ANKARA SCIENCE UNIVERSITY, RESEARCHER
Vol. 5, No. 1, July 2025
e-ISSN: 2717-9494
Research Article/Araştırma Makalesi
Doi: https://doi.org/

Bridging the Language Gap in RAG: A Case Study on Turkish Retrieval and Generation

Erdoğan BIKMAZ[1*], Mohammed BRIMAN[2], Serdar ARSLAN[3]
[1]Çankaya University, Department of Computer Engineering, Ankara, Türkiye; https://orcid.org/0009-0006-7147-4539
[2]SmartICT Bilişim A.Ş., Ankara, Türkiye; https://orcid.org/0009-0000-5785-6916
[3]Çankaya University, Department of Computer Engineering, Ankara, Türkiye; https://orcid.org/0000-0003-3115-0741

* Corresponding Author: ebikmaz@smartict.com.tr

Received: 14 May 2025; Accepted: 22 May 2025

Reference/Atıf: E. Bikmaz, M. Briman, ve S. Arslan, "Bridging the Language Gap in RAG: A Case Study on Turkish Retrieval and Generation", Researcher, c. 05, sy. 01, ss. 38–49.

Abstract

With the rise of Large Language Models (LLMs) and LLM-based Retrieval-Augmented Generation (RAG) systems, there is a high demand for developing RAG applications that utilize LLM reasoning capabilities for handling intensive text systems in multilingual settings. However, RAG components are primarily developed for the English language, which hinders their ability to retrieve and construct precise multilingual information for LLMs to answer, especially for the Turkish language. In this work, we aim to explore the effects of developing comprehensive RAG systems that handle Turkish question-answer retrieval and generation tasks. We experiment with fine-tuning two major components on Turkish data: the embedding model used for data ingestion and retrieval, and a reranker model that ranks the retrieved documents based on their relevance to a query. We evaluate four RAG systems using six evaluation metrics. Experimental results show that fine-tuning retrieval components on Turkish data improves the accuracy of LLM responses and leads to improved context construction.

Keywords: retrieval augmented generation, large language models, embedding, information retrieval

1. Introduction

Large Language Models (LLMs) have revolutionized the landscape of Natural Language Processing (NLP) due to their ability to perform complex tasks simply by being instructed through prompts. The Transformer architecture [1] enabled the emergence of capable LLMs such as GPT-4 [2], Claude 3.7 Sonnet [3], LLaMA models [4], Mistral [5], and Phi [6]. Although LLMs have demonstrated powerful reasoning capabilities through various prompting techniques [7, 8], they still suffer from hallucination problems [9]—that is, they may generate false information if the input prompt requests content that does not exist in the LLMs' training data. Thus, the hallucination problem raises concerns about the reliability of LLMs when deployed in real-world applications.

Several approaches have been proposed to reduce LLM hallucinations, with one of the most prominent and widely adopted being Retrieval Augmented Generation (RAG) [10]. RAG aims to mitigate hallucinations by incorporating external information into the prompt as "context." This added context helps the LLM provide answers involving out-of-domain knowledge. A typical RAG system consists of two main components: a retriever, which retrieves relevant embeddings [11] of context documents based on the embedding of a user query; and a generator, which uses the retrieved context along with the query to generate a response. The first stage in developing a RAG system is data ingestion, where large raw texts are split into smaller chunks, embedding vectors [11] are generated for each chunk, and these vectors are stored in a Vector Database (VDB) [12] for later semantic retrieval [13]. Semantic search refers to a similarity-based retrieval operation between the embedding vector of a user query and the document embeddings stored in the VDB. Therefore, the quality of the embedding vectors and the degree of contextual preservation between text chunk embeddings significantly influence the overall performance of a RAG system.

One of the biggest obstacles in deploying RAG applications is the limited multilingual representation for languages other than English [14]. There is a growing demand for developing RAG systems tailored to multilingual settings, particularly for the Turkish language. Although significant progress has been made in multilingual LLMs [15-16], there remains a clear lack of high-quality Turkish language support in retrieval components and embedding models. Furthermore, another critical limitation lies in the development of retrieval components that are both multilingual and capable of handling multiple domains [17].

In this work, we aim to improve the performance of RAG systems on Turkish data by fine-tuning two retrieval components: the embedding model and the reranker. Additionally, we experiment with a prepositional chunking method [18] to enhance the quality and contextual relevance of the retrieved chunks.

Our contributions can be summarized as follows:

- We fine-tune two retrieval components—a sentence embedding model and a cross-encoder reranker—on Turkish data.
- We develop four distinct RAG systems to evaluate their performance with the fine-tuned retrieval components.
- We compute six different metrics for each of the four RAG systems to assess various aspects of their performance.

2. Related Work

Examples of multilingual and domain adaptation in LLM-based RAG applications include the evaluation of RAG systems for health-related chatbots in Indian languages, which involved analyzing the performance of several multilingual LLMs [19]. However, that experiment employed a standard embedding model from OpenAI. Xu et al. [20] adapted RAG to a Chinese medical analysis task using a two-stage retrieval process and a specialized Chinese text segmentation method. For semantic search, they utilized a QWEN-based LLM [16].

