Using Language Models for Music Recommendations with Natural-Language Profiles

Paul Gagliano, Griffin Homan, Douglas Turnbull and Venkata S Govindarajan

Department of Computer Science, Ithaca College

Abstract

Recent research has shown that language models can be used effectively for recommendation. We are interested in using language models to create natural-language (NL) profiles based on users' listening habits for the purpose of providing accurate, interpretable, and steerable recommendations. In this paper, we explore the usage of these NL profiles for recommendation with both a large commercial language model and a smaller open-source model. We find that, though these methods do not perform as well as traditional recommender systems (e.g., matrix factorization) in terms of accuracy, they do produce meaningful recommendations and provide the user the ability to control their recommendations using natural language.

Keywords

music recommendation, language models, natural language profiles, popularity bias

1. Introduction

Discovering local music is challenging: most local artists are relatively obscure, and conventional recommendation interfaces struggle to surface them in a way that users trust and are able to influence. Localify.org addresses this by providing personalized, locally grounded artist and event recommendations. Building on prior work that characterized the difficulty of recommending long-tail artists and the additional challenge of contextualizing those music recommendations for users [1, 2], we explore whether language models can be used to generate high-quality local artist recommendations.

Specifically, we investigate whether the natural language (NL) user profiles (see Figure 1) produced by language models can (1) recover the quality of the recommendation comparable to more direct approaches that use a list of seed artists as a item-based profile, and (2) remain interpretable [3] and potentially steerable [4] by users, thus improving trust, scrutability, and adoption by users. To do so, we compare two item profile approaches (one that uses a traditional matrix factorization algorithm as a baseline, and one that uses a language model without the added step of creating NL profiles), introduce prompt engineering for composing descriptive and editable NL listening profiles, and perform a detailed error analysis to surface issues like popularity bias that affect recommendation accuracy.

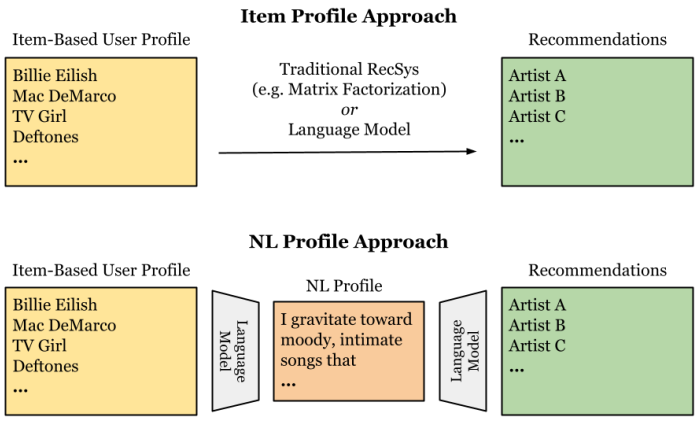


Figure 1: Recommendation Approaches

MuRS 2025: 3rd Music Recommender Systems Workshop, September 22nd, 2025, Prague, Czech Republic

gaglianopa@gmail.com (P. Gagliano); griffinhomanj@gmail.com (G. Homan); dougturnbull@gmail.com (D. Turnbull); vgovindarajan@ithaca.edu (V. S. Govindarajan)

https://griffinhoman.com/ (G. Homan); https://dougturnbull.org/ (D. Turnbull); https://venkatasg.net (V. S. Govindarajan)

0009-0005-7839-5000 (P. Gagliano); 0009-0002-0152-8598 (G. Homan); 0009-0001-7252-1855 (D. Turnbull); 0000-0003-1015-3548 (V. S. Govindarajan)

2. Related Work

Recent work has reframed recommendation to be a language task by using large language models (LLMs), either as complete recommender systems (using a variety of embedding approaches) [5] or as agents to mediate between users and traditional recommender systems [6, 7, 8]. One focus of this approach is to leverage the NL medium to first generate NL user profiles from lists of items with which a user is associated, and then use these profiles to rank-order or recommend novel items [4, 3, 9, 10]. The advantage of this approach is that the NL profiles provide a degree of scrutability (the ability to examine) and steerability (the ability to alter) through the reading and manual editing of these profiles. The generation of these NL profiles can be reframed as a prompt engineering task, allowing us to utilize techniques like providing examples (one/few-shot prompting), and providing contrasting item sets to improved the quality in terms of both readability and recommendation quality [11, 12].

