

REPUBLIC OF TÜRKİYE
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

**ENHANCING RETRIEVAL-AUGMENTED
GENERATION ACCURACY WITH DYNAMIC
CHUNKING AND OPTIMIZED VECTOR SEARCH**

Derya TANYILDIZ

MASTER OF SCIENCE THESIS
Department of Computer Engineering
Program of Computer Engineering

Supervisor
Prof. Dr. Mehmet Fatih AMASYALI

Co-supervisor
Assoc. Prof. Dr. Serkan AYVAZ

January, 2025

REPUBLIC OF TÜRKİYE
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

**ENHANCING RETRIEVAL-AUGMENTED GENERATION
ACCURACY WITH DYNAMIC CHUNKING AND
OPTIMIZED VECTOR SEARCH**

A thesis submitted by Derya TANYILDIZ in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 29.01.2025 in Department of Computer Engineering, Program of Computer Engineering.

Prof. Dr. Mehmet Fatih AMASYALI
Yildiz Technical University
Supervisor

Assoc. Prof. Dr. Serkan AYVAZ
University of Southern Denmark
Co-supervisor

Approved By the Examining Committee

Prof. Dr. Mehmet Fatih AMASYALI, Supervisor
Yildiz Technical University

Prof. Dr. Banu DIRI, Member
Yildiz Technical University

Assist. Prof. Dr. Duygu CAKIR, Member
Bahcesehir University

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of Enhancing Retrieval-Augmented Generation Accuracy With Dynamic Chunking And Optimized Vector Search supervised by my supervisor, Prof. Dr. Mehmet Fatih AMASYALI. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Derya TANYILDIZ

Signature

*Dedicated to my dear family who have been
my biggest supporters throughout my education life.*



ACKNOWLEDGEMENTS

I would like to express my profound gratitude to all those who have provided their invaluable support and guidance throughout the course of my thesis research.

First and foremost, I am deeply indebted to my thesis advisors, Prof. Dr. Mehmet Fatih Amasyalı and Assoc. Prof. Dr. Serkan Ayvaz, whose unwavering support, expert guidance, and constructive feedback have been essential to the development and completion of this work. Their profound knowledge and dedication have significantly shaped the direction of this research.

I would also like to extend my sincere thanks to the faculty and staff of the Computer Engineering Department at Yıldız Technical University for fostering an intellectually stimulating environment and for their continuous support. Their encouragement and assistance throughout my academic journey have been greatly appreciated.

Furthermore, I am grateful to my family and friends, whose constant love, understanding, and encouragement have been a constant source of strength and motivation. Their belief in my abilities has been instrumental in overcoming the challenges encountered during the course of this research.

This research would not have been possible without the contributions of all those mentioned above, and I am truly grateful for the opportunity to learn and grow throughout this process.

Derya TANYILDIZ

TABLE OF CONTENTS

LIST OF SYMBOLS	vii
LIST OF ABBREVIATIONS	viii
LIST OF FIGURES	ix
LIST OF TABLES	x
ABSTRACT	xi
ÖZET	xiii
1 INTRODUCTION	1
1.1 Advancements in Large Language Models	1
1.2 Challenges in Traditional RAG Systems	1
1.3 Proposed Optimized RAG Architecture	2
1.4 Contributions to RAG and NLP Research	2
1.5 Focus on Turkish Language and Real-World Applications	2
2 RELATED WORKS	3
2.1 Foundation of Retrieval-Augmented-Generation	3
2.2 Advances in Retrieval and Ranking Techniques	4
2.3 Challenges in Static Chunking Strategies	5
2.4 Optimization of Indexing in Vector Databases	5
2.5 Applications and Adaptations for Low-Resource Languages	6
2.6 Comparision to This Study	7
3 MATERIALS AND METHODS	9
3.1 Data and Dataset Configuration	9
3.1.1 Dataset Selection	9
3.1.2 Architecture Design for Optimized RAG Model with FAISS	11
4 RESULTS	20
4.1 Evaluation of Model Performance	20
4.2 Comparison of Indexing Methods: IVF vs. HNSW	25

4.3	Model Performance Visualization	26
4.4	Theoretical and Empirical Impact of Dynamic Chunking and Reranking on RAG Metrics	30
4.4.1	Theoretical Impact	30
4.4.2	Empirical Analysis	30
5	DISCUSSION	34
5.1	Implications of Advanced Indexing	34
5.2	Enhancing Retrieval Coherence with Dynamic Chunking	34
5.3	The Role of Cross-Encoder Re-Ranking	35
5.4	Balancing Trade-Offs in Retrieval-Augmented Systems	35
5.5	Contributions to the Turkish Language and Broader NLP Context . .	36
5.6	Future Directions for RAG Optimization	36
6	CONCLUSION	37
	REFERENCES	39
	PUBLICATIONS FROM THE THESIS	42

LIST OF SYMBOLS

AC	Answer Correctness
AR	Answer Relevancy
b	Bias
CP	Context Precision
CR	Context Recall
F	Faithfulness
h_{CLS}	Hidden state at the CLS
OS	Overall Score
$S_{\text{relevance}}$	Score of relevance
σ	Sigmoid
$\sum_{i=0}^n$	Summation notation
w_i	Weights
W	Weight Matrix

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
CLS	Classification
FAISS	Facebook AI Similarity Search
HNSW	Hierarchical Navigable Small World
IVF	Inverted File Indexing
IVFFlat	Inverted File Flat
IVFPQ	Inverted File Index with Product Quantization
LLM	Large Language Model
LLMs	Large Language Models
MoG	Mix-of-Granularity
NLP	Natural Language Processing
RAG	Retrieval-Augmented Generation
RAGAS	Retrieval-Augmented Generation Assessment

LIST OF FIGURES

Figure 3.1	System Architecture of the Enhanced RAG Model	16
Figure 4.1	Ytu-ce-cosmos/Turkish-Llama-8b-DPO-v0.1	27
Figure 4.2	Ytu-ce-cosmos/Turkish-Llama-8b-Instruct-v0.1	27
Figure 4.3	Malhajar/Mistral-7B-Instruct-v0.2-turkish	27
Figure 4.4	NousResearch/Meta-Llama-3-8B	28
Figure 4.5	Eurdem/Defne-llama3.1-8B	28
Figure 4.6	Google/gemma-2-9b-it	28

LIST OF TABLES

Table 3.1	Questions that can be answered depending on context	10
Table 4.1	Ytu-ce-cosmos/Turkish-Llama-8b-DPO-v0.1 model results . . .	21
Table 4.2	Ytu-ce-cosmos/Turkish-Llama-8b-Instruct-v0.1 model results . .	22
Table 4.3	Malhajar/Mistral-7B-Instruct-v0.2-turkish model results	23
Table 4.4	NousResearch/Meta-Llama-3-8B model results	23
Table 4.5	Eurdem/Defne-llama3.1-8B model results	24
Table 4.6	Google/gemma-2-9b-it model results	24
Table 4.7	Quantitative analysis of metric improvements of ytu-ce-cosmos/Turkish-Llama-8b-DPO-v0.1 model	31
Table 4.8	Quantitative analysis of metric improvements of ytu-ce-cosmos/Turkish-Llama-8b-Instruct-v0.1 model	32
Table 4.9	Quantitative analysis of metric improvements of malhajar/Mistral-7B-Instruct-v0.2 model	32
Table 4.10	Quantitative analysis of metric improvements of NousResearch/Meta-Llama-3-8B model	32
Table 4.11	Quantitative analysis of metric improvements of Eurdem/Defne-llama-3.1-8B model	32
Table 4.12	Quantitative analysis of metric improvements of google/gemma-2-9b-it model	33

ABSTRACT

Enhancing Retrieval-Augmented Generation Accuracy With Dynamic Chunking And Optimized Vector Search

Derya TANYILDIZ

Department of Computer Engineering
Master of Science Thesis

Supervisor: Prof. Dr. Mehmet Fatih AMASYALI

Co-supervisor: Assoc. Prof. Dr. Serkan AYVAZ

Retrieval-Augmented Generation (RAG) architectures rely on the seamless integration of efficient retrieval and ranking mechanisms to enhance the accuracy, relevance, and contextual appropriateness of generated responses. In this study, we propose a comprehensive approach to improving the response performance of RAG systems by employing advanced techniques that address critical challenges in information retrieval and generation coherence. Our methodology introduces dynamic chunking to adaptively segment documents, ensuring improved contextual coherence and reducing information fragmentation. For embedding generation, we leverage Sentence-Transformers (all-mpnet-base-v2), which produce high-dimensional, semantically rich vector representations that enhance retrieval quality. Additionally, a cross-encoder-based re-ranking mechanism refines the initial retrieval results, ensuring that only the most contextually relevant documents are passed to the generation module. Performance evaluation is conducted using RAGAS metrics, focusing on faithfulness, relevancy, correctness, and context precision, providing a holistic understanding of system capabilities. Our empirical analyses reveal the profound impact of index choice on RAG performance, emphasizing the trade-offs between retrieval efficiency and accuracy. By integrating the HNSW (Hierarchical Navigable Small World) index with re-ranking, our proposed architecture achieves a robust balance, enhancing response fidelity while maintaining computational efficiency.

The results demonstrate that advanced indexing strategies, when combined with intelligent chunking and re-ranking, significantly improve the system's ability to bridge the gap between large-scale language models and domain-specific knowledge retrieval. These insights establish a solid foundation for future research in optimizing RAG systems, especially in applications requiring high-context preservation, domain specificity, and precision. The study underscores the potential of dynamic and adaptive retrieval mechanisms as a critical enabler for next-generation conversational AI systems, paving the way for broader adoption in complex and high-stakes environments.

Keywords: Retrieval-Augmented Generation (RAG), Cross Encoders, Vector Database, Dynamic Chunking, Semantic Search



ÖZET

DİNAMİK PARÇALAMA VE OPTİMİZE EDİLMİŞ VEKTÖR ARAMASI İLE BİLGİ GETİRME DESTEKLİ ÜRETİM DOĞRULUĞUNU ARTIRMA

Derya TANYILDIZ

Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Prof. Dr. Mehmet Fatih AMASYALI

Eş-Danışman: Doç. Dr. Serkan AYVAZ

Erişim Destekli Üretim mimarileri, doğru, alakalı ve bağlama uygun yanıtlar üretmek için etkili bir geri getirme ve sıralama mekanizmasının sorunsuz entegrasyonuna dayanır. Bu çalışmada, bilgi geri getirme ve üretim tutarlılığına yönelik kritik zorlukları ele alan gelişmiş teknikleri kullanarak Erişim Destekli Üretim sistemlerinin yanıt performansını artırmak için kapsamlı bir yaklaşım öneriyoruz. Yöntemimiz, belgeleri uyarlanabilir şekilde segmente ederek bağlamsal tutarlılığı artıran ve bilgi parçalanmasını azaltan dinamik parçalama (dynamic chunking) yöntemini içermektedir. Anlamsal temsil oluşturma sürecinde, yüksek boyutlu ve anlamsal açıdan zengin vektör temsilleri üreten Sentence-Transformers (all-mpnet-base-v2) modeli kullanılmaktadır; bu da geri getirme kalitesini artırmaktadır. Ayrıca, bir çapraz kodlayıcı (cross-encoder) tabanlı yeniden sıralama mekanizması, ilk geri getirme sonuçlarını rafine ederek yalnızca bağlamsal olarak en alakalı belgelerin üretim modülüne iletilmesini sağlamaktadır. Performans değerlendirmesi, sistem yeteneklerini kapsamlı bir şekilde anlamak için bağlılık (faithfulness), alaka düzeyi (relevancy), doğruluk (correctness) ve bağlam doğruluğu (context precision) gibi metriklere odaklanan RAGAS değerlendirme ölçütleri kullanılarak gerçekleştirilmiştir. Yaptığımız ampirik analizler, Erişim Destekli Üretim performansı üzerinde indeks seçiminin derin etkisini ortaya koymakta ve geri getirme verimliliği ile doğruluk arasındaki dengeyi vurgulamaktadır. Hiyerarşik Gezilebilir Küçük Dünya (HNSW -

Hierarchical Navigable Small World) indeksi ile yeniden sıralamanın entegrasyonu sayesinde, önerilen mimarimiz, yanıt bağılılığını artırırken hesaplama verimliliğini koruyan sağlam bir dengeye ulaşmaktadır. Sonuçlar, ileri düzey indeksleme stratejilerinin, akıllı parçalama ve yeniden sıralama ile birleştirildiğinde, büyük ölçekli dil modelleri ile alan bilgisine özgü geri getirme arasındaki boşluğu kapatma yeteneğini önemli ölçüde artırdığını göstermektedir. Bu bulgular, özellikle bağlam koruma, alan özgünlüğü ve hassasiyet gerektiren uygulamalarda Erişim Destekli Üretim sistemlerini optimize etme konusunda sağlam bir temel oluşturmaktadır. Çalışma, dinamik ve uyarlanabilir geri getirme mekanizmalarının, bir sonraki nesil konuşma tabanlı yapay zeka sistemleri için kritik bir etkinleştirici olduğunu vurgulamakta ve karmaşık ve yüksek riskli ortamlarda daha geniş bir benimseme için zemin hazırlamaktadır.

