

MUST-RAG: MUSical Text Question Answering with Retrieval Augmented Generation

Daeyong Kwon¹, SeungHeon Doh¹, Juhan Nam¹

¹Graduate School of Culture Technology, KAIST, South Korea

ABSTRACT

Recent advancements in Large language models (LLMs) have demonstrated remarkable capabilities across diverse domains. While they exhibit strong zero-shot performance on various tasks, LLMs’ effectiveness in music-related applications remains limited due to the relatively small proportion of music-specific knowledge in their training data. To address this limitation, we propose **MusT-RAG**, a comprehensive framework based on Retrieval Augmented Generation (RAG) to adapt general-purpose LLMs for text-only music question answering (MQA) tasks. RAG is a technique that provides external knowledge to LLMs by retrieving relevant context information when generating answers to questions. To optimize RAG for the music domain, we (1) propose MusWikiDB, a music-specialized vector database for the retrieval stage, and (2) utilizes context information during both inference and fine-tuning processes to effectively transform general-purpose LLMs into music-specific models. Our experiment demonstrates that MusT-RAG significantly outperforms traditional fine-tuning approaches in enhancing LLMs’ music domain adaptation capabilities, showing consistent improvements across both in-domain and out-of-domain MQA benchmarks. Additionally, our MusWikiDB proves substantially more effective than general Wikipedia corpora, delivering superior performance and computational efficiency.

1. INTRODUCTION

Recent advancements in Large language models (LLMs) have demonstrated impressive capabilities across a wide range of tasks, thanks to their massive scale and ability to generalize across diverse domains. However, LLMs still face significant limitations in music-related applications due to the relatively small amount of music-specific knowledge in their training data. To effectively deploy general LLMs in music-related domains such as music recommendation systems and chatbots, a deep understanding of Music Question Answering (MQA) in text-only settings is essential. Mastering text-based MQA would enable LLMs to provide more accurate and contextually aware responses to user questions about music, ultimately enhancing the user experience in music-related applications. Developing a robust text-only music QA framework is therefore a key step

toward improving the adaptability of LLMs in the music domain.

Traditionally, domain adaptation of LLMs has often been achieved by fine-tuning them on domain-specific data [1–3]. However, this approach faces challenges in securing high-quality training data, and as model size increases, the training time and cost also rise significantly. Additionally, continuously updating the model with new knowledge remains a persistent challenge.

In this paper, we propose **MusT-RAG**, a framework that leverages Retrieval Augmented Generation (RAG) [4] techniques to enhance general-purpose LLMs for music-specific tasks. The core idea behind MusT-RAG is to augment LLMs with external knowledge retrieval mechanisms. Specifically, the model retrieves relevant external knowledge from a pre-constructed, comprehensive music-specific vector database in order to answer input questions.

For music-domain specific retrieval, we introduce **MusWikiDB**, which, to our knowledge, is the first comprehensively curated vector database designed specifically for music-related content. We explore various design choices for optimizing retrieval performance, including embedding models and chunking strategies. By incorporating this retrieval process, MusT-RAG enables LLMs to efficiently generate contextually relevant responses, drawing on specialized music knowledge to enhance performance on music-related tasks, all without requiring additional training. Furthermore, we extend the application of RAG beyond inference by incorporating contextual information during the fine-tuning process. Our empirical analysis reveals that this context-aware training enhances the model’s contextual understanding capabilities, defined as the ability to generate coherent and relevant text within a specific context [5], outperforming conventional fine-tuning approaches.

MusT-RAG demonstrated the effectiveness of RAG across all scenarios including in-domain and out-of-domain settings, as well as both fine-tuning and inference stages. By retrieving relevant **context** information from the database, MusT-RAG effectively addresses the music domain adaptation problem. Our contributions are as follows: *i)* We propose the **MusT-RAG** framework, which leverages RAG to retrieve relevant context from a music-specific database for answer generation. *ii)* We create **MusWikiDB**, the first comprehensive music-specific vector database for RAG. *iii)* We demonstrate that RAG-style fine-tuning can resolve the issue of decreased contextual understanding performance with conventional fine-tuning.