Retrieval Augmented Fine-Tuning (RAFT) [21] combines supervised fine-tuning with RAG to adapt systems to domain-specific knowledge. Rameel et al. [22] investigated RAG improvements in multilingual settings for real-world applications. Specifically, they compared paragraph-based, semantic-unit-based, topic modeling, and entity-based text splitting methods. Their findings highlighted the importance of balancing chunk size and overlap to preserve relevant information.

There are multiple methods for training and adapting embedding models to support multilingual capabilities. Early word embedding approaches explored various algorithms, such as adversarial training and pseudo-supervised refinement, to improve multilingual word-level embeddings [23]. However, the dominant approach for RAG systems is the use of full-sentence embeddings generated by Transformer-based pre-trained encoder models. Notable works in sentence embeddings include the early multilingual Universal Sentence Encoder developed by Yang et al. (2019) [24], which included Turkish in its training dataset, as well as multilingual SBERT models [11].

There are several multilingual embedding models categorized by parameter count, embedding dimensionality, and performance on the MTEB leaderboard [25-26]. Among the leading models is MiniLM [27], which uses a knowledge distillation teacher–student approach [28], applying distillation on the last layer of the Transformer teacher model to learn embedding representations. MiniLM includes multilingual data from a wide range of languages. The General Text Embedding (GTE) family of models [29] employs multi-stage contrastive learning across a diverse set of datasets. GTE models are based on a 110M-parameter BERT backbone [30], and several variants support up to 70 languages. Specifically for retrieval and RAG use cases, Wang et al. (2022) [31] introduced the e5 family of embedding models, which are tailored for RAG-style retrieval. These models use special input prefixes—“query:” for user queries and “passage:” for context documents. Additionally, e5 models are trained on multilingual datasets covering approximately 27 languages.

Reranker models are typically based on BERT-style cross-encoder architectures [30], which capture semantic relationships between queries and documents. One example is the Standalone Neural Reranking Model (SNRM), developed by Zamani et al. [32], which uses high-dimensional sparse representations for query–document pairs to perform retrieval. ColBERT, proposed by Khattab et al. [33], introduced a BERT-based reranker that employs contextualized late interaction between queries and documents to improve reranking efficiency. For fine-tuning reranker models for specific tasks, Moreira et al. [34] from NVIDIA explored the fine-tuning of both cross-encoders and decoder-based rerankers. Their results showed that fine-tuning decoder-based reranker models leads to improved accuracy compared to baseline rerankers. The development of Turkish BERT models by Kesgin et al. [35] enabled further exploration of multilingual adaptations. A notable example is the cross-encoder Turkish reranker model "turkish-colbert" [36], which was fine-tuned from a Turkish BERT cross-encoder using a Turkish-translated version of the MS MARCO dataset [37].

3. Methodology

We aim to improve the performance of RAG systems in the Turkish language by focusing on maximizing the relevance of the contextual information generated by the RAG pipeline. To achieve this, we experiment with three key components of the RAG architecture: (1) fine-tuning an embedding model on Turkish sentence pairs, (2) fine-tuning a reranker model on Turkish query–document pairs, and (3) applying a prepositional chunking strategy during the data ingestion stage to further enrich document context. We develop four distinct RAG systems, each incorporating a different combination of these components.

3.1 Fine-tuning an Embedding Model

We fine-tuned an embedding model to adapt a pre-trained multilingual model for encoding Turkish language data, enabling it to effectively embed novel and domain-specific sentences. We used an open Turkish RAG dataset containing 6,000 source documents and corresponding QA pairs for each document [38]. The significance of the QA pairs lies in the availability of ground-truth answers for each question related to the source documents.

Our embedding model of choice was multilingual-e5-large [31], selected for its multilingual capabilities, compact 384-dimensional embeddings, and suitability for retrieval tasks. The fine-tuning process involved several data preprocessing stages, including text cleaning, normalization, punctuation removal, and structuring the dataset into the required ["question," "context"] pair format for a RAG system. Additionally, model-specific prefixes were added to distinguish queries from contexts. We employed Multiple Negative Ranking as the loss function and Adam as our optimizer. Training parameters are summarized in Table 1. Finally, we will refer to our model as multilingual-e5-tr-rag.

Table 1. Fine-tuning hyperparameters for the embedding model

Parameter	Value
Base Model	multilingual-e5-large
Embedding dim	384
Training data size	5399
Learning rate	2e-5
Loss function	Multiple Negatives Ranking Loss
Evaluation metric	Recall@k (k=10)

3.2 Fine-tuning a Reranker Model

We fine-tuned a cross-encoder reranker model on a Turkish RAG dataset to improve the accuracy of reranking Turkish documents. For this purpose, we selected the "jina-reranker-v2-base-multilingual"

reranker model developed by [39] as our base model due to its multilingual capabilities. Additionally, we fine-tuned the model using resources provided by the sentence-transformer library [11].