Anahtar Kelimeler: Erişim Destekli Üretim, Çapraz Kodlayıcılar, Vektör Veritabanı, Dinamik Parçalama, Anlamsal Arama

1

INTRODUCTION

1.1 Advancements in Large Language Models

In recent years, the proliferation of large-scale pre-trained language models (LLMs) has significantly advanced the field of natural language processing (NLP). These models, with their capacity to generate coherent and contextually relevant text, have found applications across diverse domains [1].

Despite their transformative potential, LLMs rely on static, fixed knowledge representations, which limit their ability to retrieve and utilize the most current and domain-specific information. To address this challenge, Retrieval-Augmented Generation (RAG) architectures have emerged as a promising solution, integrating external retrieval mechanisms with LLMs to provide accurate and contextually relevant responses [2].

1.2 Challenges in Traditional RAG Systems

The standard RAG pipeline comprises two main stages: retrieval and generation. During the retrieval stage, relevant pieces of information are identified from a vast corpus, while the generation stage synthesizes this information into coherent responses [3].

However, traditional RAG implementations face critical limitations: **Static Chunking Strategies:** Textual data is split into uniform segments, which simplifies retrieval but often results in a loss of contextual integrity, especially for queries requiring nuanced understanding [4].

Retriever-Reader Imbalance: Retrievers are computationally intensive yet lack contextual awareness, further degrading system performance [5]. These challenges highlight the importance of advancing retrieval strategies to optimize both accuracy and efficiency.

1.3 Proposed Optimized RAG Architecture

This paper addresses these limitations by proposing an optimized RAG architecture that integrates the following key advancements: **Dynamic Chunking:** Adaptively segments textual data based on context, preserving information coherence and reducing fragmentation. **Advanced Embedding Techniques:** Leverages Sentence-Transformers (all-mpnet-base-v2) to generate high-quality embeddings for improved retrieval accuracy [6]. **State-of-the-Art Indexing Methods:** Compares FAISS indexing techniques, with empirical evidence highlighting the superior performance of the HNSW index [7]. **Cross-Encoder Re-Ranking:** Incorporates a re-ranking mechanism to ensure the retrieval of the most contextually relevant information [8]. These advancements collectively improve response quality and relevance while ensuring computational efficiency.

1.4 Contributions to RAG and NLP Research

This research makes several significant contributions to the fields of RAG and NLP: **Dynamic Chunking Strategy:** Resolves the limitations of static chunking by preserving contextual coherence and enhancing retrieval relevance. **Embedding Techniques:** Demonstrates the effectiveness of Sentence-Transformers for producing semantically rich vector representations. **Indexing Analysis:** Provides a comparative evaluation of FAISS indexing methods, with evidence supporting the advantages of the HNSW index for efficiency and accuracy. **Re-Ranking Mechanisms:** Enhances response quality through cross-encoder-based re-ranking.

1.5 Focus on Turkish Language and Real-World Applications

This study provides a comparative analysis of RAG optimization techniques, specifically in the context of the Turkish language. By evaluating key components such as embedding, chunking, indexing, and re-ranking, the research highlights the applicability and performance of these methodologies in addressing real-world information retrieval challenges for Turkish data [9]. The findings offer a practical framework for future studies, demonstrating how advanced retrieval-augmented generation systems can be effectively adapted for domain-specific applications requiring high contextual preservation and precision.

2 RELATED WORKS

The development and optimization of Retrieval-Augmented Generation (RAG) systems have gained considerable attention in recent years, primarily due to their ability to enhance the performance of large-scale pre-trained language models (LLMs) by integrating external retrieval mechanisms. This section provides an overview of the key studies related to RAG architectures, focusing on their components, limitations, and optimization strategies.

2.1 Foundation of Retrieval-Augmented-Generation

Early work on retrieval-augmented systems emphasized combining retrieval techniques with generative models to improve response relevance. Notable studies, such as Lewis et al. [10], introduced the foundational RAG pipeline, which integrates a retriever to fetch relevant information and a generator to synthesize coherent responses. This architecture was instrumental in addressing the limitations of static knowledge representations in LLMs, paving the way for more context-aware NLP applications.

Lewis et al. [10] proposed the first end-to-end RAG framework, which demonstrated how retrieval mechanisms could augment generative models for knowledge-intensive tasks. Their approach utilized dense retrievers that significantly improved retrieval accuracy compared to traditional sparse methods. In a complementary manner, Karpukhin et al. [11] introduced Dense Passage Retrieval (DPR), a method that further enhanced the retriever's efficiency by leveraging bi-encoder architectures for improved passage relevance.

Building on these advancements, Izacard and Grave [12] demonstrated how integrating passage retrieval with generative models could optimize open-domain question answering. Their work underscored the importance of combining retrieval and generation in a seamless pipeline, emphasizing the enhanced contextual relevance and coherence achieved through such hybrid systems.

These foundational studies laid the groundwork for RAG systems by addressing core challenges such as retrieving relevant information and generating contextually appropriate responses. They continue to influence modern retrieval-augmented frameworks, including dynamic chunking strategies and long-context LLMs, which aim to further refine retrieval accuracy and response quality.

2.2 Advances in Retrieval and Ranking Techniques

The retrieval stage is critical to the success of RAG systems. Traditional approaches, including BM25 [13] and dense retrieval models like DPR [11], laid the groundwork for efficient information retrieval. BM25, as introduced by Robertson and Zaragoza [13], established a probabilistic relevance framework that has been a cornerstone in information retrieval for decades. However, its reliance on sparse term matching often limits its ability to capture deeper semantic relationships.

More recent advancements have focused on embedding techniques, with Sentence-Transformers [14] gaining prominence for their ability to capture semantic similarities. Reimers and Gurevych [6] demonstrated how Sentence-BERT leverages Siamese BERT networks to generate high-quality sentence embeddings, significantly improving retrieval performance across various NLP tasks.

Additionally, the introduction of re-ranking mechanisms, such as cross-encoders, has significantly improved the quality of retrieved documents by refining ranking scores based on query-context alignment. Déjean et al. [15] conducted a thorough analysis comparing cross-encoders and LLM-based ranking methods with SPLADE. Their findings reveal that cross-encoders excel in scenarios requiring precise query-document alignment, offering superior ranking performance, particularly for complex or nuanced queries. This study underscores the critical role of advanced re-ranking strategies in enhancing the effectiveness of RAG systems.

These advances collectively highlight the evolution of retrieval and ranking techniques, moving beyond traditional sparse methods to more sophisticated approaches that integrate dense embeddings and dynamic re-ranking. As these methods continue to mature, they offer significant potential for improving the precision and relevance of information retrieval in RAG pipelines.

2.3 Challenges in Static Chunking Strategies

Static chunking, commonly employed in Retrieval-Augmented Generation (RAG) systems, often struggles with balancing contextual integrity and information granularity. Uniform segmentation can lead to challenges such as information redundancy or loss of critical contextual details, limiting retrieval accuracy and downstream task performance. Recent advancements, such as the Mix-of-Granularity (MoG) approach proposed by Zhong et al. [16], aim to overcome these limitations through dynamic chunking strategies. By adaptively segmenting text into variable-sized chunks based on semantic boundaries, MoG ensures that both broad contextual information and fine-grained details are preserved. This adaptability significantly enhances retrieval precision and model integration, outperforming static methods in terms of response accuracy and contextual relevance. Experimental results further validate MoG’s superiority, particularly in handling complex, context-sensitive queries.

Similarly, Jiang et al. [2] in their work LongRAG: Enhancing Retrieval-Augmented Generation with Long-context LLMs highlight the inherent limitations of static chunking strategies and advocate for dynamic chunking approaches. Their findings emphasize that static chunking’s fixed-length segmentation often disrupts semantic coherence and leads to suboptimal retrieval performance in long-context models. Dynamic chunking, as explored in their study, dynamically adjusts segment boundaries to align with contextual relevance, thereby preserving semantic integrity and enhancing model efficiency. The authors further demonstrate that dynamic chunking, when integrated with long-context language models, yields significant improvements in retrieval accuracy and response generation quality compared to static methods. These advancements underscore the growing necessity of adaptive chunking strategies to address the evolving demands of retrieval-augmented systems.

2.4 Optimization of Indexing in Vector Databases

Efficient indexing methods are crucial for scaling retrieval-augmented generation (RAG) systems to large corpora. Indexing techniques play a pivotal role in determining the speed and accuracy of nearest neighbor searches, which are core to RAG models. A variety of indexing strategies have been explored to balance retrieval performance, memory efficiency, and query latency.

Studies comparing different indexing methods, such as Flat, IVFPQ, and HNSW [17], have extensively analyzed the trade-offs between retrieval speed and accuracy.

Among these, the HNSW (Hierarchical Navigable Small World) index has emerged as a preferred choice for applications requiring high retrieval performance at scale. This index excels in providing efficient search capabilities while maintaining a favorable balance between speed and accuracy, as highlighted by empirical evaluations across a wide range of datasets.

Recent developments in vector indexing have introduced new paradigms for optimizing performance. For example, Curator: Efficient Indexing for Multi-Tenant Vector Databases [18] presents a novel in-memory vector index designed specifically for multi-tenant environments. Curator leverages tenant-specific clustering trees, which are compactly encoded within a shared structure. This dynamic approach adapts to the diverse vector distributions of multiple tenants, ensuring high query performance and efficient insertion and deletion operations. Furthermore, Curator maintains a low memory overhead, addressing the challenges of supporting multiple tenants in vector databases, which is particularly relevant for large-scale, multi-tenant applications.

Further research by Gupta et al. (2024) on BLISS: A Billion Scale Index using Iterative Re-partitioning [19] offers innovative methodologies for scaling index structures to handle billion-scale datasets. The authors propose an iterative re-partitioning strategy that optimizes both the indexing process and search speed without significantly increasing memory overhead. This approach is particularly valuable for large-scale systems, where managing vast amounts of data efficiently is paramount.

Finally, Rahman et al. (2024) [20] investigate the optimization of domain-specific image retrieval, benchmarking FAISS and Annoy with fine-tuned features. The study explores how fine-tuning the indexing parameters can enhance retrieval precision and efficiency for specific domains, highlighting the importance of custom-tailored indexing strategies in real-world applications.

Together, these studies underscore the importance of selecting the right indexing technique based on the dataset characteristics and the specific requirements of the RAG system, providing valuable insights into the future direction of vector database optimization.

2.5 Applications and Adaptations for Low-Resource Languages

While most RAG research has focused on English-language datasets, recent efforts have explored its applicability to low-resource languages, including Turkish.

For instance, [21] investigated embedding and retrieval methods tailored to morphologically rich languages, emphasizing the importance of language-specific adaptations for effective retrieval.

Recent studies have expanded the application of RAG techniques to low-resource languages, addressing challenges such as limited data availability and linguistic diversity. Ardic et al. (2024) [22] explored the use of RAG for information extraction from sustainability reports in Turkish, showcasing how domain-specific adaptations can improve retrieval performance in low-resource languages. Their study highlights the integration of advanced techniques such as BM25 and BERTurk to enhance the effectiveness of information retrieval, while also addressing the unique characteristics of Turkish, such as its agglutinative morphology.