¹ejmj63@kaist.ac.kr

²seunghondoh@kaist.ac.kr

³juhan.nam@kaist.ac.kr

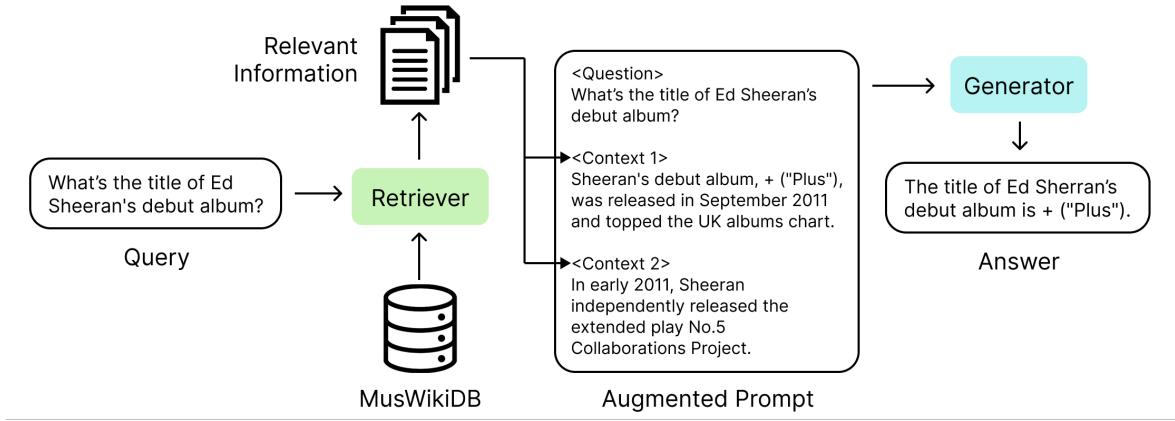


Figure 1: Overview of our **MusT-RAG** framework. The retriever searches for relevant information in **MusWikiDB** based on similarity for music-related queries, and augments the generator’s prompt with this information to generate an answer.

iv) We introduce **ArtistMus**, a benchmark designed to evaluate artist-related questions in text-only MQA tasks, addressing a gap in existing evaluations.

2. MUSIC QUESTION ANSWERING

Question Answering (QA) refers to the task of providing an appropriate answer to a given question, which is one of the Information Retrieval (IR) tasks [6]. The task is typically framed as retrieving relevant information from a collection of documents or knowledge sources to answer fact-based questions. Open-domain QA involves answering questions from a vast and varied set of topics using a large collection of general knowledge documents [7, 8]. In contrast, domain-specific QA targets specialized fields such as medicine [9], law [10], or music [11–14], where both the document set and the questions are confined to that domain. In this work, we define the MQA task as the problem of providing accurate and relevant answers to music-related questions by leveraging domain-specific musical knowledge.

Several recent studies have introduced music-related benchmarks to evaluate LLM performance. MuChoMusic [11] features 1,187 audio-based multiple-choice questions that assess both musical knowledge and reasoning capabilities. MusicTheoryBench [12] contains 372 expert-validated questions designed to evaluate advanced music knowledge and reasoning skills. TrustMus [13] comprises 400 questions across four domains—*People, Instruments and Technology, Genres, Forms, and Theory, and Culture and History*—all derived from The Grove Dictionary Online [15]. ZIQI-Eval [14] presents a comprehensive evaluation framework consisting of 14,000 comprehension tasks that span 10 major topics and 56 subtopics, encompassing a broad spectrum of music-related knowledge.

A significant shortcoming of existing benchmarks is their inadequate representation of rich metadata about tracks, artists, and albums—information crucial for everyday music listening contexts. Current text-only QA benchmarks inadequately address common music information needs, lacking comprehensive coverage of details that listeners frequently seek: complete discographies, artist col-

laboration networks, creative evolution across albums, and notable career achievements. This pronounced disparity between current MQA capabilities and the practical information demands of music consumers highlights the pressing need for benchmarks specifically designed to evaluate responses to artist-centric questions.

3. RETRIEVAL AUGMENTED GENERATION

3.1 RAG Framework

Retrieval-Augmented Generation (RAG) enhances the capabilities of LLMs by combining their generative abilities with access to external knowledge. Instead of relying solely on parametric memory, RAG retrieves relevant passages from an external database during inference time to ground responses in factual context.

3.1.1 Indexing

The first step in RAG is constructing a searchable knowledge database. This involves segmenting a large corpus into fixed-size text passages (chunking), followed by representing each passage using an embedding models. Various embedding models can be used for indexing:

Sparse Embeddings [16, 17] use term frequency-based scoring to match exact keywords, offering fast and interpretable retrieval for large-scale datasets.

Dense Embeddings [17–19] map questions and documents into a shared vector space, enabling semantic matching beyond keyword overlap.

