Distribution Agreement

In presenting this thesis as a partial fulfillment of the requirements for a degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis in whole or in part in all forms of media, now or hereafter now, including display on the World Wide Web. I understand that I may select some access restrictions as part of the online submission of this thesis. I retain all ownership rights to the copyright of the thesis. I also retain the right to use in future works (such as articles or books) all or part of this thesis.

Andrew Chung March 24, 2025

Low Resource RAG: From Slide Data Processing to RAG Systems

By

Andrew Chung

Jinho D. Choi
Advisor

Computer Science

Jinho D. Choi, Ph.D.
Advisor

Joyce C. Ho, Ph.D.
Committee Member

Sharon Sonenblum, Ph.D.
Committee Member

2025

Low Resource RAG: From Slide Data Processing to RAG Systems

By

Andrew Chung

Jinho D. Choi, Ph.D.
Advisor

An abstract of
A thesis submitted to the Faculty of the Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Computer Science

2025

Abstract

Low Resource RAG: From Slide Data Processing to RAG Systems

By Andrew Chung

Large Language Models (LLMs) have emerged as a transformative force in technological development over the past few years. These models have been widely integrated across educational, research, and business applications, serving as tools to enhance learning, a source of curiosity for research exploration, and streamline business operations in both internal and customer-facing systems. While LLMs offer diverse capabilities, one of their most sought-after application across all different sectors has been their potential to provide precise, contextual information and insights from domain-specific knowledge bases. In this context, Retrieval-Augmented Generation (RAG) has emerged as the leading framework for leveraging LLMs' capabilities while maintaining accuracy and reliability.

To advance the understanding and development of successful retrieval-augmented generation systems, we examine various components to identify essential elements and potential performance improvements across different methodologies. Through collaboration with Hyundai, we develop a low-resource domain retrieval-augmented generation system designed to answer questions about automotive safety collision tests using information from multimodal slides. Our approach introduces a novel, language model-centric data processing pipeline that effectively transforms slide information into textual content suitable for retrieval and answer generation. We evaluate the performance of different state-of-the-art retrieval-augmented generation frameworks on our processed data, as well as different variations of embedding models. To assess our system's effectiveness, we generate synthetic question-answer pairs from our refined data to test the accuracy of different retrieval models. Furthermore, we create additional synthetic question-answer pairs specifically targeting the multimodal table and chart information extracted from the slides. Our findings indicate that utilizing fine-tuned embedding models and language models with the original retrieval-augmented generation framework achieves the highest accuracy. We also finetune Vision Large Language Models to see if open-sourcing our data processing pipeline is possible. We conclude by outlining next steps to encourage research toward developing open-source retrieval-augmented generation frameworks for low-resource domains.

Low Resource RAG: From Slide Data Processing to RAG Systems

By

Andrew Chung

Jinho D. Choi, Ph.D.
Advisor

A thesis submitted to the Faculty of the Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Computer Science

2025

Acknowledgments

I would like to acknowledge Dr. Jinho Choi for his continuous guidance and sound advice, as well as his unwavering patience and fostering my intellectual curiosity in my academic journey. I would also like to thank undergraduate student Ellie Paek and the PhD students Grace Byun and Nayoung Choi for their assistance and collaborative efforts. I am deeply grateful to Shinsun Lee for his trust in my capabilities with the Hyundai Project and for providing me the opportunity to collaborate with our talented group of people on this unique and rewarding initiative. This thesis would not be possible without everyone's expertise and enthusiasm for this project.

Last but definitely not least, I am grateful to my family, whose steadfast emotional, financial, and spiritual support has enabled me to pursue my education and research at Emory University with undivided focus and dedication. None of this would be possible without them.

Thank You All.

Contents

1 Introduction **1**

1.1 Background and Motivation 1

1.2 Research Objective 4

1.3 Thesis Statement 5

2 Background **6**

2.1 Background and trends in NLP and information systems 6

2.2 Exploring Retrieval-Augmented Generation 7

2.3 Embedding Models 10

2.4 Fine-tuning 11

2.5 Multimodal Data Processing in RAG 12

2.5.1 Synthetic QA and Data Generation 14

3 Approach **15**

3.1 Data Processing 15

3.1.1 Dataset 15

3.1.2 Slide Data Processing 17

3.1.3 Additional Headers 18

3.1.4 Synthetic QA Generation 19

3.1.5 Evaluation of Synthetic Data 20

3.2 Embedding Model - BGE-M3 21
3.2.1 Fine-Tuning 22
3.3 RAG Frameworks 22
3.3.1 Frameworks 22
3.3.2 Storage of Data 23
3.3.3 Evaluation of Retrieval 25
3.4 Open Source VLLMs for OCR Processing 26
3.4.1 Fine-tuning VLLM 27
3.4.2 Data Preparation 27
3.4.3 Evaluation of VLLM 28

4 Experiments **32**
4.1 Evaluation of RAG and Embedding Models 32
4.2 Evaluation of Table and Chart Data 35
4.3 Evaluation of VLLM 36

5 Analysis **39**
5.1 Synthetic Data 39
5.1.1 Qualitative Analysis 39
5.1.2 Quantitative Analysis 44
5.2 RAG Frameworks & Retrieval Analysis 46
5.2.1 Embedding Models 46
5.2.2 RAG Frameworks 46
5.3 Qwen 2.5 VL Fine-tuning Analysis 47
5.3.1 Image Dimensions 47
5.3.2 Base Model vs. Finetuned Model Data 48
5.3.3 LLM Judge Preference - Qualitative Analysis 49

6 Conclusion **51**

6.0.1 Future Directions 51

6.0.2 Conclusion 52

A Appendix **54**

A.0.1 Converting image-to-text prompt: 54

A.0.2 Correction of results: 56

Bibliography **57**

List of Figures

1.1 The general concept of RAG. Utilize an external database to have domain-specific/updated information. 2

2.1 The retrieval-augmented generation framework. A user query is inputted into the system, where it allows for document retrieval and answers the user question based on retrieved documents through LLM answer generation. 8

3.1 An example of potential elements that exist within slide data. Includes complex tables, charts, and more. 16

3.2 A 3-stage process for data processing of slide data. Each process further refines or extrapolates synthetic data from the original slide presentation. 17

3.3 Example Header that was prepended to all slide chunks 19

3.4 Each slide of each slide presentation is stored as a chunk in the database and given a dense vector representation through the embedding model. 24

4.1 Bar graph showing the distribution of rankings assigned by an LLM-as-Judge evaluator. The chart uses the the original synthetic data as ground truth, comparing the best-performing finetuned models against our original synthetic data. 38

5.1 Each color represents a phrase or word as it goes through the process and shows character correction. The text in the process 2 results matches the original slide, while process 1 contains mistakes. 40

5.2 The Red box indicates a structural change that occurs through the self-correction step in process 2. 41
5.3 An image is given to the LLM to generate a descriptive explanation that showcases the image's purpose. 42
5.4 An image of a chart is given to the LLM to generate a descriptive portrayal of the chart, picking out key values and parameters. 43
5.5 We highlight different corresponding areas of each of the tables in different colors, showcasing the correctness of each of the outputs from the different finetune variants. 49

List of Tables

3.1 Raw data statistics by source, along with the corresponding number of generated Q&A pairs. 15

4.1 Comparison of RAG vs. HyDE for Vanilla-BGE and Fine-Tune BGE. 32
4.2 Vanilla BGE-M3 33
4.3 BGE-M3 Fine-tuned on slide and explanation data 34
4.4 BGE-M3 Fine-tuned on slide data 34
4.5 Example results for Fine-Tune BGE on slides vs. explanations. 35
4.6 Comparison of different image dimension configurations for finetuning Qwen2.5-VL-7B, grouped by configuration. 36
4.7 Comparison of BERT-Score metrics (in percentages) grouped by image dimensions. 37

5.1 Token Count Analysis Statistics(n=63) 45

Chapter 1

Introduction

1.1 Background and Motivation

Large Language Models (LLMs) have emerged as powerful tools for question answering across diverse domains. Various benchmarks evaluate LLMs on tasks ranging from mathematics and chemistry to logical reasoning. However, LLMs are inherently constrained by their training data, leading to three critical limitations in their question-answering capabilities:

1. Inability to access post-training information and current events
2. Persistence of outdated information as presumed facts
3. Limited comprehension of domain-specific knowledge for specialized applications

While newer LLM iterations incorporate more recent information, the challenge of domain specificity remains significant for organizations seeking to leverage LLM-based question-answering systems for specialized use cases. Creating and customizing information systems for specific data types or organizational requirements continues to be a pressing challenge. Moreover, continuous LLM training proves impractical for most applications due to its substantial computational and financial demands. In response to these

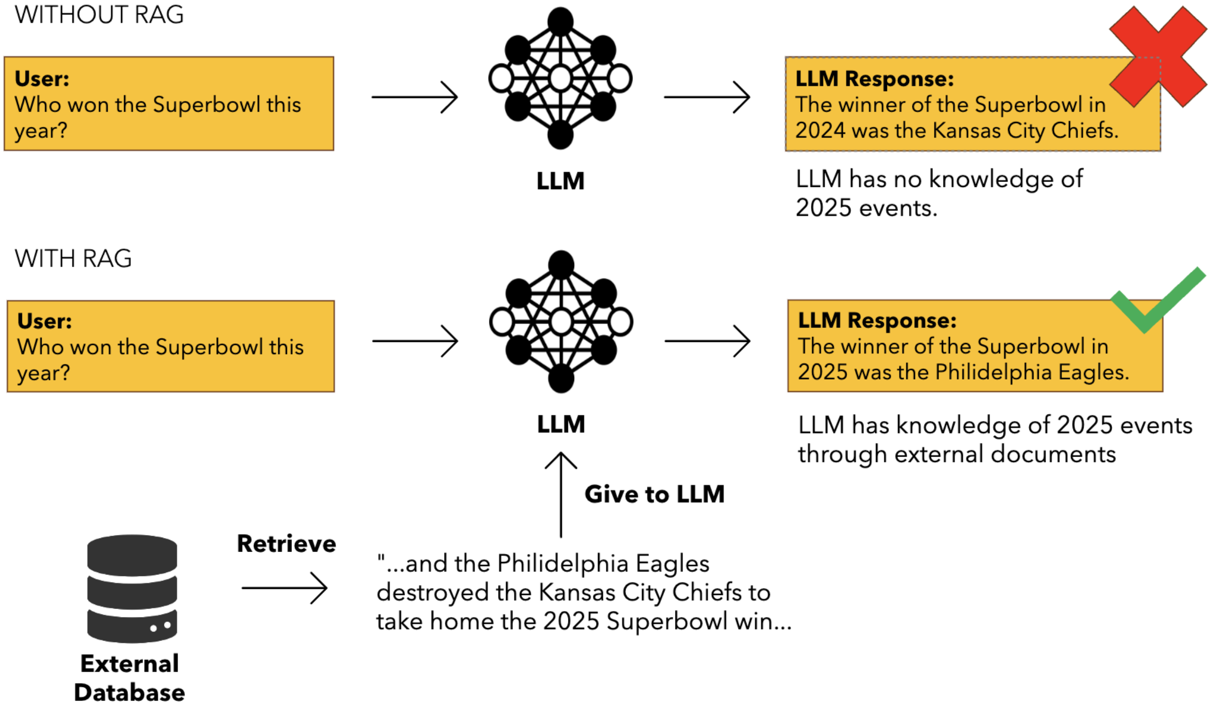

WITHOUT RAG
User:
Who won the Superbowl this year?
LLM
LLM Response:
The winner of the Superbowl in 2024 was the Kansas City Chiefs.
LLM has no knowledge of 2025 events.
WITH RAG
User:
Who won the Superbowl this year?
LLM
LLM Response:
The winner of the Superbowl in 2025 was the Philidelphia Eagles.
LLM has knowledge of 2025 events through external documents
Give to LLM
Retrieve
"...and the Philidelphia Eagles destroyed the Kansas City Chiefs to take home the 2025 Superbowl win...
External Database

Figure 1.1: The general concept of RAG. Utilize an external database to have domain-specific/updated information.

limitations, retrieval-augmented generation [1] (RAG) has emerged as the current state-of-the-art solution.

RAG addresses the challenge of incorporating novel information by retrieving relevant contextual information from an external database based on user queries. This retrieved information is then provided to an LLM alongside the query, enabling the generation of informed responses. Following RAG's introduction, numerous frameworks and methodologies have been proposed, each claiming superior performance. In our research collaboration with Hyundai, we faced the challenge of developing a question-answering system for novel multimodal data in a low-resource domain. This led us to investigate various RAG frameworks, embedding models, and data processing methodologies to create an effective system for handling multimodal data in safety collision testing and automotive applications. Additionally, we were tasked to build an open-source RAG system without utilizing closed-source models like GPT or Claude due to privacy and security concerns.

A notable aspect of our research involves processing visual information, including tabular data and crash test imagery. Our findings suggest that while limitations exist, visual information can be integrated through text conversion, enabling the development of a functional RAG system. Furthermore, we aimed to develop a RAG model specifically optimized for the Korean language, adding another layer of complexity to our low-resource domain challenge. We find that the fine-tuning of LLMs and embedding models lead to significant improvements and usability in Korean-based RAG systems and allows the models to learn about the low resource car domain. Thus, our primary objective is to develop a sustainable RAG system and robust data conversion pipeline capable of continuously integrating new test data while providing accurate and fluent responses to user queries in Korean. To investigate this, we explore the following questions:

1. How can recent advances in multimodal large language models be leveraged to effectively process diverse data types (visual, positional, and graphical) within retrieval-augmented generation systems?

2. What methods can improve the quality and usability of textual representations derived from multimodal information, and how do these enhancements affect retrieval performance and answer generation?

3. What factors influence the effectiveness of different RAG architectures across languages and domains, and what optimization strategies can improve performance in multilingual settings?

4. With the recent advancement of open-source vision large language models, are we able to capture the multimodal data through text as well as existing closed source models?

1.2 Research Objective

This thesis was developed in collaboration with Hyundai Motor Company as part of a broader research initiative [2]. The work focuses primarily on data processing methodologies and the evaluation of various retrieval-augmented generation frameworks. Our goal is to develop a system capable of effectively answering user questions based on multimodal vehicle collision data. The key contributions of this work include:

1. We develop a novel, LLM-centric data processing pipeline that can effectively convert slide information into a usable text format that takes into account positional, graphical, and visual information presented on slide images.
2. We conduct qualitative and quantitative analyses of LLM-generated data representations and evaluate retrieval performance across various data preprocessing strategies, including the incorporation of explanatory text and handling of tabular and graphical elements.
3. We systematically evaluate the retrieval performance of various Korean-language embedding models to assess the current capabilities of available open-source solutions for multilingual information retrieval
4. We conduct benchmarking of state-of-the-art RAG frameworks for the retrieval aspect using our domain-specific data to evaluate and compare retrieval effectiveness across different methodological approaches in Korean
5. We finetune Qwen2.5-VL-7B [3], a SOTA vision large language model to create an open source solution for our data processing pipeline.

Due to the nature of our domain, our research explores how retrieval-augmented generation systems perform within low-resource domains, such as the car domain. Through this research, we hope to present a reliable, open-source solution that can be applied for all different low-resource domains that contain multimodal information.

1.3 Thesis Statement

Our thesis investigates retrieval-augmented generation in low-resource environments and a develops method for creating synthetic textual data that represents multimodal elements. We evaluate various RAG methods and analyze how retrieval performance changes across different components of the RAG pipeline.

Chapter 2

Background

2.1 Background and trends in NLP and information systems

Natural Language Processing [4] (NLP) represents a significant intersection of computer science, computational linguistics, and machine learning dedicated to enabling computational systems to understand, interpret, and generate human language. This field has experienced unprecedented growth in recent years, primarily driven by advancements in deep learning architectures and the emergence of LLMs.