Our work sits at the intersection of these threads. We evaluate LLMs both as recommenders and as interface layers by embedding user behavior into editable NL profiles, using established prompt engineering techniques, and studying how intrinsic data biases [13, 14] , such as artist popularity, affect model behavior in long-tail music recommendation settings.

3. Experimental Setup

The evaluation of the experiments in this paper was carried out using data from Localify.org[1] users who signed up with their Spotify accounts. This data is comprised of the "heavy rotation" artists (artists to which a user listens frequently) of 192 Localify users, and contains 3551 artists. Each user must have at least 10 artists in their heavy rotation, but there is no upper limit on the number of artists in this set[2]. Henceforth, we will refer to the artists in these heavy rotations as "seed artists".

During evaluation, each user's seed artists were split into two groups at random. The first group was used as their seed artists in the recommendation, and the second group was placed in the candidate set. A set of artists (i.e. distractors) of the same size as the split seed groups was then taken from the seed artists of other users and placed in the candidate set as well, and the language model was asked to rank the candidate set. The score for each recommendation was found by calculating the area under the ROC curve (AUC) [15] of a binary list representing the ranked artists, where 1 (a true positive) represents an artist that was recovered from the user's seeds, and 0 (a true negative) represents an artist that is from distractor set. AUC is a common evaluation metric for evaluating the quality of ranked data, useful because it evaluates the accuracy of the order of a list of recommendations where randomly ranking items has an expected value of 0.5, and a perfect ranking (all true positives rank ahead of all true negatives) is 1.0.

In our previous work [1], we establish Alternating Least Squares Matrix Factorization (MF) [16] as a useful algorithm for local music recommendation, and the algorithm used currently by Localify. With the evaluation metric used in this paper, MF performs with an AUC of 0.82. We will use this as a baseline by which to compare the recommendation techniques introduced in this paper.

4. Direct Recommendations Without NL Profiles

In order to establish a baseline for recommendation performance using language models with our evaluation method, we first generate recommendations by providing a language model directly with a list of seed artists for a user. We then generated recommendations by providing both artist names and their associated genres. The prompt used for these approaches can be seen in B.1 and the results can be seen in Table 1.

[1] https://localify.org

[2] In order to ensure anonymity, 20% of all users' seed artists were removed at random prior to any experiments or processing of data. This ensures that the sets of user seeds cannot be traced back to the user with which they are associated, even by system administrators.

We ran these experiments on two models; gpt-4o-mini via the OpenAI API and gemma-3-4b-it running on a local GPU workstation. When we provided the model with artist genres to contextualize the names, neither model's accuracy improved by a substantial amount. In the case of gemma-3-4b-it, providing genres improved the accuracy in some iterations of the experiment and degraded the accuracy in others.

Table 1: Average AUC (with standard error) for 192 Users with Item-based Profiles

Model	Artists Names	Artist Names & Genres
gpt-4o-mini	0.75 (± 0.02)	0.77 (± 0.02)
gemma-3-4b-it	0.69 (± 0.02)	0.68 (± 0.02)
MF (baseline)	0.82 (± 0.02)	–

5. Initial Listening Profiles

Our first experiment with embedding user seeds into a natural language (NL) profile used a naive prompt as a starting point for these experiments. We asked the model to construct a natural language profile of a user's listening habits given a list of artists that this user is known to listen to. The prompt for this can be found in B.3. We then passed this profile, along with a list of candidate artists, to the model in a new context window and asked it to rank the candidate set. Though we expected the quality of the recommendations to degrade by removing the set of seed artists, the experiments with these initial natural language listening profiles showed no substantial change in accuracy from the experiments that provided the seed artists directly in the prompt (see Table 2).

6. Prompt Engineering

We performed a series of prompt engineering experiments to try and improve the quality of our natural language profiles. We hypothesized that, in so doing, we would improve the quality of the recommendations. We also wanted to make these profiles as human-readable and concise as possible in the interest of future research on steerability [4].

6.1. Example

Research shows that LM-generated summaries can be improved by providing the model with contrasting examples [12]. We applied this to our listening profiles by providing the model with the known artists of n random other users from the test set, and asking it to write the profile paying special attention to the difference between the target user and the other users. We tried this both with 3 contrasting users and with 5 contrasting users, and did not notice a substantial improvement with either approach. The prompt for this approach can be seen in B.4.