Audio-Text Joint Embeddings [20–24] extend this further by jointly embedding text with the audio modality. By leveraging contrastive learning between audio and text, they can serve as more domain-specialized text embedding models for music-related tasks.

3.1.2 Retrieval

Formally, the retriever R is defined as a function:

$$R : (q, D) \rightarrow c$$

where q is the input question, D is the entire database of text passages, and $c \subset D$ is the filtered context consisting of the top- k passages, such that $|c| = k \ll |D|$. Each passage ¹ $p \in D$ is scored based on its similarity to the input question using cosine similarity between their embeddings:

$$\text{sim}(q, p) = \frac{E(q) \cdot E(p)}{\|E(q)\| \|E(p)\|}$$

Here, $E(\cdot)$ denotes an embedding function that maps both questions and passages into a shared vector space. The retriever ranks all passages in D by their similarity scores and selects the top- k passages to form c , which serve as the external context for the generation step.

3.1.3 Generation

The retrieved context c is provided to a generator LLM, which produces an output sequence using next-token prediction. Each token x_i is generated conditioned on the input query q , the retrieved context c , and the previously generated tokens $x_{<i}$:

$$p(x_1, \dots, x_n | q, c) = \prod_{i=1}^n p_\theta(x_i | [q, c; x_{<i}])$$

This structure enables the model to dynamically incorporate external knowledge during inference, improving factual accuracy and adaptability without retraining.

3.2 RAG vs. Fine-tuning

LLMs often struggle with specialized tasks such as MQA due to limited exposure to domain-specific knowledge during pretraining. To address this, two primary domain adaptation strategies are commonly used: fine-tuning and RAG. Fine-tuning is akin to a closed-book exam: the model internalizes domain knowledge during training and must rely solely on that knowledge at inference. While effective for learning structured formats or stylistic patterns [26, 27], it is resource-intensive and inflexible when adapting to new or frequently changing knowledge. In contrast, RAG is like an open-book exam: the model dynamically retrieves relevant information from an external knowledge source during inference. This enables LLMs to access up-to-date and specialized information without retraining. Prior studies [28, 29] show that RAG improves factual accuracy, mitigates hallucinations, and provides greater transparency by allowing source verification. It is also more scalable and economically efficient, as it does not require updating model parameters [27]. These benefits are especially useful in rapidly evolving domains like music, where new artists, compositions, and styles continuously emerge.

3.3 RAG with Fine-tuning

While fine-tuning typically relies on question-answer pairs, it does not always emphasize learning to extract relevant information from the context provided alongside the question. In standard fine-tuning, the model is trained to

directly map a question to its answer without fully leveraging any external context that might be available. As a result, the model may struggle to utilize background information effectively, especially when answering questions that require specialized or up-to-date knowledge.

To address this limitation, we adopt a RAG-style fine-tuning approach using a dataset consisting of *(context, question, answer)* triples. Unlike standard QA fine-tuning, which relies solely on the question, our method introduces an external relevant passage p for the input question q . This enables the model to learn how to incorporate relevant contextual information during answer generation. Both approaches share the same next-token prediction objective, but differ in the input they condition on. In standard fine-tuning, the model is trained as follows:

$$\mathcal{L}_{\text{QA Fine-tuning}} = - \sum_{i=1}^n \log p_\theta(x_i | [q; x_{<i}]),$$

where the model predicts each answer token x_i based only on the question and the previously generated tokens. In contrast, RAG-style fine-tuning conditions the generation not only on the question but also on the relevant passages as context:

$$\mathcal{L}_{\text{RAG Fine-tuning}} = - \sum_{i=1}^n \log p_\theta(x_i | [q, c; x_{<i}]),$$

where c is the relevant passage retrieved from an external corpus. By incorporating c as an additional context, the model is encouraged to utilize external knowledge when generating answers. This strategy improves the model’s ability to ground its responses in retrieved evidence, leading to more accurate and contextually appropriate answers. During RAG fine-tuning, we used gold passages with high relevance to the answers, ensuring the model learns to effectively utilize contextual information.

4. DATASET

4.1 MusWikiDB

To address the lack of a music-specific vector database for RAG in MQA, we developed **MusWikiDB**. We began by collecting music-related content from Wikipedia across seven categories: *artists*, *genres*, *instruments*, *history*, *technology*, *theory*, and *forms*. These categories were selected to cover a broad spectrum of music knowledge, providing a well-rounded foundation for answering music-related questions. The data was collected with a page depth of 2, which allowed us to capture detailed subtopics and related information. We split the content into sections such as *background*, *biography*, and *history*. We then removed sections shorter than 60 tokens to ensure the remaining text had enough context for meaningful retrieval.