Further contributions to the field include Shahnaz and Rony (2023) [21], who developed the TraSe architecture to enhance RAG performance for Bengali, another low-resource language. The study proposes a specialized retrieval-augmented generation framework that adapts to the syntactic and semantic properties of Bengali, thereby improving the efficiency of NLP tasks in this language. Their work emphasizes the importance of tailoring RAG models to the linguistic nuances of low-resource languages to maximize their performance.

Additionally, Jin et al. (2024) [18] introduced Curator, an indexing solution designed for multi-tenant vector databases, which holds promise for improving RAG systems in low-resource language scenarios. Curator’s efficient indexing and retrieval mechanisms, particularly its adaptation to tenant-specific clustering trees, offer a scalable solution for managing diverse language models. By optimizing indexing for multi-tenant environments, Curator can enhance the retrieval capabilities of RAG systems when applied to low-resource languages with limited computational resources.

Together, these studies underscore the growing importance of adapting RAG models to the unique challenges posed by low-resource languages. Tailoring retrieval and generation techniques to specific linguistic features, domain-specific needs, and computational constraints is crucial for advancing the state of NLP in these underrepresented languages.

2.6 Comparison to This Study

This research extends and refines the methodologies employed in prior studies by addressing critical limitations associated with static chunking strategies and

imbalances between the retriever and reader components. While existing literature predominantly focuses on generic, multilingual, or high-resource datasets, our study is uniquely tailored to the intricacies of Turkish datasets, which present distinct linguistic and contextual challenges.

1

In addition, this study leverages Hierarchical Navigable Small World (HNSW) indexing, which provides significant improvements in computational efficiency and scalability for vector search operations. Combined with a cross-encoder-based re-ranking mechanism, our framework achieves superior performance in terms of contextual relevance and retrieval precision. By aligning these components within a cohesive system, this research not only surpasses existing methodologies but also sets a new benchmark for retrieval and question-answering tasks, particularly in underexplored languages like Turkish.

3

MATERIALS AND METHODS

3.1 Data and Dataset Configuration

For this study, we utilized the Metin/WikiRAG-TR dataset, a curated corpus designed to evaluate the effectiveness of RAG architectures in retrieving and generating contextually accurate responses [23]. To ensure the evaluation focused on the model’s ability to retrieve and leverage relevant context, a subset of 15 targeted questions was carefully selected from the dataset. These questions were specifically designed such that generating a correct answer would be infeasible without access to the corresponding context. This configuration ensured that the responses relied heavily on the retrieved context, minimizing the possibility of context-independent, generic answers from the model.

3.1.1 Dataset Selection

When examining the literature, it becomes evident that RAG architectures have demonstrated considerable success with English datasets, often achieving significant improvements with minimal optimization. However, such advancements remain less explored in low-resource languages. In this study, we employed a Turkish dataset, a language that is morphologically complex and structurally distinct from English, with the aim of enhancing the performance of RAG architectures for Turkish. The experimental setup was meticulously designed to evaluate the core strengths of the RAG framework using a set of carefully curated questions that necessitated precise semantic understanding and alignment with the given context (Table 3.1). The evaluation process focused on three key dimensions: **Context Recall**, which measures the retrieval module’s ability to identify and extract the most relevant chunks; **Answer Faithfulness**, which ensures that the generated responses are firmly grounded in the provided context; and **Model Accuracy**, which assesses the overall correctness of the answers relative to the original dataset. This comprehensive methodology provided a robust foundation for assessing and improving the integration of retrieval and generative capabilities

in the RAG architecture, particularly in the context of a low-resource language like Turkish.

Table 3.1 Questions that can be answered depending on context

#	Questions
1	Bu modelin belirsiz bir pazarı tanımlamak için kullanılmasının sebebi nedir? <i>What is the reason for using this model to define an uncertain market?</i>
2	Gimlé'nin yerleşim yeri hakkında farklı kaynaklarda çelişkiler var mı? Eğer varsa, bu çelişkiler nelerdir? <i>Are there contradictions about the location of Gimlé in different sources? If so, what are these contradictions?</i>
3	Egzotik hayvan tanımının zamanla değişebileceğini söyleyen yazı, bunun nedenlerini açıklıyor mu? <i>Does the article that suggests the definition of exotic animals may change over time explain the reasons for this?</i>
4	Yazıda bahsedilen kask türlerinden bazıları örneklendirilmiş. Peki, bu çeşitli kask türleri hangi farklı ihtiyaçlara göre tasarlanmıştır? <i>Some types of helmets mentioned in the article have been exemplified. So, for what different needs are these various types of helmets designed?</i>
5	Yazıda tam kare sayılar ile karesel sayılar arasındaki fark açıklanmış. Peki, bu iki kavram arasında bir ilişki var mıdır? <i>The difference between perfect square numbers and square numbers has been explained in the article. Is there a relationship between these two concepts?</i>
6	Fonksiyonlar hangi alanlarda kullanılır? Metinde birkaç örnek verilmiş, bunlardan başka hangi alanlarda kullanılabilir? <i>In which fields are functions used? The article provides a few examples, but in what other fields can they be applied?</i>
7	Yazıda abaküs ve hesap makinesi gibi araçlardan bahsediliyor. Peki, temel aritmetik işlemleri tarih boyunca insanlar nasıl yapıyordu? <i>The article mentions tools like the abacus and calculator. So, how did people perform basic arithmetic operations throughout history?</i>
8	U-20 denizaltısının Türkiye'ye ulaştırılmasının ardından Türk yetkililer tarafından neden reddedildiği belirtilmiştir. Peki, bu reddetmenin sebepleri neler olabilirdi? <i>After the U-20 submarine was delivered to Turkey, the reasons for its rejection by Turkish officials were mentioned. What could these reasons have been?</i>
9	Yazıda telemetri için kullanılan kablosuz sistemlerden örnekler verilmiş. Peki ya bu sistemlerin avantajları nelerdir? <i>Examples of wireless systems used for telemetry are provided in the article. What are the advantages of these systems?</i>
10	Grendel destanda genellikle nasıl tasvir ediliyor? <i>How is Grendel generally portrayed in the epic?</i>

3.1.2 Architecture Design for Optimized RAG Model with FAISS

The proposed architecture aims to enhance the response accuracy and retrieval efficiency of Retrieval-Augmented Generation (RAG) systems by incorporating advanced embedding techniques, dynamic chunking, and optimized vector database indexing. The integration of these components ensures that the system delivers high-quality and contextually relevant responses [24]. Figure 3.1 illustrates an overview of the proposed system architecture with the enhanced RAG model.

3.1.2.1 Document Management Pipeline

The document management pipeline comprises several interdependent stages designed to optimize the retrieval and generation of information. The process begins with Data Ingestion and Chunking, where input documents are processed using a document loader to prepare them for subsequent operations. In contrast to conventional static chunking methods, the proposed approach employs dynamic chunking, which segments the text based on semantic coherence. The data chunked with the Turkish data set is as follows.

Static Chunking splits text into fixed lengths (e.g. 100 words) with no overlap. This method cuts text content regardless of its context.

Static Chunking (100 words) example

Paragraph 1:

Chunk 1: HH 24-26, Herbig-Haro cisimleri olan HH 24, HH 25 ve HH 26'yı içeren bir moleküler bulut ve yıldız oluşum bölgesidir. Bu bölge, gökyüzünde bilinen en yüksek yoğunlukta astrofiziksel jetlere sahiptir.

HH 24-26 is a molecular cloud and star-forming region containing the Herbig-Haro objects HH 24, HH 25, and HH 26. This region has the highest intensity astrophysical jets known in the sky.

Chunk 2: Moleküler bulut, Avcı takımyıldızındaki Orion B moleküler bulutunun bir parçası olan yaklaşık 1400 ışık yılı uzaklıktaki L1630 karanlık bulutsusunda yer alır. Bölgede, iki adet 0 sınıfı ve bir adet I sınıfı olmak üzere birden fazla önyıldız bulunmaktadır.

The molecular cloud is located in the dark nebula L1630, about 1,400 light-years away, which is part of the Orion B molecular cloud in the constellation Orion. The region contains multiple protostars, two class 0 and one class I.

Chunk 3: Bu bölgedeki Herbig-Haro cisimleri bu üç önyıldız tarafından hareketlendirilmektedir.

The Herbig-Haro objects in this region are animated by these three protostars.

Paragraph 2:

Chunk 1: İskandinav mitolojisinde, Gimlé, ölenlerin gittiği ve Ragnarök'ten hayatta kalan değerli kişilerin yaşayacaklarının önceden söylendiği bir yerdir. Nesir Edda'da ve Manzum Edda şiiri "Völuspá" da bahsedilir.

In Norse mythology, Gimlé is a place where the dead go and where it is foretold that the worthy survivors of Ragnarök will live. It is mentioned in the Prose Edda and the Verse Edda poem "Völuspá".

Chunk 2: Asgard'ın güneşten daha güzel en güzel yeri olarak tanımlanır. Tanrıların krallığı Asgard'da Gimlé, doğru insanların öldüklerinde gittikleri altın çatılı bir binadır.

It is described as the most beautiful place in Asgard, more beautiful than the sun. In Asgard, the kingdom of the gods, Gimlé is a golden-roofed building where righteous people go when they die.

Chunk 3: Snorri Sturluson, Gimlé'yi ışık elflerinin yaşadığı üçüncü cennet olarak tanımlar.

Snorri Sturluson describes Gimlé as the third heaven, where the light elves lived.

Dynamic Chunking splits the text by a specified maximum length (e.g. 200 words) and overlap size (e.g. 30 words). This method aims to increase the preservation of context.

Dynamic Chunking (200 words, 30 words) example

Paragraph 1:

Chunk 1: HH 24-26, Herbig-Haro cisimleri olan HH 24, HH 25 ve HH 26'yı içeren bir moleküler bulut ve yıldız oluşum bölgesidir. Bu bölge, gökyüzünde bilinen en yüksek yoğunlukta astrofiziksel jetlere sahiptir. Moleküler bulut, Avcı takımyıldızındaki Orion B moleküler bulutunun bir parçası olan yaklaşık 1400 ışık yılı uzaklıktaki L1630 karanlık bulutsusunda yer alır.

HH 24-26 is a molecular cloud and star-forming region containing the Herbig-Haro objects HH 24, HH 25, and HH 26. This region has the highest-intensity astrophysical jets known in the sky. The molecular cloud is located approximately

1,400 light-years away in the dark nebula L1630, part of the Orion B molecular cloud in the constellation Orion.

Chunk 2: Moleküler bulut, Avcı takımyıldızındaki Orion B moleküler bulutunun bir parçası olan yaklaşık 1400 ışık yılı uzaklıktaki L1630 karanlık bulutsusunda yer alır. Bölgede, iki adet 0 sınıfı ve bir adet I sınıfı olmak üzere birden fazla önyıldız bulunmaktadır. Bu bölgedeki Herbig-Haro cisimleri bu üç önyıldız tarafından hareketlendirilmektedir.

The molecular cloud is located in the dark nebula L1630, about 1,400 light-years away, which is part of the Orion B molecular cloud in the constellation Orion. The region contains multiple protostars, two class 0 and one class I. The Herbig-Haro objects in this region are animated by these three protostars.

Paragraph 2:

Chunk 1: İskandinav mitolojisinde, Gimlé, ölenlerin gittiği ve Ragnarök'ten hayatta kalan değerli kişilerin yaşayacaklarının önceden söylendiği bir yerdir. Nesir Edda'da ve Manzum Edda şiiri "Völuspá" da bahsedilir ve Asgard'ın güneşten daha güzel en güzel yeri olarak tanımlanır. Tanrıların krallığı Asgard'da Gimlé, doğru insanların öldüklerinde gittikleri altın çatılı bir binadır.