Table 2: Recommendation Accuracy with NL Profiles

Experiment	Accuracy (gpt)	Accuracy (gemma)
No Profile	0.75 (± 0.02)	0.69 (± 0.02)
Contrast (3)	0.76 (± 0.02)	-
Contrast (5)	0.76 (± 0.02)	-
Example	0.76 (± 0.02)	0.70 (± 0.02)
First-Person	0.76 (± 0.02)	-
List	0.76 (± 0.02)	-
All (Human Example)	0.74 (± 0.02)	-
All (Generated Example)	**0.77** (± 0.02)	0.68 (± 0.02)
MF (baseline)	0.82 (± 0.02)	0.82 (± 0.02)

6.2. Contrast

Research shows that providing a language model with an example output (one-shot prompting) can improve the quality of the response [11]. We tried this approach with two different example listening profiles. The first was human-written, and the second was generated by the language model, and then iterated upon to create an optimal prompt. We found that the accuracy of the model degraded slightly when provided with the human-written

example, and improved slightly when provided with the optimal generated example. However, both of these changes were within two standard errors of the original score, and are not statistically significant. The prompt for this approach can be seen in B.5.

6.3. First-Person, List

Our next two experiments with prompt engineering put more of an emphasis on human readability than recommendation quality. We wanted to see if the quality would hold up if the listening profile was formatted as a list, and if it was written in first person. This is because lists are easy to edit and augment, and users will likely want to write in first person if they are editing and adding to their listening profile. Both of these experiments improved the accuracy by around 0.01, which is not significant given the standard error.

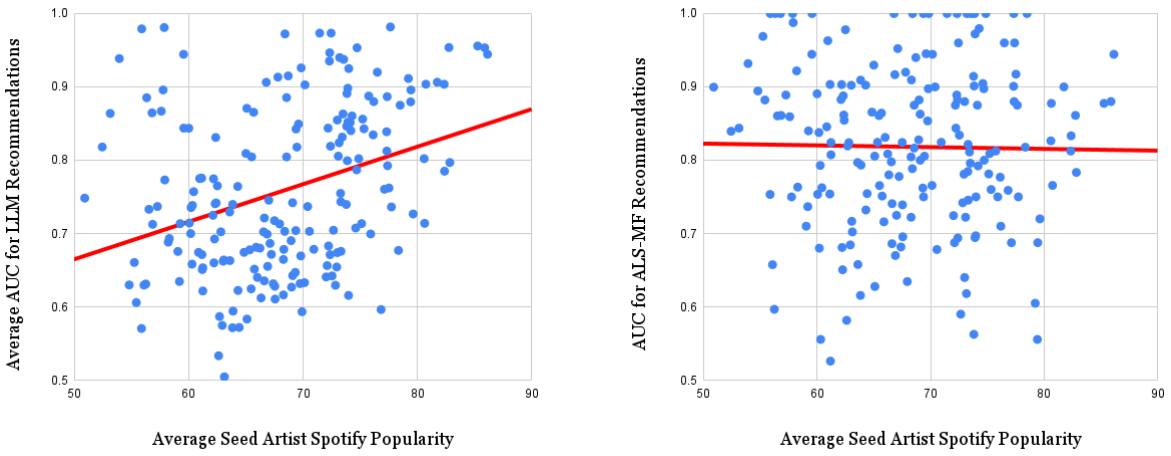
6.4. All Techniques

Our final experiments with prompt engineering combined all of these approaches. We also wrote the prompts while keeping in mind OpenAI's guidelines[3] for prompt engineering using their API. In this experiment, we found an accuracy of around 0.77, which is not a statistically significant improvement over our previous LLM-based approaches. We also tried this experiment with an accompanying system prompt, but did not see a significant difference in the AUC score. As such, although we will continue to use these prompt engineering approaches due to the slight improvement in accuracy, prompt engineering does not improve accuracy by a statistically significant amount in our case. The prompt for this approach can be seen in B.6.