Table 1 compares our proposed MusWikiDB with the Wikipedia corpus [8]. While MusWikiDB contains fewer pages (31K vs 3.2M) and has a smaller vocabulary size (786K vs 21.5M), it consists exclusively of music-specialized text information.

¹ A passage refers to a portion of a document relevant to a query [25].

	MusWikiDB	Wikipedia Corpus [8]
# Pages	31K	3.2M
# Passages	629.2K	21M
Total tokens	65.5M	2.1B
Vocab Size	786K	21.5M

Table 1: MusWikiDB and Wikipedia Corpus [8] statistics.

Based on the ablation study in Section 6.3, the text was then split into segments of up to 128 tokens, with a 10% overlap between adjacent passages, to preserve context between passages. For embedding, we employed BM25 [16], a classical and highly effective algorithm for ranking text relevance, which helped build an efficient index for MusWikiDB. This allowed us to quickly retrieve relevant information during RAG-based inference, improving the accuracy and relevance of answers. The resulting MusWikiDB provides a scalable, up-to-date knowledge base that enhances the performance of RAG in MQA tasks, allowing the system to answer complex, domain-specific music-related questions with more accuracy and context.

4.2 ArtistMus

The existing text-only MQA benchmarks have focused on multimodal music understanding [11, 12] or musicology topics such as melody, chords, and history [12, 13]. However, there has been no benchmark that focuses on music metadata, particularly the artist, which is crucial in music listening contexts [30, 31]. Therefore, we created the **ArtistMus** to test the performance of LLMs in artist-related QA, using artist-related data from MusWikiDB.

We grouped sections into five categories: *biography*, *career*, *discography*, *artistry*, and *collaborations*. Token lengths ranging from 500 to 2000 were considered. Genre normalization [32] was applied by first converting all genre labels to lowercase, and then removing spaces, hyphens (-), and slashes (/). We obtained 48 root genres from [33], and after retaining only the data corresponding to the top 300 most frequent genres, each genre was mapped to the 20 final genre labels. To extract artists’ regional information, we provided the abstract of pages to the Llama 3.1 8B Instruct [34] to extract information on the country of the artist. The country list was obtained from the *pycountry* library. Then, we select a diverse range of 500 artists based on *topic*, *genre*, and *country*. Country was set as the highest priority, with a preference for artists from minor countries. Subsequently, popular genres and topics were replaced with less common ones. We generated one factual and one contextual question for each artist to evaluate the LLM’s factuality and contextual understanding. To construct these questions, we provided GPT-4o [35] with the corresponding section text. Factual questions focus on verifiable details such as dates, names, or events, whereas contextual questions require reasoning or synthesis across multiple pieces of information within the passage.

We validate the generated questions based on two criteria: *Music Relevance* and *Faithfulness*. For Music Relevance, questions that did not pertain to musical aspects

were excluded except important details such as the artist’s birthplace. For Faithfulness, GPT-4o was asked to verify whether the question and answer could be derived from the provided text. Finally, 1,000 multiple-choice questions passing human validation were generated. We randomly reassigned the correct answers, ensuring an even distribution by assigning 250 correct answers to each option.

5. EXPERIMENTS

5.1 Benchmarks

For evaluation, we used two datasets: ArtistMus (in-domain) and TrustMus (out-of-domain). Performance on factual and contextual questions was separately measured on the ArtistMus. For TrustMus, evaluation was conducted across four categories: People (Ppl), Instrument & Technology (IT), Genre, Forms, and Theory (GFT), and Culture & History (CH), each comprising 100 questions. All evaluations use a multiple-choice QA format.

5.2 Models

We compare zero-shot and QA fine-tuned models with our proposed RAG inference and RAG fine-tuned models to evaluate MQA performance. Following [11], we consider a response incorrect if it deviates from the expected format.

Zero-shot Baselines We evaluated GPT-4o [35] (API-based), Llama 3.1 8B Instruct [34] (open-source), and two music-specific models: MuLLaMA [36] and ChatMusician [12]. MuLLaMA is designed to handle audio based question answering. ChatMusician specializes in music understanding and generation with ABC notation.

QA Fine-tuning We fine-tune the Llama 3.1 8B Instruct [34] on 8K multiple-choice QA pairs that were generated from MusWikiDB.