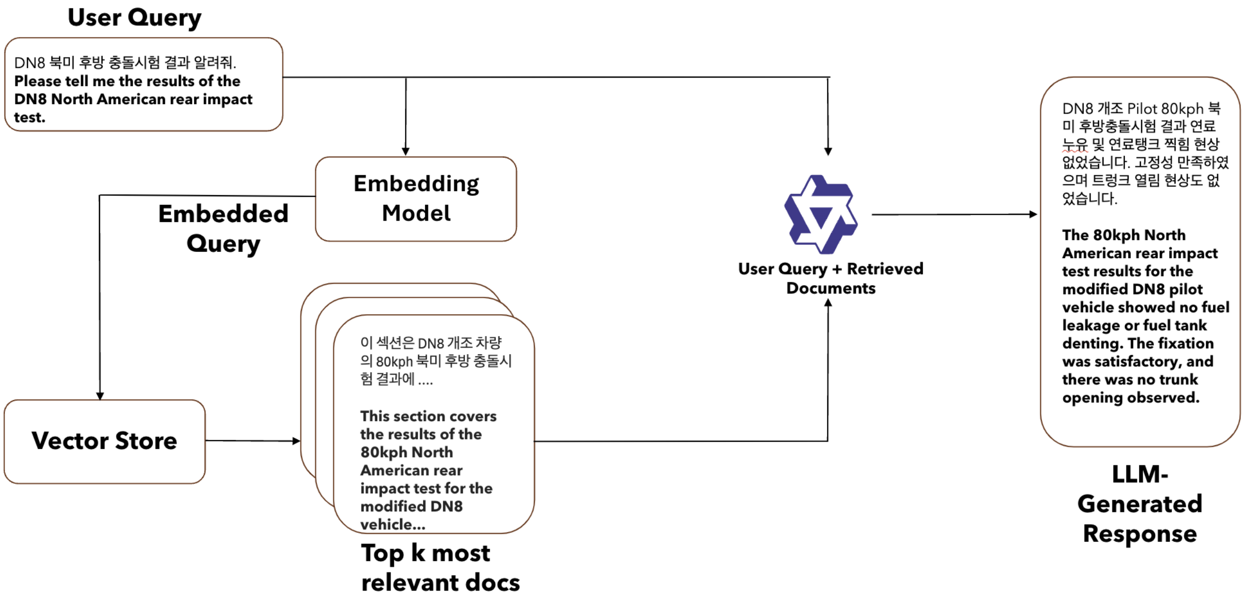
LLMs [5] have fundamentally transformed the NLP landscape, establishing themselves as the state-of-the-art approach for a diverse range of language-related tasks. These models, trained on vast corpora of text data, have demonstrated remarkable capabilities in understanding context, generating coherent responses, and performing complex linguistic tasks with human-like proficiency. Their impact extends beyond academic research, permeating various industrial applications through multiple implementation strategies: fine-tuning for domain-specific tasks [6] [2], deployment as autonomous agents capable of executing complex workflows [7], and integration into conversational systems that facilitate natural human-computer interaction [8].

Concurrent with the rise of LLMs, sophisticated embedding models [9] have been developed that capture semantic relationships by representing linguistic units—words, sentences, or entire documents—as dense vectors in high-dimensional space. This development from sparse to dense embedding representations have allowed for embedding models to encode richer semantic information, effectively mapping the complex relationships between linguistic elements while preserving their contextual meaning. This has enabled semantic search capabilities that transcend traditional keyword-based and sparse representation approaches. By computing similarity metrics between query vectors and document vectors, these models can identify conceptually related content even when exact term matches are absent.

There has been a significant shift across industries toward developing robust internal search systems for proprietary or restricted documents that cannot be indexed by conventional web search engines [10]. This trend reflects growing organizational needs to efficiently access and leverage institutional knowledge contained within private repositories. The combination of embedding models and LLMs has catalyzed a more natural language interface. Modern systems can now interpret user queries expressed in conversational language, retrieve relevant information based on semantic understanding, and generate comprehensive answers that synthesize information from multiple sources.

2.2 Exploring Retrieval-Augmented Generation

Retrieval-augmented generation [1](RAG), at its core, is a methodology enabling users to obtain domain-specific or up-to-date answers from an LLM. By accessing retrieved documents from an external database, the LLM generates answers based on this input rather than solely relying on internal knowledge. The core components of RAG can be distilled to: user queries, LLMs, embedding models, and a vector store or external database. Since RAG's emergence, researchers have modified various aspects of this framework. Works like



DN8 북미 후방 충돌시험 결과 알려줘.
Please tell me the results of the DN8 North American rear impact test.

이 섹션은 DN8 개조 차량의 80kph 북미 후방 충돌시험 결과에 ....

This section covers the results of the 80kph North American rear impact test for the modified DN8 vehicle...

DN8 개조 Pilot 80kph 북미 후방충돌시험 결과 연료 누유 및 연료탱크 찍힘 현상 없었습니다. 고정성 만족하였으며 트렁크 열림 현상도 없었습니다.

The 80kph North American rear impact test results for the modified DN8 pilot vehicle showed no fuel leakage or fuel tank denting. The fixation was satisfactory, and there was no trunk opening observed.

Figure 2.1: The retrieval-augmented generation framework. A user query is inputted into the system, where it allows for document retrieval and answers the user question based on retrieved documents through LLM answer generation.

RAG-FUSION [11] and Hypothetical Document Embeddings [12] (HyDE) have refined user queries through hypothetical answer generation and multi-query techniques. Other approaches enhance answer generation by fine-tuning LLMs for specific domain knowledge and implementing self-evaluation and correction mechanisms [13]. Embedding models have been fine-tuned and retrieval results re-ranked based on document relevance [2] [14]. Additionally, data preprocessing has been optimized through synthetic data generation, hierarchical structuring, and diverse chunking strategies using semantic or paragraph-based segmentation [15]. Methodologies like Recursive Abstractive Processing for Tree Organized Retrieval [16] (RAPTOR) have become a staple method for developing databases for long context retrieval through creation of synthetic generation based on clustered documents.

From the numerous RAG-oriented frameworks available, we selected two approaches that we believe would achieve optimal retrieval quality on our synthetically generated data. These frameworks include the original retrieval-augmented generation framework, Hypothetical Document Embeddings (HyDE), and Recursive Abstractive Processing for Tree Organized Retrieval(RAPTOR). Beyond the baseline RAG framework, the other three approaches enhance the retrieval component through either data manipulation or query manipulation/rewriting.

The HyDE framework improves retrieval performance—subsequently enhancing answer accuracy—by pre-generating a hypothetical answer to the user query. This generated answer, despite potentially containing hallucinated information, typically incorporates domain-specific terminology and concepts relevant to the user's inquiry. This approach enhances retrieval performance because the embedding generated from this hypothetical answer often exhibits stronger correlation with embeddings in the vector database through increased lexical and semantic similarity. We leverage this answer generation approach using our LLM fine-tuned on the vehicle collision safety domain. We attempt to utilize different variations of the HyDE framework by including questions

Additionally, we implemented the RAPTOR framework, which generates synthetic larger-context summaries for clustered documents. This approach theoretically enhances retrieval by providing higher-level conceptual information alongside specific details, giving the model a more comprehensive understanding of the cross-document context relevant to the user's query. By clustering semantically related documents and generating abstracted summaries of these clusters, RAPTOR creates a hierarchical knowledge representation that bridges the gap between granular document-level information and broader thematic connections. This multi-level retrieval strategy potentially improves answer generation by supplying the LLM with both the specific details needed for accuracy and the contextual framework necessary for coherence, particularly beneficial when responding to complex queries that span multiple related concepts within the vehicle collision safety domain. However, this framework was ultimately dropped, as it was not possible to fairly evaluate against the other models.

2.3 Embedding Models

RAG systems are inherently text-based, as most embedding models have historically utilized text to generate embeddings. The evolution of these embedding models began with transformer-based architectures like BERT [17] (Bidirectional Encoder Representations from Transformers), which revolutionized NLP by introducing contextualized word representations. Significant advances emerged with Siamese BERT networks [18], which utilize twin BERT encoders with shared weights to learn similarity between text pairs, dramatically improving retrieval performance for semantic search applications. These BERT-based embedding models established the foundation for semantic search by encoding textual information into dense vector representations that capture contextual meaning.

In recent times, embedding models have evolved beyond traditional text-based approaches, with the Beijing Academy of Artificial Intelligence (BAAI) leading advancements through its BGE series, particularly BGE-M3 [19]. BGE-M3 is a multi-functional embedding model that supports dense, sparse, and multi-vector retrieval, enabling more flexible and accurate information retrieval in RAG systems. Built on a transformer-based architecture similar to XLM-RoBERTa, it leverages self-knowledge distillation to refine embeddings across different retrieval tasks. Its multilingual capabilities, which were particularly important for our Korean-based RAG system, and ability to process long-form documents make it highly effective for cross-lingual and large-scale knowledge augmentation, enhancing the precision and contextual relevance of retrieved information in RAG workflows.

As RAG frameworks evolved, the limitations of text-only processing became apparent, particularly when dealing with diverse information sources containing images, audio, and video. This recognition prompted the development of multimodal embedding models capable of processing and aligning representations across different modalities. Models such as CLIP [20](Contrastive Language-Image Pre-training) pioneered this approach by jointly training on image-text pairs, enabling semantic search across both textual and visual content. However, even with current advancements there was much to desire for image embedding models and vision large language models(VLLM).

Despite these advancements, challenges remain in effectively integrating multimodal information within RAG systems. The biggest issue was due to VLLM image understanding, as at the point of the experiments open-source VLLMs were not advanced enough to coherently understand complex images and slides. This was further shown when utilizing Korean, as many of these open-source VLLMs and image embedding models were much more proficient in English, with a significant decrease in performance in languages like Korean. This ultimately led us to convert all visual information that we have into textual information.

2.4 Fine-tuning

Fine-tuning [21] LLMs and embedding models have proven to be very effective in recent times, and have been used across all different domains and applications. Through fine-tuning the weights of a model, or learnable parameters, are able to be adjusted to better align with specific domain knowledge and task requirements. This process allows pre-trained models to adapt to specialized contexts while maintaining the foundational knowledge acquired during their initial training phase.

For LLMs, fine-tuning typically involves exposing the model to domain-specific corpora and training it to generate responses that reflect the linguistic patterns, terminology, and knowledge structures unique to that domain. In the context of vehicle crash collision safety, this means training the model to understand and accurately reproduce specialized technical vocabulary, safety regulations, engineering principles, and analytical frameworks common in this field. The fine-tuning process can be further enhanced through instruction-tuning [22], where the model learns to follow specific formats and patterns for question answering based on the anticipated user queries.

Embedding models benefit similarly from domain-specific fine-tuning [19], though the objective differs slightly. Here, the goal is to optimize the vector space representation to better capture the semantic relationships particular to the domain. By fine-tuning on pairs of domain-specific texts with known semantic relationships, the embedding model learns to map related concepts closer together in the vector space, even when those relationships might be obscure or technical. This is particularly valuable in specialized fields like vehicle safety, where technical terms may have precise relationships that general-purpose embedding models might not fully capture.

2.5 Multimodal Data Processing in RAG

In our context, utilizing multimodal data requires the conversion of non-textual elements into textual components. Due to limitations in both language support and the integration of images in an open-source setting, converting all images into textual descriptions allows us to leverage state-of-the-art (SOTA) multilingual textual embedding models and incorporate the Korean language more effectively. Additionally, the limitations of open-source vision-language models [23] (VLMs) played a crucial role in our data processing decisions. At the initial stage of experimentation, the available vision-language models were not of sufficient quality to ensure reliable retrieval and generation, making a text-centric approach more viable.

Previous research has explored the integration of image embeddings and textual elements within retrieval-augmented generation (RAG) systems. Notably, models such as MuRAG [24] (Multimodal Retrieval-Augmented Generation) have attempted to retrieve both image and textual information to enhance knowledge-grounded generation. Similarly, frameworks like REVEAL [25] have introduced retrieval-augmented pretraining strategies that incorporate multimodal knowledge to improve generative outputs. However, these approaches primarily rely on direct multimodal embeddings and fusion rather than converting images into text for retrieval purposes.

To our knowledge, no prior research has specifically investigated the use of image-to-text conversion within retrieval-augmented generation systems. Furthermore, there has not been any formal research on slide presentation data processing and utilizing different complex multimodal elements within. There has been evaluation of tabular data within a text format, but the evaluations simply focused on the table data rather than any other elements [26]. While this method is commonly employed in industry for practical applications—such as image captioning, OCR-based document processing, and search engine optimization—there has not been an in-depth quality analysis of how image-to-text transformations affect retrieval performance and generative accuracy within RAG architectures. Our work aims to address this gap by systematically evaluating the impact of textualized visual data on retrieval quality, particularly in resource-constrained language environments such as Korean.

Our research makes a novel contribution by textualizing complex multimodal elements including tabular data, charts, and images through detailed explanations and direct content extraction. This approach presented significant challenges, as converting these diverse visual elements into coherent and retrievable text required developing specialized processing techniques. Our evaluation indicates that this textualization method proved moderately effective for improving retrieval quality, particularly when handling Korean-language content.

By structuring multimodal data into a purely textual format, we aim to enhance the alignment between visual knowledge and language models while ensuring compatibility with existing retrieval and generation architectures. This approach not only improves retrieval efficiency but also allows for seamless integration with Korean-language NLP systems. Future research may explore hybrid approaches that balance text-based image representations with multimodal embeddings to further optimize retrieval performance.

2.5.1 Synthetic QA and Data Generation

Synthetic QA generation [27] has emerged as a critical technique in addressing data scarcity challenges for training and evaluating NLP systems. Traditional QA datasets rely on human annotation, which is costly, time-consuming, and often limited in scope. Synthetic data [28] generation offers a scalable alternative by algorithmically creating customizable datasets tailored to specific domains.

Recent advances in LLMs have significantly improved synthetic data quality [29]. Models can now generate natural-sounding questions and answers that closely mimic human content across diverse domains. This capability has enabled wider adoption of synthetic data for both training and evaluation purposes.

In our work, we leveraged synthetic QA generation to create an evaluation dataset tailored to our synthetically generated domain data. This approach provided a controlled evaluation environment with known ground truth, enabling precise measurement of system performance. The generated QA pairs facilitated detailed analysis of retrieval mechanisms in RAG systems, helping identify strengths and weaknesses in finding relevant context from our synthetic corpus.

Additionally, our synthetic QA dataset supported the fine-tuning of both embedding models, LLMs, and VLLMs [2] [19] [30] , allowing them to better align with our specific domain requirements and query patterns. By generating questions targeting specific aspects of our synthetic data, we ensured comprehensive coverage of various information types and complexity levels.

This synthetic data generation approach formed a crucial component of our evaluation framework, providing insights that would have been difficult to obtain through manually created datasets alone. As LLM capabilities continue to advance, synthetic data represents an increasingly valuable technique for developing robust, domain-specialized AI systems.

Chapter 3

Approach

3.1 Data Processing

Data processing was a critical component of our retrieval-augmented generation system. Due to limitations in open-source VLLM image understanding capabilities and our system's Korean language requirements, we developed a data processing pipeline to convert image data into textual markdown format for use in our vectorstore.

Type	**Source**	**File**	**Slide**	**Q&A Pairs**
PPT	Test Report	1,463	4,662	59,402
	Meeting Report	249	882	7,696
		Page	**Chapter**	**Q&A Pairs**
PDF	Textbook	404	81	1,505
	Total		5,625	68,603

Table 3.1: Raw data statistics by source, along with the corresponding number of generated Q&A pairs.

3.1.1 Dataset

Our dataset comprised 1712 slide presentations containing 5,544 individual slides with diverse complex elements including images, tables, charts, text, and various combinations thereof. There was a lot of company domain-specific knowledge like internal collision test naming conventions, abbreviations, and classified information that an LLM could not have been previously trained on. We also incorporated a textbook with 404 pages on vehicle crash collision safety. The textbook processing was handled by another lab member and falls outside the scope of this thesis. This data was all in the Korean language with minimal english content, adding an additional challenge for OCR processing.

