

From Novice to Expert: Generating Audience-Dependent Concert Moderations with RAG-LLMs

Kerstin Denecke

Bern University of Applied Sciences

Quellgasse 21, 2502 Biel

Switzerland

kerstin.denecke@bfh.ch

Abstract

In this paper, we study the capabilities of large language models (LLMs) to adapt a concert moderation to diverse expertise levels of listeners. Our proof-of-concept concert moderator is based on retrieval-augmented generation (RAG) and uses few-shot audience modelling to infer listener’s expertise. We study the capabilities of the system to adapt to three different listener’s expertise levels. Two open domain LLMs are compared: gpt-oss:20b and llama3. The recognised differences among the models suggest that they vary in how directly they reproduce versus paraphrase retrieved information while maintaining semantic alignment.

1 Introduction

Live concert moderation plays a crucial role in shaping listeners’ understanding and engagement. However, listeners have a wide range of musical backgrounds, from novices who are unfamiliar with technical terminology to expert musicians who expect stylistic and analytical insight. Recent advances in large language models (LLMs), conversational interfaces, and query comprehension provide new possibilities for music recommendation (Yun and Lim, 2025; Epure et al., 2025). They showed to help listeners to clarify implicit needs and support unique exploration of music (Yun and Lim, 2025). They can also help in modelling user preferences, listening intents, emotion and mood (Epure et al., 2025). Based on these initial results, we hypothesise that LLMs can be used for concert moderation, i.e. generate personalised explanations to musical pieces, adapted to different audience groups, and able to respond to ad hoc questions from the audience.

Despite the growing interest in artificial intelligence-generated explanations and recommendation systems in cultural contexts (Singh et al., 2024), to the best of our knowledge no studies

have examined LLMs as real-time concert moderators. General purpose LLMs were tested for music recommendation (Yun and Lim, 2025). In this work, we want to test retrieval-augmented generation (RAG) (Gao et al., 2023) to realise a concert moderator. RAG can ground LLMs in reliable musical sources, reducing hallucinations and enabling contextually rich commentary. The system integrates: (i) a structured knowledge base that combines musicological literature with piece-specific metadata, (ii) a dynamic audience modelling module that classifies listener expertise using few-shot prompting, and (iii) audience-adapted generation rules that control complexity, technical terminology, and explanatory depth. The moderator engages in conversation with the audience, offering piece introductions, intermission commentary, and live responses to audience inquiries.

2 Material and methods

In the following, we describe the system components. All components were implemented in Python and run locally using an Ollama-hosted LLM and a separate embedding model (mxbai-embed-large) for retrieval. We tested and compared to LLMs: llama3 (Grattafiori et al., 2024), gpt-oss:20b (OpenAI, 2025).

2.1 Knowledge base

As knowledge base for the proof-of-concept implementation, we downloaded the *Cambridge Companion to the Organ* (Thistlethwaite and Webber, 1999). The volume contains chapters on organ music from different regions and historical periods (from before 1700 to after 1800), as well as two chapters describing the pipe organ as an instrument and the fundamentals of organ playing. This choice was pragmatic, as the PDFs are accessible with an institutional license; in future work and for a publicly available moderator, we plan to replace this

with copyright-free material.

All pages of each chapter were concatenated and split into overlapping text chunks of approximately 750 characters. Each chunk was stored in an in-memory document store together with metadata (document identifier, chapter title, section index). During system initialization, each chunk was embedded using the `mxbai-embed-large` embedding model via Ollama’s `/api/embeddings` endpoint, and the resulting vectors were cached in a NumPy array for efficient nearest-neighbour search.

In addition to the textbook material, a hand-crafted JSON file specified the concert program and structured information about the performed pieces and composers. For each piece, the JSON file contained fields such as title, composer, year of composition, key, catalogue number, estimated duration, liturgical context, registration tips, short program notes, and anecdotes. For each composer, the file contained biographical metadata (name, dates, era) and a short biography with optional “fun facts”. On startup, this JSON file was parsed into `Piece` and `Composer` data classes. Each instance was converted into a textual document (e.g., “Piece Title”, “Composer”, “Program Notes”, “Registration Tips”) and embedded into the same vector. This ensured that both textbook knowledge and concert-specific metadata were available to the RAG pipeline as separate, searchable sources.

2.2 Audience modelling module

To adapt explanations to different listener profiles, we implemented an explicit audience modelling module that infers an audience expertise level and conditions all subsequent generation on this profile.

At the beginning of each concert session (before the first piece introduction), the moderator poses three short calibration questions to the audience via the conversational interface: 1) A self-report of familiarity with classical/organ music (e.g., “almost none”, “I go to some concerts”, “I am a musician”), 2) A preference regarding explanation style (stories and imagery vs. musical theory and structure vs. a mix of both), 3) An open prompt to name a composer, piece, or musical term of interest (allowing “I’m not sure”).

