Interactive Playlist Generation from Titles

Eléa Vellard*
Enzo Charolois–Pasqua*
elea.vellard@eurecom.fr
enzo.charolois-pasqua@eurecom.fr
EURECOM
Sophia Antipolis, France

Youssra Rebboud
youssra.rebboud@eurecom.fr
EURECOM
Sophia Antipolis, France

Pasquale Lisena
pasquale.lisena@eurecom.fr
EURECOM
Sophia Antipolis, France

Raphael Troncy
raphael.troncy@eurecom.fr
EURECOM
Sophia Antipolis, France

Abstract

This demo presents an interactive playlist recommendation system that relies exclusively on playlist titles. By fine-tuning a transformer-based language model on clustered playlists, we enable real-time playlist generation for a given title, relying on the semantic meaning of known playlists' and tracks' titles. The playlist title provided in input is freely expressed in natural language in a user-friendly web interface. The system is lightweight, fast, and fully accessible through a simple web page.

CCS Concepts

• **Information systems → Language models**; **Recommender systems**.

Keywords

Language model, Playlist generation, NLP

ACM Reference Format:
Eléa Vellard, Enzo Charolois–Pasqua, Youssra Rebboud, Pasquale Lisena, and Raphael Troncy. 2025. Interactive Playlist Generation from Titles. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems (RecSys '25), September 22–26, 2025, Prague, Czech Republic.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3705328.3759336

1 Introduction

Recommender systems play a key role in modern music platforms, but playlist curation remains a manual and time-consuming task for users. Playlist titles often provide strong cues about mood, genre, or intended use, such as "workout" or "chill", which can be leveraged to generate music recommendations, particularly in cold-start scenarios. In other cases, playlist titles can be more complex or descriptive, combining multiple concepts, emotional states, or specific listening contexts, such as songs for late-night drives in the rain or relaxing indie music for studying. These richer, free-form titles still convey valuable semantic information that can guide meaningful recommendations, provided the system is capable of understanding their nuanced language. This highlights the importance of building models that can handle both short, keyword-like titles and longer, expressive descriptions.

Prior studies [5, 8] have shown that playlist titles influence user expectations and selection behaviors. However, most research focuses on offline benchmarks without providing live systems for real-time evaluation.

This demo introduces a playlist generation system that uses playlist titles as the sole input, implementing the system described in [1]. It enables users to test the recommendations in real time through a web interface, offering a practical and deployable solution. The system is containerized for portability and can be deployed on any machine with Docker.

*Both authors contributed equally to this work.

2 Method

Fine-tuning a SBERT model. Our system is based on a fine-tuned Sentence-BERT (SBERT) model [7], which captures the semantic similarity between playlist titles. This model was fine-tuned on playlists derived from the Million Playlist Dataset (MPD) [2], released by Spotify. The playlists were grouped using K-Means clustering, leveraging the cosine distance computed on the SBERT embeddings of their track titles. In other words, playlists including often the same track would appear in the same cluster, making it a semantically homogeneous *document* inside a *training corpus* (the cluster set). Through iterative experimentation, we determined the optimal number of clusters to be 200. To ensure semantic consistency, we excluded clusters whose average internal similarity fell below an empirical threshold set to 0.02. The remaining cohesive clusters were used in the fine-tuning process of SBERT, treating each cluster as a textual document. In this context, titles within the same cluster were considered analogous to words appearing within the same document. The fine-tuning used cross-entropy loss.

Embedding and Retrieval. When a user submits a playlist title, the system:

- Encodes the title into a 384-dimensional embedding using the fine-tuned SBERT model;
- Retrieves the 50 most similar playlists based on cosine similarity;

- Applies a voting mechanism on the tracks within these playlists, that simply returns the 10 most frequently occurring tracks.

The system accepts any user-provided title as input, including titles not previously seen during training, and automatically retrieves the most relevant playlists from its indexed collection. Experiments on the MPD yielded an R-Precision of 0.1332 and an NDCG of 0.4311. This results show a significant improvement upon the state of the art for recommendation systems based solely on playlist titles, as is shown in Table 1.

Method	R-Precision	NDCG
Monti et al. [6] (Only title)	0.0837	0.1260
Faggioli et al. [3] (Only title)	0.1093	0.2451
Kim et al. [4] (Only title)	0.0760	0.1866
Our method	**0.1556**	**0.2825**

Table 1: Comparison between our method and the state of the art

3 A Demo Web Application

Implementation. The pipeline of the demo is represented in Figure 1. The main components are

- a *back-end*, wrapping the recommender system;
- a *front-end*, realised with standard web technologies;
- a web server realised in Flask that realise the communication between the other components through API calls returning JSON results.

For portability, the system is provided as a self-contained Docker image based on `python:3.11-slim` with a footprint under 6 GB, including the machine learning model.