In Norse mythology, Gimlé is a place where the dead go and where it is foretold that the worthy survivors of Ragnarök will live. It is mentioned in the Prose Edda and the Verse Edda poem "Völuspá" and is described as the most beautiful place in Asgard, more beautiful than the sun. In Asgard, the kingdom of the gods, Gimlé is a golden-roofed building where the righteous go when they die.

Chunk 2: Asgard'ın güneşten daha güzel en güzel yeri olarak tanımlanır. Tanrıların krallığı Asgard'da Gimlé, doğru insanların öldüklerinde gittikleri altın çatılı bir binadır. Snorri Sturluson, Gimlé'yi ışık elflerinin yaşadığı üçüncü cennet olarak tanımlar.

It is described as the most beautiful place in Asgard, more beautiful than the sun. In Asgard, the kingdom of the gods, Gimlé is a golden-roofed building where righteous people go when they die. Snorri Sturluson describes Gimlé as the third heaven, where the light elves live.

This approach preserves contextual relationships within the text, ensuring that relevant information remains intact within each chunk, thereby avoiding fragmentation. By maintaining the semantic integrity of the content, dynamic chunking enhances retrieval accuracy significantly [25].

Subsequently, the dynamically chunked text undergoes embedding using the Sentence-Transformers model (all-mpnet-base-v2) [14], a cutting-edge model optimized for generating dense vector representations. These embeddings effectively capture semantic similarities, enabling the system to retrieve conceptually relevant documents rather than relying solely on superficial keyword matching.

The embeddings are then stored and managed within a FAISS (Facebook AI Similarity Search) vector database, which supports efficient and scalable similarity searches. Two indexing strategies, IndexIVFFlat and IndexHNSWFlat, are implemented and evaluated to determine their effectiveness. The IndexIVFFlat method partitions the vector space into clusters, offering a trade-off between computational efficiency and retrieval precision. In contrast, the IndexHNSWFlat method constructs a hierarchical navigable small-world graph, known for its superior performance in tasks requiring precise semantic retrieval. Performance evaluations demonstrate that HNSW indexing consistently outperforms IVFFlat, excelling in both retrieval speed and accuracy [26].

This pipeline, integrating dynamic chunking, advanced embedding techniques, and efficient vector database management, establishes a robust framework for document management, particularly in applications requiring high retrieval accuracy and semantic fidelity.

3.1.2.2 LLM Prompting and Answer Generation Pipeline

The response generation pipeline leverages document embeddings to deliver accurate and contextually aligned answers to user queries. The process begins with Query Processing and Retrieval, where user queries are transformed into embeddings using the same Sentence-Transformers model as employed during document embedding. This ensures compatibility between query embeddings and the stored document embeddings. A similarity search is then conducted in the FAISS vector database, retrieving the top_k1 most contextually relevant chunks. These chunks serve as the initial candidate set of data segments that align with the user query.

Once the chunks are retrieved, they undergo a Re-Ranking stage, in which a cross-encoder is employed to refine their relevance. The cross-encoder [27], typically a transformer-based model, evaluates each chunk's semantic and contextual alignment with the user query. This evaluation results in a relevance score for each chunk, which is computed by concatenating the query and chunk

texts and passing them through the model. The model produces a scalar relevance score $S_{\text{relevance}}$, indicating how well each chunk aligns with the query. The score is calculated as:

$$S_{\text{relevance}} = (\sigma(W \cdot h_{\text{CLS}} + b))$$

where h_{CLS} represents the hidden state corresponding to the *CLS* token after the query-chunk pair is processed, and W and b are learnable parameters. The sigmoid function ensures that the relevance score is between 0 and 1, with higher values indicating greater relevance.

The top_k2 chunks with the highest relevance scores are then selected for further processing. This re-ranking step ensures that only the most pertinent information is retained for the final stage, thereby enhancing the overall accuracy and reliability of the system. By incorporating both the initial similarity search and the fine-tuned relevance re-ranking, the pipeline ensures that the generated responses are not only contextually appropriate but also semantically aligned with the user query, improving the system’s precision and effectiveness.

In the final stage, Response Generation, the re-ranked chunks are utilized as contextual input to a fine-tuned Large Language Model (LLM), such as Turkish-Llama-8b-DPO-v0.1 [28] or Turkish-Llama-8b-Instruct-v0.1 [29]. The LLM synthesizes the contextual information with the user query to generate coherent and accurate responses. This synthesis process prioritizes faithfulness, relevance, and correctness, aligning with the predefined evaluation metrics discussed in [30].

By integrating query processing, dynamic retrieval, cross-encoder re-ranking, and advanced LLM capabilities, this pipeline ensures a robust and efficient framework for delivering high-quality responses to user queries.

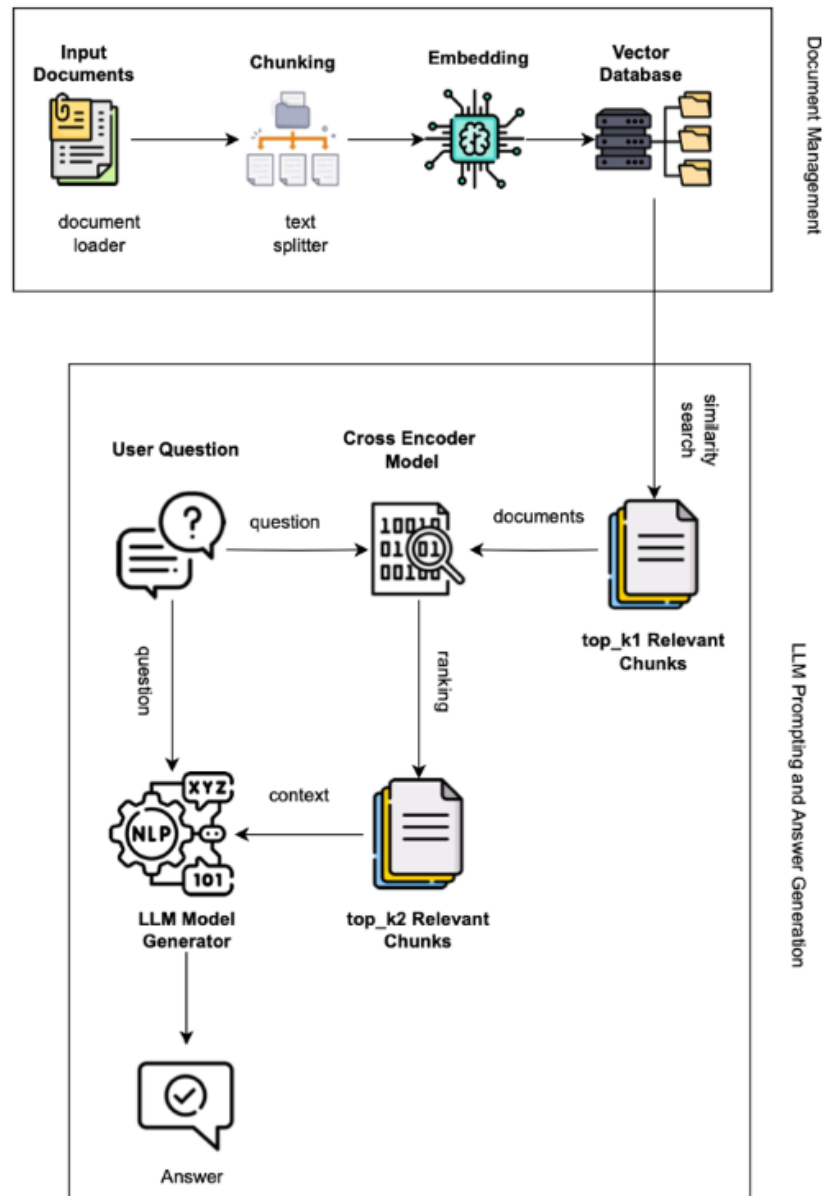


Figure 3.1 System Architecture of the Enhanced RAG Model

3.1.2.3 Framework for Evaluating RAG Systems: The RAGAS Metrics

The Retrieval-Augmented Generation Answer Scoring (RAGAS) framework provides a structured methodology for evaluating the performance of RAG-based systems. It incorporates several metrics that assess the semantic accuracy, contextual consistency, and retrieval quality of the generated answers. Each metric focuses on a distinct aspect of system performance, ensuring a comprehensive evaluation.

Faithfulness (F) measures the extent to which the generated answers align with the retrieved context, ensuring factual consistency. A higher faithfulness score indicates that the generated response avoids hallucination or inaccuracies. Mathematically, faithfulness is assessed by comparing the generated answer with the retrieved context to verify whether the information is correctly conveyed. Higher faithfulness scores indicate a lower degree of discrepancy between the answer and the context, ensuring that the response is consistent with the retrieved data. This process often involves the use of similarity measures, such as cosine similarity or other text comparison metrics, to quantitatively assess the alignment between the response and the context. So, a response's faithfulness is determined by its factual accuracy and consistency with the provided context, with a higher score reflecting a more reliable and contextually accurate answer:

$$F = \frac{\text{Number of Faithful Answers}}{\text{Total Number of Answers}}$$

Answer Relevancy (AR) quantifies the degree of semantic alignment between the generated response and the original query. It assesses whether the generated answer is directly relevant to the question posed, ensuring that the response appropriately addresses the user's intent and provides useful information. A high AR score indicates that the response is contextually appropriate and effectively answers the query. Mathematically, AR is computed by measuring the semantic congruence between the response and the question, often using techniques such as cosine similarity or other natural language processing methods to determine the degree of relevance. This metric is essential in evaluating the quality of response generation models, particularly in tasks such as question answering, where providing accurate and contextually appropriate answers is crucial. So, Answer Relevancy evaluates how well the generated response aligns with the question, with a higher score indicating a more relevant and effective answer that directly addresses the query.

$$AR = \frac{\text{Number of Relevant Answers}}{\text{Total Number of Answers}}$$

Answer Correctness (AC) measures the factual accuracy of the generated response, evaluating the model's ability to produce answers that are both truthful and relevant to the query. This metric is crucial in determining whether the generated output reflects the correct information, ensuring that the response is not only contextually appropriate but also factually precise. Mathematically, AC is determined by comparing the factual content of the generated answer to a ground truth or verified source, assessing whether the information aligns with established facts. A high AC score indicates that the response is accurate and aligned with the expected factual knowledge. So, Answer Correctness evaluates the factual reliability of the generated response, with a higher score signifying that the answer is both factually accurate and semantically relevant to the question posed. This metric is indispensable for ensuring that response generation models produce reliable and trustworthy outputs.

$$AC = \frac{\text{Number of Correct Answers}}{\text{Total Number of Answers}}$$

Context Recall (CR) evaluates the effectiveness of the retrieval process by measuring the proportion of retrieved contexts that contain the relevant and correct information required to answer the query. This metric is vital for assessing the quality of the information retrieval system, as it reflects the system's ability to retrieve the appropriate context from which a valid answer can be generated. Mathematically, CR is calculated by comparing the retrieved contexts to the ground truth, determining what percentage of the retrieved contexts contain the necessary and accurate information. A high CR score indicates that a large proportion of the retrieved contexts are relevant and contribute effectively to answering the question. So, Context Recall assesses the quality of the retrieval process, with a higher score indicating that the system retrieves more relevant and accurate information essential for generating correct responses. This metric is essential for understanding the retrieval system's performance in information retrieval-based tasks.

$$CR = \frac{\text{Number of Relevant Contexts Containing the Correct Answer}}{\text{Total Number of Relevant Contexts}}$$

Context Precision (CP) measures the relevance of the retrieved contexts by calculating the proportion of useful and relevant contexts among all the retrieved

documents. This metric is crucial for evaluating the quality of the retrieval process, as it reflects the system’s ability to return only those contexts that are truly pertinent to answering the query. Mathematically, CP is determined by comparing the retrieved contexts to the ground truth, calculating the percentage of relevant contexts among all the retrieved documents. A higher CP score indicates that fewer irrelevant contexts are included in the retrieval process, signifying a more efficient and accurate retrieval system. So, Context Precision assesses the relevance of the retrieved contexts, with a higher score indicating that the system retrieves more pertinent and useful information to effectively answer the query. This metric is essential for evaluating the effectiveness of the information retrieval system in tasks where relevance plays a critical role in producing accurate responses.

$$CP = \frac{\text{Number of Relevant Contexts Containing the Correct Answers}}{\text{Total Number of Retrieved Contexts}}$$

The Overall Score (OS) represents an aggregated performance measure derived from a weighted average of the individual metrics. The weights (w_i) are application-specific and emphasize the relative importance of each metric. The overall score is computed as:

$$(OS) = w_1 \cdot F + w_2 \cdot AR + w_3 \cdot AC + w_4 \cdot CR + w_5 \cdot CP, \quad \text{where} \quad \sum_{i=1}^5 w_i = 1$$

The RAGAS framework is integral to the systematic evaluation of RAG-based systems. It provides a robust mechanism for identifying key performance aspects, such as faithfulness deficiencies or suboptimal retrieval precision. Furthermore, the framework facilitates benchmarking by enabling standardized comparisons of different RAG models across identical datasets. RAGAS also serves as a guide for optimizing RAG systems by offering actionable insights into retrieval and generation processes. For instance, it highlights areas where retrieval strategies such as dynamic chunking or cross-encoder-based re-ranking could be improved. By quantifying the impact of such advanced techniques, RAGAS establishes a solid foundation for enhancing retrieval-augmented generation architectures in diverse application domains.