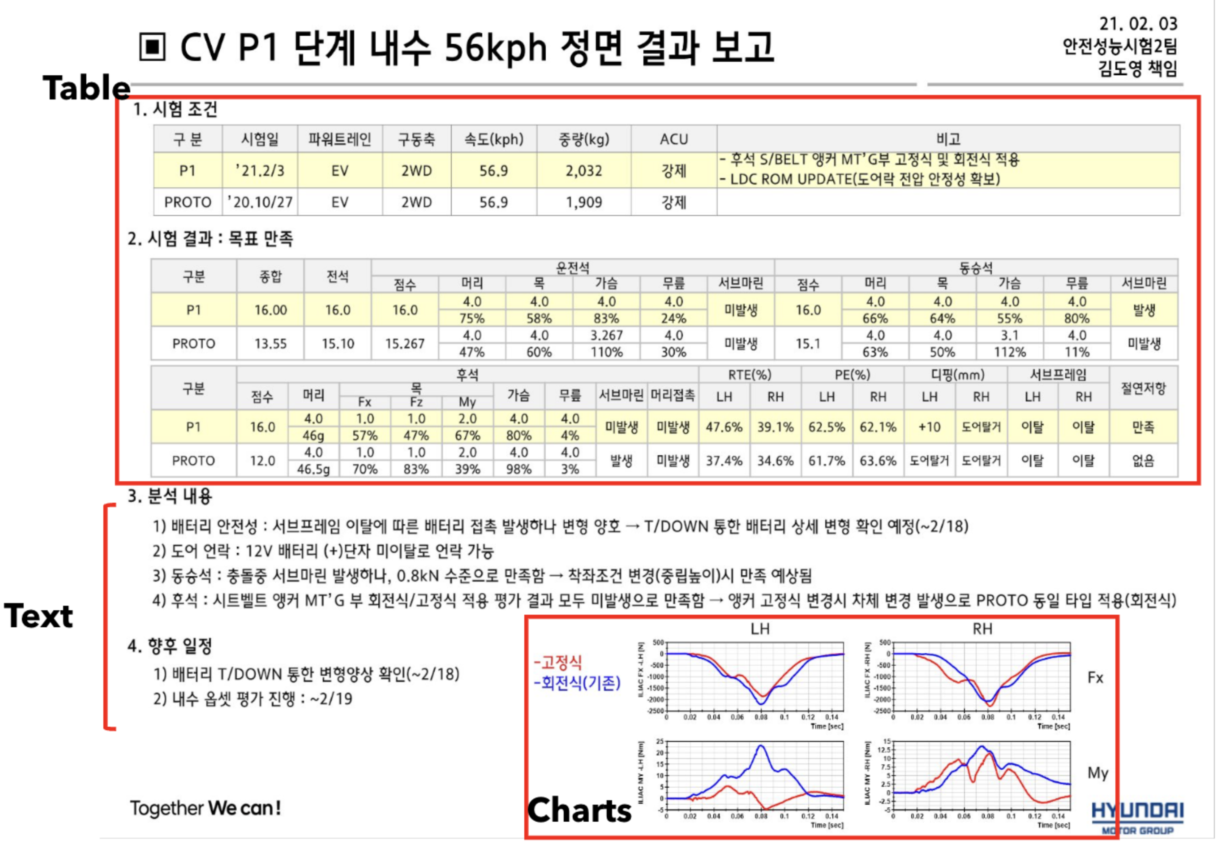


Table
Text
Charts
LH
RH
Fx
My
-고정식
-회전식(기존)
ILIAC FX -LH [N]
ILIAC FX -RH [N]
ILIAC MY -LH [Nm]
ILIAC MY -RH [Nm]
Time [sec]

▣ CV P1 단계 내수 56kph 정면 결과 보고

21. 02. 03
안전성능시험2팀
김도영 책임

1. 시험 조건

구 분	시험일	파워트레인	구동축	속도(kph)	중량(kg)	ACU	비고
P1	'21.2/3	EV	2WD	56.9	2,032	강제	- 후석 S/BELT 앵커 MT'G부 고정식 및 회전식 적용 - LDC ROM UPDATE(도어락 전압 안정성 확보)
PROTO	'20.10/27	EV	2WD	56.9	1,909	강제	

2. 시험 결과 : 목표 만족

구분	종합	전석	운전석						동승석					
			점수	머리	목	가슴	무릎	서브마린	점수	머리	목	가슴	무릎	서브마린
P1	16.00	16.0	16.0	4.0 75%	4.0 58%	4.0 83%	4.0 24%	미발생	16.0	4.0 66%	4.0 64%	4.0 55%	4.0 80%	발생
PROTO	13.55	15.10	15.267	4.0 47%	4.0 60%	3.267 110%	4.0 30%	미발생	15.1	4.0 63%	4.0 50%	3.1 112%	4.0 11%	미발생

구분	후석									RTE(%)		PE(%)		디핑(mm)		서브프레임		절연저항
	점수	머리	목			가슴	무릎	서브마린	머리접촉	LH	RH	LH	RH	LH	RH	LH	RH	
			Fx	Fz	My													
P1	16.0	4.0 46g	1.0 57%	1.0 47%	2.0 67%	4.0 80%	4.0 4%	미발생	미발생	47.6%	39.1%	62.5%	62.1%	+10	도어탈거	이탈	이탈	만족
PROTO	12.0	4.0 46.5g	1.0 70%	1.0 83%	2.0 39%	4.0 98%	4.0 3%	발생	미발생	37.4%	34.6%	61.7%	63.6%	도어탈거	도어탈거	이탈	이탈	없음

3. 분석 내용

1) 배터리 안전성 : 서브프레임 이탈에 따른 배터리 접촉 발생하나 변형 양호 → T/DOWN 통한 배터리 상세 변형 확인 예정(~2/18)
2) 도어 언락 : 12V 배터리 (+)단자 미이탈로 언락 가능
3) 동승석 : 충돌중 서브마린 발생하나, 0.8kN 수준으로 만족함 → 착좌조건 변경(중립높이)시 만족 예상됨
4) 후석 : 시트벨트 앵커 MT'G 부 회전식/고정식 적용 평가 결과 모두 미발생으로 만족함 → 앵커 고정식 변경시 차체 변경 발생으로 PROTO 동일 타입 적용(회전식)

4. 향후 일정

1) 배터리 T/DOWN 통한 변형양상 확인(~2/18)
2) 내수 옵셋 평가 진행 : ~2/19

Together We can!

Figure 3.1: An example of potential elements that exist within slide data. Includes complex tables, charts, and more.

The multimodal nature of the slide content necessitated specialized processing techniques for images, charts, and tables. Our objective was to generate high-quality synthetic textual markdown data suitable for effective retrieval and for fine-tuning both the embedding model and the LLM.

3.1.2 Slide Data Processing

In order to process the data, we utilized a 3-stage data processing pipeline that mitigated many different issues that existed at the time of when we processed the data. The 3-stage pipeline presented in figure 3.2 showcase an LLM-centric pipeline that utilizes both the original slide presentation data as well as the images to create quality data extractions and explanations. The pipeline leverages both Python-based tools and advanced LLMs, specifically Claude 3.5 Sonnet.

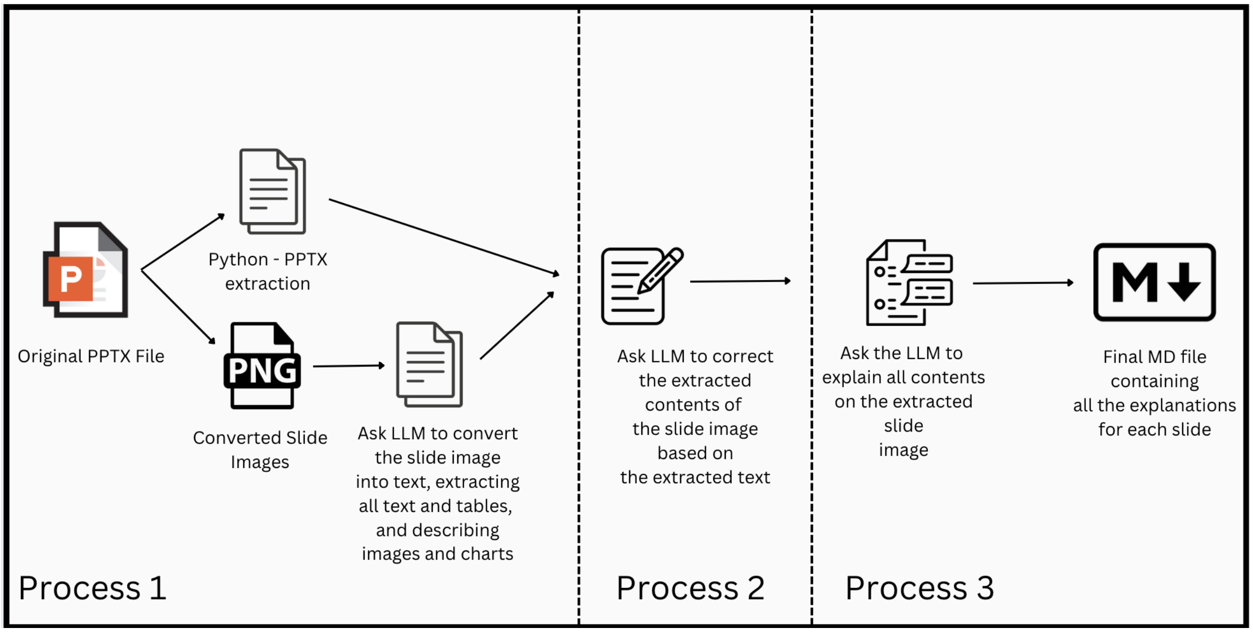

Original PPTX File
Python - PPTX extraction
Converted Slide Images
Ask LLM to convert the slide image into text, extracting all text and tables, and describing images and charts
Process 1
Ask LLM to correct the extracted contents of the slide image based on the extracted text
Process 2
Ask the LLM to explain all contents on the extracted slide image
Final MD file containing all the explanations for each slide
Process 3

Figure 3.2: A 3-stage process for data processing of slide data. Each process further refines or extrapolates synthetic data from the original slide presentation.

Process 1: Data Extraction and Initial Processing

The original PPTX file is processed in two parallel streams:

- Python-pptx is employed to extract textual content directly from the slides, converting it into an initial markdown format.
- Simultaneously, each slide is converted into a PNG image.

Claude Sonnet 3.5 is then tasked with analyzing the slide images. It extracts comprehensive information, including all text, tables, and detailed descriptions of images and charts.

Process 2: Comparative Analysis and Error Correction

In this stage, the LLM performs a comparative analysis between its extracted content from the slide images and the text directly extracted by Python-pptx. This crucial step helps identify and rectify OCR errors, misspellings, and formatting inconsistencies. The LLM corrects the extracted contents of the slide image based on this comparison, ensuring high fidelity to the original content.

Process 3: Expert-Level Explanation Synthesis

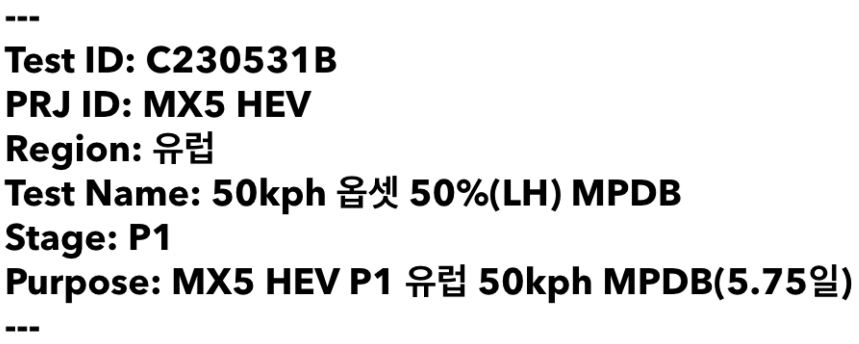
In this stage the LLM generates detailed, expert-level explanations for all contents on each extracted slide. Importantly, this process incorporates context from the previous two slides ($n = 2$), allowing for more coherent and contextually relevant explanations. The output is a final Markdown (MD) file containing comprehensive explanations for each slide.

This systematic approach transforms complex visual presentations into structured, contextually rich markdown content. By combining Python-based text extraction, image processing, and multi-stage LLM analysis, we ensure a comprehensive conversion optimized for both human readability and machine processing in RAG frameworks.

The original hypothesis for utilizing explanations of the data rather than the raw data extraction itself, was that table data would contain many unnecessary characters that offer no value for the embeddings generated by our embedding model. However, after evaluation of the data, it was shown that the explanations had no tangible benefit for both retrieval and fine-tuning of the different embedding models. In the future it would be wise to skip stage three, but was kept here as it is an interesting point of exploration.

3.1.3 Additional Headers

In addition to the slide presentation data, there was a spreadsheet that contained specific information about the different test reports and presentations such as ID numbers, region



Test ID: C230531B
PRJ ID: MX5 HEV
Region: 유럽
Test Name: 50kph 옵셋 50%(LH) MPDB
Stage: P1
Purpose: MX5 HEV P1 유럽 50kph MPDB(5.75일)

Figure 3.3: Example Header that was prepended to all slide chunks

where testing took place, and the names and purpose of the test report. As shown in figure 3.3, these small sections of testing information was prepended to all slide chunks, providing extra keywords and topics that could relate better to user queries for specific tests or examples of tests.

3.1.4 Synthetic QA Generation

Using our generated synthetic data, we created synthetic question-answer pairs that served multiple purposes. These synthetic QA pairs enabled us to fine-tune both our LLM and embedding model, which were not previously trained on crash-collision safety domain knowledge or our classified dataset, and severed as evaluation queries for retrieval and answer generation. This approach provided the LLM with enhanced context for retrieved data during answer generation and allowed the embedding model to learn contrastive elements between different data chunks.

To generate these synthetic QA pairs, we employed an LLM with carefully engineered prompts to create multiple distinct questions based on each chunk of textbook content and slide data. The questions comprehensively covered various content elements from each source, including image descriptions and tabular/chart data.

We also created a synthetic QA dataset to specifically test files that contained table or chart extractions. The QA pairs generated for this part of the evaluation set specifically focused on specific details within the tables and charts that were extracted from slide images. This was done in order to focus on the evaluation of multimodal data converted into text format, as this was a crucial part of the overarching Hyundai Project.

We deliberately generated these QA pairs in Korean to closely simulate the real-world use case, as our target users would be inputting queries in Korean and expecting responses in the same language. This approach ensured our system would function effectively in its intended language environment.

3.1.5 Evaluation of Synthetic Data

To evaluate our synthetic data, we utilized qualitative and quantitative analysis. We take a deep dive into Korean OCR performance, as well as qualitative analysis on table/chart extraction. We utilize token counts to compare the changes between process 1 and process 2 from figure 3.2, giving a quantitative perspective on the amount of change between both the first and second processes; this allows us to calculate the total difference in tokens, overall change, and largest differentials in token count between the first and second processes of each document to name a few. We also calculate hallucination through the number of tokens that were processed incorrectly across 63 different documents. We also perform qualitative analysis on the generated files through looking at the differences between process 1 and process 2 documents, as well as looking at the results generated for each of the different modalities present within the documents.

We perform special analysis, specifically on domain-specific abbreviations. Throughout the slide documents, it was noted that there were many different abbreviations that are company specific. For example, terms such as RTE meaning restraint energy or DAB meaning driver airbag were company specific abbreviations used across all slide presentations. When utilizing these terms, we evaluated the rate at which they were incorrectly interpreted by the LLM for explanation generation for 18 predefined abbreviations.

3.2 Embedding Model - BGE-M3

For our experiments, we attempted to use many different embedding models within our framework. This included models such as KoSimCSE, KoBERT, as well as SNU-Bert [31] which are all models specifically designed to understand and utilize Korean text. However, upon our preliminary testing, these models underperformed significantly, achieving very low precision and MAP. With that in mind, we utilized the state-of-the-art BGE-M3 embedding model, which allows for not only multilingual embeddings, but cross-lingual embeddings. This model proved to be superior in our use case, and significantly improved as we fine-tuned the model on different variations of the dataset, further described in Chapter 4. The vector representations generated by BGE-M3 allows for the usage of cosine similarity between vectors, calculated as:

$$\text{Cosine Similarity}(v_1, v_2) = \frac{v_1 \cdot v_2}{||v_1|| \times ||v_2||} \tag{3.1}$$

By calculating the cosine similarity between two vectors we are able to obtain a score that compares the query to each document in the vectorstore, allowing us to rank documents by relevance to the user's query. Higher cosine similarity scores indicate greater semantic similarity between the query and document vectors.

In general, for a question q, the model retrieves the top n chunks from m chunks $\mathbf{D}$, based on the cosine similarity between the normalized embeddings of q and each chunk $d_i \in \mathbf{D}$, as follows:

$$\begin{gathered}
q_{[\text{CLS}]} \in \mathbb{R}^{1\times d}; \quad \mathbf{D}_{[\text{CLS}]} \in \mathbb{R}^{m\times d} \\
\hat{q}_{[\text{CLS}]} = \frac{q_{[\text{CLS}]}}{||q_{[\text{CLS}]}||}; \quad \hat{\mathbf{D}}_{[\text{CLS}]} = \frac{\mathbf{D}_{[\text{CLS}]}}{||\mathbf{D}_{[\text{CLS}]}||} \\
\text{Similarity}(q, \mathbf{D}) = \hat{q}_{[\text{CLS}]} \cdot \hat{\mathbf{D}}_{[\text{CLS}]}^{\top} \in \mathbb{R}^{1\times m} \\
\mathbf{D}_{\text{top}_n} = \text{Sort}\left(\mathbf{D},\ \text{key} = \text{Similarity}(q, \mathbf{D})\right)[:n]
\end{gathered} \tag{3.2}$$

The BGE-M3 model's architecture is based on a transformer encoder with 580 million parameters, trained on a diverse multilingual corpus spanning 100+ languages. Its cross-lingual capabilities were particularly valuable for our application, as it could effectively handle both Korean and English technical terminology within the same embedding space. This allowed us to maintain consistent semantic relationships regardless of the language mixture in our slides.