To guarantee fast response times and efficient retrieval during user interaction, the application performs a warm-up phase at startup. This procedure ensures that all critical components are preloaded into memory, minimizing computational overhead during runtime. Specifically, the system first loads the Sentence-BERT tokenizer, enabling immediate text encoding of any playlist title provided by the user. It then loads the fine-tuned model and initializes an index containing the precomputed embeddings of all known playlists in the dataset. This in-memory index allows for rapid nearest-neighbor retrieval using cosine similarity, making real-time recommendations possible with minimal latency[1].

User Interaction. The user interface of the demo (Figure 2) is intentionally minimal and intuitive, designed to focus the user's attention on the core functionality: playlist continuation based on a title. The landing page presents a simple, clean layout with a central input field where users can freely enter a playlist title, expressed in natural language. The interface supports a wide range of inputs, from concise labels such as workout or chill to more complex, descriptive titles like relaxing indie songs for studying on a rainy day.

[1]An average of 270 ms (min 177, max 392) over 10 distinct calls, including network latency.

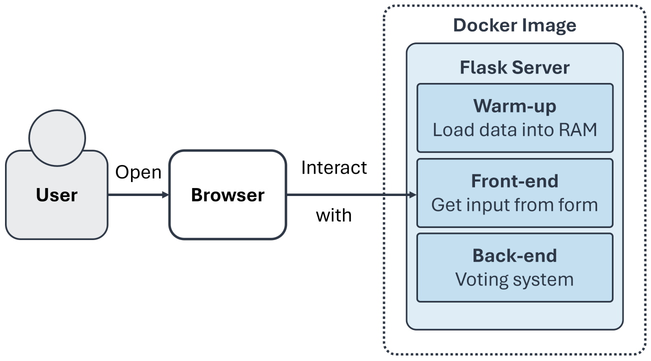
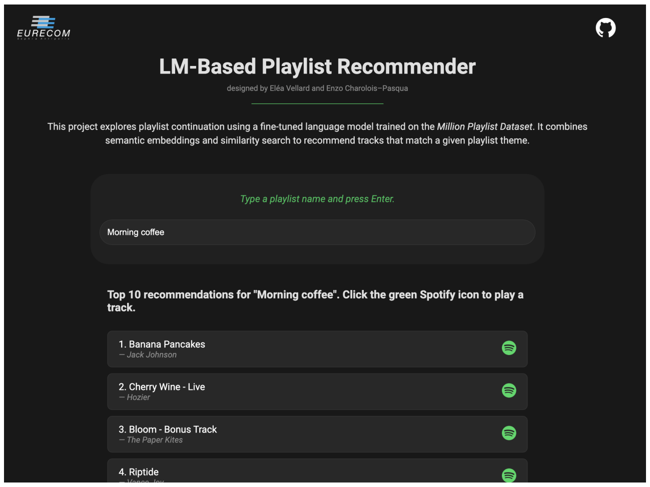


Figure 1: Diagram representing the overview of the demo pipeline.



EURECOM

LM-Based Playlist Recommender

designed by Eléa Vellard and Enzo Charolois–Pasqua

This project explores playlist continuation using a fine-tuned language model trained on the *Million Playlist Dataset*. It combines semantic embeddings and similarity search to recommend tracks that match a given playlist theme.

Type a playlist name and press Enter.

Morning coffee

Top 10 recommendations for "Morning coffee". Click the green Spotify icon to play a track.

1. Banana Pancakes
— Jack Johnson

2. Cherry Wine - Live
— Hozier

3. Bloom - Bonus Track
— The Paper Kites

4. Riptide

Figure 2: The UI of the demo application

Upon submitting the title, the system displays a list of ten recommended tracks, each presented with the track name and artist. For each recommended song, a clickable Spotify icon is provided, enabling users to immediately listen to the track via Spotify's web or desktop application. This direct playback option enhances the interactivity of the demo and facilitates real-time validation of the recommendations.

The interface is lightweight, responsive, and refreshes instantly when a new title is provided. It is deliberately free of additional elements, sidebars, or complex controls, offering a focused and seamless user experience. This design supports rapid experimentation and makes the system accessible to both technical and non-technical audiences.

4 Conclusion and Future Work

This demo provides a user-centric experience of a title-based playlist recommender system. The full implementation of the system and the web app is publicly available at https://github.com/elea-vellard/DEMO-playlist-generation, while the demo is deployed at: https://playlist-recommendation.tools.eurecom.fr/. A detailed description of the method and a more extensive evaluation and discussion is included in the main paper [1].