4

RESULTS

4.1 Evaluation of Model Performance

This section provides an in-depth analysis of the performance of the proposed Retrieval-Augmented Generation (RAG) framework, leveraging FAISS for semantic retrieval. The evaluation encompasses a comparative assessment of two FAISS indexing strategies: IndexIVFFlat, which partitions the vector space into clusters to enhance retrieval efficiency, and IndexHNSWFlat, which constructs a hierarchical navigable small world graph to enable more precise nearest-neighbor search. The performance is systematically evaluated using key metrics such as faithfulness, answer correctness, context recall, context precision, and answer relevancy. Furthermore, the effects of dynamic chunking and cross-encoder re-ranking on retrieval performance are examined. The results underscore the significant contributions of these techniques in improving faithfulness and answer correctness by maintaining semantic consistency and emphasizing contextually relevant responses. However, a notable trade-off is observed in terms of context precision and answer relevancy, potentially attributable to overlapping or redundant information introduced during the retrieval process.

4.1.0.1 Experimental Evaluation and Results of RAG Models

In this study, a comprehensive benchmark was developed by evaluating six different models using the WikiRAG-TR dataset, which is a Turkish text dataset. The models considered in this evaluation were as follows: ytu-ce-cosmos/Turkish-Llama-8b-DPO-v0.1, ytu-ce-cosmos/Turkish-Llama-8b-Instruct-v0.1, google/gemma-2-9b-it [31], Eurdem/Defne-llama3.1-8B [32], NousResearch/Meta-Llama-3-8B [33], and mistralai/Mistral-7B-Instruct-v0.2 [34]. For each of these models, a question-answering system was developed utilizing the Retrieval-Augmented Generation (RAG) architecture, specifically tailored to the Turkish dataset.

To enhance the performance of the models, advanced techniques such as cross-encoder based re-ranking and dynamic chunking were implemented. These methods aim to improve the quality and accuracy of the generated answers by refining the ranking of retrieved documents and adjusting the chunking strategy dynamically during the retrieval process. Additionally, various indexing methods in vector databases were compared to evaluate their impact on the retrieval process and the overall performance of the RAG-based models.

As a result of the experiments conducted to enhance the performance of each model, the benchmarks for scenarios involving dynamic chunking and cross-encoder re-ranking with IVF and HNSW indexing, as well as those without these techniques, are presented in the table below. Specifically, the baseline model and the configurations with top_k=15 and top_k=30 utilize only the RAG architecture and vector databases, whereas the configurations with IVF index and HNSW index incorporate both dynamic chunking and cross-encoder re-ranking techniques.

Table 4.1 Ytu-ce-cosmos/Turkish-Llama-8b-DPO-v0.1 model results

Model Name	Faithfulness	Answer Relevancy	Answer Correctness	Context Recall	Context Precision
ytu-ce-cosmos/Turkish-Llama-8b-DPO-v0.1 no indexing	0.30	0.73	0.51	0.71	0.93
ytu-ce-cosmos/Turkish-Llama-8b-DPO-v0.1 with top_k=15	0.15	0.25	0.29	0.71	0.93
ytu-ce-cosmos/Turkish-Llama-8b-DPO-v0.1 with top_k=30	0.14	0.19	0.30	0.70	0.93
ytu-ce-cosmos/Turkish-Llama-8b-DPO-v0.1 with IVF index	0.53	0.40	0.71	0.58	0.87
ytu-ce-cosmos/Turkish-Llama-8b-DPO-v0.1 with HNSW index	0.61	0.39	0.67	0.74	0.87

As illustrated in Table 4.1, the performance of the ytu-ce-cosmos/Turkish-Llama-8b-DPO-v0.1 model is assessed using various indexing methods. In the baseline scenario, no vector database is employed (referred to as no indexing). For configurations utilizing the IndexFlatL2 index, top_k parameters are set to 15 and 30. The indexing mechanism in these cases involves retrieving the top-k nearest neighbors based on the Euclidean distance in the vector space. Here, the top_k parameter determines how many of the closest vectors are retrieved for a given query. Unlike re-ranking approaches, where parameters such as top_k1 and top_k2 define the number of candidates selected and re-ranked, respectively, the top_k parameter in IndexFlatL2 directly controls the size of the retrieved set and does not involve any subsequent re-ranking processes. Furthermore, advanced indexing techniques, such as IVF and HNSW, are employed to enhance retrieval efficiency and scalability.

The baseline model achieves moderate faithfulness (0.30) and answer correctness (0.51), with high context recall (0.71) and context precision (0.93), indicating good context retrieval but room for improvement in accuracy. top_k=15: Faithfulness

drops to 0.15, answer relevancy to 0.25, and answer correctness to 0.29, showing a loss in both context fidelity and answer accuracy. top_k=30: Performance worsens further, with faithfulness (0.14) and answer relevancy (0.19) declining. This indicates that restricting context retrieval with top_k negatively impacts the model’s performance. IVF indexing improves faithfulness (0.53) and answer correctness (0.71), but reduces context recall (0.58), suggesting a trade-off between context retrieval and accuracy. HNSW indexing shows the best performance, with faithfulness (0.61) and context recall (0.74) improving, though answer correctness (0.67) remains moderate. This indicates that HNSW enhances context alignment and retrieval efficiency.

Table 4.2 Ytu-ce-cosmos/Turkish-Llama-8b-Instruct-v0.1 model results

Model Name	Faithfulness	Answer Relevancy	Answer Correctness	Context Recall	Context Precision
ytu-ce-cosmos/Turkish-Llama-8b-Instruct-v0.1 no indexing	0.25	0.50	0.52	0.83	0.87
ytu-ce-cosmos/Turkish-Llama-8b-Instruct-v0.1 with top_k=15	0.13	0.06	0.30	0.72	0.87
ytu-ce-cosmos/Turkish-Llama-8b-Instruct-v0.1 with top_k=30	0.12	0.12	0.30	0.81	0.93
ytu-ce-cosmos/Turkish-Llama-8b-Instruct-v0.1 with IVF index	0.51	0.38	0.41	0.73	0.87
ytu-ce-cosmos/Turkish-Llama-8b-Instruct-v0.1 with HNSW index	0.64	0.40	0.43	0.76	0.87

As shown in Table 4.2, the performance of the ytu-ce-cosmos/Turkish-Llama-8b-Instruct-v0.1 model is evaluated with different indexing methods: top_k, IVF and HNSW. The model’s performance is measured using faithfulness, answer relevancy, answer correctness, context recall and context precision. Without any indexing, the model shows moderate performance with a faithfulness of 0.25 and answer correctness of 0.52, but performs well in context recall (0.83) and context precision (0.87), suggesting it retrieves relevant information, though with some loss of accuracy and faithfulness. Applying the top k parameter reduces performance significantly, especially in faithfulness (0.13) and answer relevancy (0.06). This suggests limiting the number of retrieved contexts impacts both context adherence and relevance. With IVF indexing, faithfulness improves to 0.51, but answer correctness (0.41) remains low. Context recall (0.73) also shows moderate improvement, indicating that IVF indexing helps in maintaining context but does not significantly enhance answer accuracy. HNSW indexing results in the highest faithfulness (0.64) and context recall (0.76), showing notable improvements in context alignment and information retrieval. However, answer correctness (0.43) still shows only modest improvement.

As presented in Table 4.3, the malhajar/Mistral-7B-Instruct-v0.2-turkish model’s performance varies significantly across configurations, including baseline, top_k, IVF and HNSW indexing methods. The baseline model demonstrates limited faithfulness (0.01) and answer correctness (0.24) but strong context recall (0.76)

Table 4.3 Malhajar/Mistral-7B-Instruct-v0.2-turkish model results

Model Name	Faithfulness	Answer Relevancy	Answer Correctness	Context Recall	Context Precision
malhajar/Mistral-7B-Instruct-v0.2-turkish no indexing	0.01	0.26	0.24	0.76	0.93
malhajar/Mistral-7B-Instruct-v0.2-turkish with top_k=15	0.06	0.17	0.30	0.73	0.94
malhajar/Mistral-7B-Instruct-v0.2-turkish with top_k=30	0.13	0.12	0.30	0.74	0.86
malhajar/Mistral-7B-Instruct-v0.2-turkish with IVF index	0.42	0.38	0.43	0.77	0.89
malhajar/Mistral-7B-Instruct-v0.2-turkish with HNSW index	0.45	0.43	0.46	0.81	0.89

and context precision (0.93). Introducing top_k improves faithfulness (0.06 for top_k=15, 0.13 for top_k=30) but slightly reduces context precision, indicating a trade-off between response alignment and context accuracy. IVF indexing significantly enhances faithfulness (0.42) and answer correctness (0.43) while maintaining high context recall (0.77). HNSW indexing achieves the best overall results, with the highest faithfulness (0.45), improved answer correctness (0.46), and increased context recall (0.81), demonstrating superior alignment between retrieved context and generated responses. While IVF improves response accuracy, HNSW indexing outperforms all methods by providing the most balanced and effective improvements across metrics, particularly in faithfulness and context recall.

Table 4.4 NousResearch/Meta-Llama-3-8B model results

Model Name	Faithfulness	Answer Relevancy	Answer Correctness	Context Recall	Context Precision
NousResearch/Meta-Llama-3-8B no indexing	0.06	0.48	0.36	0.82	0.93
NousResearch/Meta-Llama-3-8B with top_k=15	0.15	0.13	0.27	0.75	0.93
NousResearch/Meta-Llama-3-8B with top_k=30	0.12	0.12	0.27	0.78	0.93
NousResearch/Meta-Llama-3-8B with IVF index	0.51	0.38	0.70	0.68	0.86
NousResearch/Meta-Llama-3-8B with HNSW index	0.57	0.43	0.68	0.83	0.86

Table 4.4 presents the performance of the NousResearch/Meta-Llama-3-8B model under various configurations, including baseline, top_k, IVF and HNSW indexing methods. The baseline model achieves moderate faithfulness (0.06), strong context recall (0.82), and high context precision (0.93). Using top_k values (15 and 30) improves faithfulness (0.15 and 0.12, respectively) but reduces answer relevancy (0.13 for top_k=15). These results indicate that top_k enhances faithfulness at the expense of relevancy and correctness. With IVF indexing, faithfulness improves significantly to 0.51, and answer correctness rises to 0.70. However, context recall decreases to 0.68, suggesting a trade-off between enhanced response accuracy and retrieval effectiveness. HNSW indexing yields the highest faithfulness (0.57) and improves answer relevancy (0.43). Context recall also reaches 0.83, matching the baseline performance, while maintaining a stable context precision of 0.86. This demonstrates that HNSW effectively balances response quality and contextual

alignment. Both IVF and HNSW indexing outperform the baseline and top_k configurations. While IVF enhances accuracy, HNSW indexing provides superior performance across all key metrics, particularly in faithfulness and context recall, making it the optimal indexing strategy for this model.