Audience responses are collected as free-text and concatenated into a short “survey” string. This string is then passed to the LLM in a dedicated classification prompt. The classifier prompt uses a few-shot strategy: it defined the target label set (novice / intermediate / expert), provides three il-

lustrative survey–label pairs as examples (one per level), and then asks the model to output exactly one label for the current audience. The model was instructed to use conversational cues such as explicit self-description (“I’m an organist”, “I just like the sound”), mention of technical terminology (“fugue”, “modal language”, “registration”), and the sophistication of the named interests to choose between the three levels. The resulting label is post-processed heuristically (e.g., normalising to lowercase and falling back to novice in case of malformed output) to obtain a single categorical expertise level for the current session. This level is then used to dynamically construct the system prompt for all subsequent LLM calls. Specifically, we defined a base system prompt describing the role of the moderator (warm, concise introductions; use of RAG context; avoidance of hallucinated facts). On top of this, we appended an audience-profile block that specified style and complexity constraints:

- **Novice:** avoid jargon; if a technical term is necessary, define it in one simple sentence; prioritise stories, analogies, and felt experience over theory.
- **Intermediate:** assume familiarity with basic musical concepts (e.g., keys, themes, fugue); mix narrative and light structural/harmonic comments; keep explanations compact.
- **Expert:** allow concise use of specialist terminology (e.g., registration schemes, modal language, contrapuntal techniques); focus on insight and specificity while maintaining brevity for live settings.

The same audience level conditioned both the piece moderations and the answers to spontaneous audience questions, ensuring consistent adaptation of explanatory depth across the entire concert.

2.3 Retrieval-augmented generation pipeline

The system’s core interaction loop followed a retrieval-augmented generation paradigm. Given a generation task (e.g., “introduce the next piece” or “answer the following audience question”), the pipeline proceeded as follows:

1. **Query formulation.** For piece moderations, the system constructs a query containing the piece title, composer, and a short description

of the desired output (e.g., era/style, one listening tip, one factual highlight). For audience questions, the query consists of the question text itself.

2. **Vector retrieval.** The query is embedded with the same embedding model as the knowledge base. Cosine similarity between the query vector and all document vectors in the in-memory store is computed, and the top- k (default $k = 5$) most similar chunks are selected.
3. **Context construction.** The retrieved chunks are concatenated into a “context block” that prefixed each chunk with a simple label (e.g., piece title, composer name, or source file and section) and the retrieval score.
4. **LLM generation.** The final call to the chat model combines: (i) the audience-adapted system prompt, (ii) the context block with retrieved knowledge, and (iii) a task-specific user message (e.g., “Please introduce the next piece succinctly for a *novice/intermediate/expert* audience. . .” or “Audience question: . . . Answer briefly and accurately for a *novice/intermediate/expert* audience.”). The LLM is instructed not to contradict the provided context and not to invent dates or catalogue numbers. Generation used a moderate temperature (0.6) to balance fluency and determinism.

In the current implementation, audience inputs are collected via a text-based terminal interface (standing in for speech recognition), while outputs from the LLM are both printed to the console and synthesized using an offline text-to-speech engine (pyttsx3). In a real-world deployment, the same back-end architecture can be paired with a microphone-based speech-to-text front end (e.g., Whisper) and a loudspeaker output, allowing the moderator to function as a live, voice-based guide between pieces, during Q&A segments, and in intermission commentary.

2.4 Evaluation of audience adaptation

With the implementation described above, we want to study whether LLMs can reliably adapt the style and complexity of musical explanations to different audience expertise levels and to what extent does RAG impact on factual grounding relative to a no-RAG condition.

For each piece in the concert program (comprising 2 pieces in this experiment), we automatically generated five versions of the moderator’s output: EXPERT, INTERMEDIATE, NOVICE, NO RAG, GENERIC. For each piece, the system generated a concise moderation in all five conditions as well as answers to three standardised audience questions (e.g., “What should I listen for?”, “Why is this piece important?”, “What makes this difficult to play?”). All answers were generated by the two LLMs separately.

For judging the linguistic complexity, we measured total token count of response, number of sentences and average sentence length, type-token ratio (lexical diversity, calculated by number of unique words divided by number of words), and Flesch-Kincaid Grade Level and Flesch-Reading Ease Score (readability index, calculated with <https://goodcalculators.com/flesch-kincaid-calculator/>). We expected novice outputs to be structurally simpler and expert outputs to show higher complexity.