To prepare the reranking dataset, we used the queries and documents from the original dataset [38] and applied the "Hard Negative Mining" (HNM) process [40], which selects documents that may appear to be relevant to a given query but are not. This process improves the reranker model's ability to differentiate between relevant and irrelevant query-document pairs. The resulting dataset consists of (query, document, label) triples, with a binary label column indicating whether the query-document pairs are relevant.

To evaluate the fine-tuning results, we utilized the BEIR retrieval evaluation benchmark [41]. Our fine-tuned model will be referred to as jina-reranker-multilingual-wiki-tr-rag.

Table 2. Fine-tuning hyperparameters for the reranking model

Parameter	Value
Base Model	jina-reranker-v2-base-multilingual
Embedding dim	1024
HNM training data size	26004
Training data size	5399
Learning rate	2e-5
Loss function	Binary Cross Entropy Loss
Evaluation metric	Cross-Encoder Nano BEIR Evaluator

3.3 Evaluation Metrics

This experiment aims to evaluate the quality of RAG systems in terms of generating factual answers and measuring the utilization of retrieved context. We assessed each of the four RAG systems from three perspectives: the relationship between the generated LLM answer and the retrieved contexts, the relevance between the queries and the retrieved contexts, and the similarity between the LLM-generated answer and the ground truth answer. We used RAG-specific metrics from the Retrieval Augmented Generation Assessment (RAGAS) [42] and standard NLP metrics such as ROUGE-N [43] and BERTScore [44].

To assess the similarity between the LLM-generated answers and the ground truth, we employed ROUGE-N and BERTScore. Both ROUGE-N and BERTScore compute recall, precision, and F1 scores between two texts (e.g., LLM-generated answer and ground truth). ROUGE-N calculates the similarity based on overlapping n-gram units, while BERTScore uses BERT-based embeddings to evaluate similarity, considering semantic meaning.

For RAG-specific evaluations, we utilized four RAGAS metrics: Faithfulness, Answer Relevance, Context Recall, and Context Precision.

- **Faithfulness**: Faithfulness measures the factual consistency between the LLM-generated answer and the retrieved context.
- **Answer Relevance**: This metric evaluates the relevance between the LLM-generated answer and the original question.
- **Context Recall**: Context Recall assesses whether the necessary documents have been retrieved to answer the question. It is computed by comparing the retrieved context to the ground truth.
- **Context Precision**: Context Precision measures the signal-to-noise ratio between the question and the retrieved context.

3.3 RAG Systems

We developed a RAG system [10] consisting of two main stages: the data ingestion stage, where data undergoes a splitting process to break down large texts into smaller, manageable chunks, and the retrieval and generation stage, where a user query triggers a retrieval process to fetch the most relevant text chunks. All information is placed into a prompt template containing both the original user query and the retrieved chunks (see the prompt template in Table A1). The final prompt is then passed to an LLM for generating the answer. The RAG setup is illustrated in Figure 1.

In the data ingestion stage, we used both Recursive and Prepositional chunking methods in our RAG systems to compare their performance. We compared the effectiveness of a simple recursive separator-based splitting method with an LLM-based chunking method. Our goal was to explore how much context is preserved between the two methods and how the resulting text chunks impact the quality of the retrieval stage.

Recursive chunking [45] is a standard text-splitting method that uses separators in its process. It splits the text recursively until the smallest possible sentence is reached. However, due to its simplicity, some chunks may lose important context.

Prepositional chunking [18] is a model-based chunking method in which an LLM is prompted to remove ambiguity from a piece of text by following a set of chunking instructions. As the name suggests, prepositional chunking creates factual, concise, and self-contained sentences, with the aim of improving retrieval results and helping LLMs generate more accurate responses. An example is given in Table 3.

Table 3. Prepositional chunking applied to a piece of Turkish text

Original Chunk	Prepositional Chunks
San Miguel de Allende, Meksika'nın iç bölgelerinde yer alan Guanajuato Eyaleti'nin doğu kesiminde bulunan bir şehir ve belediyedir. Bajío'nun makrobölgelerinin bir parçasıdır. Meksiko'dan 274 km (170 mi), Guanajuato Eyaleti'nin başkentinden 97 km (60 mi) uzaklıkta yer alır. Eski adı San Miguel el Grande olan şehir, 1826'da yapılan değişikle Ignacio Allende'nin anısına günümüzdeki adı almıştır.	San Miguel de Allende, Meksika'nın iç bölgelerinde yer almaktadır. San Miguel de Allende, bir şehir ve belediyedir. San Miguel de Allende, Bajío'nun makrobölgelerinin bir parçasıdır.

We applied prepositional chunking to our documents using "gemini-2.0-flash-lite" [46] as our LLM of choice due to its multilingual capabilities. Table 3 provides an example of prepositional chunking applied to a Turkish text. The prompt used to apply the chunking process is provided in Table A1 in Appendix A.

To experiment with the effects of embedding and reranker models on the performance of a RAG system, we developed four different versions, each employing a combination of text splitting and embedding models. We use two different text splitting methods: a simple recursive method with a defined set of separators, and a prepositional chunking method that utilizes an LLM to split the text into self-contained, factual sentences. Additionally, we use "gemma-3-27b-it" [47] as the primary generation LLM due to its multilingual capabilities.