Table 4.5 Eurdem/Defne-llama3.1-8B model results

Model Name	Faithfulness	Answer Relevancy	Answer Correctness	Context Recall	Context Precision
Eurdem/Defne-llama3.1-8B no indexing	0.06	0.18	0.32	0.74	0.93
Eurdem/Defne-llama3.1-8B with top_k=15	0.13	0.06	0.30	0.74	0.87
Eurdem/Defne-llama3.1-8B with top_k=30	0.13	0.06	0.30	0.72	0.87
Eurdem/Defne-llama3.1-8B with IVF index	0.46	0.37	0.73	0.68	0.86
Eurdem/Defne-llama3.1-8B with HNSW index	0.53	0.43	0.78	0.72	0.86

Table 4.5 summarizes the performance of Eurdem/Defne-llama3.1-8B under various configurations, including baseline, top_k, IVF and HNSW indexing methods. The baseline model achieves low faithfulness (0.06) and moderate context recall (0.74). Using top_k=15 and top_k=30 slightly improves faithfulness (0.13) but has minimal effect on other metrics, suggesting limited gains from top_k adjustments. IVF indexing substantially enhances faithfulness (0.46) and answer correctness (0.73). However, context recall declines to 0.68, indicating a trade-off between improved response accuracy and retrieval coverage. HNSW indexing delivers the highest faithfulness (0.53), answer relevancy (0.43), and answer correctness (0.78). It also maintains competitive context recall (0.72) and context precision (0.86), demonstrating a strong balance between accuracy and contextual alignment. While IVF indexing offers significant gains in accuracy, HNSW indexing consistently achieves superior overall performance, making it the most effective method for this model.

Table 4.6 highlights the performance of google/gemma-2-9b-it under baseline, top_k, IVF, and HNSW configurations. The baseline model shows moderate faithfulness (0.38) and low context recall (0.40). Adjusting top_k values improves context recall significantly, with top_k=30 achieving 0.77. However, both faithfulness and answer correctness decrease (to 0.14 and 0.28, respectively),

Table 4.6 Google/gemma-2-9b-it model results

Model Name	Faithfulness	Answer Relevancy	Answer Correctness	Context Recall	Context Precision
google/gemma-2-9b-it no indexing	0.38	0.58	0.36	0.40	0.55
google/gemma-2-9b-it with top_k=15	0.18	0.13	0.29	0.70	0.93
google/gemma-2-9b-it with top_k=30	0.14	0.20	0.28	0.77	0.93
google/gemma-2-9b-it with IVF index	0.37	0.71	0.53	0.68	0.86
google/gemma-2-9b-it with HNSW index	0.48	0.73	0.56	0.79	0.87

suggesting reduced accuracy. IVF indexing substantially enhances answer relevancy (0.71) and answer correctness (0.53), while maintaining competitive faithfulness (0.37). However, it offers only moderate improvements in context recall (0.68). HNSW indexing delivers the highest faithfulness (0.48), answer relevancy (0.73), and answer correctness (0.56). It also significantly improves context recall (0.79), indicating a strong balance between response accuracy and contextual alignment. While IVF indexing enhances specific metrics, HNSW indexing achieves superior overall performance, particularly in faithfulness and contextual retrieval, making it the preferred method for this model.

4.2 Comparison of Indexing Methods: IVF vs. HNSW

In the comparative analysis of the IVF and HNSW indexing methods, it is evident that HNSW generally outperforms IVF in several key metrics, particularly in context recall, answer correctness, and context precision. This is primarily due to HNSW's superior ability to conduct more accurate nearest neighbor searches, which ensures better alignment of context and responses. The refined search capabilities of HNSW allow models to retrieve more relevant context and generate responses with higher accuracy, making it particularly advantageous for tasks involving complex or nuanced queries.

In contrast, IVF indexing tends to provide faster and more efficient context retrieval, especially in scenarios where large-scale data processing is required. However, IVF's performance is often compromised in terms of answer correctness and faithfulness. While it is effective in grouping semantically similar vectors, this efficiency comes at the cost of reduced precision in response generation. Models utilizing IVF show improvements in context recall but struggle to maintain high answer accuracy, particularly in more intricate or detailed queries.

For instance, models such as google/gemma-2-9b-it and Eurdem/Defne-llama3.1-8B exhibit clear advantages when employing HNSW indexing, especially in terms of answer correctness and context recall. These models benefit from the precise search mechanism of HNSW, which improves both the semantic accuracy of responses and the retention of relevant context. On the other hand, the use of IVF indexing in these models results in higher context recall but lower accuracy and faithfulness, demonstrating the trade-offs between efficiency and precision.

Similarly, models like NousResearch/Meta-Llama-3-8B and malhajar/Mistral-7B-Instruct-v0.2-turkish show that while IVF indexing provides

a moderate boost in context recall, HNSW indexing is consistently superior in generating correct and precise answers. The advantages of HNSW in these models are particularly pronounced in tasks requiring a high level of accuracy and context alignment.

In summary, HNSW indexing is generally the preferred choice when precision and context alignment are critical, as it enhances both answer correctness and context retention. IVF, however, remains a viable option in large-scale applications where efficiency in context retrieval is prioritized over the absolute accuracy of responses.

4.3 Model Performance Visualization

In addition to the quantitative evaluation, the performance of each model variant was visually analyzed through radar charts. These charts provide a comprehensive view of how the models perform across several RAGAS metrics, with a focus on variations induced by different indexing configurations. The metrics considered include faithfulness, answer relevancy, answer correctness, context recall, and context precision.

ytu-ce-cosmos/Turkish-Llama-8b-DPO-v0.1 Model Performance Comparison

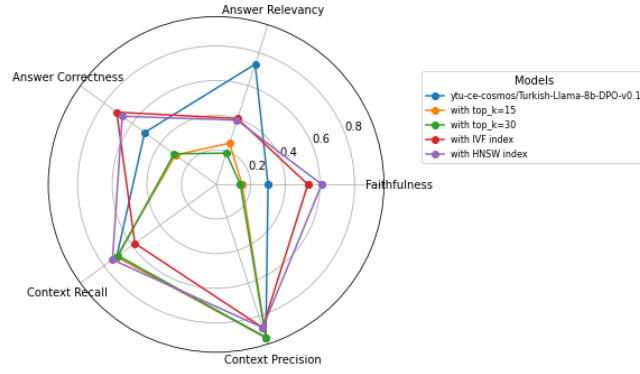


Figure 4.1 Ytu-ce-cosmos/Turkish-Llama-8b-DPO-v0.1

ytu-ce-cosmos/Turkish-Llama-8b-Instruct-v0.1 Model Performance Comparison

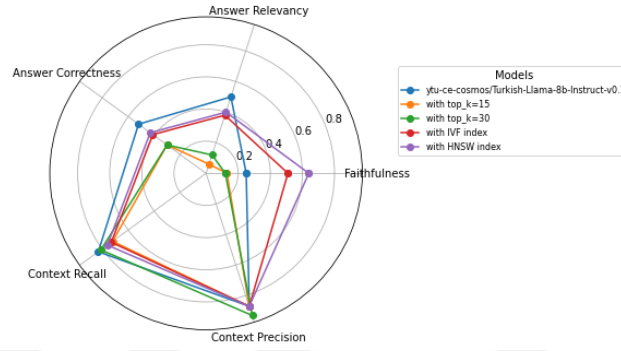


Figure 4.2 Ytu-ce-cosmos/Turkish-Llama-8b-Instruct-v0.1

malhajar/Mistral-7B-Instruct-v0.2-turkish Model Performance Comparison

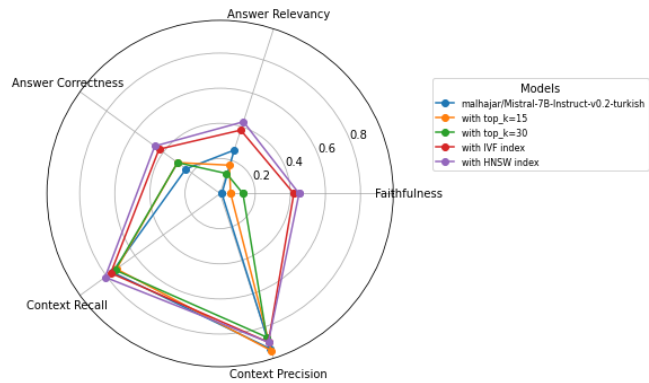


Figure 4.3 Malhajar/Mistral-7B-Instruct-v0.2-turkish

NousResearch/Meta-Llama-3-8B Model Performance Comparison

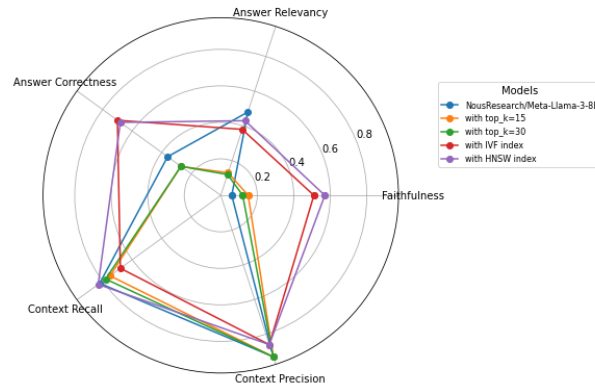


Figure 4.4 NousResearch/Meta-Llama-3-8B

Eurdem/Defne-llama3.1-8B Model Performance Comparison

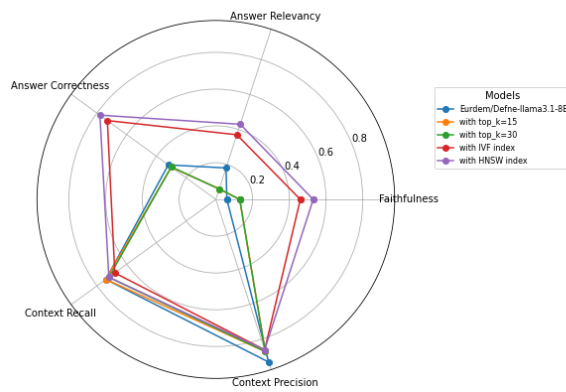


Figure 4.5 Eurdem/Defne-llama3.1-8B

google/gemma-2-9b-it Model Performance Comparison

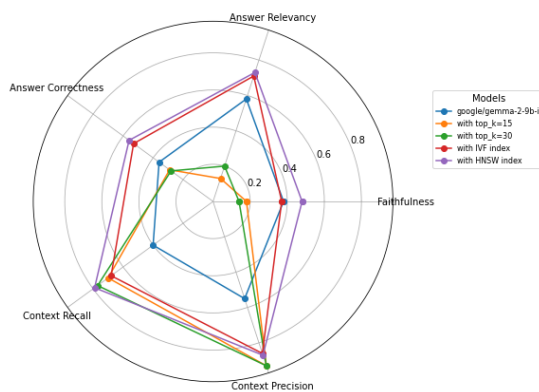


Figure 4.6 Google/gemma-2-9b-it

The introduction of indexing methods such as IVF and HNSW generally led to significant improvements in context recall across most of the models. In particular, models utilizing HNSW indexing exhibited a pronounced enhancement in context recall, owing to the method's ability to capture more refined semantic distinctions. This capability enabled a more comprehensive retrieval of relevant information. For instance, Figure 4.6 shows the google/gemma-2-9b-it model with HNSW indexing, which demonstrated a substantial increase in context recall, rising from 0.40 in the baseline model to 0.79. Similarly, Figure 4.5 presents the Eurdem/Defne-llama3.1-8B model with HNSW indexing, which achieved a context recall score of 0.72, although slightly lower than the 0.74 achieved with IVF indexing. Nonetheless, this still highlighted a significant improvement in the model's ability to retrieve relevant data.