RAG Inference We use Llama 3.1 8B Instruct [34] as our base model and implement RAG at inference-time using MusWikiDB as the retrieval database.

RAG Fine-tuning We performed RAG fine-tuning using a dataset in the form of (*context*, *question*, *answer*), by augmenting the original QA fine-tuning dataset with additional context. The target model and all other training settings were kept identical to those used in QA fine-tuning.

5.3 Training Configurations

The models are trained for one epoch using LoRA [37] with 8-bit quantization with the following hyperparameter settings: batch size = 2, gradient accumulation steps = 4, learning rate = 3e-5, weight decay = 0.005, warmup ratio = 0.1, cosine scheduler [38], AdamW [39] optimizer, r = 16, alpha = 16, and dropout = 0.1. For the ArtistMus dataset, half of the artists were included in the training data (Seen), while the other half were excluded (Unseen).

5.4 Retriever Configurations

To select the optimal retriever configuration MusWikiDB, we performed an ablation study using the ArtistMus

Model	Params	Seen	Factual		All	Contextual		All
			Unseen			Seen	Unseen	
<i>Baseline Models (zero-shot)</i>								
GPT-4o [35]	N/A	70.0	64.8	67.4	93.2	92.8	93.0	
ChatMusician [12]	7B	28.0	25.2	26.6	78.8	67.6	73.2	
MuLLaMA [36]	7B	27.2	25.2	26.2	38.4	40.0	39.2	
Llama 3.1 8B Instruct [34]	8B	40.0	38.0	39.0	87.6	82.8	85.2	
<i>Domain Adaptation Models (Llama 3.1 8B Instruct)</i>								
QA Fine-tuning	8B	41.2	38.8	40.0	81.6	78.8	79.7	
RAG Inference (Ours)	8B	81.2	82.8	82.0	89.6	88.0	88.8	
RAG Fine-tuning (Ours)	8B	81.6	83.2	82.4	92.4	91.6	92.0	

Table 2: Performance on the **ArtistMus** benchmark. *Seen* refers to data with artists present in training data, while *Unseen* contains new artists. This distinction applies only to domain adaptation models. For baseline models, all data is unseen.

Model	Params	Ppl	IT	GFT	CH	All
<i>Baseline Models (zero-shot)</i>						
GPT-4o [35]	N/A	48.0	47.0	57.0	60.0	53.0
ChatMusician [12]	7B	18.0	20.0	26.0	24.0	20.0
MuLLaMA [36]	7B	25.0	15.0	18.0	21.0	19.8
Llama 3.1 8B Instruct [34]	8B	36.0	24.0	41.0	42.0	35.8
<i>Domain Adaptation Models (Llama 3.1 8B Instruct)</i>						
QA Fine-tuning	8B	32.0	21.0	39.0	36.0	32.0
RAG Inference (Ours)	8B	33.0	40.0	44.0	46.0	40.8
RAG Fine-tuning (Ours)	8B	33.0	38.0	46.0	49.0	41.5

Table 3: Performance on out-of-domain (OOD) **TrustMus** benchmark. Four categories are: People (Ppl), Instrument & Technology (IT), Genre, Forms, and Theory (GFT), and Culture & History (CH).

benchmark. We varied the passage size (128, 256, 512 tokens) and embedding models (BM25 [16], Contriever [19], CLAP [20]). For CLAP, we increased the token limit without additional training. To ensure a fair comparison, we constrained the total token budget to 1024 by adjusting the number of retrieved passages: top-8 for 128-token passages, top-4 for 256, and top-2 for 512.

6. RESULT

6.1 In-domain Performance

Zero-shot Baselines As shown in Table 2, all models performed significantly worse on factual questions than on contextual ones, indicating challenges in recalling concrete information such as names or dates. GPT-4o [35] outperformed Llama [34] by 28.4% in factual performance, though the gap narrowed to 7.8% for contextual understanding. Despite being music-specific, both ChatMusician [12] and MuLLaMA [36] showed relatively low performance. ChatMusician slightly underperformed compared to Llama, while MuLLaMA exhibited the lowest scores, likely due to its lack of training on the MQA task and poor instruction-following capabilities.

QA Fine-tuning Comparing QA fine-tuning with zero-shot performance, factual performance improved by 1.0%, but contextual performance decreased by 5.5%. This sug-

gests that while QA fine-tuning is effective in helping the model retain information from the training data, it may also reduce the overall inference capability.