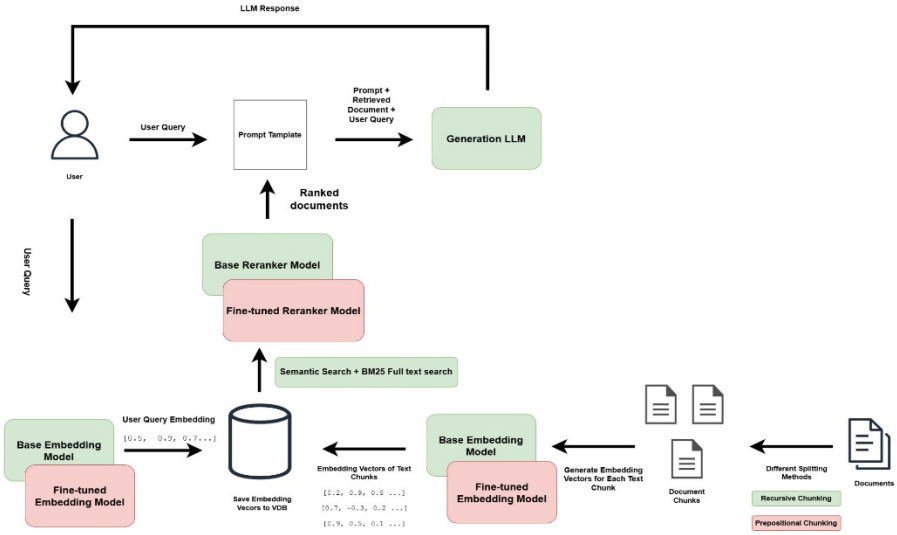

Figure 1. RAG system overview. The green boxes show the base version of each component and the pink boxes show our custom componants.

We used a hybrid search pipeline for the retrieval process, combining a cosine similarity-based semantic search retriever with a BM25 keyword retriever [48]. All vector embeddings were stored in a Chroma DB vector database [49].

The configurations of the RAG systems are as follows:

- **Base RAG:** Recursive chunking + paraphrase-multilingual-MiniLM-L12-v2 base embedding model [50] + semantic search + ColBERTv2.0 reranking [33].
- **RAG V1:** Recursive chunking + fine-tuned multilingual-e5-tr-rag embedding model [50] + hybrid search + ColBERTv2.0 reranking [33].
- **RAG V2:** Recursive chunking + fine-tuned multilingual-e5-tr-rag embedding model [50] + hybrid search + fine-tuned jina-reranker-multilingual-wiki-tr-rag reranking.
- **RAG V3:** Prepositional chunking + fine-tuned multilingual-e5-tr-rag embedding model [50] + hybrid search + fine-tuned jina-reranker-multilingual-wiki-tr-rag reranking.

The RAG evaluation dataset consists of 600 samples taken from the test set of the Turkish RAG dataset [38]. Each sample includes three columns: Context, Question, and Answer. The Context represents the source document, the Question is a query related to the Context, and the Answer is the corresponding ground-truth response.

4. Experimentation Setup

In terms of datasets, we use 600 test samples from the Turkish RAG dataset [38]. For evaluation purposes, the dataset was transformed into an evaluation format consisting of the following columns:

- **Question:** Questions about the information presented in the source documents.
- **Answer:** The LLM-generated answer from the RAG system.
- **Contexts:** The set of retrieved documents via Semantic Search.
- **Ground Truth:** The ground truth answers present in the dataset.

Each of the four metrics presented in Section 3.3—namely, Faithfulness, Answer Relevance, Context Recall, Context Precision, ROUGE-N, and BERTScore—is applied to the outputs of each of the four RAG systems.

4. Experimentation Results and Analysis

By examining the results in Figure 2, we observe that RAG V2 outperforms the other three configurations, including the baseline RAG system. Notably, RAG V2 also produces higher-quality LLM-generated responses, as indicated by the ROUGE-N and BERTScore metrics. These findings suggest that fine-tuning retrieval components—such as embedding models and rerankers—on domain-specific language significantly enhances the retrieval effectiveness in RAG systems. Consequently, the improved relevance and quality of the retrieved context lead to more accurate and informative responses generated by the LLM.

Interestingly, Figure 2 also reveals that RAG V3, which incorporates the prepositional chunking method, performs worse than both the baseline and the other RAG configurations. We hypothesize that the use of short, self-contained sentences—while intended to enhance clarity—may negatively impact the richness of the contextual information captured by the embedding vectors. As a result, this could lead to suboptimal retrieval, with relevant documents being missed due to insufficient contextual cues.