In our implementation, we configured the model to generate 1,024-dimensional embeddings, which provided sufficient representational capacity for the technical domain while maintaining reasonable computational efficiency.

3.2.1 Fine-Tuning

To fine-tune BGE-M3, we utilized synthetically generated QA pairs derived from our dataset. Two variants were created: one incorporating explanatory with the slide extractions and another excluding the explanations. Both variants were fine-tuned using dense retrieval methodologies, focusing on optimizing the encoder for dense embeddings.

BGE-M3 employs contrastive learning to effectively distinguish between similar and dissimilar documents in relation to a query. In our approach, each query was paired with a positive document and negative documents. Negative samples were selected through random sampling, ensuring they were contextually relevant yet distinct from the positive samples. This fine-tuning process led to notable improvements in retrieval performance.

3.3 RAG Frameworks

3.3.1 Frameworks

For our retrieval-augmented generation frameworks, we initially planned to use Hypothetical Document Embeddings (HyDE) cite, Recursive Abstractive Processing for Tree Organized Retrieval (RAPTOR) cite, and the original RAG framework as a baseline. However, we ultimately dropped RAPTOR because its higher-level summaries—generated during database build time—made it impossible to create precise citations for the underlying text chunks. Since a single summary can encompass multiple chunks, it would lead to citations for several different slides, which would be unhelpful to the end user. Citations were a crucial requirement in our system, and RAPTOR's approach hindered this need. Moreover, developing a fair evaluation method based on the synthetically generated data proved challenging, as it would require human evaluation and be impractical for comparison against our automated approach for the remaining two methods.

We also explored modifications to the HyDE framework to enhance retrieval performance. In its traditional implementation, HyDE generates a hypothetical response to a query and then uses that response to retrieve relevant documents. In our modified approach, we concatenate the original question with the generated answer before performing retrieval. Both theoretically and practically, this combined input better captures the query's intent even when the original query lacks domain specific terminology, resulting in more accurate document matches. Considering these factors, we ultimately evaluated three methods: the original RAG framework, standard HyDE, and our modified HyDE that includes the question with the hypothetical answer.

3.3.2 Storage of Data

In order to utilize our data for retrieval in our different RAG systems, we need to store our data. Many different vector databases and indexes were tested, such as ChromaDB or FAISS, however, these different vector database indexings had their own optimizations that degraded performance. The trade-offs of the optimization, which was almost always related to the speed of lookup times, was not significant enough for our data which was on the smaller side with less than 10,000 chunks of textual data.

In terms of how the data was chunked and stored in the database, we utilized each slide image as a chunk.

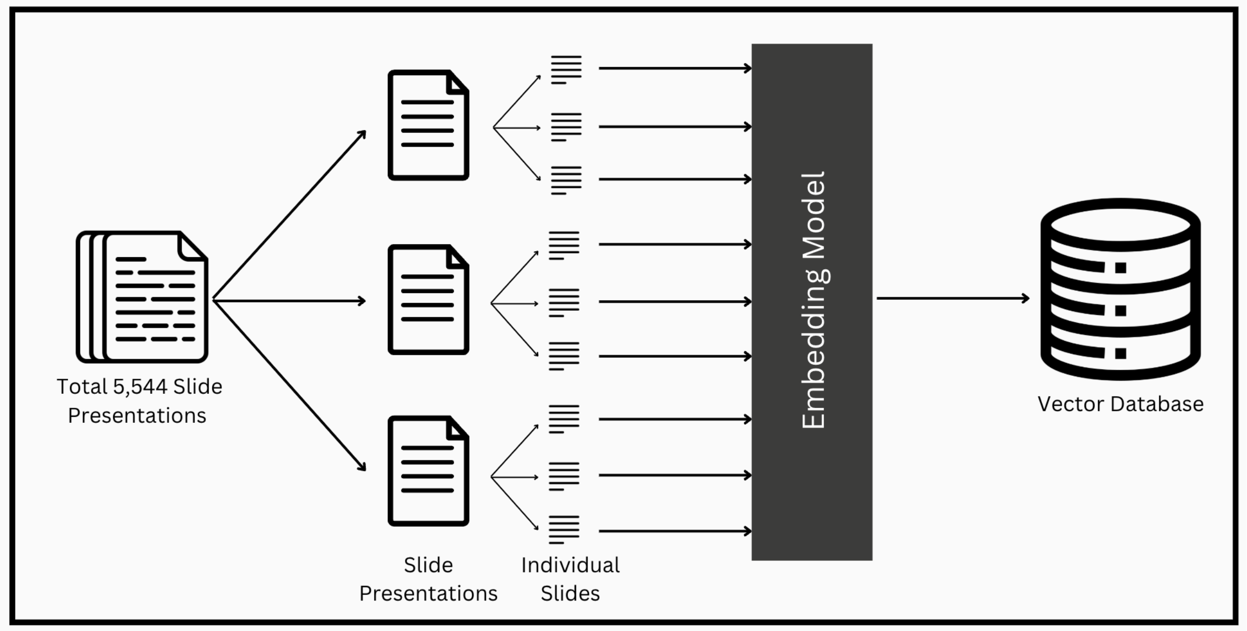


Figure 3.4: Each slide of each slide presentation is stored as a chunk in the database and given a dense vector representation through the embedding model.

$$\text{Total Chunks} = \sum_{i=1}^{N} s_i \tag{3.3}$$

The equation above represents the total number of chunks, where s_i represents the number of slides in presentation i. Each of these chunks were encoded with a dense vector representation of the textual elements through our fine-tuned embedding models. Separate databases were created for each of the different variations of the embedding models and sets of stored data, as the encoding of the dense vector representation would change between them. The textbook data was stored in a similar manner, where each section of the book was considered a single chunk and stored into the database. These vectors are later used with cosine similarity to retrieve documents that can best answer the user query. For storage efficiency and experimental flexibility, we utilized NumPy's .npy file format, which provided optimal I/O performance while maintaining the structural integrity of our vector representations.

3.3.3 Evaluation of Retrieval

To evaluate our retrieval quality, we utilize mean average precision (MAP) and success rate as our primary evaluation metrics.

Mean Average Precision (MAP) measures the quality of document ranking across multiple queries:

$$\text{MAP@}k = \frac{1}{|Q|} \sum_{q \in Q} \text{AP@}k(q) \tag{3.4}$$

where $|Q|$ represents the total number of queries in our evaluation set, and AP@$k(q)$ is the average precision for a specific query q up to rank position k.

Average Precision (AP) for an individual query is calculated as:

$$\text{AP@}k(q) = \frac{1}{\min(m, k)} \sum_{i=1}^{k} P(i) \times \text{rel}(i) \tag{3.5}$$

where m is the number of relevant documents (in our case, always 1), $P(i)$ is the precision at cutoff position i, and rel(i) is an indicator function that equals 1 if the document at rank i is relevant to query q, and 0 otherwise.

Success rate provides a simpler metric that measures whether the relevant document appears in the top-k retrieved results:

$$\text{Success@}k = \begin{cases} 1 & \text{if } \sum_{i=1}^{k} \text{rel}[i] > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

We evaluate MAP at $k = 1, 5,$ and 10, and success rate at $k = 5$ and 10. These metrics at different k values help us understand both the precision of our highest-ranked results and the overall coverage of relevant documents within various retrieval depths.

For our evaluation methodology, we leveraged synthetic question-answer pairs specifically generated to correspond with individual slide chunks. This approach established a controlled experimental environment where each query had exactly one designated correct document in the corpus. We considered a document retrieved successfully only if it matched the specific slide chunk from which the question was derived.

This single-source ground truth assumption was implemented to ensure objective and consistent evaluation across different model variations. It eliminated the subjectivity of determining which additional documents might contain partial or alternative answers to a given query. Consequently, traditional metrics like recall were not applicable to our evaluation framework, as we deliberately constrained the relevant document set to exactly one item per query, rather than attempting to identify all potentially relevant documents in the corpus.

3.4 Open Source VLLMs for OCR Processing

Over the course of the past six months, there has been tremendous growth and improvement in the open-source community, especially for vision LLMs. Beginning with the release of Llama 3.2 generation of models, open-source vision models had a large increase in accuracy and usability. However, even then these models were the most capable in the English language rather than other non-Latin alphabet languages like Korean or Japanese. With the release of Qwen2.5-VL, the capabilities and possibility of open-source OCR extraction became much more apparent and its abilities to OCR and understand languages other than English grew tremendously.

With the data extracted through our data processing pipeline from Section 3.1.2, we explored the possibility of fine-tuning an open-source VLLM to skip over process 1 and immediately get useful synthetic data that can be used as a source of data for synthetic QA generation and integration into a RAG system as data within the external database. This would give us an open source solution for data generation, and a much more privatized way for companies like Hyundai to generate data to put into a RAG database. We fine-tune the Qwen2.5-VL-7B model for OCR and synthetic data generation from our slide presentation images.

3.4.1 Fine-tuning VLLM

In order to fine-tune a VLLM, it is very similar to an LLM, in that you need quality data to shift the weights or parameters of a model to suit your task. However, there is now the added input of an image that is fed in along with the text prompt. Since we want the VLLM to have an increased understanding in OCR processing of presentation slides, we created a dataset consisting of slide images paired with their correctly extracted textual content.

The fine-tuning process involved optimizing the model to recognize and extract text and create descriptions from slides while maintaining the semantic relationships between visual elements. We employed full fine-tuning in order to improve OCR and synthetic data generation accuracy. We fine-tune Qwen2.5-VL 7B for its performance in OCR processing and visual understanding. We employ two H200 GPUs contained 144GB of memory for each H200.

3.4.2 Data Preparation

To prepare the data, we set up the training examples in the following format:

```
"messages": [
    {"role": "system", "content": SYSTEM_PROMPT},
    {"role": "user", "content": [TEXT_PROMPT, IMAGE_PLACEHOLDER]},
    {"role": "assistant", "content": MARKDOWN_ANSWER}
],
"image_path": IMAGE_PATH
```

Where ”messages” contains the system prompt given to the VLLM, the textual prompt with instructions, the image placeholder indicating where the slide image should be processed, and the expected answer containing the extracted text. The ”image_path” field provides the location of the slide image file to be tokenized and inserted at the IMAGE_PLACEHOLDER position during training.

Each training example pairs a slide image with its corresponding OCR output, allowing the model to learn the mapping between visual slide elements and their textual representation. This format follows the standard conversational structure required for fine-tuning VLLMs while incorporating the image processing capabilities needed for the OCR task.

In order to utilize our images, we utilized the recommended resizing from Qwen, which recommended 448×448 dimensions. We tested increasingly higher resolutions up to 2400×1800 to determine optimal performance across various image sizes. The actual size that is inputted into the VLLM may differ, as Qwen automatically estimates pixel sizes to the nearest multiple of 28.

3.4.3 Evaluation of VLLM

Our evaluation strategy focused on assessing multiple fine-tuned variants with different image resolution configurations. We trained our model on 4,742 data points, reserving 447 for evaluation purposes. For this generation task, we employed several different metrics. In order to get a sense of quality of the OCR, we utilized standard OCR metrics which includes character error rate (CER) and word error rate (WER). These equations can be written as follows:

$$\mathrm{CER} = \frac{\mathrm{S}_c + \mathrm{D}_c + \mathrm{I}_c}{\mathrm{N}_c} \tag{3.7}$$

where S_c, D_c, and I_c represent the number of character substitutions, deletions, and insertions, respectively, required to transform the predicted text into the reference text, and N_c is the total number of characters in the reference text.

$$\mathrm{WER} = \frac{\mathrm{S}_w + \mathrm{D}_w + \mathrm{I}_w}{\mathrm{N}_w} \tag{3.8}$$

where S_w, D_w, and I_w represent the number of word substitutions, deletions, and insertions, respectively, and N_w is the total number of words in the reference text.

We also utilize the BERT-Score similarity metric to evaluate semantic similarity between the generated and reference texts. While character and word error rates capture surface-level differences, they fail to account for semantic equivalence or contextual meaning. BERT-Score addresses this limitation by leveraging contextual embeddings from pre-trained language models to measure similarity at a deeper semantic level.

This metric is particularly valuable for our VLLM evaluation because OCR errors may preserve semantic meaning despite character-level differences. For instance, substituting "organization" with "organisation" would significantly impact CER and WER metrics despite maintaining identical meaning. This limitation becomes particularly pronounced when evaluating descriptions of images or charts, where two semantically equivalent descriptions that identify identical key elements may use entirely different phrasing or structural organization, resulting in severely penalized WER and CER scores despite their functional equivalence. BERT-Score [32] allows us to calculate precision, recall, and F1 scores based on contextual embeddings and shows higher correlation with human judgments compared to traditional string-matching metrics, making it especially suitable for evaluating text with potential paraphrasing or minor variations that preserve the core information. The BERT-Score is computed as:

$$\mathrm{P}_{\mathrm{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{y_j \in y} \cos(\mathbf{e}_{x_i}, \mathbf{e}_{y_j}) \tag{3.9}$$

$$\mathrm{R}_{\mathrm{BERT}} = \frac{1}{|y|} \sum_{y_j \in y} \max_{x_i \in x} \cos(\mathbf{e}_{x_i}, \mathbf{e}_{y_j}) \tag{3.10}$$

$$\mathrm{F}_{\mathrm{BERT}} = 2 \cdot \frac{\mathrm{P}_{\mathrm{BERT}} \cdot \mathrm{R}_{\mathrm{BERT}}}{\mathrm{P}_{\mathrm{BERT}} + \mathrm{R}_{\mathrm{BERT}}} \tag{3.11}$$

where $\mathbf{e}_{x_i}$ and $\mathbf{e}_{y_j}$ are the contextual embeddings of tokens x_i and y_j in the candidate and reference texts, respectively.

In this context, **BERT-Score Precision** ($\mathrm{P}_{\mathrm{BERT}}$) measures how well the generated (candidate) tokens align with the most semantically similar tokens in the reference text. It evaluates whether the generated output contains relevant information by checking if each token in the candidate has a close match in the reference text. A high precision score suggests that the generated text contains words or phrases that are contextually similar to those in the reference, reducing the likelihood of extraneous or irrelevant information.

Similarly, **BERT-Score Recall** ($\mathrm{R}_{\mathrm{BERT}}$) assesses how well the reference tokens are covered by the generated text. It measures whether the important elements in the reference text are retained in the candidate output by checking if each reference token has a semantically similar counterpart in the generated text. High recall indicates that the generated text captures most of the meaningful content from the reference, ensuring completeness.

Since precision and recall often have trade-offs—where high precision may lead to overly concise outputs and high recall may introduce verbosity—**BERT-Score F1** ($\mathrm{F}_{\mathrm{BERT}}$) provides a balanced measure by taking their harmonic mean. A high $\mathrm{F}_{\mathrm{BERT}}$ score indicates that the generated text not only contains relevant information (precision) but also comprehensively retains the key aspects of the reference text (recall), resulting in an overall semantically faithful representation.

Unlike traditional precision, recall, and F1-score in information retrieval (which rely on exact token matching), **BERT-Score** leverages contextual embeddings to compare words based on meaning rather than surface similarity. This allows it to effectively account for paraphrasing, synonym usage, and slight rewording while still maintaining the core meaning of the text. As a result, BERT-Score provides a more robust evaluation metric for OCR-extracted data, especially when assessing descriptive text, where different but semantically equivalent expressions may be used to describe the same visual content.

Lastly, we evaluated documents quality through an LLM-as-Judge methodology [33]. Building on our successful application of this evaluation approach for RAG system answer generation [2], we adapted the technique to assess our slide extraction responses. The evaluation used the original slide extraction document from the data processing pipeline as the reference standard, comparing responses from different fine-tuned model variants. Rankings were determined based on similarity to the original slide documents from our closed-source data processing pipeline. We compared the 448×336, 1280×960, and 2400×1800 model variants, excluding the 448×448 fine-tune due to its inferior performance relative to the other three fine-tuned models. For these evaluations, we utilized GPT-4o as the impartial judge, as the evaluation data was processed with Claude and Qwen. To counter positional bias within the prompt response, the positions of the documents were randomly shuffled each time the documents were sent to the model.

Qualitative analysis is also done on the highest performing image dimension fine-tune models to evaluate the quality of explanations and overall OCR processing. We compared our original extracted documents to those produced by the best performing models to see the quality difference.