A limitation of our system is its sensitivity to input quality, as nonsensical user inputs may lead to less relevant recommendations. To mitigate these risks, future work could explore the integration of input validation and moderation mechanisms to filter out nonsensical or objectionable inputs. Additionally, implementing user feedback loops and safety protocols could help enhance the robustness and reliability of the system in handling a wide range of input qualities.

Future developments of this system will focus on several key improvements. First, we aim to extend the system to support playlist continuation not only from titles but also from track-based seeds, possibly integrating also collaborative filtering signals. We also plan to improve the track ranking strategy beyond the current voting mechanism. Incorporating learning-to-rank approaches or using more advanced retrieval models could help refine the order of recommended tracks, moving beyond simple frequency-based selection to a more nuanced assessment of relevance.

Also, we intend to give users greater control over the recommendation process, enabling them to dynamically prioritize diversity, precision, or thematic alignment directly through the demo interface, based on their personal preferences and exploration goals. At the same time, user feedback – explicit or implicit (direct clicks on the songs) – can be a valuable source for incorporating continuous learning in the system, going beyond predictions made on the static clusters. Lastly, we intend to optimize the deployment and scalability of the system, reducing model size, parallelizing search operations, and integrating caching strategies.

Acknowledgments

This work was supported by the French Public Investment Bank (Bpifrance) i-Demo program within the LettRAGraph project (Grant ID DOS0256163/00). No AI tools were used for data analysis, experimentation, or the formulation of conclusions, except in the ways it is described in the paper itself.

Author Contributions

Eléa Vellard: Methodology, Software, Writing – review & editing; **Enzo Charolois–Pasqua:** Methodology, Software, Writing – review & editing; **Youssra Rebboud:** Supervision, Methodology, Writing – review & editing; **Pasquale Lisena:** Conceptualization, Supervision, Methodology, Writing – original draft; **Raphael Troncy:** Supervision, Writing – review & editing;

References

[1] Enzo Charolois-Pasqua, Eléa Vellard, Youssra Rebboud, Pasquale Lisena, and Raphael Troncy. 2025. A Language Model-Based Playlist Generation Recommender System. In *Proceedings of the 19th ACM Conference on Recommender Systems* (Prague, Czech Republic) *(RecSys '25)*. ACM, New York, NY, USA, 11. doi:10.1145/3705328.3748053

[2] Ching-Wei Chen, Paul Lamere, Markus Schedl, and Hamed Zamani. 2018. Recsys challenge 2018: automatic music playlist continuation. In *Proceedings of the 12th ACM Conference on Recommender Systems* (Vancouver, British Columbia, Canada) *(RecSys '18)*. Association for Computing Machinery, New York, NY, USA, 527–528. doi:10.1145/3240323.3240342

[3] Guglielmo Faggioli, Mirko Polato, and Fabio Aiolli. 2018. Efficient Similarity Based Methods For The Playlist Continuation Task. In *Proceedings of the ACM Recommender Systems Challenge 2018* (Vancouver, BC, Canada) *(RecSys Challenge '18)*. Association for Computing Machinery, New York, NY, USA, Article 15, 6 pages. doi:10.1145/3267471.3267486

[4] Jaehun Kim, Minz Won, Cynthia C. S. Liem, and Alan Hanjalic. 2018. Towards Seed-Free Music Playlist Generation: Enhancing Collaborative Filtering with Playlist Title Information. In *Proceedings of the ACM Recommender Systems Challenge 2018* (Vancouver, BC, Canada) *(RecSys Challenge '18)*. Association for Computing Machinery, New York, NY, USA, Article 14, 6 pages. doi:10.1145/3267471.3267485

[5] Zhi Li, Min Song, Shen Duan, and Zhe Wang. 2022. Are users attracted by playlist titles and covers? Understanding playlist selection behavior on a music streaming platform. *Journal of Innovation & Knowledge* 7, 3 (2022), 100212. doi:10.1016/j.jik.2022.100212

[6] Diego Monti, Enrico Palumbo, Giuseppe Rizzo, Pasquale Lisena, Raphaël Troncy, Michael Fell, Elena Cabrio, and Maurizio Morisio. 2018. An Ensemble Approach of Recurrent Neural Networks using Pre-Trained Embeddings for Playlist Completion. In *Proceedings of the ACM Recommender Systems Challenge 2018* (Vancouver, BC, Canada) *(RecSys Challenge '18)*. Association for Computing Machinery, New York, NY, USA, Article 13, 6 pages. doi:10.1145/3267471.3267484

[7] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 3982–3992. doi:10.18653/v1/D19-1410

[8] Ali Yürekli, Alper Bilge, and Cihan Kaleli. 2021. Exploring playlist titles for cold-start music recommendation: an effectiveness analysis. *Journal of Ambient Intelligence and Humanized Computing* 12, 11 (2021), 10125–10144. doi:10.1007/s12652-020-02777-3