To evaluate how strongly answers relied on information retrieved from the knowledge base, we computed context-overlap rate as proportion of content words in the generated text that also appear in the retrieved context block, and similarity to context, defined as cosine similarity between embeddings of the output and its context. NO RAG outputs were expected to show substantially lower values on both measures.

To verify that changes induced by adaptation primarily affected style rather than factual content, we computed embedding-based cosine similarity between each adapted output and the corresponding GENERIC output for the same piece and question. High similarity indicates stable factual content, while observable stylistic shifts were expected across audience profiles.

3 Evaluation results

The results show that gpt-oss:20b answers have more tokens in average and are 1 sentence longer (see table 1). Overall, the sentences are longer than those generated by llama3. The type-token ratio is similar for both LLMs. We can recognise a larger context overlap in answers of llama3. The similarity to generic is quite high for both LLMs (gpt-oss:20b: 93-97%, llama3: 92-95%). The similarity to context is higher for answers by gpt-oss:20b. As expected, the Flesch-Kincaid Grade Level is higher

LLM	condition	tokens	sentences	average sentence length	type-token-ratio	context overlap	similarity to generic	similarity to context	Flesch-Kinaid Grade Level	Flesch Reading Ease Score
gpt-oss:20b	Expert	191.9	9	21.4	0.69	0.36	0.93	0.85	12.08	39.175
gpt-oss:20b	Intermediate	196.7	9.33	22.05	0.65	0.34	0.97	0.85	10.7	49.7
gpt-oss:20b	Novice	203.4	10.25	20.23	0.631	0.304	0.95	0.845	8.775	62.58
gpt-oss:20b	Generic	147.8	6.5	23.55	0.69	0.39	0.99	0.84	10.925	48.25
gpt-oss:20b	No Rag	129.8	7	19.56	0.74	0	0.93	0	10.01	51.96
Llama3	Expert	163.4	8.25	19.82	0.69	0.46	0.95	0.79	11.8	41.9
Llama3	Intermediate	175.9	8.75	20.24	0.644	0.41	0.95	0.79	9.93	55.575
Llama3	Novice	133	7.375	18.45	0.697	0.41	0.926	0.76	9.44	56.84
Llama3	Generic	142.9	7	20.69	0.67	0.42	1	0.782	10.225	53.45
Llama3	No Rag	112.4	5.75	19.45	0.75	0	0.924	0	10.875	48.34

Table 1: Evaluation results. Per model we averaged the results from 7 answers per condition. **GENERIC**: The base system prompt was used without any audience-profile block; RAG remained active. **NOVICE**: System prompt included the novice audience profile. **INTERMEDIATE**: System prompt included the intermediate audience profile. **EXPERT**: System prompt included the expert audience profile. **NORAG**: Same base system prompt as the **GENERIC** condition but with retrieval disabled; no context block was provided to the LLM.

for answers in expert level. For gpt-oss:20b answers, the three levels are well recognisable in a Flesch-Kinaid Grade Level increasing from novice to expert. In llama3 answers, the Flesch-Kinaid Grade Level for intermediate and novice-targeted answers are quite similar.

In terms of content provision it can be recognised that gpt-oss:20b creates answers often with enumerations (e.g. "Technical demands", "Registration:", "Tempo:"). In contrast, llama3 provides a well readable text without any explicit enumerations. Llama3 appears more empathetic with starting phrases such as "What a wonderful piece", while gpt-oss:20b answers are more factual. The complete evaluation results are available at Zenodo (<https://10.5281/zenodo.18421171>)

4 Discussion and conclusions

The evaluation results indicate that audience-adaptive generation largely behaves as intended. Outputs tailored to novice listeners were shorter, contained fewer sentences, and scored lower on readability indices, reflecting a simpler linguistic structure. Conversely, outputs targeted to experts showed higher token counts, more complex syntax, and higher Flesch-Kinaid Grade Levels. This confirms that the system modulates explanatory complexity across expertise levels.

The differences among the models suggest that

models vary in how directly they reproduce versus paraphrase retrieved information while maintaining semantic alignment. At the same time, we observed high similarity between the outputs of the adapted conditions and the generic baseline. This indicates that audience adaptation primarily affects style and terminology rather than the underlying factual content. The high similarity-to-context scores across all RAG-enabled conditions support this, demonstrating that both models maintain strong grounding in the retrieved musicological sources. The combination of curated program metadata and thematically relevant PDFs establishes appropriate boundaries for the domain while enabling flexibility in answering audience questions. Future systems could also incorporate cultural background cues to adapt examples, metaphors, and contextual framing.