An examination of the RAGAS Faithfulness scores [42] for all four systems, as presented in Figure 2, reveals that RAG V3 exhibits the lowest level of factual consistency between the LLM-generated responses and the retrieved context. This metric, which measures the degree to which the generated answer aligns with the retrieved supporting evidence, indicates that the prepositional chunking method used in RAG V3 may hinder the model's ability to ground its responses in contextually accurate information compared to the other configurations.

$$Faithfulness\ Score = \frac{\text{Number of claims in the response supported by the retrieved context}}{Total\ number\ of\ claims\ in\ the\ response} \quad (1)$$

Tables A2 and A3 in Appendix A illustrate the outputs of the RAG V2 and RAG V3 systems for a representative sample from our dataset [38]. In Table A2, RAG V2 achieves a Faithfulness score of 1.0, with the response explicitly grounded in the first retrieved chunk—denoted by "(1)" —demonstrating effective alignment between context and answer. The recursive chunking method employed by RAG V2 produced semantically rich and coherent chunks, enabling the LLM to generate a well-supported response. In contrast, Table A3 shows that RAG V3, which utilized prepositional chunking, retrieved short, atomic sentences that often lacked sufficient contextual information. Although prepositional chunking aims to produce concise and self-contained units, in this case it resulted in lower-quality context that negatively affected the LLM's output. This outcome highlights that, despite the use of fine-tuned retrieval components, the quality of the initial chunking strategy can significantly influence the overall performance of a RAG system.

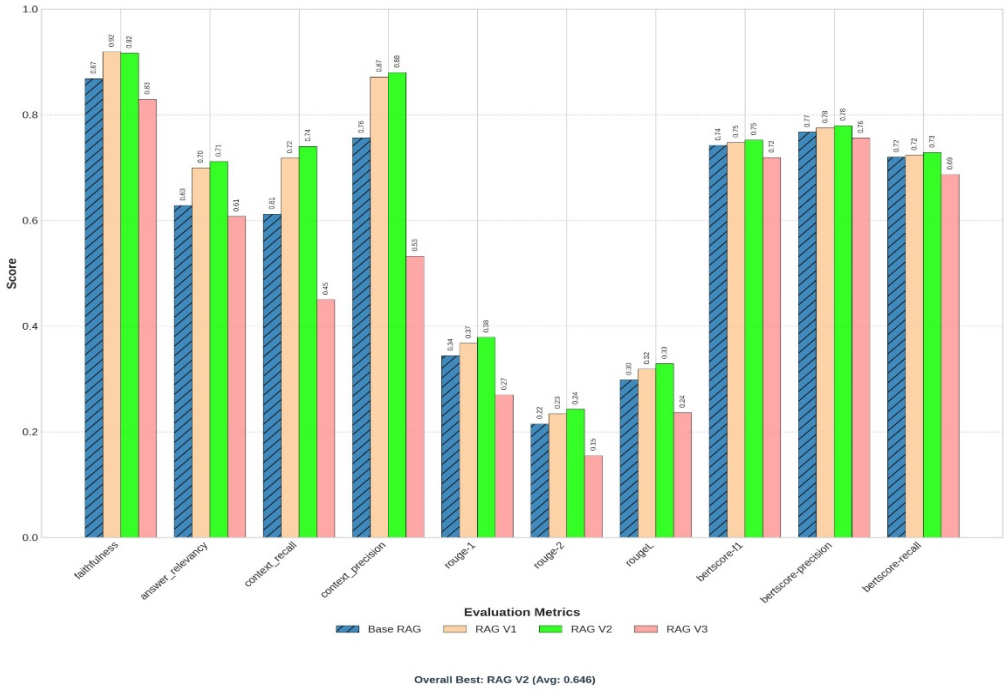

Figure 2. RAGAS, ROUGE-N, and BERTScore metric results for each RAG system

6. Conclusion

In this work, we explored the impact of fine-tuning retrieval components for a domain-specific language, such as Turkish, on improving both retrieval performance and LLM-generated response quality. We fine-tuned a multilingual embedding model and a reranker model on Turkish text with the goal of enhancing both embedding quality and reranking effectiveness. Additionally, we examined the effectiveness of a novel chunking method called prepositional chunking. To evaluate the impact of these components, we developed four different RAG systems and assessed their performance using six key metrics: faithfulness, answer relevance, context recall, context precision, ROUGE-n, and BERTScore. Our experimental results demonstrate that further adapting multilingual base models to domain-specific languages significantly improves retrieval performance on the target tasks. We show that further fine-tuning multilingual embedding and reranking models drastically aligns the embedding space to our specific Turkish language domain and enhances the retrieval quality for better LLM generation. Moreover, we demonstrate that using a standard chunking method with custom fine-tuned retrieval components performs better than using LLM-based chunking methods such as prepositional chunking. Future studies will focus on combining both embedding (bi-encoders) and reranking (cross-encoder) architectures to create fusion retrieval and reranking models for improving the accuracy of retrieval components for the Turkish Language.

Appendix A: Tables and Figures.