Chapter 4

Experiments

4.1 Evaluation of RAG and Embedding Models

In our experiments, we evaluate three main frameworks: the original RAG framework, HyDE, and a variant of HyDE that uses both the question and the answer as the retrieval query. We also consider three variants of the BGE-M3 embedding model: the original vanilla model, one fine-tuned on both slide and explanation data, and another fine-tuned solely on slide data. We measure performance using Mean Average Precision (MAP) and Success at various retrieval depths (1, 5, and 10 documents). We assume each test query has exactly one relevant document. To ensure consistency in our HyDE evaluations, we use a file of pre-generated answers for each test query.

We begin by experimenting with data that combines slides and explanations. Specifically, we test the RAG and HyDE (answer-only) frameworks using both the vanilla and explanation-fine-tuned BGE-M3 embedding models.

Model	**Method**	**MAP@1**	**MAP@5**	**Success@5**	**MAP@10**	**Success@10**
Vanilla-BGE	RAG	28.28	38.11	54.32	39.41	64.06
Vanilla-BGE	HyDE	18.08	24.62	35.67	25.65	43.30
Fine-Tune BGE	RAG	**47.60**	**59.27**	**77.86**	**60.32**	**85.52**
Fine-Tune BGE	HyDE	21.28	29.16	42.49	30.42	51.95

Table 4.1: Comparison of RAG vs. HyDE for Vanilla-BGE and Fine-Tune BGE.

Through the above results, we can see that between RAG and HyDE, HyDE performs worse even with the fine-tuned embedding models. At MAP@10, there is a 30% difference between RAG and HyDE performance for the fine-tuned variation of the BGE-M3 embedding model. It is also worthy to note that Success@10 for the RAG model is at 85.52%, meaning that when k is 10, the golden document is retrieved within the top 10 documents 85% of the time. Also, fine-tuning significantly increases the performance across both methodologies, with RAG having a significant increase in retrieval quality. However, there is still more to be desired from these retrieval metrics.

We further our evaluations by testing not only data with both the slide extractions and explanations, but also data that only contains the slide extractions.

Test Data	**Method**	**MAP@1**	**MAP@5**	**Success@5**	**MAP@10**	**Success@10**
Slide	RAG	**29.83**	**39.74**	**56.05**	**40.89**	**64.65**
Slide + expl	RAG	28.58	37.91	53.69	39.25	63.69

Table 4.2: Vanilla BGE-M3

In Table 4.2, we observe that the vanilla embedding model achieves slightly higher retrieval performance when explanations are excluded from the dataset. Based on our preliminary experiments with various dataset formats, we extended our evaluation by fine-tuning an additional version of the BGE-M3 embedding model exclusively on slide extraction data. If document retrieval improves with data containing only slide extractions, a model fine-tuned solely on this content may yield even better results. Additionally, we evaluated the HyDE retrieval method using a combination of both the question and answer across all three embedding models, given that the performance of the original HyDE framework indicated room for improvement. We adjust our experiments based on the results shown in tables 4.1 and 4.2 to further test different frameworks, data formats, and embedding models.

Table 4.3 shows the results for various combinations of data and frameworks using an embedding model fine tuned on slide extraction and explanation data. The findings

Test Data	**Method**	**MAP@1**	**MAP@5**	**Success@5**	**MAP@10**	**Success@10**
Slide	RAG	**47.46**	**59.06**	**77.35**	**60.23**	**85.97**
Slide	HyDE(A)	21.58	29.60	43.36	30.78	52.12
Slide	HyDE(QA)	33.62	44.26	61.74	45.62	71.95
Slide + expl	RAG	**47.79**	**59.23**	**77.44**	**60.34**	**85.51**
Slide + expl	HyDE(A)	21.44	29.35	42.66	30.65	52.40
Slide + expl	HyDE(QA)	34.00	44.46	61.56	45.82	71.70

Table 4.3: BGE-M3 Fine-tuned on slide and explanation data

indicate that each matching framework performs similarly across both datasets, with RAG achieving the highest performance on both the slides only and the slides with explanations. Although HyDE lags behind base RAG, it has a notable performance increase, by 11 to 13%, when the questions are appended along with the generated hypothetical answers. Due to the comparable performance between the two datasets, we continue testing on both sets. Although the embedding model is fine tuned on the explanation and slide dataset, the overall performance remains similar no matter the type of data that is being retrieved.

We continue our experiments by fine-tuning the BGE-M3 embedding model specifically on slide data, without utilizing any explanations. From the table shown below, we can see a significant increase in both MAP and Success performance across all the different frameworks and datasets.

Test Data	**Method**	**MAP@1**	**MAP@5**	**Success@5**	**MAP@10**	**Success@10**
Slide	RAG	**60.42**	**71.09**	**87.09**	**71.73**	**91.79**
Slide	HyDE(A)	26.40	34.64	48.18	35.63	55.45
Slide	HyDE(QA)	42.56	53.63	71.10	54.70	78.95
Slide + expl	RAG	58.26	69.00	85.16	69.79	90.99
Slide + expl	HyDE(A)	26.51	34.52	47.53	35.59	55.29
Slide + expl	HyDE(QA)	42.44	53.28	70.36	54.31	77.98

Table 4.4: BGE-M3 Fine-tuned on slide data

RAG continues to perform the best across both slide data and the combination of slide and explanations. However, we can see the massive performance increase when utilizing the embedding model that is fine-tuned on slide data rather than the combination dataset. The performance of all the frameworks increase significantly, ranging between a 5% - 13%

percent increase across the metrics. However, with the embedding model fine tuned on slide data, RAG with the slide data performs significantly better than all the frameworks and datasets tested, sitting at 60.42% MAP at 1, and over 90% success rate at top 10 documents retrieved.

4.2 Evaluation of Table and Chart Data

Given that the multimodal aspect is integral to our data processing pipeline, we focus on evaluating the chart and table data extracted from the markdown text. We select a subset of our test dataset containing tables and charts using string matching, and generate synthetic QA pairs following the method described in Section. Using these questions, we assess retrieval accuracy on data both with and without explanations. For this evaluation, we employ the basic RAG framework, as it demonstrated the best performance among the three frameworks tested.

Model	**Data**	**MAP@1 (%)**	**MAP@10 (%)**
BGE-M3(Slides)	slide	33.28	45.63
BGE-M3(Slides)	slide + Expl	**34.52**	**46.52**
BGE-M3(Expl)	slide	31.68	43.15
BGE-M3(Expl)	slide + Expl	33.60	45.44

Table 4.5: Example results for Fine-Tune BGE on slides vs. explanations.

From Table 4.5, the best-performing variant of our RAG framework was the BGE-M3 model fine-tuned on slide data, using a database that included both the original data and the corresponding explanations. Interestingly, even with the inclusion of explanations, the model fine-tuned solely on slide extractions outperformed the one specifically intended to process explanation data.

4.3 Evaluation of VLLM

We evaluate the performance of different variations of the Qwen2.5-VL-7B model through our test data containing 447 data points. There are 4 models that were finetuned on 4 different image dimensions: 448x448, 448x336, 1280x960, and 2400x1800 pixel sizes. We begin by calculating common OCR metrics to get a sense of the overall performance of each of the models:

Base or FT	**Image Size**	**Avg. CER (%)**	**Avg. WER (%)**
Base	448x448	111.52	111.74
Finetuned	448x448	99.02	118.10
	448x336	78.89	97.88
	1280x960	63.87	80.86
	2400x1800	**59.62**	**78.62**

Table 4.6: Comparison of different image dimension configurations for finetuning Qwen2.5-VL-7B, grouped by configuration.

An overall trend within the table is developed, as the larger the dimensions of the image, the smaller the error rates across both characters and words. It is also notable that the recommended 448x448 pixel dimension given by Qwen is not ideal, as it tends to perform significantly worse, even against image dimensions of similar size like 448x336. While these metrics signify poor performance, it is also important to consider that our OCR process is not strictly relating to text; it also contains descriptions of the multimodal elements, as well as table generation. When utilizing metrics like the ones above that require exact matching, the error rate may misrepresent the contents within the OCR document. It is also important to note that the error rate goes over 100% due to hallucinations, as a significant larger token count can lead to multiple substitutions, deletions, and insertions being necessary.

Due to the limitation in the basic OCR metrics, as exact characters are required to achieve a good score, we also utilize BERT-Score to get a better sense of the semantic similarities between our golden documents and the new outputs for each of the tested model variants. We also test each of the image dimensions on the base, or unfinetuned, Qwen2.5-VL-7B model to compare the finetuned variants. We calculate Bert-Score precision, recall, and F1 and attain the following results:

Image Size	**Base (%)**			**Finetuned (%)**		
	BS-P	BS-R	BS-F1	BS-P	BS-R	BS-F1
448x448	7.27	6.67	6.92	74.01	71.25	72.52
448x336	10.38	9.65	9.96	77.31	76.46	76.83
1280x960	5.48	4.99	5.21	**82.26**	**81.39**	**81.76**
2400x1800	3.82	3.34	3.55	**82.39**	**81.47**	**81.88**

Table 4.7: Comparison of BERT-Score metrics (in percentages) grouped by image dimensions.

Across all finetuned models, there is a significant increase in precision, recall, and F1 score for its corresponding image dimension. Across each row, there is between a 60% - 78% increase in contextual similarity with the original golden document. The 448x448 image dimensions continues to perform the worst, while the 1280x960 and 2400x1800 finetuned models perform the best, with 2400x1800 image dimension only slightly outperforming its lower dimension counterpart. There is a positive effect overall for all finetune variants, and finetuning helps to significantly increase semantic similarity to closed-source generated documents.

We further explore the quality of our data by employing an LLM-as-Judge methodology. Two distinct judging systems are evaluated: one where the golden answer is the original Claude-extracted slide document, and another where the golden answer is the original slide image. To mitigate potential positional bias, we randomize the order of the documents before including them in the evaluation prompt.

For the evaluation using the golden document, outputs from the 1280×960, 2400×1800, and 448×336 finetuned variants are compared against the original document.

In the bar chart, where we compare the finetuned outputs to the original Claude-processed document, both the 1280×960 and 2400×1800 finetunes substantially outperform the smaller 448×336 variant. Although 2400×1800 is twice the resolution of 1280×960, the

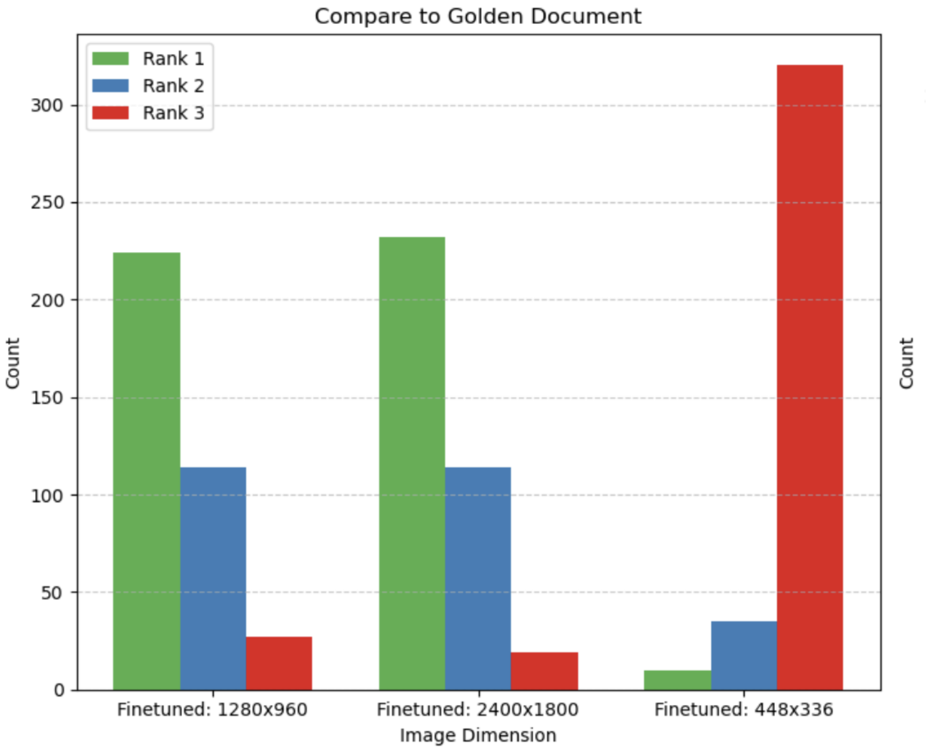


Figure 4.1: Bar graph showing the distribution of rankings assigned by an LLM-as-Judge evaluator. The chart uses the the original synthetic data as ground truth, comparing the best-performing finetuned models against our original synthetic data.

LLM judges them to be rather similar which suggests that beyond a certain threshold, increasing the resolution does not proportionally boost quality.

Chapter 5

Analysis

5.1 Synthetic Data

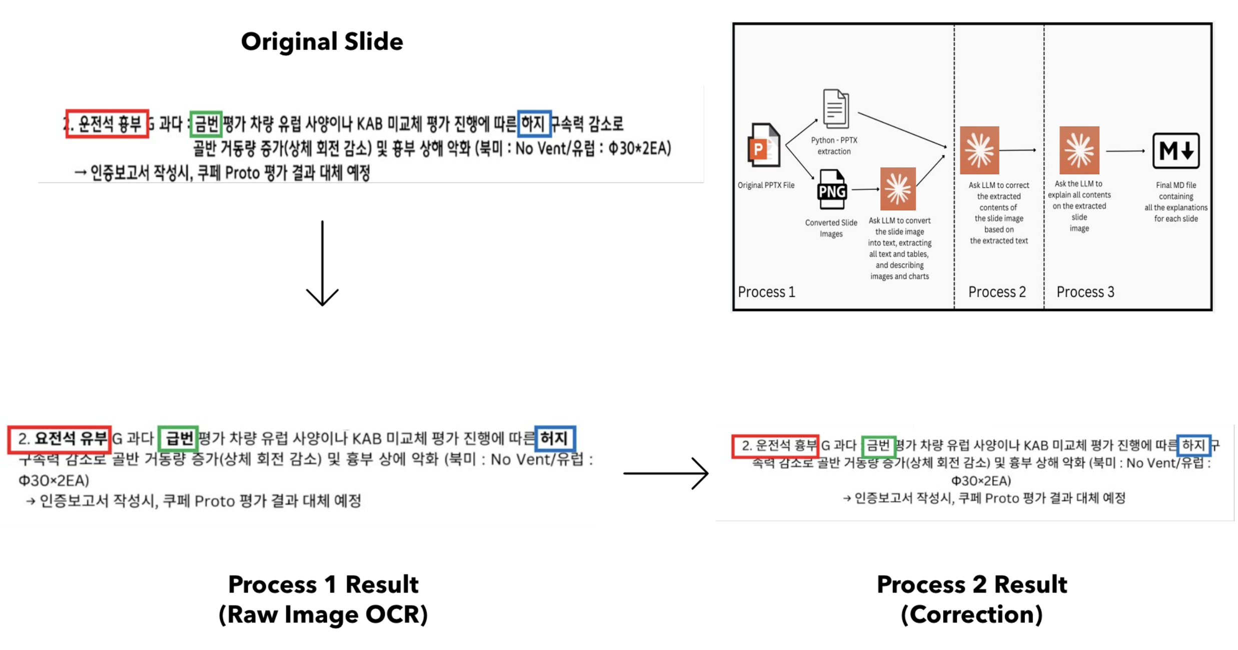
With our generated synthetic data, we evaluate the quality of the extracted data through qualitative and quantitative analysis. We also delve into the different between process 1 documents, documents that have no correction, and process 2 documents, documents that have been corrected with text directly extracted from the slide presentations. We find the following commonalities and improvements in our processing pipeline.

5.1.1 Qualitative Analysis

From our qualitative analysis we find different examples of the usefulness of the OCR and data correction that occurs in process 2, as well as the results of chart and image extractions.