However, the gains in context recall came with a minor trade-off in context precision. The models utilizing IVF and HNSW indexing displayed slight reductions in context precision compared to the standard configurations. For example, Figure 4.6 also shows that the google/gemma-2-9b-it with HNSW indexing maintained a high precision score of 0.87, reflecting a balanced trade-off between recall and precision. In contrast, Figure 4.5 illustrates the Eurdem/Defne-llama3.1-8B model with IVF indexing, which showed a slight decrease in precision, dropping from 0.93 in the baseline model to 0.86. These findings underscore the inherent trade-off between recall and precision when optimizing retrieval strategies, particularly when enhancing recall through advanced indexing methods.

The performance variations across different models further emphasize the interplay between architectural design and indexing techniques in shaping retrieval effectiveness. For instance, Figure 4.3 shows the malhajar/Mistral-7B-Instruct-v0.2-turkish model with HNSW indexing, which exhibited notable gains in faithfulness (0.45) and answer relevancy (0.43), while maintaining a high level of context precision (0.89). Conversely, Figure 4.2 presents the ytu-ce-cosmos/Turkish-Llama-8b-Instruct-v0.1 model with IVF indexing, which demonstrated substantial improvements in answer correctness (0.73) and answer relevancy (0.46), albeit at the expense of a slight reduction in context precision (0.86).

These results underscore the critical importance of selecting the most suitable indexing method based on the specific requirements of the application. An optimal approach should consider both context recall and context precision to ensure the best overall model performance across various tasks.

4.4 Theoretical and Empirical Impact of Dynamic Chunking and Reranking on RAG Metrics

4.4.1 Theoretical Impact

Dynamic chunking and reranking are pivotal techniques in enhancing the performance of Retrieval-Augmented Generation (RAG) architectures, each contributing to the improvement of specific evaluation metrics in distinct ways.

Dynamic chunking involves partitioning documents into smaller, contextually meaningful segments, optimizing the retrieval process. By dividing documents into information-dense chunks, the system is able to more precisely identify and retrieve relevant information. This segmentation results in a more granular understanding of the content, significantly improving both Context Recall and Context Precision. The ability to target specific, meaningful portions of text allows for better contextual representation, which in turn facilitates the retrieval of highly relevant information while minimizing the inclusion of irrelevant data. This refinement ensures that the model can access nuanced details without overwhelming the retrieval pipeline with unnecessary content.

Reranking, on the other hand, operates as a secondary layer of optimization that prioritizes the most contextually pertinent chunks after the initial retrieval process. This is typically achieved through advanced scoring mechanisms, such as cross-encoder models, which refine the ranking of candidate responses. The application of reranking enhances Answer Relevancy by ensuring that only the most contextually aligned content is considered for response generation. Furthermore, reranking contributes to Answer Correctness by filtering out potentially erroneous or misleading information. Additionally, reranking plays a crucial role in improving Faithfulness by ensuring that the selected content is both reliable and faithful to the original query, thereby reducing the risk of generating hallucinated or irrelevant responses.

In summary, dynamic chunking enhances the retrieval process by improving the granularity of context, thereby optimizing Context Recall and Context Precision. Reranking, in turn, refines the selection of the most relevant and faithful content, improving Answer Relevancy, Answer Correctness, and Faithfulness.

4.4.2 Empirical Analysis

To assess the practical implications of these theoretical insights, experiments were conducted using two distinct FAISS indexing methods—FAISS IVF and

FAISS HNSW. Both of these methods were integrated with dynamic chunking and reranking functionalities to explore their impact on the performance of the RAG system. These indexing methods were specifically chosen for their complementary advantages in terms of retrieval accuracy and computational efficiency.

FAISS IVF (Inverted File) is particularly well-suited for large-scale datasets due to its hierarchical clustering approach. This technique allows for efficient retrieval, even from extensive data pools, without significant performance degradation. On the other hand, FAISS HNSW (Hierarchical Navigable Small World) indexing leverages proximity graphs to enable high recall rates and precise nearest-neighbor searches. The characteristics of FAISS HNSW make it especially valuable for RAG systems, where high fidelity in response retrieval is crucial. The strategic selection of these two indexing methods highlights an effort to combine the strengths of hierarchical clustering with proximity-based indexing, which together can significantly enhance RAG metrics.

The experiments were designed to evaluate the effects of dynamic chunking and reranking on several key RAG metrics. The findings from these experiments provide empirical support for the theoretical claims, demonstrating that the combination of these techniques—when applied with appropriate indexing methods—substantially improves performance across various evaluation metrics. These results underscore the value of integrating dynamic chunking and reranking into RAG systems, as they lead to more accurate, relevant, and faithful responses.

Table 4.7 Quantitative analysis of metric improvements of ytu-ce-cosmos/Turkish-Llama-8b-DPO-v0.1 model

Metric	RAG Only	RAG + IVF (Dynamic Chunking + Re-Ranking)	RAG + HNSW (Dynamic Chunking + Re-Ranking)	% Improvement (IVF vs. Baseline)	% Improvement (HNSW vs. Baseline)
Faithfulness	0.39	0.53	0.61	+35.9%	+56.4%
Answer Relevancy	0.31	0.40	0.39	+29.0%	+25.8%
Answer Correctness	0.55	0.71	0.67	+29.1%	+21.8%
Context Recall	0.46	0.58	0.74	+26.1%	+60.9%
Context Precision	0.78	0.87	0.87	+11.5%	+11.5%

Table 4.8 Quantitative analysis of metric improvements of ytu-ce-cosmos/Turkish-Llama-8b-Instruct-v0.1 model

Metric	RAG Only	RAG + IVF (Dynamic Chunking + Re-Ranking)	RAG + HNSW (Dynamic Chunking + Re-Ranking)	% Improvement (IVF vs. Baseline)	% Improvement (HNSW vs. Baseline)
Faithfulness	0.41	0.51	0.64	+24.4%	+56.1%
Answer Relevancy	0.30	0.38	0.40	+26.7%	+33.3%
Answer Correctness	0.31	0.41	0.43	+32.3%	+38.7%
Context Recall	0.58	0.73	0.76	+25.9%	+31.0%
Context Precision	0.78	0.87	0.87	+11.5%	+11.5%

Table 4.9 Quantitative analysis of metric improvements of malhajar/Mistral-7B-Instruct-v0.2 model

Metric	RAG Only	RAG + IVF (Dynamic Chunking + Re-Ranking)	RAG + HNSW (Dynamic Chunking + Re-Ranking)	% Improvement (IVF vs. Baseline)	% Improvement (HNSW vs. Baseline)
Faithfulness	0.30	0.42	0.45	+40.0%	+50.0%
Answer Relevancy	0.29	0.38	0.43	+31.0%	+48.3%
Answer Correctness	0.33	0.43	0.46	+30.3%	+39.4%
Context Recall	0.56	0.77	0.81	+37.5%	+44.6%
Context Precision	0.80	0.89	0.89	+11.3%	+11.3%

Table 4.10 Quantitative analysis of metric improvements of NousResearch/Meta-Llama-3-8B model

Metric	RAG Only	RAG + IVF (Dynamic Chunking + Re-Ranking)	RAG + HNSW (Dynamic Chunking + Re-Ranking)	% Improvement (IVF vs. Baseline)	% Improvement (HNSW vs. Baseline)
Faithfulness	0.34	0.51	0.57	+50.0%	+67.6%
Answer Relevancy	0.27	0.38	0.43	+40.7%	+59.3%
Answer Correctness	0.55	0.70	0.68	+27.3%	+23.6%
Context Recall	0.48	0.68	0.83	+41.7%	+72.9%
Context Precision	0.76	0.86	0.86	+13.2%	+13.2%

Table 4.11 Quantitative analysis of metric improvements of Eurdem/Defne-llama-3.1-8B model

Metric	RAG Only	RAG + IVF (Dynamic Chunking + Re-Ranking)	RAG + HNSW (Dynamic Chunking + Re-Ranking)	% Improvement (IVF vs. Baseline)	% Improvement (HNSW vs. Baseline)
Faithfulness	0.35	0.46	0.53	+31.4%	+51.4%
Answer Relevancy	0.29	0.37	0.43	+27.6%	+48.3%
Answer Correctness	0.63	0.73	0.78	+15.9%	+23.8%
Context Recall	0.52	0.68	0.72	+30.8%	+38.5%
Context Precision	0.76	0.86	0.86	+13.2%	+13.2%

Table 4.12 Quantitative analysis of metric improvements of google/gemma-2-9b-it model

Metric	RAG Only	RAG + IVF (Dynamic Chunking + Re-Ranking)	RAG + HNSW (Dynamic Chunking + Re-Ranking)	% Improvement (IVF vs. Baseline)	% Improvement (HNSW vs. Baseline)
Faithfulness	0.28	0.37	0.48	+32.1%	+71.4%
Answer Relevancy	0.61	0.71	0.73	+16.4%	+19.7%
Answer Correctness	0.43	0.53	0.56	+23.3%	+30.2%
Context Recall	0.52	0.68	0.79	+30.8%	+51.9%
Context Precision	0.74	0.86	0.87	+16.2%	+17.6%

The empirical analysis highlights the significant contributions of dynamic chunking, reranking, and advanced indexing techniques like FAISS HNSW in enhancing the performance of RAG systems. HNSW indexing consistently demonstrated superior faithfulness by aligning retrieved content more accurately with source material, ensuring the integrity and reliability of responses. While reranking modestly improved answer relevancy, further optimization could amplify its impact. Answer correctness saw consistent, albeit modest, improvements with HNSW indexing, reinforcing its role in refining response accuracy.

Dynamic chunking, particularly when paired with HNSW, significantly boosted context recall by enabling the retrieval of more comprehensive and relevant information. At the same time, context precision remained consistently high, indicating an effective balance between relevant content extraction and noise reduction. These findings validate the complementary roles of dynamic chunking and reranking, combined with advanced indexing, in improving critical RAG metrics such as faithfulness, correctness, and contextual relevance.

5

DISCUSSION

The findings of this study underscore the transformative potential of optimized Retrieval-Augmented Generation (RAG) architectures in addressing the inherent limitations of conventional language models when applied to domain-specific and high-context environments. By integrating advanced retrieval, indexing, and ranking mechanisms, this research presents a robust framework that significantly enhances the performance of RAG systems across critical metrics, including faithfulness, answer correctness, and context recall.

5.1 Implications of Advanced Indexing

The empirical results demonstrate the superiority of the HNSW (Hierarchical Navigable Small World) index in improving retrieval precision and maintaining computational efficiency. Its consistent outperformance in terms of faithfulness highlights its capacity to align retrieved content with source materials, thereby ensuring the integrity and reliability of generated responses. These results suggest that the hierarchical structure and navigability of HNSW indexing contribute to more effective nearest-neighbor searches, which are essential for preserving semantic coherence in retrieval-augmented architectures.

While the IndexIVFFlat offered notable advantages in retrieval speed, its performance in contextual alignment and faithfulness was comparatively limited. This underscores the trade-off between retrieval efficiency and accuracy—a critical consideration for scaling RAG systems in real-world applications requiring high degrees of precision and reliability.

5.2 Enhancing Retrieval Coherence with Dynamic Chunking

Dynamic chunking emerged as a pivotal technique for preserving contextual coherence during the segmentation of textual data. By adaptively partitioning

documents into semantically relevant chunks, this method reduces information fragmentation, which has traditionally hindered the effectiveness of static chunking strategies. The combination of dynamic chunking with HNSW indexing achieved significant improvements in context recall, enabling the retrieval of a broader and more comprehensive range of relevant information.

This finding is particularly relevant for applications involving complex queries or domain-specific knowledge, where nuanced understanding and contextual continuity are paramount. However, the potential risk of overlapping or redundant information due to finer-grained chunking strategies warrants further investigation, particularly concerning its impact on context precision and downstream generation quality.

5.3 The Role of Cross-Encoder Re-Ranking

The inclusion of a cross-encoder-based re-ranking mechanism was instrumental in refining the initial retrieval results, ensuring that only the most contextually relevant documents were passed to the generation module. While the improvements in answer relevancy were modest, the consistency of these gains across different configurations reinforces the utility of re-ranking mechanisms in retrieval pipelines.