Table A1. Prepositional chunking prompt in Turkish

Prepositional Chunking Prompt
Lütfen aşağıdaki metni basit, kendi içinde anlamlı önermelere ayırın. Her bir önerme aşağıdaki kriterleri karşılamalıdır:

1. Tek Bir Gerçeği İfade Etmeli: Her önerme, belirli bir gerçeği veya iddiayı ifade etmelidir.
2. Bağlama İhtiyaç Duymadan Anlaşılır Olmalı: Önerme, başka bir bağlama ihtiyaç duymadan anlaşılır olmalıdır.
3. Zamir Değil, Tam İsimler Kullanılmalı: Zamirler veya belirsiz ifadeler yerine tam varlık isimleri kullanılmalıdır.
4. İlgili Tarih/Sıfatları İçermeli: Uygunsa, gerekli tarih, zaman ve niteleyiciler eklenerek önerme kesin hale getirilmelidir.
5. Tek Özne-Yüklem İlişkisi İçermeli: Tek bir özne ve ona karşılık gelen bir eylem veya özellik üzerine odaklanılmalı; bağlaçlar veya birden fazla yargı içermemelidir.

Table A2. Faithfulness Evaluation Example for RAG V2 sample

Parameter	Value
User Input	Havayolu Serbestleştirme Kanunu, havacılık endüstrisinde hangi değişiklikleri beraberinde getirdi?
Response	Havayolu Serbestleştirme Kanunu, bilet fiyatları, rotalar ve yeni havayollarının pazara girişi üzerindeki devlet kontrolünü kaldırdı ve yolcuları piyasa güçlerine maruz bıraktı (1).
Retrieved Contexts	Airline Deregulation Act (Türkçe: Havayolu Serbestleştirme Kanunu), 1978'de Amerika Birleşik Devletleri başkanı Jimmy Carter tarafından imzalanan, ticari havacılık'daki bilet fiyatları, yolları ve yeni havayolların pazara girişi üzerindeki devlet kontrolünü kaldırma amaçlı federal kanun. Civil Aeronautics Board'un denetleme gücü yavaş yavaş azaltılmış, sonunda havacılık endüstrisinde yolcuları piyasa güçlerine maruz bırakmıştır. Fakat kanun, Federal Aviation Administration'ın (FAA) tüm hava emniyet yönleri. Pseudostaurastrum, ışınlılar grubunda yer alan bir Protista. Aralarında 3-4 boşluk olan her bir pseudostaurastrum'un boşlukları birbirine eşittir. Enternasyonal turuncu havacılık endüstrisinde nesneleri çevresinden ayırt etmek için kullanılan, güvenlik turuncusuna benzeyen, ama daha derin ve kırmızı bir tona sahip olan renktir. Liberizm ya da özgürcülük, filozof Benedetto Croce tarafından kullanılan ve İtalyan-Amerikan siyaset bilimci Giovanni Sartori tarafından İngilizce olarak popüler olan laissez-faire kapitalizminin ekonomik doktrini için kullanılan bir terimdir. Ekonomik doktrin laissez-faire kapitalizmini ima eden terim, filozof Benedetto Croce tarafından ilk kez kullanılmıştır ve siyaset bilimci Giovanni Sartori tarafından popülarize edilmiştir. Kavram İtalyanca liberismo kelimesinden türetilmiştir.
Faithfulness	1.0

Table A3. Faithfulness Evaluation Example for RAG V3 sample

Parameter	Value
User Input	Havayolu Serbestleştirme Kanunu, havacılık endüstrisinde hangi değişiklikleri beraberinde getirdi?
Response	Havayolu Serbestleştirme Kanunu, bilet fiyatları, rotalar ve yeni havayollarının pazara girişi üzerindeki devlet kontrolünü kaldırdı (1).
Retrieved Contexts	Airline Deregulation Act, havacılık endüstrisinde yolcuları piyasa güçlerine maruz bırakmıştır. Airline Deregulation Act, bilet fiyatları, yollar ve yeni havayollarının pazara girişi üzerindeki devlet kontrolünü kaldırmayı amaçlamaktadır. Airline Deregulation Act, 1978'de Amerika Birleşik Devletleri başkanı Jimmy Carter tarafından imzalanmıştır. Enternasyonal turuncu, havacılık endüstrisinde nesneleri çevresinden ayırt etmek için kullanılan bir renktir. Türk Kadastro Kanunu, kadastroyu tanımlar.
Faithfulness	0.0

Contribution of Researchers

Erodoğan BIKMAZ: Implementation, software, evaluation, and writing.

Mohammed BRIMAN Implementation, software, evaluation, and writing.

Serdar ARSLAN: Review and editing.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] F. J. Cazorla *et al.*, "PROXIMA: Improving Measurement-Based Timing Analysis through Randomisation and Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L ., Polosukhin, I.: Attention is All you Need. In: Conference on Neural Information Processing Systems (NeurIPS) (2017). https://papers.nips.cc/paper*files/paper/*2017*/hash/*3*f*5*ee*243547*dee*91*fbd*053*c*1*c*4*a*845*aa*− *Abstract.html*

[2] OpenAI: GPT-4 Technical Report. arXiv preprint (2023). https://arxiv.org/abs/ 2303.08774

[3] Unknown: Introducing Claude 3.5 Sonnet. https://www.anthropic.com/news/ claude-3-5-sonnet. No date provided in citation (n.d.)