Text correction

The two-stage process also demonstrates remarkable efficacy in correcting Optical Character Recognition (OCR) errors and typos, particularly in multilingual contexts. A salient example observed in our Korean language test cases shows the system's ability to differ-


Original PPTX File
Python - PPTX extraction
Converted Slide Images
Ask LLM to convert the slide image into text, extracting all text and tables, and describing images and charts
Ask LLM to correct the extracted contents of the slide image based on the extracted text
Ask the LLM to explain all contents on the extracted slide image
Final MD file containing all the explanations for each slide
Process 1
Process 2
Process 3

Original Slide

2. 운전석 흉부 G 과다 : 금번 평가 차량 유럽 사양이나 KAB 미교체 평가 진행에 따른 하지 구속력 감소로
골반 거동량 증가(상체 회전 감소) 및 흉부 상해 악화 (북미 : No Vent/유럽 : Φ30*2EA)
→ 인증보고서 작성시, 쿠페 Proto 평가 결과 대체 예정

2. 요전석 유부 G 과다 급번 평가 차량 유럽 사양이나 KAB 미교체 평가 진행에 따른 허지
구속럭 감소로 골반 거동량 증가(상체 회전 감소) 및 흉부 상에 악화 (북미 : No Vent/유럽 :
Φ30×2EA)
→ 인증보고서 작성시, 쿠페 Proto 평가 결과 대체 예정

Process 1 Result
(Raw Image OCR)

2. 운전석 흉부 G 과다 금번 평가 차량 유럽 사양이나 KAB 미교체 평가 진행에 따른 하지 구
속력 감소로 골반 거동량 증가(상체 회전 감소) 및 흉부 상해 악화 (북미 : No Vent/유럽 :
Φ30×2EA)
→ 인증보고서 작성시, 쿠페 Proto 평가 결과 대체 예정

Process 2 Result
(Correction)

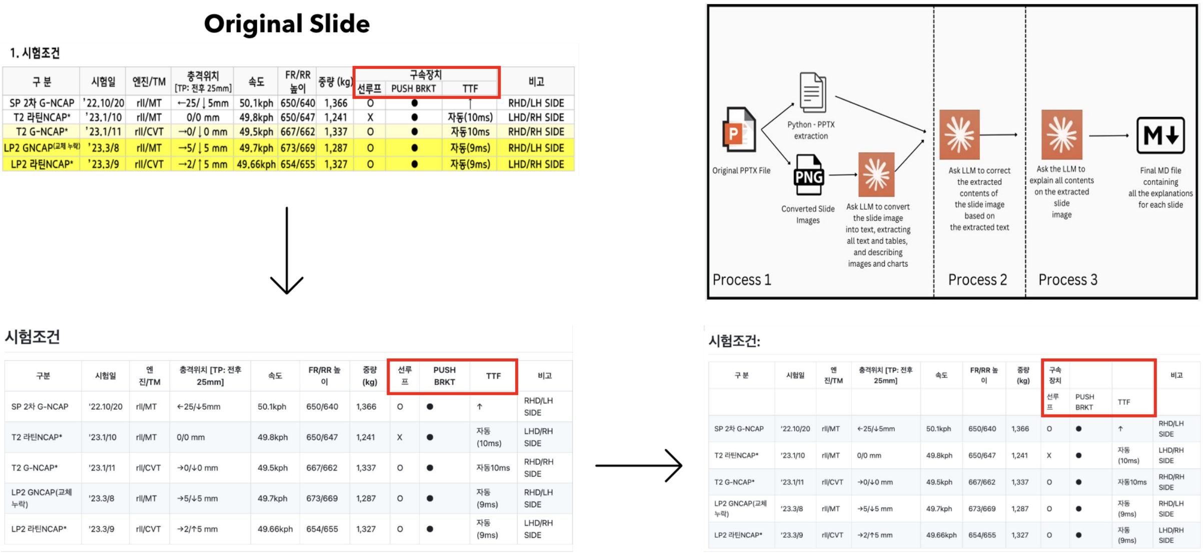
Figure 5.1: Each color represents a phrase or word as it goes through the process and shows character correction. The text in the process 2 results matches the original slide, while process 1 contains mistakes.

entiate between visually similar but semantically distinct characters. For instance, the correction of ”힘” (power) to ”휠” (wheel) illustrates the system’s enhanced contextual understanding and linguistic nuance recognition.

Perhaps most notably, LLM employed in our pipeline exhibits an improved capacity for contextual comprehension. This is evidenced by its ability to discern and reconcile differences between the initially extracted text and the LLM-processed slide information. Such contextual awareness is crucial for accurate interpretation and representation of slide content, especially in cases where direct translation or transcription might lead to errors.

Table correction

In addition to OCR corrections, process 2 also addresses errors in table structure. Although the LLM accurately recognizes the table contents, its initial extraction can sometimes result in a disorganized structure. As illustrated in Figure 5.2, a missing component is reinstated during process 2, which notably alters the table’s interpretation. While the



Original Slide

1. 시험조건

구 분	시험일	엔진/TM	충격위치 [TP: 전후 25mm]	속도	FR/RR 높이	중량 (kg)	구속장치			비고
							선루프	PUSH BRKT	TTF	
SP 2차 G-NCAP	'22.10/20	rII/MT	←25/↓5mm	50.1kph	650/640	1,366	O	●	↑	RHD/LH SIDE
T2 라틴NCAP*	'23.1/10	rII/MT	0/0 mm	49.8kph	650/647	1,241	X	●	자동(10ms)	LHD/RH SIDE
T2 G-NCAP*	'23.1/11	rII/CVT	→0/↓0 mm	49.5kph	667/662	1,337	O	●	자동10ms	RHD/RH SIDE
LP2 GNCAP(교체 누락)	'23.3/8	rII/MT	→5/↓5 mm	49.7kph	673/669	1,287	O	●	자동(9ms)	RHD/LH SIDE
LP2 라틴NCAP*	'23.3/9	rII/CVT	→2/↑5 mm	49.66kph	654/655	1,327	O	●	자동(9ms)	LHD/RH SIDE

Original PPTX File
Python - PPTX extraction
Converted Slide Images
PNG
Ask LLM to convert the slide image into text, extracting all text and tables, and describing images and charts
Ask LLM to correct the extracted contents of the slide image based on the extracted text
Ask the LLM to explain all contents on the extracted slide image
Final MD file containing all the explanations for each slide
Process 1
Process 2
Process 3

시험조건

구분	시험일	엔진/TM	충격위치 [TP: 전후 25mm]	속도	FR/RR 높이	중량 (kg)	선루프	PUSH BRKT	TTF	비고
SP 2차 G-NCAP	'22.10/20	rII/MT	←25/↓5mm	50.1kph	650/640	1,366	O	●	↑	RHD/LH SIDE
T2 라틴NCAP*	'23.1/10	rII/MT	0/0 mm	49.8kph	650/647	1,241	X	●	자동 (10ms)	LHD/RH SIDE
T2 G-NCAP*	'23.1/11	rII/CVT	→0/↓0 mm	49.5kph	667/662	1,337	O	●	자동10ms	RHD/RH SIDE
LP2 GNCAP(교체 누락)	'23.3/8	rII/MT	→5/↓5 mm	49.7kph	673/669	1,287	O	●	자동 (9ms)	RHD/LH SIDE
LP2 라틴NCAP*	'23.3/9	rII/CVT	→2/↑5 mm	49.66kph	654/655	1,327	O	●	자동 (9ms)	LHD/RH SIDE

Process 1 Result
(Raw Image OCR)

시험조건:

구 분	시험일	엔진/TM	충격위치 [TP: 전후 25mm]	속도	FR/RR 높이	중량 (kg)	구속 장치			비고
							선루프	PUSH BRKT	TTF	
SP 2차 G-NCAP	'22.10/20	rII/MT	←25/↓5mm	50.1kph	650/640	1,366	O	●	↑	RHD/LH SIDE
T2 라틴NCAP*	'23.1/10	rII/MT	0/0 mm	49.8kph	650/647	1,241	X	●	자동 (10ms)	LHD/RH SIDE
T2 G-NCAP*	'23.1/11	rII/CVT	→0/↓0 mm	49.5kph	667/662	1,337	O	●	자동10ms	RHD/RH SIDE
LP2 GNCAP(교체 누락)	'23.3/8	rII/MT	→5/↓5 mm	49.7kph	673/669	1,287	O	●	자동 (9ms)	RHD/LH SIDE
LP2 라틴NCAP*	'23.3/9	rII/CVT	→2/↑5 mm	49.66kph	654/655	1,327	O	●	자동 (9ms)	LHD/RH SIDE

Process 2 Result
(Correction)

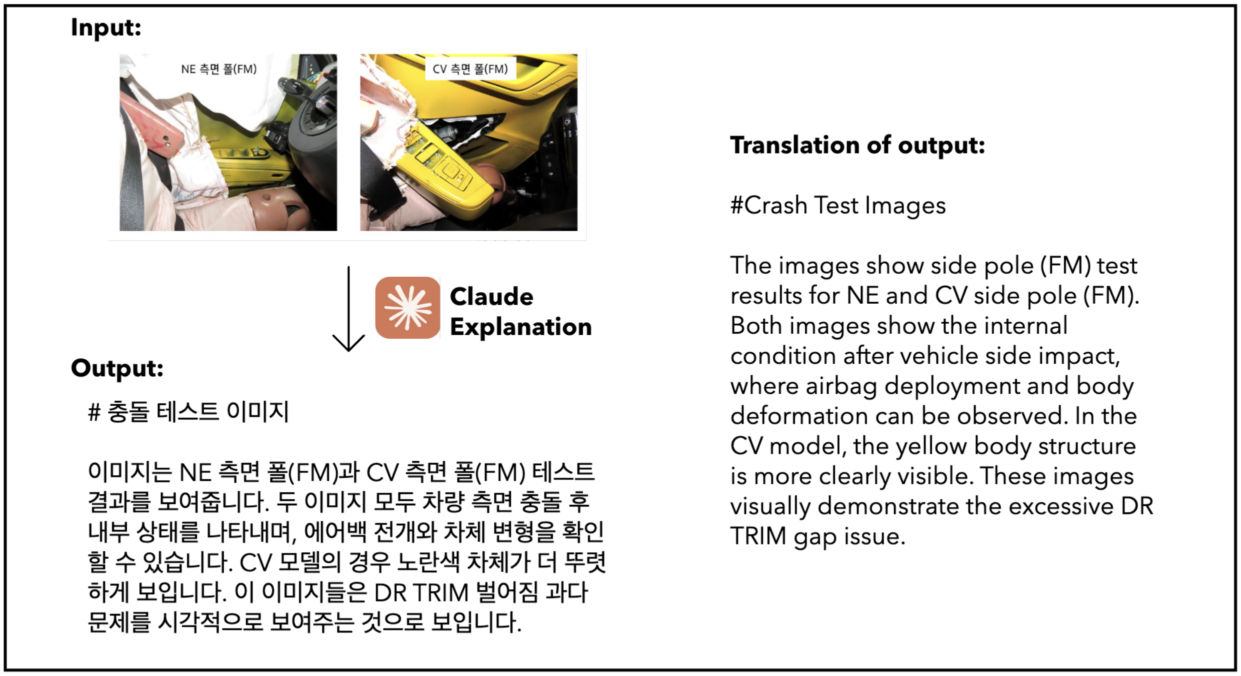
Figure 5.2: The Red box indicates a structural change that occurs through the self-correction step in process 2.

resulting formatting may appear suboptimal—yielding three separate boxes at the top rather than a single merged box—this is solely due to the limitations of markdown table formatting and is deemed acceptable for our purposes. Capturing complex the complex table structures is significantly enhanced with the self correction procedure and there are many instances where the above example occurs.

Image Extraction

For the images, we prompt the LLM to interpret and describe the content based on the slide's context. Our data processing pipeline yields satisfactory results, generating detailed explanations of the visual content. Although the analysis may not match the precision of a Hyundai Engineer, it adds significant value by offering a descriptive understanding and identifying tested car components and clarifying what is depicted.

However, there are times when the image explanation is very generic. In these cases, it simply describes basic visual elements, such as colors or objects that appear in the image,



Input:

Output:

충돌 테스트 이미지

이미지는 NE 측면 폴(FM)과 CV 측면 폴(FM) 테스트 결과를 보여줍니다. 두 이미지 모두 차량 측면 충돌 후 내부 상태를 나타내며, 에어백 전개와 차체 변형을 확인할 수 있습니다. CV 모델의 경우 노란색 차체가 더 뚜렷하게 보입니다. 이 이미지들은 DR TRIM 벌어짐 과다 문제를 시각적으로 보여주는 것으로 보입니다.

Translation of output:

#Crash Test Images

The images show side pole (FM) test results for NE and CV side pole (FM). Both images show the internal condition after vehicle side impact, where airbag deployment and body deformation can be observed. In the CV model, the yellow body structure is more clearly visible. These images visually demonstrate the excessive DR TRIM gap issue.

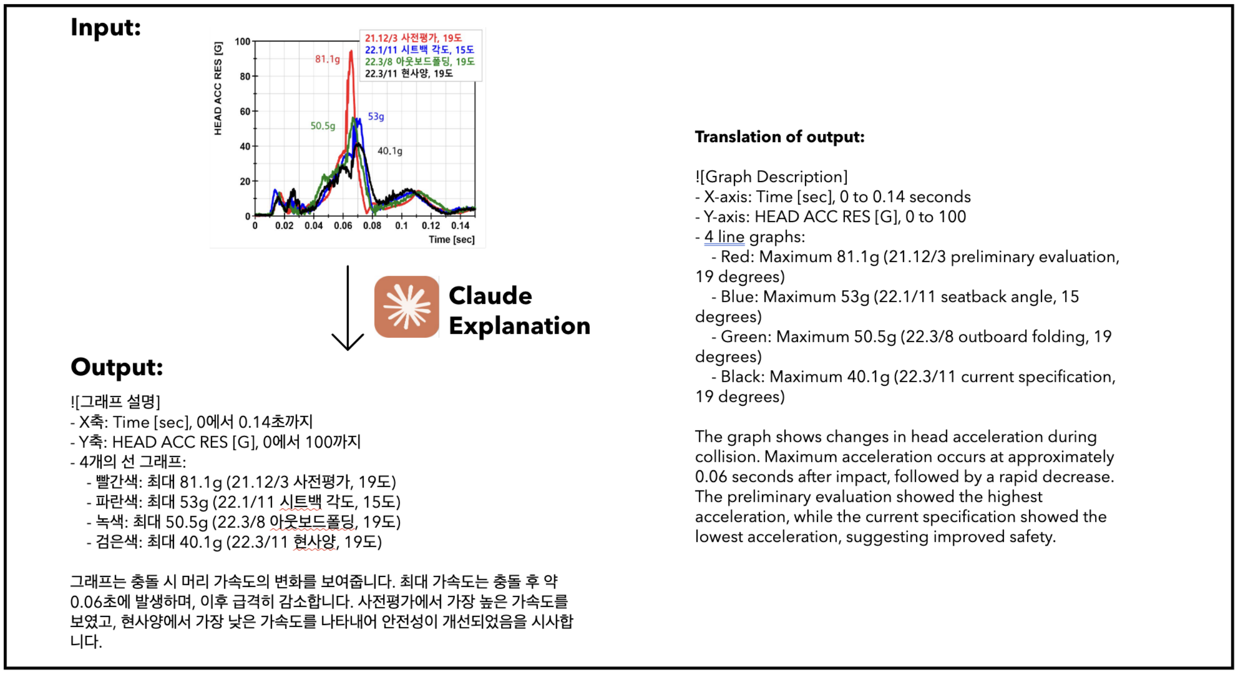
Figure 5.3: An image is given to the LLM to generate a descriptive explanation that showcases the image's purpose.

instead of providing an explanation that draws on the slide context. While these outputs may be useful, they are not as significant as the more detailed explanations generated earlier. It is important to note that the explanations are not incorrect; they simply do not offer the level of detail that an engineer might provide.

Chart Extraction

For the charts, we prompt the LLM to interpret and accurately represent the numerical values as well as any legend or key associated with the chart. The model is expected to identify elements such as the x and y axis and note any highlighted values or trends present in the graph. It is also able to relate the chart to the context of the overall slide image that is sent to the LLM.

In some cases, the analysis of charts can be overly broad or may contain inaccuracies. This issue tends to arise when there are several charts in a single slide image or when a chart contains multiple components, making it challenging to determine which data points



Input:

Output:

![그래프 설명]
- X축: Time [sec], 0에서 0.14초까지
- Y축: HEAD ACC RES [G], 0에서 100까지
- 4개의 선 그래프:
  - 빨간색: 최대 81.1g (21.12/3 사전평가, 19도)
  - 파란색: 최대 53g (22.1/11 시트백 각도, 15도)
  - 녹색: 최대 50.5g (22.3/8 아웃보드폴딩, 19도)
  - 검은색: 최대 40.1g (22.3/11 현사양, 19도)

그래프는 충돌 시 머리 가속도의 변화를 보여줍니다. 최대 가속도는 충돌 후 약 0.06초에 발생하며, 이후 급격히 감소합니다. 사전평가에서 가장 높은 가속도를 보였고, 현사양에서 가장 낮은 가속도를 나타내어 안전성이 개선되었음을 시사합니다.