7. Error Analysis

To gain insight into why these recommendations are less effective than traditional matrix factorization, we tracked the performance for different users' seed artist sets across our multiple experiments. Identifying features of seed artist sets that correlate with high recommendation accuracy could inform the development of more effective recommendation techniques. As such, we recorded the scores for individual sets of user seeds across six experiments; the original recommendation experiment with no

[3]https://platform.openai.com/docs/guides/prompt-engineering



(a) Average Popularity vs. Average AUC for Natural Language Recommendations

(b) Average Popularity vs. AUC for ALS-MF Recommendations

Figure 2: Average popularity vs. Average AUC of Natural Language Recommendations as compared to Baseline ALS-MF Recommendations

embedded listening profile, the prompt engineering experiment using examples, the prompt engineering experiment using contrasting items, and the three combined prompt engineering experiments (one of which used a human-written example, one of which used a solid generated example, and one of which used a generated example and a system prompt). The goal was not to see the accuracy of these experiments, but to analyze how individual users' recommendations were performing under a variety of conditions.

We found that the average score for recommendations using a specific user's seed artists was correlated positively (r = 0.337) with the mean Spotify popularity[4] for those artists (i.e., the more popular artists a user listens to, the more accurate their recommendations will be). A graph illustrating this finding can be found in figure 2a. Each point in the graph represents the average artist popularity for a user's item-based profile vs. the average accuracy of their recommendations over 6 LLM prompts (see Appendix B). For our baseline ALS-MF evaluation, we found that the correlation coefficient for average artist popularity and AUC was only r = -0.015. A graph illustrating this finding can be found in figure 2b. This suggests that our LLM-based recommendation approach is more susceptible to popularity bias than our MF baseline.

8. Popularity Bias

Given the potential for popularity bias revealed by these findings, we perform an experiment specifically focused on determining the influence of artist popularity on LLM-based recommendations.

Table 3: Accuracy with NL Profiles by Persona

Experiment	Accuracy (gpt)	Accuracy (Baseline MF)
Full Test Set	0.76 (± 0.02)	0.82 (± 0.02)
Low Popularity	0.74 (± 0.03)	0.81 (± 0.02)
Medium Popularity	0.76 (± 0.04)	0.83 (± 0.02)
High Popularity	0.81 (± 0.03)	0.82 (± 0.02)

To start, we divide the set of artists in our dataset into three roughly equal-sized groups based on their Spotify popularity; low popularity artists with a score below 64, medium popularity artists with a score between 64 and 73, and high popularity artists with a score above 73.

We used these popularity thresholds to divide each user's seed artists into three groups, each of which is considered an individual user *persona* for this experiment. After filtering these personas by size, we were left with 110 low popularity personas, 82 medium popularity personas, and 109 high popularity personas. We then used these personas to run our best-performing approach, "All (Generated Example)", over the three popularity groups. The results of this experiment can be seen in Table 3.

These results (see Table 3) suggest a popularity bias; the score for the recommendations using medium popularity personas is similar to our original experiment score, the score for the recommendations using low popularity personas is slightly lower, and the score for the recommendations using high popularity personas is similar to the MF baseline.

9. Conclusion and Future Work

In this paper, we explored a variety of methods for generating recommendations using NL user profiles and language models. It is clear that, with these methods, recommendations made with language models do not perform as well (in terms of recommendation accuracy) as recommendations made with traditional methods. However, these listening profiles provide a potential for user control that would not be possible with conventional methods. As such, we believe that language models could be a useful tool for scrutable and steerable recommendation.

There are some limitations on the work we were able to do with NL user profiles. There are an infinite number of approaches to prompt engineering we could have tried, and there are many more language models (both commercial-grade and open-source) that we could have used for the experiments. These are both avenues that could lead to better accuracy and more readable listening profiles.

[4]https://developer.spotify.com/documentation/web-api/reference/get-an-artist

Future research will include attempting to improve recommendation performance through language model training and fine-tuning. We are also interested in exploring hybrid recommendation which uses traditional recommendation algorithms to rank order candidate artists and then uses language models to help explain recommendations [2]. Finally, we plan to explore scrutability and steerability which involves conducting user studies to explore different ways users can benefit from natural language profiles.

Acknowledgments

This research was supported by NSF Award IIS-2312866. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors. We'd like to thank Joyce Zhou and Thorsten Joachims for their helpful discussions related to this research.

Declaration on Generative AI

During the preparation of this work, the authors used X-GPT-4 for grammar and spell-checking. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] D. Turnbull, A. Trainor, G. Homan, E. Richards, K. Bentley, V. Conrad, P. Gagliano, C. Raineault, T. Joachims, Localify. org: Locally-focus music artist and event recommendation, in: Proceedings of the 17th ACM Conference on Recommender Systems, 2023, pp. 1200–1203.