RAG Inference By utilizing RAG inference without additional training, we were able to address the low factual performance that was an issue with previous LLMs. It demonstrated a 14.6% higher factual performance compared to GPT-4o [35]. Contextual performance improved by 3.6% compared to zero-shot, but was still 4.2% lower than GPT-4o.

RAG Fine-tuning The model fine-tuned on the RAG-style dataset showed improvements in both types of questions. Compared to RAG inference, factual performance improved by 0.4%, and contextual performance improved by 3.2%. This demonstrates that by learning to leverage context, the model not only improves its memory of information present in the training data but also enhances its overall contextual understanding ability. It exhibited a remarkable 15.0% higher factual performance compared to GPT-4o, and only 1.0% lower contextual performance, which is nearly equivalent. Considering factors such as the model size, amount of training data, and the extent of training, this is an exceptionally high performance.

Embedding	Passage Size	Factual	Contextual
Gold (Upper Bound)		97.8	97.0
BM25 [16]	512	82.0	88.8
	256	82.8	88.0
	128	82.2	89.0
Contriever [19]	512	46.6	81.0
	256	55.6	84.2
	128	58.2	86.6
CLAP [20]	512	41.2	79.6
	256	41.0	84.0
	128	41.8	84.0

Table 4: Llama 3.1 8B Instruct [34] RAG performance on **ArtistMus**, by different passage size and embeddings.

6.2 Out-of-domain Performance

To validate the effectiveness of the MusT-RAG in out-of-domain scenarios, we conducted experiments by changing the benchmark from ArtistMus to TrustMus [13], using the same framework with in-domain evaluation. The results are presented in Table 3.

Zero-shot Baselines A similar trend was observed in the zero-shot evaluation for the in-domain setting. MuL-LaMA [36] and ChatMusician [12] performed worse than the random baseline (25%), which is due to incorrect answers being counted when the models failed to follow instructions. Given that the overall zero-shot performance closely aligns with the factual scores from the in-domain evaluation, we infer that TrustMus mostly consists of factual questions. The Llama 3.1 8B Instruct [34] model scored 17.2% lower than GPT-4o [35].

QA Fine-tuning The QA fine-tuned model showed a 3.8% decrease in performance compared to zero-shot, which can be attributed to the fact that models trained on artist data tend to forget information about out-of-domain topics, such as *Instrument* and *Genre*.

RAG Inference RAG inference led to an 5.0% performance improvement over zero-shot, demonstrating that MusT-RAG framework is also helpful for out-of-domain data, such as *The Grove Dictionary Online* [15], which is the basis for the TrustMus benchmark.

RAG Fine-tuning The RAG fine-tuned model showed a 0.7% improvement over RAG inference, even with the same artist data used for QA fine-tuning. This supports the fact that the RAG fine-tuning method, which incorporates context, enhances the model’s robustness in contextual understanding, even for out-of-domain data.

6.3 Ablation Study: Retriever Configurations

Table 4 shows the results of the RAG inference for the Llama 3.1 8B Instruct [34] with various passage sizes and embeddings, evaluated under the same total computation budget for fair comparison. The performance on contextual questions tended to improve as the passage size decreased across all embedding models. In contrast, for

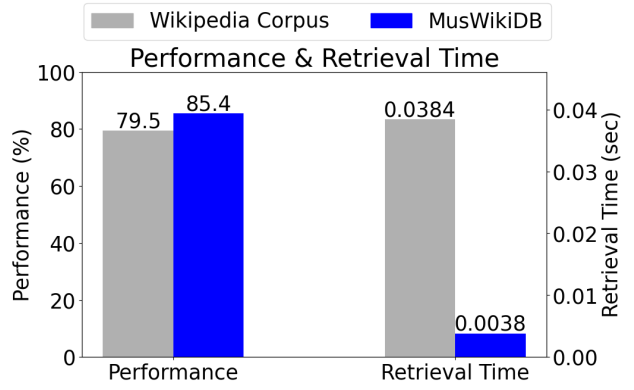


Figure 2: RAG performance and retrieval time for Wikipedia Corpus [8] and MusWikiDB.

factual questions, only Contriever [19] showed clear improvements with shorter passages, while BM25 [16] and CLAP [20] showed little to no change in performance across different passage lengths. For factual questions, there was a significant performance gap between BM25 and the other two dense embeddings. This is likely because ArtistMus places high importance on music entities such as artist and albums. Overall, the best performance was achieved using BM25 with a passage size of 128. When compared to the gold context, the factual performance was 15.6% lower, and the contextual performance was 8.0% lower. In Figure 2, we compare the RAG inference performance using the Wikipedia corpus [8] and MusWikiDB. The results show that MusWikiDB achieves a 10x faster retrieval speed and 5.9% higher performance.