Translation of output:

![Graph Description]
- X-axis: Time [sec], 0 to 0.14 seconds
- Y-axis: HEAD ACC RES [G], 0 to 100
- 4 line graphs:
  - Red: Maximum 81.1g (21.12/3 preliminary evaluation, 19 degrees)
  - Blue: Maximum 53g (22.1/11 seatback angle, 15 degrees)
  - Green: Maximum 50.5g (22.3/8 outboard folding, 19 degrees)
  - Black: Maximum 40.1g (22.3/11 current specification, 19 degrees)

The graph shows changes in head acceleration during collision. Maximum acceleration occurs at approximately 0.06 seconds after impact, followed by a rapid decrease. The preliminary evaluation showed the highest acceleration, while the current specification showed the lowest acceleration, suggesting improved safety.

Figure 5.4: An image of a chart is given to the LLM to generate a descriptive portrayal of the chart, picking out key values and parameters.

correspond to a specific key. This limitation highlights the challenges in optical character recognition and the visual understanding capabilities of even the most advanced language models, such as Claude.

Example Generation

As discussed in the explanation generation section, we initially attempted to produce slide explanations that ignore insignificant markdown formatting characters, such as those found in markdown tables. Although these explanations do not detail every individual cell, they provide an overarching summary of key trends and significant figures present in the table. The explanation process was mostly for these tables, as well as to potentially create a more coherent understanding of the overall premise of the slide and give more descriptive value of the talking points present. While this explanation generation process can reduce the overall retrieval performance of our RAG systems, we present the concept for those interested in replicating or extending this work in the future.

5.1.2 Quantitative Analysis

Domain-specific Abbreviations

Throughout our data quality analysis for the explanations, we noticed an interesting pattern in the synthetic data: the use of abbreviations. Numerous domain specific and company specific abbreviations appear in the slide presentations. These abbreviations refer to various car and car crash safety terms, but they may be unclear to an LLM because it lacks the internal expertise found in companies like Hyundai.

To investigate further, we randomly sampled 50 documents that include 18 predetermined abbreviations provided by a Hyundai engineer and evaluated whether the abbreviations were correctly interpreted in the explanations. Among 338 instances of use, 12 instances (approximately 3.55%) were misinterpreted. Further analysis revealed that all 12 errors were associated with 4 of the 18 abbreviations. These abbreviations include RTE, which stands for Restraint Energy, DAB for Driver Airbag, TTF for Time to Fire, and C/PAD for Crash Pad. The abbreviation most frequently misinterpreted was RTE, which was often read as Return to Earth, a common meaning of the RTE acronym. This error accounted for over half of the misinterpretations.

Token Count Analysis: Raw OCR vs. Self-Correction Synthetic Data

Further evidence of the improvements between the first and second prompts can be observed in the changes in token count. An analysis of 63 files revealed that 46 files (73%) experienced changes in their token counts. Specifically, 31 markdown files showed an increase in token count, while 15 files demonstrated a decrease. The average difference in token count was 4.63 tokens, with a standard deviation of 25.55.

At first glance, the average change of 4.63 tokens might not seem substantial. However, the high standard deviation of 25.55 is particularly noteworthy. This large spread indicates significant variability in the changes across files, suggesting that some files underwent

Metric	**Value**
Total tokens (1st CSV)	27745
Total tokens (2nd CSV)	28037
Overall token change	292
Mean token difference	4.63
Std. dev. of difference	25.55
Files with increased tokens	31
Files with decreased tokens	15
Files with no change	17
Largest increase in tokens	23.68%
Largest decrease in tokens	14.24%
Hallucination by token count	6.74%

Table 5.1: Token Count Analysis Statistics(n=63)

much more extensive modifications than others. This variability is crucial to our understanding of the process. Larger increases in token counts represent cases where the second prompt successfully captured missing information, such as previously omitted table data or misrepresented content. These instances align with our earlier observations regarding improved table extraction and content representation. It is also important to note, that even the files that contained no token count change had small typo corrections as well.

Conversely, some files experienced relatively large decreases in token count. While this might initially seem concerning, closer examination of these cases is necessary to understand the nature of these reductions. These decreases represent the correction of redundancies or the refinement of improperly extracted content, rather than the loss of valuable information.

It is important to note that after the self-correction step, the hallucination rate by token count was 6.74%. This indicates that while we can correct many typographical errors and inaccuracies, some errors remain. To calculate this statistic, we tokenized the text and manually checked each word or phrase for inaccuracies. The percentage of hallucination was calculated for each document and averaged together for the final result.

The wide range of token count changes underscores the complex nature of the refinement process between the first and second prompts. It highlights the system's ability to make substantial corrections where needed, while also performing more subtle refinements in other cases. This nuanced approach contributes to the overall improvement in the quality and accuracy of the extracted content, reinforcing the value of our two-stage prompting method.

5.2 RAG Frameworks & Retrieval Analysis

5.2.1 Embedding Models

Among the fine-tuned BGE-M3 embedding models, training solely on slide-extracted data consistently yielded the highest performance across both datasets. Although the inclusion of explanation data provided a performance boost, it did not match the retrieval quality achieved by the slide-only model. This result is surprising given our initial hypothesis that explanations would offer a clearer and more cohesive view of the slide content. One possible explanation is that the increase in document length and complexity - when slides are combined with explanations - makes learning an effective vector representation more challenging. Alternatively, the explanation data may not contribute significant additional semantic value, as they often contain repetitive information despite some novel details in table or chart descriptions. Overall, it could be said the short and concise slide extractions allow for the embedding model to more clearly understand patterns or contrasts within our data.

5.2.2 RAG Frameworks

Among the tested frameworks, the original RAG significantly outperformed both variants of the HyDE methodology. This finding is unexpected, considering that many HyDE variants have been reported to surpass the original RAG in performance. One potential reason is that the embedding model was not fine-tuned to use answers as part of the user query, but rather only the question itself. Furthermore, incorporating more domain-specific terms from the hypothetical answer generation did not seem to improve the performance, showcased especially when both the question and answer are combined before utilizing the query to search for documents. Although alternative RAG frameworks might be more effective in different scenarios, our results indicate that even a widely used variant of RAG performs under the original RAG framework in our specific use case. The best performance was achieved by combining RAG with an embedding model fine-tuned solely on slide data, while restricting retrieval exclusively to slide content without explanations. We are able to create an effective RAG system that pulls in relevant documents on low-resource domain-specific data.

5.3 Qwen 2.5 VL Fine-tuning Analysis

5.3.1 Image Dimensions

Our findings indicate that larger image dimensions significantly improve performance in image understanding, OCR, and descriptive analysis of multimodal elements on slides. It is important to note that, contrary to Qwen's recommended 448×448 image dimension size, we do not utilize a 1:1 image dimension ratio. Instead, to maintain the aspect ratio of the original slide image, we used a 4:3 image dimension (448×336), which could have led to the performance difference between synthetic data generation with 448×448 and 448×336 image sizes. Although the width was consistent and the size of 448x448 is larger, the difference between a square shape and a rectangular shape seems to significantly alter performance.

We also note that at a certain point in image resolution, there are diminishing returns. While the gap between the 448×336 fine tuned variant and the 1280×960 fine tuned variant was very noticeable, the performance difference between the 1280×960 fine tuned variant and the 2400×1800 fine tuned Qwen2.5-VL-7B model led to much smaller differences in metrics and performance across the LLM-as-Judge evaluation.

5.3.2 Base Model vs. Finetuned Model Data

As shown in Table 4.7, there is a notable difference in semantic similarity when comparing data generated by the base model to the golden document versus data from the fine-tuned model. Fine-tuning the Qwen model with our own images has led to key improvements, including enhanced table generation, higher-quality image descriptions, improved chart interpretation, and more accurate OCR capabilities.

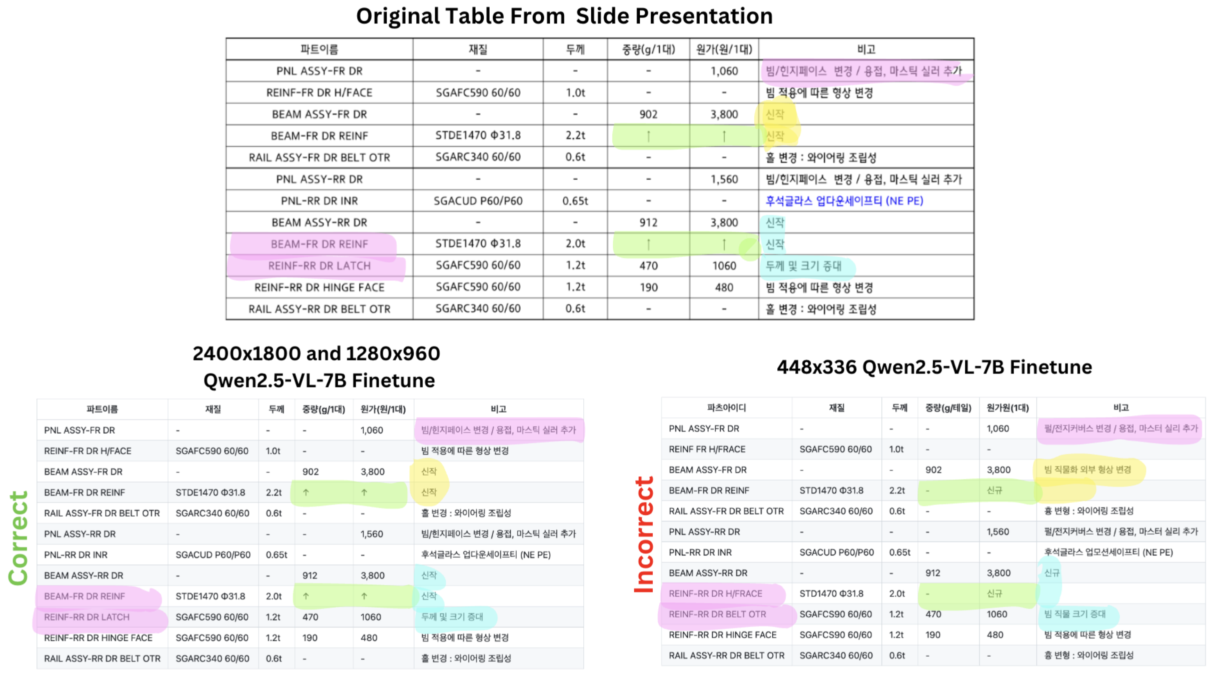
To assess these enhancements, we conducted a qualitative comparison of 50 documents generated by both the base and 2400x1800 variants of the fine-tuned models, focusing on various aspects of OCR and synthetic data generation. One immediate improvement is a reduction in spelling errors. We compared 50 documents from each model to our originally processed documents—omitting any word that does not appear in the golden document—and found that the base model's output contained an average of 2.6875 mispelled words, whereas the fine-tuned model's output averaged 1.0625 mispelled words. In effect, for every three spelling errors in the base model's synthetic data, there is only about one error in the fine-tuned variant.

During this evaluation, we also observed an interesting phenomenon: occasionally, Korean words were rendered in Chinese characters and vice versa. Although rare, this pattern was consistently noted across our data. This behavior may be attributed to the origins of the Qwen model, which was developed in China and is likely trained on extensive Chinese corpora and documents. Also, across both datasets but much more frequent in the base synthetic data, the markdown tables are sometimes formatted with ”¡br¿” rather than the ”—” character.

We also assessed the number of missing components or data points that were not captured from the images. As expected, the fine-tuned models successfully extracted all elements present in the slide presentations. In contrast, the base model occasionally missed certain aspects, such as image and chart descriptions, table extractions, or OCR text. Among the 50 documents reviewed, 6 exhibited missing features. Notably, both the fine-tuned and base models sometimes produced generic or placeholder-like image descriptions. However, it is important to note that the fine-tuned data tends to include more "placeholders" rather than actual descriptions for image data. This occurs because the golden documents sometimes contained placeholders like "[IMG CAR CRASH].jpg" instead of detailed descriptions, leading the model to learn these patterns. To mitigate this issue, future improvements could involve using cleaner documents and incorporating synthetic data.

Overall, fine-tuning significantly enhances the modeling of our golden data, improving both character recognition and information extraction from image presentations. Further improvements could be achieved by utilizing the full three-step processing pipeline rather than attempting to extract all information in a single input-output sequence through the LLM.

5.3.3 LLM Judge Preference - Qualitative Analysis


Correct
Incorrect

Original Table From Slide Presentation

파트이름	재질	두께	중량(g/1대)	원가(원/1대)	비고
PNL ASSY-FR DR	-	-	-	1,060	빔/힌지페이스 변경 / 용접, 마스틱 실러 추가
REINF-FR DR H/FACE	SGAFC590 60/60	1.0t	-	-	빔 적용에 따른 형상 변경
BEAM ASSY-FR DR	-	-	902	3,800	신작
BEAM-FR DR REINF	STDE1470 Φ31.8	2.2t	↑	↑	신작
RAIL ASSY-FR DR BELT OTR	SGARC340 60/60	0.6t	-	-	홀 변경 : 와이어링 조립성
PNL ASSY-RR DR	-	-	-	1,560	빔/힌지페이스 변경 / 용접, 마스틱 실러 추가
PNL-RR DR INR	SGACUD P60/P60	0.65t	-	-	후석글라스 업다운세이프티 (NE PE)
BEAM ASSY-RR DR	-	-	912	3,800	신작
BEAM-FR DR REINF	STDE1470 Φ31.8	2.0t	↑	↑	신작
REINF-RR DR LATCH	SGAFC590 60/60	1.2t	470	1060	두께 및 크기 증대
REINF-RR DR HINGE FACE	SGAFC590 60/60	1.2t	190	480	빔 적용에 따른 형상 변경
RAIL ASSY-RR DR BELT OTR	SGARC340 60/60	0.6t	-	-	홀 변경 : 와이어링 조립성

2400x1800 and 1280x960 Qwen2.5-VL-7B Finetune

파트이름	재질	두께	중량(g/1대)	원가(원/1대)	비고
PNL ASSY-FR DR	-	-	-	1,060	빔/힌지페이스 변경 / 용접, 마스틱 실러 추가
REINF-FR DR H/FACE	SGAFC590 60/60	1.0t	-	-	빔 적용에 따른 형상 변경
BEAM ASSY-FR DR	-	-	902	3,800	신작
BEAM-FR DR REINF	STDE1470 Φ31.8	2.2t	↑	↑	신작
RAIL ASSY-FR DR BELT OTR	SGARC340 60/60	0.6t	-	-	홀 변경 : 와이어링 조립성
PNL ASSY-RR DR	-	-	-	1,560	빔/힌지페이스 변경 / 용접, 마스틱 실러 추가
PNL-RR DR INR	SGACUD P60/P60	0.65t	-	-	후석글라스 업다운세이프티 (NE PE)
BEAM ASSY-RR DR	-	-	912	3,800	신작
BEAM-FR DR REINF	STDE1470 Φ31.8	2.0t	↑	↑	신작
REINF-RR DR LATCH	SGAFC590 60/60	1.2t	470	1060	두께 및 크기 증대
REINF-RR DR HINGE FACE	SGAFC590 60/60	1.2t	190	480	빔 적용에 따른 형상 변경
RAIL ASSY-RR DR BELT OTR	SGARC340 60/60	0.6t	-	-	홀 변경 : 와이어링 조립성

448x336 Qwen2.5-VL-7B Finetune

파츠아이디	재질	두께	중량(g/테일)	원가원(1대)	비고
PNL ASSY-FR DR	-	-	-	1,060	펄/전지커버스 변경 / 용접, 마스터 실리 추가
REINF FR H/FRACE	SGAFC590 60/60	1.0t	-	-	
BEAM ASSY-FR DR	-	-	902	3,800	빔 직물화 외부 형상 변경
BEAM-FR DR REINF	STD1470 Φ31.8	2.2t	-	신규	
RAIL ASSY-FR DR BELT OTR	SGARC340 60/60	0.6t	-	-	흉 변형 : 와이어링 조립성
PNL ASSY-RR DR	-	-	-	1,560	펄/전지커버스 변경 / 용접, 마스터 실리 추가
PNL-RR DR INR	SGACUD P60/P60	0.65t	-	-	후석글라스 업모션세이프티 (NE PE)
BEAM ASSY-RR DR	-	-	912	3,800	신규
REINF-RR DR H/FRACE	STD1470 Φ31.8	2.0t	-	신규	
REINF-RR DR BELT OTR	SGAFCS90 60/60	1.2t	470	1060	빔 직물 크기 증대
REINF-RR DR HINGE FACE	SGAFCS90 60/60	1.2t	190	480	빔 적용에 따른 형상 변경
RAIL ASSY-RR DR BELT OTR	SGARC340 60/60	0.6t	-	-	흉 변형 : 와이어링 조립성

Figure 5.5: We highlight different corresponding areas of each of the tables in different colors, showcasing the correctness of each of the outputs from the different finetune variants.