[2] P. Gagliano, G. Homan, C. Raineault, R. Ayambem, B. Burns, D. Turnbull, Localify.org: Contextualizing long-tail music for local artist discovery, in: LBD Proceedings of the 2024 ISMIR Conference, 2024.

[3] F. Radlinski, K. Balog, F. Diaz, L. Dixon, B. Wedin, On natural language user profiles for transparent and scrutable recommendation, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022. URL: https://dl.acm.org/doi/10.1145/3477495.3531873.

[4] J. Zhou, T. Joachima, Language-based user profiles for recommendation, in: WSDM 2024 Workshop on Large Language Models for Individuals, Groups, and Society, 2024. URL: https://arxiv.org/pdf/2402.15623.

[5] S. Oramas, A. Ferraro, A. Sarasua, F. Gouyon, Talking to your recs: Multimodal embeddings for recommendation and retrieval, in: MuRS 2024: 2nd Music Recommender Systems Workshop, 2024.

[6] K. Bao, J. Zhang, X. Lin, Y. Zhang, W. Wang, F. Feng, Large language models for recommendation: Past, present, and future, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024. URL: https://dl.acm.org/doi/abs/10.1145/3626772.3661383.

[7] W. Xu, Y. Shi, Z. Liang, X. Ning, K. Mei, K. Wang, X. Zhu, M. Xu, Y. Zhang, iagent: Llm agent as a shield between user and recommender systems, 2025. URL: https://arxiv.org/abs/2502.14662. doi:10.48550/arXiv.2502.14662. arXiv:2502.14662.

[8] X. Zhu, Y. Wang, H. Gao, W. Xu, C. Wang, Z. Liu, K. Wang, M. Jin, L. Pang, Q. Weng, P. S. Yu, Y. Zhang, Recommender systems meet large language model agents: A survey, Foundations and Trends® in Privacy and Security 7 (2025) 247–396. URL: http://dx.doi.org/10.1561/3300000050. doi:10.1561/3300000050.

[9] J. Ramos, H. A. Rahmani, X. Wang, X. Fu, A. Lipani, Transparent and scrutable recommendations using natural language user profiles, in: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2024. URL: https://aclanthology.org/2024.acl-long.753/.

[10] E. Penaloza, O. Gouvert, H. Wu, L. Charlin, Tears: Textual representations for scrutable recommendations, 2024. URL: https://arxiv.org/abs/2410.19302.

[11] B. Meskó, Prompt engineering as an important emerging skill for medical professionals: Tutorial, Journal of Medical Internet Research (2023). URL: https://www.jmir.org/2023/1/e50638/PDF.

[12] X. Gao, K. Das, Customizing language model responses with contrastive in-context learning, in: Association for the Advancement of Artificial Intelligence Conference, 2024. URL: https://arxiv.org/abs/2401.17390v1.

[13] H. Abdollahpouri, M. Mansoury, R. Burke, B. Mobasher, The unfairness of popularity bias in recommendation, arXiv preprint arXiv:1907.13286 (2019).

[14] A. Trainor, D. Turnbull, Popularity degradation bias in local music recommendation, in: Proceedings of the MuRS Workshop at the 17th ACM Conference on Recommender Systems (RecSys 2023), 2023.

[15] A. P. Bradley, The use of the area under the roc curve in the evaluation of machine learning algorithms, Pattern recognition 30 (1997) 1145–1159.

[16] Y. Hu, Y. Koren, C. Volinsky, Collaborative filtering for implicit feedback datasets, in: 2008 Eighth IEEE international conference on data mining, Ieee, 2008, pp. 263–272.

A. Experiment Code

The code used for this paper can be found on GitHub at https://github.com/JimiLab/localify-nlp.

B. Prompts

This section enumerates prompts used for LLM artist recommendation. B.1 is used for item-based profile recommendation. B.2 is the generic prompt for natural language (NL) profile recommendation. B.3 - B.6 are the prompts used to generate NL profiles using different prompt engineering techniques. B.7 is used in conjunction with the recommendation prompts to achieve a model response that can be parsed by our evaluation procedure.

B.1. Prompt used for Direct Recommendations

You are an expert in music recommendation. Your specialty is in ranking a list of artists by how similar each one is to a different set of artists that someone already knows.

You are presented with a client who frequently listens to the following artists:

`{seeds}`

You are asked to use your expert knowledge of these artists to rank the following artists (on which you are also an expert) in order from most recommended to least recommended:

`{candidates}`

B.2. Prompt used for Recommendations with Listening Profiles

You are an expert in music recommendation. Your specialty is in ranking a list of artists based on a textual listening profile that you are provided.