[4] Grattafiori: The Llama 3 herd of models. arXiv preprint (2024). https://arxiv. org/abs/2407.21783

[5] Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., De Las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L.R., Lachaux, M.A., Stock, P., Scao, T.L., Lavril, T., Wang, T., Lacroix, T., Sayed, W.E.: Mistral 7B. arXiv preprint (2023). https://arxiv.org/abs/2310. 06825

[6] Abdin, M., Aneja, J., Behl, H., Bubeck, S., Eldan, R., Gunasekar, S., Harrison,

M., Hewett, R.J., Javaheripi, M., Kauffmann, P., Lee, J.R., Lee, Y.T., Li, Y., Liu, W., Mendes, C.C.T., Nguyen, A., Price, E., Gustavo, D.R., Saarikivi, O., Salim, A., Shah, S., Wang, X., Ward, R., Wu, Y., Yu, D., Zhang, C., Zhang, Y.: PHI-4 Technical Report. arXiv preprint (2024). https://arxiv.org/abs/2412.08905

[7] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., Zhou, D.: Chain-of-Thought prompting elicits reasoning in large language models. arXiv preprint (2022). https://arxiv.org/abs/2201.11903

[8] Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T.L., Cao, Y., Narasimhan, K.: Tree of Thoughts: Deliberate Problem Solving with Large Language Models. arXiv preprint (2023). https://arxiv.org/abs/2305.10601

[9] Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., Liu, T.: A survey on hallucination in large language models: principles, taxonomy, challenges, and open questions. ACM Transactions on Office Information Systems (2024) arXiv:2311.05232. Preprint available at https://arxiv.org/abs/2311.05232

[10] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Ku¨ttler, H., Lewis, M., Yih, W.T., Rockt¨aschel, T., Riedel, S., Kiela, D.: Retrieval-Augmented Generation for Knowledge-Intensive NLP tasks. arXiv preprint (2020). https:// arxiv.org/abs/2005.11401

[11] Reimers, N., Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv preprint (2019). https://arxiv.org/abs/1908.10084

[12] Jegou, H., Douze, M., Johnson, J.: Faiss: A library for efficient similarity search. Facebook Engineering blog. Published March 29, 2017, but cited as 2018 (2018). https://engineering.fb.com/2017/03/29/data-infrastructure/ faiss-a-library-for-efficient-similarity-search/

[13] Unknown: What is Semantic Search? https://cohere.com/llmu/ what-is-semantic-search. No date provided in citation (n.d.)

[14] Ahmad, S.R.: Enhancing multilingual information retrieval in mixed human resources environments: a RAG model implementation for multicultural enterprise. arXiv preprint (2024). https://arxiv.org/abs/2401.01511

[15] Ustu¨n, A., Aryabumi, V., Yong, Z.X., Ko, W.Y., D'souza, D., Onilude, G., Bhan-¨ dari, N., Singh, S., Ooi, H.L., Kayid, A., Vargus, F., Blunsom, P., Longpre, S., Muennighoff, N., Fadaee, M., Kreutzer, J., Hooker, S.: Aya model: An Instruction finetuned Open-Access Multilingual language model. arXiv preprint (2024). https://arxiv.org/abs/2402.07827

[16] Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., al.: QWEN2 Technical Report. arXiv preprint (2024). https://arxiv.
org/abs/2407.10671

[17] Chirkova, N., Rau, D., D´ejean, H., Formal, T., Clinchant, S., Nikoulina, V.: Retrieval-augmented generation in multilingual settings. arXiv preprint (2024). https://arxiv.org/abs/2407.01463

[18] Chen, W.H.C.S.Y.W.M.K.Z.X.Z.H..Y.D. T.: Dense X retrieval: What retrieval granularity should we use? arXiv preprint (2023). https://arxiv.org/abs/2312. 06648

[19] Gumma, V., Raghunath, A., Jain, M., Sitaram, S.: HEALTH-PARIKSHA: Assessing RAG models for health chatbots in Real-World multilingual settings. arXiv preprint (2024). https://arxiv.org/abs/2410.13671

[20] Xu, P., Wu, H., Wang, J., Lin, R., Tan, L.: Traditional Chinese Medicine Case Analysis System for High-Level Semantic Abstraction: Optimized with Prompt and RAG. arXiv preprint (2024). https://arxiv.org/abs/2411.15491

[21] Zhang, T., Patil, S.G., Jain, N., Shen, S., Zaharia, M., Stoica, I., Gonzalez, J.E.: RAFT: Adapting Language Model to Domain Specific RAG. arXiv preprint (2024). https://arxiv.org/abs/2403.10131

[22] Zhang, T., Patil, S.G., Jain, N., Shen, S., Zaharia, M., Stoica, I., Gonzalez, J.E.: RAFT: Adapting Language Model to Domain Specific RAG. arXiv preprint. This
uses the same source as the 'raft' entry, labelled 'chunking' in the prompt list. (2024). https://arxiv.org/abs/2403.10131

[23] Chen, X., Cardie, C.: Unsupervised multilingual word embeddings. arXiv preprint (2018). https://arxiv.org/abs/1808.08933

[24] Yang, Y., Cer, D., Ahmad, A., Guo, M., Law, J., Constant, N., Abrego, G.H., Yuan, S., Tar, C., Sung, Y.H., Strope, B., Kurzweil, R.: Multilingual universal sentence encoder for semantic retrieval. arXiv preprint (2019). https://arxiv.org/abs/1907.04307

[25] Muennighoff, N., Tazi, N., Magne, L., Reimers, N.: MTEB: Massive Text Embedding Benchmark. arXiv preprint (2022). https://arxiv.org/abs/2210.07316

[26] Unknown: MTEB: Massive Text Embedding Benchmark. https://huggingface.co/ blog/mteb. No date provided in citation (n.d.)

