand and the Crevice Tool together into the holder on the right side of the body as shown.
6. Attach the Combination Tool into the holder on the left side of the body as shown.
7. Attach the Bracket Brush on the front of the handle and clip the Mini Turbo Brush into the Bracket Brush as shown.



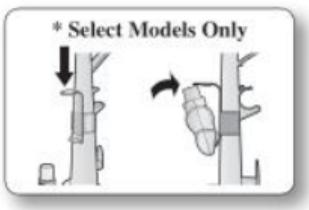


Figure A.4: An example page from one of the manuals chosen for evaluation. V