7. CONCLUSION

In this paper, we presented **MusT-RAG**, a retrieval-augmented framework that enhances text-only Music Question Answering (MQA) by adapting general-purpose LLMs to the music domain. By retrieving relevant passages from a music-specific database and incorporating them into the generation context, MusT-RAG effectively mitigates the factuality limitations commonly observed in LLMs. As a result, our method achieves substantial improvements over GPT-4o [35], particularly in factual accuracy. Beyond simple retrieval, we further demonstrated that RAG-style fine-tuning outperforms traditional QA fine-tuning by improving both factual and contextual performance. Our final model achieves a 15.0% gain in factual performance over GPT-4o while maintaining comparable performance in contextual tasks. Importantly, MusT-RAG shows strong generalization capabilities. On the out-of-domain benchmark TrustMus [13], it delivers a 5.7% performance improvement over the zero-shot baseline, underscoring its robustness across diverse music-related QA scenarios. To facilitate future work in this underexplored domain, we release two key resources: **MusWikiDB**, a music-specific retrieval corpus, and **ArtistMus**, a benchmark focused on artist-level musical knowledge. We hope these contributions will drive further progress in developing accurate and domain-aware LLMs for music understanding and beyond.

8. REFERENCES

- [1] C. Jeong, “Fine-tuning and utilization methods of domain-specific llms,” *arXiv preprint arXiv:2401.02981*, 2024.
- [2] S. S. Sahoo, J. M. Plasek, H. Xu, Ö. Uzuner, T. Cohen, M. Yetisgen, H. Liu, S. Meystre, and Y. Wang, “Large language models for biomedicine: foundations, opportunities, challenges, and best practices,” *Journal of the American Medical Informatics Association*, vol. 31, no. 9, pp. 2114–2124, 2024.
- [3] N. Satterfield, P. Holbrook, and T. Wilcox, “Fine-tuning llama with case law data to improve legal domain performance,” *OSF Preprints*, 2024.
- [4] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela, “Retrieval-augmented generation for knowledge-intensive nlp tasks,” 2021. [Online]. Available: <https://arxiv.org/abs/2005.11401>
- [5] Z. Zhao, E. Monti, J. Lehmann, and H. Assem, “Enhancing contextual understanding in large language models through contrastive decoding,” *arXiv preprint arXiv:2405.02750*, 2024.
- [6] A. M. N. Allam and M. H. Haggag, “The question answering systems: A survey,” *International Journal of Research and Reviews in Information Sciences (IJR-RIS)*, vol. 2, no. 3, 2012.
- [7] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “Squad: 100,000+ questions for machine comprehension of text,” *arXiv preprint arXiv:1606.05250*, 2016.
- [8] V. Karpukhin, B. Oguz, S. Min, P. S. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, “Dense passage retrieval for open-domain question answering,” in *EMNLP (1)*, 2020, pp. 6769–6781.
- [9] A. Pal, L. K. Umapathi, and M. Sankarasubbu, “Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering,” in *Conference on health, inference, and learning*. PMLR, 2022, pp. 248–260.
- [10] I. Chalkidis, A. Jana, D. Hartung, M. Bommarito, I. Androutsopoulos, D. M. Katz, and N. Aletras, “Lexglue: A benchmark dataset for legal language understanding in english,” *arXiv preprint arXiv:2110.00976*, 2021.
- [11] B. Weck, I. Manco, E. Benetos, E. Quinton, G. Fazekas, and D. Bogdanov, “Muchomusic: Evaluating music understanding in multimodal audio-language models,” *arXiv preprint arXiv:2408.01337*, 2024.
- [12] R. Yuan, H. Lin, Y. Wang, Z. Tian, S. Wu, T. Shen, G. Zhang, Y. Wu, C. Liu, Z. Zhou *et al.*, “Chatmusician: Understanding and generating music intrinsically with llm,” *arXiv preprint arXiv:2402.16153*, 2024.
- [13] P. Ramoneda, E. Parada-Cabaleiro, B. Weck, and X. Serra, “The role of large language models in musicology: Are we ready to trust the machines?” 2024. [Online]. Available: <https://arxiv.org/abs/2409.01864>
- [14] J. Li, L. Yang, M. Tang, C. Chen, Z. Li, P. Wang, and H. Zhao, “The music maestro or the musically challenged, a massive music evaluation benchmark for large language models,” *arXiv preprint arXiv:2406.15885*, 2024.
- [15] S. Sadie and J. Tyrrell, *The New Grove Dictionary of Music and Musicians*, 2nd edition, D. Root, Ed. London: Macmillan Publishers, 2001, accessed 05-05-2024. [Online]. Available: <http://www.oxfordmusiconline.com>
- [16] S. E. Robertson and S. Walker, “Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval,” in *SIGIR’94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*. Springer, 1994, pp. 232–241.
- [17] P. BehnamGhader, V. Adlakha, M. Mosbach, D. Bahdanau, N. Chapados, and S. Reddy, “Llm2vec: Large language models are secretly powerful text encoders,” *arXiv preprint arXiv:2404.05961*, 2024.
- [18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019, pp. 4171–4186.
- [19] G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave, “Unsupervised dense information retrieval with contrastive learning,” *arXiv preprint arXiv:2112.09118*, 2021.
- [20] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [21] I. Manco, E. Benetos, E. Quinton, and G. Fazekas, “Contrastive audio-language learning for music,” in *IS-MIR*, 2022.
- [22] S. Doh, M. Won, K. Choi, and J. Nam, “Toward universal text-to-music retrieval,” in *ICASSP 2023-2023*

- IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [23] S. Doh, M. Lee, D. Jeong, and J. Nam, “Enriching music descriptions with a finetuned-llm and metadata for text-to-music retrieval,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 826–830.
- [24] S. Wu, Z. Guo, R. Yuan, J. Jiang, S. Doh, G. Xia, J. Nam, X. Li, F. Yu, and M. Sun, “Clamp 3: Universal music information retrieval across unaligned modalities and unseen languages,” *arXiv preprint arXiv:2502.10362*, 2025.
- [25] C. Wade and J. Allan, “Passage retrieval and evaluation,” *Center for Intelligent Information Retrieval Department of Computer Science University of Massachusetts Amherst, MA*, vol. 1003, 2005.
- [26] K. Shuster, S. Poff, M. Chen, D. Kiela, and J. Weston, “Retrieval augmentation reduces hallucination in conversation,” *arXiv preprint arXiv:2104.07567*, 2021.
- [27] S. Borgeaud, A. Mensch, J. Hoffmann, T. Cai, E. Rutherford, K. Millican, G. B. Van Den Driessche, J.-B. Lespiau, B. Damoc, A. Clark *et al.*, “Improving language models by retrieving from trillions of tokens,” in *International conference on machine learning*. PMLR, 2022, pp. 2206–2240.
- [28] M. Yasunaga, A. Bosselut, H. Ren, X. Zhang, C. D. Manning, P. S. Liang, and J. Leskovec, “Deep bidirectional language-knowledge graph pretraining,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 37 309–37 323, 2022.
- [29] Y. Wang, P. Li, M. Sun, and Y. Liu, “Self-knowledge guided retrieval augmentation for large language models,” *arXiv preprint arXiv:2310.05002*, 2023.
- [30] J. H. Lee, “Analysis of user needs and information features in natural language queries seeking music information,” *Journal of the American Society for Information Science and Technology*, vol. 61, no. 5, pp. 1025–1045, 2010.
- [31] S. Doh, K. Choi, D. Kwon, T. Kim, and J. Nam, “Music discovery dialogue generation using human intent analysis and large language models,” *arXiv preprint arXiv:2411.07439*, 2024.
- [32] H. Schreiber, “Improving genre annotations for the million song dataset,” in *ISMIR*, 2015, pp. 241–247.
- [33] —, “Genre ontology learning: Comparing curated with crowd-sourced ontologies,” in *ISMIR*, 2016, pp. 400–406.
- [34] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan *et al.*, “The llama 3 herd of models,” *arXiv preprint arXiv:2407.21783*, 2024.
- [35] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, “Gpt-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [36] S. Liu, A. S. Hussain, C. Sun, and Y. Shan, “Music understanding llama: Advancing text-to-music generation with question answering and captioning,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 286–290.
- [37] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, “Lora: Low-rank adaptation of large language models,” *ICLR*, vol. 1, no. 2, p. 3, 2022.
- [38] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, “Transformers: State-of-the-art natural language processing,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: <https://www.aclweb.org/anthology/2020.emnlp-demos.6>
- [39] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017.