[27] Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., Zhou, M.: MINILM: Deep
Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. arXiv preprint (2020). https://arxiv.org/abs/2002.10957

[28] Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint (2015). https://arxiv.org/abs/1503.02531

[29] Li, Z., Zhang, X., Zhang, Y., Long, D., Xie, P., Zhang, M.: Towards General Text Embeddings with Multi-stage Contrastive Learning. arXiv preprint (2023). https://arxiv.org/abs/2308.03281

[30] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint (2018). https://arxiv.org/abs/1810.04805

[31] Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., Wei, F.: Text embeddings by Weakly-Supervised contrastive pre-training. arXiv preprint (2022). https://arxiv.org/abs/2212.03533

[32] Hamed, Z., Mostafa, D., Bruce, C.W., Erik, L.-M., Jaap, K.: From Neural ReRanking to Neural Ranking learning a sparse representation for inverted indexing. ACM Proceedings, 497–506 (2018)

[33] Khattab, O., Zaharia, M.: ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. arXiv preprint (2020). https://arxiv.
org/abs/2004.12832

[34] De Souza P Moreira, G., Ak, R., Schifferer, B., Xu, M., Osmulski, R., Oldridge, E.: Enhancing QA Text Retrieval with Ranking Models: Benchmarking, fine-tuning and deploying Rerankers for RAG (2024). https://arxiv.org/abs/2409.07691

[35] Kesgin, H.T., Yuce, M.K., Amasyali, M.F.: Developing and evaluating tiny to medium-sized turkish bert models. arXiv preprint arXiv:2307.14134 (2023)

[36] ytu-ce-cosmos/turkish-colbert· Hugging Face. https://huggingface.co/ ytu-ce-cosmos/turkish-colbert

[37] Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., Rosenberg, M., Song, X., Stoica, A., Tiwary, S., Wang, T.: MS MARCO: A Human Generated MAchine Reading
COmprehension Dataset (2016). https://arxiv.org/abs/1611.09268

[38] MeTin/WIKIRAG-TR · Datasets at Hugging Face. https://huggingface.co/ datasets/Metin/WikiRAG-TR

[39] Gu¨nther, M., Ong, J., Mohr, I., Abdessalem, A., Abel, T., Akram, M.K., Guzman, S., Mastrapas, G., Sturua, S., Wang, B., Werk, M., Wang, N., Xiao, H.: JINA Embeddings 2: 8192-Token General-Purpose text embeddings for long documents (2023). https://arxiv.org/abs/2310.19923

[40] Robinson, J., Chuang, C.-Y., Sra, S., Jegelka, S.: Contrastive Learning with Hard Negative Samples (2020). https://arxiv.org/abs/2010.04592

[41] Thakur, N., Reimers, N., Ru¨ckl´e, A., Srivastava, A., Gurevych, I.: BEIR: a heterogenous benchmark for zero-shot evaluation of information retrieval models (2021). https://arxiv.org/abs/2104.08663

[42] Es, S., James, J., Espinosa-Anke, L., Schockaert, S.: RAGAS: Automated Evaluation of Retrieval Augmented Generation. arXiv preprint (2023). https://arxiv. org/abs/2309.15217

[43] Lin, C.Y.: ROUGE: a package for automatic evaluation of summaries. In: Proceedings of the Meeting of the Association for Computational Linguistics, Barcelona, Spain, pp. 74–81 (2004)

[44] Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: BERTScore: Evaluating Text Generation with BERT. arXiv preprint (2019). https://arxiv.org/abs/ 1904.09675

[45] LangChain: Recursively split by character. https://python.langchain.com/v0.1/ docs/modules/data connection/document transformers/recursive text splitter/. No date provided in citation (n.d.)

[46] Gemini Team: Gemini: a family of highly capable multimodal models. arXiv preprint (2023). https://arxiv.org/abs/2312.11805

[47] Team, G. et. al.: Gemma 3 Technical Report (2025). https://arxiv.org/abs/2503.19786

[48] Amati, G.: BM25, pp. 257–260 (2009). https://doi.org/10.1007/978-0-387-39940-9\{ 9 . https://doi.org/10.1007/978-0-387-39940-9₉21

[49] Chroma-Core: GitHub - chroma-core/chroma: the AI-native open-source embedding database. https://github.com/chroma-core/chroma

[50] sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2 · Hugging Face. https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2