The relatively limited impact of re-ranking on answer relevancy suggests that additional fine-tuning or the incorporation of more sophisticated scoring methodologies could further optimize this component. Exploring hybrid approaches that combine traditional ranking models with neural re-ranking techniques may offer a promising avenue for future research.

5.4 Balancing Trade-Offs in Retrieval-Augmented Systems

A critical insight from this study lies in the observed trade-offs between key performance metrics. While the integration of dynamic chunking and HNSW indexing markedly improved faithfulness and context recall, these advancements occasionally introduced redundant information, thereby marginally affecting context precision and answer relevancy. This trade-off highlights the challenges of balancing granularity in retrieval mechanisms with the need for precise and non-overlapping information.

Addressing these trade-offs requires a more nuanced approach to retrieval optimization, potentially involving hybrid indexing strategies or adaptive retrieval thresholds that dynamically balance precision and recall based on query complexity.

5.5 Contributions to the Turkish Language and Broader NLP Context

This study also makes a notable contribution to the application of RAG systems in the Turkish language, which presents unique linguistic challenges due to its agglutinative structure and rich morphological features. By leveraging advanced embedding techniques (e.g., Sentence-Transformers) and adaptive retrieval mechanisms, the proposed architecture demonstrates its versatility and adaptability to non-English datasets.

These findings not only validate the applicability of the proposed framework to real-world Turkish data but also establish a foundation for broader adoption of optimized RAG systems in other low-resource and morphologically complex languages.

5.6 Future Directions for RAG Optimization

The integration of dynamic chunking, advanced indexing, and re-ranking represents a significant step forward in the optimization of RAG systems. However, the observed limitations and trade-offs call for continued exploration of complementary techniques, such as:

Dynamic embedding strategies that adapt to varying query complexities and domain-specific requirements.

Hybrid retrieval architectures that combine the efficiency of clustering-based methods with the precision of graph-based indexing.

Fine-grained re-ranking models leveraging reinforcement learning to optimize contextual relevance.

The results of this study pave the way for further innovations in retrieval-augmented systems, particularly in applications demanding high contextual preservation, domain specificity, and precision. By addressing the identified challenges and trade-offs, future research can extend the capabilities of RAG architectures, enabling their deployment in increasingly complex and high-stakes environments, such as healthcare, legal analysis, and scientific research.

6 CONCLUSION

This study provides a detailed exploration of methodologies to significantly enhance the response accuracy of Retrieval-Augmented Generation (RAG) architectures by incorporating cutting-edge techniques, including dynamic chunking, cross-encoder-based re-ranking, and FAISS HNSW indexing. By employing a Turkish domain-specific dataset and focusing on questions that demand precise contextual understanding, this research underscores the critical contributions of these techniques to the overall performance of RAG models.

The results reveal that dynamic chunking is instrumental in overcoming the inherent limitations of static chunking approaches. By adaptively segmenting context into granular and task-specific chunks, dynamic chunking facilitates higher fidelity in context retrieval, leading to marked improvements in RAGAS metrics such as context recall and context precision. This advancement ensures that the retrieved information is not only relevant but also optimally aligned with the query's specific requirements, which is particularly vital for complex or ambiguous queries.

The integration of cross-encoder-based re-ranking further amplified this impact by refining the retrieval pipeline to prioritize the most semantically relevant chunks. The experimental findings indicate that cross-encoder re-ranking plays a transformative role in improving answer correctness and faithfulness, directly addressing the challenges posed by noisy or incomplete retrieved contexts. This enhanced filtering capability ensures that RAG architectures can deliver responses that are both accurate and contextually grounded, demonstrating a substantial boost across all evaluated RAGAS metrics.

Additionally, the adoption of FAISS HNSW indexing emerged as a cornerstone of the proposed enhancements. Compared to traditional IVF methods, HNSW indexing consistently outperformed by delivering higher context recall and precision while maintaining computational efficiency. Its hierarchical approach to nearest-neighbor search enables a more refined and accurate retrieval process,

which directly translates into elevated response quality. The synergy between dynamic chunking, cross-encoder re-ranking, and HNSW indexing resulted in a robust framework that achieved a significant leap in RAGAS metric performance, setting a new benchmark for response success in RAG systems.

The RAGAS evaluation framework proved invaluable in quantifying these improvements. Faithfulness, answer relevance, and context recall metrics consistently reflected the profound impact of the proposed techniques, reinforcing their effectiveness in addressing the critical quality dimensions of RAG architectures. These findings highlight the transformative potential of combining advanced indexing, adaptive chunking, and re-ranking techniques to optimize RAG models for high-stakes applications where precision and reliability are non-negotiable.

In summary, this research demonstrates that leveraging dynamic chunking, cross-encoder-based re-ranking, and FAISS HNSW indexing can significantly elevate the response success of RAG architectures, as evidenced by substantial improvements across RAGAS metrics. These advancements not only enhance the fidelity and relevance of retrieved information but also establish a robust foundation for deploying RAG systems in domains such as legal, medical, and academic sectors, where accuracy and contextual understanding are paramount. Future research could explore the scalability of these techniques across multilingual and larger-scale datasets, as well as integrate emerging neural re-ranking models to further augment retrieval performance.

REFERENCES

- [1] W. X. Zhao *et al.*, “A survey of large language models,” *arXiv preprint arXiv:2303.18223*, 2023.
- [2] Z. Jiang, X. Ma, W. Chen, “Longrag: Enhancing retrieval-augmented generation with long-context llms,” *arXiv preprint arXiv:2406.15319*, 2024.
- [3] D. Lewis, E. Perez, H. Penedones, *et al.*, “Retrieval-augmented generation for knowledge-intensive nlp tasks,” *arXiv preprint arXiv:2005.11401*, 2020.
- [4] Y. Liu, M. Ott, N. Goyal, *et al.*, “Roberta: A robustly optimized bert pretraining approach,” *arXiv preprint arXiv:1907.11692*, 2019.
- [5] J. Kwiatkowski, T. Palomaki, *et al.*, “Natural questions: A benchmark for question answering research,” *Transactions of the Association for Computational Linguistics*, 2019.
- [6] N. Reimers, I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” *Proceedings of EMNLP*, 2019.
- [7] J. Johnson, M. Douze, H. Jégou, “Billion-scale similarity search with gpus,” *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019. doi: 10.1109/TBDATA.2019.2931291.
- [8] H. Nogueira, K. Cho, “Passage re-ranking with bert,” *arXiv preprint arXiv:1901.04085*, 2019.
- [9] E. Aktolga, K. Balog, M. de Rijke, “A probabilistic retrieval model for information finding queries,” *ECIR*, 2011.
- [10] P. Lewis *et al.*, “Retrieval-augmented generation for knowledge-intensive nlp tasks,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [11] V. Karpukhin *et al.*, “Dense passage retrieval for open-domain question answering,” *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6769–6781.
- [12] G. Izacard, E. Grave, “Leveraging passage retrieval with generative models for open domain question answering,” *arXiv preprint arXiv:2007.01282*, 2021.
- [13] S. Robertson, H. Zaragoza, “The probabilistic relevance framework: Bm25 and beyond,” *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.
- [14] W. Song, T. Tan, Y. Qin, X. Lu, T. Liu, “Mpnet: Masked and permuted pre-training for language understanding,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. [Online]. Available: <https://huggingface.co/sentence-transformers/all-mpnet-base-v2>.

- [15] H. Déjean, S. Clinchant, T. Formal, “A thorough comparison of cross-encoders and llms for reranking splade,” *arXiv preprint arXiv:2403.10407v1*, Mar. 2024.
- [16] Z. Zhong, H. Liu, X. Cui, X. Zhang, Z. Qin, “Mix-of-granularity: Optimize the chunking granularity for retrieval-augmented generation,” *arXiv preprint arXiv:2406.00456v1 [cs.LG]*, 2024, [Accessed: Dec. 6, 2024]. [Online]. Available: <https://arxiv.org/abs/2406.00456>.
- [17] M. Douze *et al.*, “The faiss library,” *arXiv preprint arXiv:2401.08281v2*, Sep. 2024, Accessed: Dec. 6, 2024.
- [18] Y. Jin *et al.*, “Curator: Efficient indexing for multi-tenant vector databases,” *arXiv*, 2024.
- [19] G. Gupta, M. Tharun, A. Shrivastava, A. J. Smola, “Bliss: A billion scale index using iterative re-partitioning,” Rice University, ThirdAI Corp., Amazon Web Services, Tech. Rep., 2024.
- [20] M. S. Rahman, S. M. E. Rabbi, M. M. Rashid, “Optimizing domain-specific image retrieval: A benchmark of faiss and annoy with fine-tuned features,” *arXiv*, 2024.
- [21] A. Shahnaz, M. A. T. Rony, “Empowering low-resource languages: Trase architecture for enhanced retrieval-augmented generation in bangla,” *ResearchGate*, 2023, Accessed: Dec 21, 2024. [Online]. Available: https://www.researchgate.net/publication/383670478_Empowering_Low-Resource_Languages_TraSe_Architecture_for_Enhanced_Retrieval-Augmented_Generation_in_Bangla.
- [22] O. Ardic, M. U. Ozturk, I. Demirtas, S. Arslan, “Information extraction from sustainability reports in turkish through rag approach,” *2024 32nd Signal Processing and Communications Applications Conference (SIU)*, 2024, pp. 1–4. doi: 10.1109/SIU61531.2024.10600994.
- [23] Metin, *Wikirag-tr*, [Online], Available: <https://huggingface.co/datasets/Metin/WikiRAG-TR>, 2024.
- [24] Y. Gao *et al.*, “Retrieval-augmented generation for large language models: A survey,” *arXiv preprint arXiv:2312.10997v5*, Mar. 2024, Accessed: Dec. 6, 2024.
- [25] I. S. Singh, R. Aggarwal, I. Allahverdiyev, A. Akalin, K. Zhu, S. O’Brien, “Chunkrag: Novel llm-chunk filtering method for rag systems,” *arXiv preprint arXiv:2410.19572v4*, Nov. 2024, Accessed: Dec. 6, 2024.
- [26] X. Ma, T. Teofili, J. Lin, “Anserini gets dense retrieval: Integration of lucene’s hnsw indexes,” *arXiv preprint arXiv:2304.12139v1*, Apr. 2023, Accessed: Dec. 6, 2024.
- [27] K. Wang, N. Thakur, N. Reimers, I. Gurevych, “Gpl: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval,” *arXiv preprint arXiv:2112.07577v3*, Apr. 2022, Accessed: Dec. 10, 2024.
- [28] ytu-ce-cosmos, *Turkish-llama-8b-dpo-v0.1*, [Online], Available: <https://huggingface.co/ytu-ce-cosmos/Turkish-Llama-8b-DPO-v0.1>, 2024.

- [29] ytu-ce-cosmos, *Turkish-llama-8b-instruct-v0.1*, [Online], Available: <https://huggingface.co/ytu-ce-cosmos/Turkish-Llama-8b-Instruct-v0.1>, 2024.
- [30] H. T. Kesgin *et al.*, “Optimizing large language models for turkish: New methodologies in corpus selection and training,” *2024 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 2024, pp. 1–6.
- [31] Google, *Gemma 2.9b it*, [Online], Available: <https://huggingface.co/google/gemma-2-9b-it>, 2024.
- [32] Eurdem, *Defne-llama3.1-8b*, [Online], Available: <https://huggingface.co/Eurdem/Defne-llama3.1-8B>, 2024.
- [33] NousResearch, *Meta-llama-3-8b*, [Online], Available: <https://huggingface.co/NousResearch/Meta-Llama-3-8B>, 2024.
- [34] Mistralai, *Mistral-7b-instruct-v0.2*, [Online], Available: <https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2>, 2024.

PUBLICATIONS FROM THE THESIS

Conference Papers

1. D. Tanyildiz, S. Ayvaz, M. F. Amasyali, “Enhancing Retrieval-Augmented Generation Accuracy with Dynamic Chunking and Optimized Vector Search”, 4th International Conference on Design, Research and Development (RDCONF 2024), 19-20 December 2024. <https://doi.org/10.56038/oprd.v5i1.516>