The content accuracy was not explicitly checked in our evaluation. In table 2, we show three examples of the generated replies to the question "What should I listen for in "Präludium und Fuge in C-Dur, BWV 545" by Johann Sebastian Bach?". We selected the three conditions "Expert", "Novice" and "No RAG". It can be seen that the language of the answer targeting novices is very metaphoric while in expert-level, it uses many domain-specific terms such as augmentation, inversion, ritornello-like. In terms of correctness, the novice-level answer can be considered correct - mostly because

No.	Answer
1	<p>What a wonderful piece! As you listen to the Präludium und Fuge in C-Dur, BWV 545, I encourage you to pay attention to the way the melody flows like a river, building momentum and energy as it goes. You might also notice how the fugue section takes that same melody and weaves it together with other musical threads, creating a rich tapestry of sound.</p> <p>In simple terms, think of the prelude as a gentle stream that grows into a mighty river, and the fugue as the river’s many tributaries coming together to create a beautiful, complex whole.</p>
2	<p>What a delight to explore this magnificent work with you! In the Präludium und Fuge in C-Dur, BWV 545, I recommend listening for the prelude’s bright, architecturally clear architecture, showcasing Bach’s mastery of counterpoint and harmony. Pay attention to the way the principal chorus registration adds depth and clarity to the intricate figuration.</p> <p>As we transition into the fugue, note how Bach subjects the theme to a range of procedures, including inversion, augmentation, and inversion with augmentation, creating a truly compelling contrapuntal argument. The angular chromaticism of the subject is striking, but it’s Bach’s formal strategy that makes this piece stand out.</p> <p>Finally, appreciate the way the fugue’s structure unfolds, with its ritornello-like episodes and clever use of counterpoint to create a sense of tension and release. And don’t forget to listen for the subtle nuances in registration, which add texture and variety to the overall performance.</p>
3	<p>What a wonderful piece! In the Präludium und Fuge in C-Dur, BWV 545, I recommend listening for the way Bach’s masterful hands weave together intricate counterpoint and soaring melody. Pay attention to the way the fugue subject is expertly developed throughout the piece, with each entry adding new layers of texture and interest. And don’t miss the majestic, sweeping arpeggios that punctuate the prelude – they’re a perfect showcase for the organ’s expressive capabilities!</p>

Table 2: Two answers generated by Llama 3 to the question: What should I listen for in "Präludium und Fuge in C-Dur, BWV 545" by Johann Sebastian Bach? in 1) novice level and 2) expert level and 3) without RAG

of its generic and metaphoric character, reducing the risk of false information. The version without RAG is very generic without specific information on the piece. The expert-level answer contains some wrong information. For example, there is no "ritornello" in the fugue. Further, the phrase "Bach subjects the theme to inversion, augmentation, and inversion with augmentation" is incorrect. One of the main elements of the fugue is the syncopation between the voices.

A more in-depth evaluation of the accuracy is still outstanding. For this purpose, the knowledge base will be extended by more detailed information on the musical pieces. In its current knowledge base, there is only the information on the pieces in the JSON file; the Cambridge Companion to the Organ describes the musical epochs but contains almost no information on specific musical pieces. Therefore, the knowledge base might be insufficient to answer the audience’s questions appropriately, resulting in these generic answers as the examples show.

Additional limitations of the current implementation will inform future work. Piece and com-

poser information was manually prepared for this prototype. Integrating structured music metadata from open knowledge graphs, such as MusicBrainz or Wikidata, could improve scalability and enable the automatic enrichment of concerts with additional contextual facts. User studies with authentic classical music audiences would provide valuable insights into acceptance, trust, and perceived usefulness.

In summary, our findings show that RAG-grounded LLMs can offer stylistically adaptive and factually grounded concert moderation. However, it still has to be assessed whether listeners will accept such system and in which contexts it could be successfully implemented.

References

- Elena V. Epure, Yashar Deldjoo, Bruno Sguerra, Markus Schedl, and Manuel Moussallam. 2025. [Music recommendation with large language models: Challenges, opportunities, and evaluation](#). *Preprint*, arXiv:2511.16478.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen

- Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, and et al. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- OpenAI. 2025. [gpt-oss-120b and gpt-oss-20b model card](#). *Preprint*, arXiv:2508.10925.
- Pushpdeep Singh, Mayur Patidar, and Lovekesh Vig. 2024. [Translating across cultures: LLMs for intralingual cultural adaptation](#). In *Proceedings of the 28th Conference on Computational Natural Language Learning*, pages 400–418, Miami, FL, USA. Association for Computational Linguistics.
- Nicholas Thistlethwaite and Geoffrey Webber, editors. 1999. *The Cambridge Companion to the Organ*, 1 edition. Cambridge University Press.
- Sojeong Yun and Youn-kyung Lim. 2025. [User experience with llm-powered conversational recommendation systems: A case of music recommendation](#). In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA. Association for Computing Machinery.