Our evaluation reveals a clear preference pattern: GPT-4o consistently favors data generated by Qwen models fine-tuned on larger image dimensions compared to our original synthetic data. Closer analysis confirms this preference is justified, as the quality of content from smaller-dimension models is demonstrably lower, particularly in table generation tasks.

The smaller-dimension models exhibit a higher rate of hallucination, frequently producing incorrect cell values and header organizations. Figure 5.5 illustrates this disparity—while both dimension variants maintain correct table structure, critical content differences emerge upon inspection. In the highlighted example, the original table contains ”신작” in the third and fourth cells of the last column. The larger-dimension model correctly preserves this term, whereas the smaller-dimension variant incorrectly substitutes ” 빔 직물화 외부 형상 변경” in the third column and leaves the second column blank. Similar errors appear consistently throughout the table.

For image descriptions, quality remains relatively consistent across all three synthetic data variants. Each variant produces detailed explanations that capture the maximum visual information permitted by the VLLM's capabilities. Formatting analysis reveals another advantage of larger-dimension files: they demonstrate more consistent document organization patterns compared to the lower-dimension models. This consistency manifests in header usage, text organization, chart interpretations, and data arrangement when converting visual elements like bar charts into tabular format. This finding aligns with our quantitative observations, where we recorded very similar BERT scores when comparing these documents to the original Claude-processed synthetic data.

OCR character recognition remains highly consistent across all models, with only minor errors in text extraction. In general, larger image dimensions improve the accuracy of visual extractions, particularly for table and chart data, making them preferable for the LLM.

Chapter 6

Conclusion

6.0.1 Future Directions

Following our experiments, several research avenues and potential improvements can be explored to further enhance retrieval-augmented generation (RAG) systems for low-resource domains:

1. **Qwen2.5-VL-72B Variant** – In our exploration of open-source data processing, we utilized the 7B variant of the Qwen2.5-VL model series. While effective, the performance may have been constrained by the model's size. Future research could investigate the benefits of scaling to larger variants, such as the 72B model, which may enable more comprehensive information extraction and improved processing capabilities.

2. **Advancing RAG Frameworks** – Although we evaluated RAG frameworks that best suited our use case, the rapid development of new retrieval-augmented generation methodologies presents opportunities for further improvements. Exploring and integrating newer frameworks could yield significant gains in retrieval accuracy, response quality, and multimodal integration.

3. **True Multimodality** – Rather than converting all multimodal elements into text,

future work could explore direct multimodal processing. At the time of our research, multimodal embeddings, vision-language models, and embedding techniques were still in their early stages. However, with advancements in vision-language (VL) models, a more sophisticated multimodal RAG approach could improve retrieval effectiveness by directly leveraging visual, positional, and textual information.

4. **Enhanced Data Refinement** – While synthetic data generated using Claude proved valuable, further refinements could enhance data quality and retrieval accuracy. Future research could explore alternative methods for synthetic data generation, including leveraging multiple AI models, incorporating domain-specific fine-tuning, and refining preprocessing techniques to improve the consistency and relevance of extracted information.

By addressing these research directions, future work can build on our findings to develop more robust, efficient, and scalable RAG systems for low-resource domains.

6.0.2 Conclusion

This thesis explored the development of a retrieval-augmented generation (RAG) system for low-resource domains, focusing on multimodal data processing in the automotive safety sector in Korean. Through collaboration with Hyundai, we designed and evaluated a novel data processing pipeline that converts slide-based visual information into structured textual representations, enhancing retrieval and question-answering capabilities. Additionally, we examined variations in retrieval and fine-tuned data within different RAG frameworks, systematically testing multiple retrieval strategies and embedding model fine-tuning techniques to determine the most effective approach for domain-specific applications.

Our findings demonstrate that fine-tuning both language and embedding models significantly improves retrieval accuracy and information extraction, particularly when applied to specialized datasets. Notably, fine-tuning on slide-based data yielded superior performance compared to models incorporating additional explanations. Furthermore, our experiments underscore the importance of larger image dimensions in improving OCR accuracy and structured data extraction, emphasizing the need for high-resolution processing in multimodal RAG applications.

Building on these insights, we explored advancements in vision language models, fine-tuning Qwen2.5-VL-7B with various image dimensions to replicate our closed-source data processing pipeline. The fine-tuned models exhibited significant improvements over the base model, validated through qualitative and quantitative analyses. While challenges remain in optimizing model output quality, our results demonstrate meaningful progress in fine-tuning open-source models, particularly in Korean OCR. Integrating this model into our three-stage processing pipeline could further enhance retrieval performance and data quality.

Despite these improvements, multimodal processing still presents challenges. While tabular structure extraction improved with fine-tuning, further refinements are needed for table processing, chart interpretation, and image description quality. Developing a more systematic approach for extracting and structuring chart data could enhance overall retrieval effectiveness.

Ultimately, this work advances the development of robust, scalable RAG frameworks for low-resource domains, providing a foundation for future innovations in information retrieval, question-answering systems, and AI-driven knowledge management. We are able to convert multimodal data into a usable format that can be retrieved and utilized in LLMs. It is possible to develop an effective RAG application for low-resource domains, even in languages like Korean where resources for LLMs, embedding models, and data are scarce.

Appendix A

Appendix

We utilized two different prompts to process the synthetic data, converting from images to text.

A.0.1 Converting image-to-text prompt:

This prompt was used to extract all elements on the slide image into usable textual elements.

```
"""

1. Examine the provided image of a PowerPoint slide from a car collision safety report.

2. Convert all content into markdown format, preserving the original structure and information

3. Use a single hashtag (  # ) for section headings only. Do not use any variation of hashtags
anything else. Only use one hashtag for each section heading. All heading MUST have section co
underneath it. There should never be two headings in a row without content in between.

4. Each graphic element type should have a section heading. This means tables, graphs, and imag
   should each have their own section heading. Text should be included in the appropriate secti
   or if different enough, have its own section.
```

5. Structure your output as follows:

```
< markdown_conversion >
# Heading appropriate for section content underneath
[section content goes here]
< / markdown_conversion >
```

6. For each element type:

Text:

- Convert to markdown, maintaining structure, bullet points, and numbering.
- Include all text: titles, subtitles, and body.

Tables:

- Convert to markdown table format.
- Maintain original content and structure.
- Provide a detailed interpretation after each table.
- For images in cells, describe them within the appropriate cell in markdown.

Graphs and Images:

- For graphs, try to recreate the data in markdown table format.
- Describe in detail: type, labels, data representation, trends.
- Explain significance in the context of car collision safety under it.

Remember:

- Do not use unnecessary affirmations or filler phrases.
- Do not include personal opinions or anecdotes.
- Use markdown for code snippets if applicable.

```
"""
```

A.0.2 Correction of results:

This prompt was utilized in Process 2, where we attempted to fix the OCR errors within the slide extraction.

```
'''
Original Analysis:
{concatenated_text}


Extracted Text:
{extracted_slide}


Instructions:
1. Review the Original Analysis for typos and errors.
2. You MUST maintain the content and structure of the Original Analysis.
3. Correct any typos or obvious errors found in the Original Analysis.
4. KEEP ALL SECTIONS OF THE ORIGINAL ANALYSIS INTACT(<markdown_conversion>).
5. If you encounter a table in both the Original Analysis and the Extracted Text that are similar,
   replace the table in the Original Analysis with the one from the Extracted Text.
6. If a table from Extracted Text does not exist in the Original Analysis, add it to the corrected analy
6. If the table has image explanations, ensure that the image explanations are included in the correcte
7. Use the Extracted Text as reference for fact checking and typo checking.
8. Fix the exper analysis after all the corrections are made.
9. Return ONLY the CORRECTED version of the Original Analysis in its ENTIRETY.
10. Remove any sections or text that explainn or have the text "Together We Can" or "HYUNDAI MOTOR GROU
'''
```

Bibliography

[1] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Matthijs van der Leeuw, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. 2020.

[2] Nayoung Choi, Grace Byun, Andrew Chung, Ellie S. Paek, Shinsun Lee, and Jinho D. Choi. Trustworthy answers, messier data: Bridging the gap in low-resource retrieval-augmented generation for domain expert systems. `https://doi.org/10.48550/arXiv.2502.19596`, Feb 2025. arXiv:2502.19596v1 [cs.AI].

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and et. al Junyang Lin. Qwen2.5-vl technical report. arXiv preprint, Feb 2025. Submitted on 19 Feb 2025.

[4] DaeHun Nyang David Mohaisen Marwan Omar, Soohyeon Choi. Robust natural language processing: Recent advances, challenges, and future directions. *arXiv preprint*, Jan 2022. Submitted on 3 Jan 2022.

[5] Narjes Nikzad Meysam Chenaghlu Richard Socher Xavier Amatriain Jianfeng Gao

Shervin Minaee, Tomas Mikolov. Large language models: A survey. *arXiv preprint*, Feb 2024. Submitted on 9 Feb 2024 (v1), last revised 20 Feb 2024 (this version, v2).

[6] Cheonsu Jeong. Fine-tuning and utilization methods of domain-specific llms. *Journal of Intelligence and Information Systems*, Jan 2024. doi: 10.48550/arXiv.2401.02981. Submitted on 1 Jan 2024 (v1), last revised 24 Jan 2024 (this version, v2).

[7] Saket Kumar Tala Talaei Khoei Aditi Singh, Abul Ehtesham. Agentic retrieval-augmented generation: A survey on agentic rag. *arXiv preprint*, Jan 2025. Submitted on 15 Jan 2025 (v1), last revised 4 Feb 2025 (this version, v3).

[8] Emmett Coin Diego Gosmar, Deborah A. Dahl. Conversational ai multi-agent interoperability, universal open apis for agentic natural language multimodal communications. *arXiv preprint*, Jul 2024. Submitted on 28 Jul 2024.

[9] Xiaolong Huang Linjun Yang Rangan Majumder Furu Wei Liang Wang, Nan Yang. Improving text embeddings with large language models. *arXiv preprint*, May 2024. Submitted on 31 Dec 2023 (v1), last revised 31 May 2024 (this version, v3).

[10] Pengfei He Yue Xing Yiding Liu Han Xu Jie Ren Shuaiqiang Wang Dawei Yin Yi Chang Jiliang Tang Shenglai Zeng, Jiankun Zhang. The good and the bad: Exploring privacy issues in retrieval-augmented generation (rag). *arXiv preprint*, Feb 2024. Submitted on 23 Feb 2024.

[11] Zackary Rackauckas. Rag-fusion: a new take on retrieval-augmented generation. *International Journal on Natural Language Computing (IJNLC)*, 13(1), Feb 2024. doi: 10.48550/arXiv.2402.03367. Submitted on 31 Jan 2024 (v1), revised 21 Feb 2024 (v2), arXiv:2402.03367v2 [cs.IR].

[12] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*, 2022.

[13] Yizhong Wang Avirup Sil Hannaneh Hajishirzi Akari Asai, Zeqiu Wu. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint*, Oct 2023. Submitted on 17 Oct 2023, 30 pages, 2 figures, 12 tables.

[14] Shishir Subedi Sujit Khanna. Tabular embedding model (tem): Finetuning embedding models for tabular rag applications. *arXiv preprint*, Apr 2024. Submitted on 28 Apr 2024.

[15] Kai Zhang Shiwei Tong Qi Liu Zhaofeng Liu Hao Yu, Aoran Gan. Evaluation of retrieval-augmented generation: A survey. *arXiv preprint*, May 2024. Submitted on 13 May 2024 (v1), last revised 3 Jul 2024 (this version, v2).

[16] Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. `https://arxiv.org/abs/2401.18059v1`, January 2024. arXiv:2401.18059v1 [cs.CL], CC BY 4.0.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423/`.

[18] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November

2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL `https://aclanthology.org/D19-1410/`.

[19] Peitian Zhang Kun Luo Defu Lian Zheng Liu Jianlv Chen, Shitao Xiao. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint*, Feb 2024. Submitted on 5 Feb 2024 (v1), last revised 28 Jun 2024 (this version, v4).

[20] Chris Hallacy Aditya Ramesh Gabriel Goh Sandhini Agarwal Girish Sastry Amanda Askell Pamela Mishkin Jack Clark Gretchen Krueger Ilya Sutskever Alec Radford, Jong Wook Kim. Learning transferable visual models from natural language supervision. *arXiv preprint*, Feb 2021. Submitted on 26 Feb 2021.

[21] Vincent Y. Zhao Kelvin Guu Adams Wei Yu Brian Lester Nan Du Andrew M. Dai Quoc V. Le Jason Wei, Maarten Bosma. Finetuned language models are zero-shot learners. *arXiv preprint*, Sep 2022. Submitted on 3 Sep 2021 (v1), last revised 8 Feb 2022 (this version, v5).

[22] Xiaoya Li Sen Zhang Xiaofei Sun Shuhe Wang Jiwei Li Runyi Hu Tianwei Zhang Fei Wu Guoyin Wang Shengyu Zhang, Linfeng Dong. Instruction tuning for large language models: A survey. *arXiv preprint*, Aug 2024. Submitted on 21 Aug 2023 (v1), last revised 1 Dec 2024 (this version, v8).

[23] Anurag Ajay Alexander C. Li Adrien Bardes Suzanne Petryk Oscar Mañas Zhiqiu Lin Anas Mahmoud Bargav Jayaraman Mark Ibrahim Melissa Hall Yunyang Xiong Jonathan Lebensold Candace Ross Srihari Jayakumar Chuan Guo Diane Bouchacourt Haider Al-Tahan Karthik Padthe Vasu Sharma Hu Xu Xiaoqing Ellen Tan Megan Richards Samuel Lavoie Pietro Astolfi Reyhane Askari Hemmat Jun Chen Kushal Tirumala Rim Assouel Mazda Moayeri Arjang Talattof Kamalika Chaudhuri Zechun Liu Xilun Chen Quentin Garrido Karen Ullrich Aishwarya Agrawal Kate Saenko Asli

Celikyilmaz Vikas Chandra Florian Bordes, Richard Yuanzhe Pang. An introduction to vision-language modeling. *arXiv preprint*, May 2024. Submitted on 27 May 2024.

[24] Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William W. Cohen. Murag: Multimodal retrieval-augmented generator for open question answering over images and text. *arXiv preprint*, October 2022. Submitted on 6 Oct 2022, last revised 20 Oct 2022.

[25] Ziniu Hu, Ahmet Iscen, Chen Sun, Zirui Wang, Kai-Wei Chang, Yizhou Sun, Cordelia Schmid, David A. Ross, and Alireza Fathi. Reveal: Retrieval-augmented visual-language pre-training with multi-source multimodal knowledge memory. *arXiv preprint*, December 2022. Submitted on 10 Dec 2022, last revised 3 Apr 2023.

[26] Dehai Min, Nan Hu, Rihui Jin, Nuo Lin, Jiaoyan Chen, Yongrui Chen, Yu Li, Guilin Qi, Yun Li, Nijun Li, and Qianren Wang. Exploring the impact of table-to-text methods on augmenting LLM-based question answering with domain hybrid data. In Yi Yang, Aida Davani, Avi Sil, and Anoop Kumar, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 464–482, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-industry.41. URL `https://aclanthology.org/2024.naacl-industry.41/`.

[27] Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. On LLMs-driven synthetic data generation, curation, and evaluation: A survey. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11065–11082, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.658. URL `https://aclanthology.org/2024.findings-acl.658/`.

[28] Siamak Shakeri, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. End-to-end synthetic data generation for domain adaptation of question answering systems. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5445–5460, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.439. URL `https://aclanthology.org/2020.emnlp-main.439/`.

[29] Pierre Boyeau, Anastasios N. Angelopoulos, Nir Yosef, Jitendra Malik, and Michael I. Jordan. Autoeval done right: Using synthetic data for model evaluation. *arXiv preprint*, March 2024. Submitted on 9 Mar 2024, last revised 28 May 2024.

[30] Ziije Zhong, Linqing Zhong, Zhaoze Sun, Qingyun Jin, Zengchang Qin, and Xiaofan Zhang. Synthet2c: Generating synthetic data for fine-tuning large language models on the text2cypher task. *arXiv preprint*, June 2024. Submitted on 15 Jun 2024, last revised 26 Jan 2025.

[31] Kichang Yang. Transformer-based korean pretrained language models: A survey on three years of progress. *arXiv preprint*, January 2024. Undergraduate Student, Soongsil University.

[32] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint*, April 2019. Submitted on 21 Apr 2019, last revised 24 Feb 2020.

[33] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint*, June 2023. Submitted on 9 Jun 2023, last revised 24 Dec 2023.