You are able to perfectly rank these artists in order of preference, where preference is defined by how much a user matching your provided listening profile will enjoy that artist.

You are presented with a client with the following listening profile:

”””

`{seeds}`

”””

You are asked to use your expert knowledge of musical artists and your complete understanding of the listening profile to rank the following artists (on which you are also an expert) in order from most recommended to least recommended:

`{candidates}`

B.3. Prompt used for Initial Listening Profiles

You are an expert in describing people's music listening habits. You are presented with a client who listens to the following artists:

`{seeds}`

Give a textual description of this person's listening habits, without using artist names.

B.4. Prompt used for Listening Profiles by Example

You are an expert in describing people's music listening habits. You are presented with a client who listens to the following artists:

`{seeds}`

Give a textual description of this person's listening habits, without using artist names. Write this description according to the following example, but the details should correspond to the user I have provided.

”””

`{example}`

”””

B.4.1. Human-Written Example

My music taste is a mix of pop, musical theatre, r&b and dance/electronic. I used to be in theatre, so I love a great singer, and I really value the quality of voice and the skill of the band. As such, I love when artists use real instruments and studio musicians. I do also like music at the other side of the spectrum that's electronic, but not synth trying to be instruments. I like upbeat music, so I don't usually listen to sad songs but I will if the vocals are amazing, and I love a dreamy relaxing r&b track. I also love a rap feature on a pop song.

B.4.2. Generated Example

My musical taste weaves velvety jazz-infused neo-soul with driving, synth-heavy electronic grooves that effortlessly blend warmth and futurism. I gravitate toward smoky, upright-bass-led lounge numbers that evoke intimate club corners, alongside pulsating house tracks that ignite late-night dancefloors. I refresh my sets with experimental ambient textures and lo-fi hip-hop beats that layer nostalgic vinyl crackle over head-nodding rhythms, while occasional avant-garde free-jazz injections add daring dissonance. This balance of cozy soulfulness and boundary-pushing sonic exploration speaks to my love of music that soothes and stimulates in equal measure. The result is a listening profile rooted in sophistication and spontaneity, comfort and curiosity, offering a journey that feels both familiar and thrilling.

B.5. Prompt used for Listening Profiles by Contrast

You are an expert in describing people's music listening habits. You are presented with a client who listens to the following artists:

`{seeds}`

You are also presented with the following other users' familiar artists:

`{other_seeds}`

Give a textual description of this person's listening habits, without using artist names. Focus on how your client's listening habits differ from the habits of the other users (what makes them unique). Do not directly mention these other users, and do not pander to the client. Give an accurate description that gives the best possible summary of the artists that they like. The description should be designed such that a third party could use it to make artist recommendations.

B.6. Prompt used for Listening Profiles by All Techniques

You are an expert in describing people's music listening habits. You are experienced in writing clear, concise, and descriptive summaries of people's listening habits based on a list of artists that they like.

You are presented with a client who listens to the following artists:

`{seeds}`

You are also presented with the following other users' familiar artists:

`{other_seeds}`

Give a textual description of this person's listening habits, without using artist names. Focus on how your client's listening habits differ from the habits of the other users (what makes them unique). Do not directly mention these other users, and do not pander to the client. Give an accurate description that gives the best possible summary of the artists that they like. The description should be designed such that a third party could use it to make artist recommendations, and it should follow the following example:
"""

`{example}`
"""

as a structure and guide, but the details should pertain to the client it is written for. The description should be written in first-person, from the perspective of the client.

B.7. Post-amble used for Parseable Model Responses

You must present these recommendations in a very specific way. Each candidate artist that you are recommending has an integer ID associated with them. The key is as follows:

`{candidate_key}`

You must finish your response by listing your recommendations only using these ids, separated by commas, and surrounded by <>. An example recommendation would be as follows:

`<#,#,#,#,#,#,#,#,#,#,...>`

With each hashtag replaced by the artist you recommend in that position. You must abide by the following rules:

- Rank all of the artists in the candidate set I provided you, not just however many are in my example recommendation

- Do not include any artists not in that candidate set

- Your response must be formatted as I have said, in a list of comma separated ids (governed by the key I provided) and surrounded by angle brackets/chevrons (<>)