Predicting User Intents and Musical Attributes from Music Discovery Conversations

Anonymous ACL submission

Abstract

Intent classification is a text understanding task that identifies user needs from input text queries. While intent classification has been extensively studied in various domains, it has not received much attention in the music domain. In this paper, we investigate intent classification models for *music discovery conversation*, focusing on pre-trained language models. Rather than only predicting functional needs: *intent classification*, we also include a task for classifying musical needs: *musical attribute classification*. Additionally, we propose a method of concatenating previous chat history with just single-turn user queries in the input text, allowing the model to understand the overall conversation context better. Our proposed model significantly improves the F1 score for both user intent and musical attribute classification, and surpasses the zero-shot and few-shot performance of the pretrained Llama 3 model.

1 Introduction

Intent classification is a Natural Language Processing (NLP) task that identifies the purpose of user input in conversational systems [18] and virtual assistants [20]. It determines what the user wants to achieve, enabling the system to respond appropriately and enhance user interaction. Textual user queries are classified into predefined intent categories, typically through the use of discriminative models. For example, in the input *"I want jazz songs to listen to with my dad,"* the system should predict an initial playlist request and apply relevant filters. Additionally, the conversational system should identify that the genre and user are related musical attributes (Figure 1).

The intent classification task has been actively researched alongside advancements in pre-trained language models [20]. Early models focused on task-specific approaches using various features, including sparse representations [13], word embeddings [15, 1], and BERT-style models [8]. More recently, large language models (LLMs) [19, 7], which are decoder-only transformer models with billions of parameters, have demonstrated strong performance in intent classification through in-context few-shot learning [12].

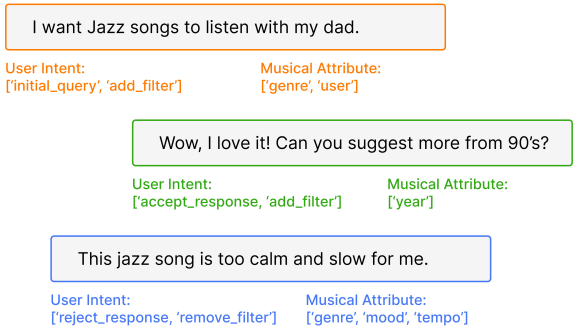

Figure 1: Examples of intent classification for music conversation. Using the given user query as input, the **user intents** and **musical attributes** of each conversation are predicted.

Thanks to these advancements in the NLP domain, research on intent classification has been conducted in various fields such as banking [11], travel scheduling [17], and movie recommendation [2] to achieve better user query understanding. However, intent classification has received very little attention in the music domain. While research on general domain intent classification [4] partially covers aspects of musical intent, it only considers two intents (i.e., *PlayMusic* and *AddToPlaylist*) and is limited to single-turn user queries. Accurately identifying the user's intent and musical needs in music discovery dialogues plays a critical role in enhancing the usability of the chat interface and overall user satisfaction.

To address this issue, we propose, for the first time, an intent classification task within the context of conversational music retrieval. Our main contributions are as follows: 1) Following prior research [6] that conducted a qualitative analysis

of music conversations, we introduce two music-specific intent classification tasks: predicting 8 interface control labels and 15 music attribute labels. 2) We apply and conduct a comparative analysis of various intent classification methodologies from the existing NLP domain. Our findings highlight that current open-source LLMs (e.g., LLaMa3) exhibit more weaknesses than task-specific fine-tuning models in the music conversation domain. 3) Additionally, we identify the optimal use of chat history to improve intent classification performance.

2 Intent Classification Frameworks

In our study, intent classification refers to the task of categorizing user input queries into user intent and musical attributes. We used sparse representation, word embedding, DistilBERT, and Llama for the intent classification task.

2.1 Sparse Representation

We utilize two types of sparse representations: Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF). BoW represents a document by counting word frequencies ignoring grammar and word order, and creates a vector based on a predefined vocabulary. TF-IDF improves on BoW by weighting words based on their frequency in the document relative to their frequency across all documents, giving higher scores to more unique words.

2.2 Word Embedding

For dense representation, we used the skip-gram version of Word2Vec [14], which was trained on the 100B word Google News Corpus and consists of 300-dimensional word vectors for 3M vocabularies. The input sentence is split into words using a whitespace tokenizer, and each word is transformed into a sequence of features through word embedding lookup. The sequence of features is aggregated into a global sentence feature via average pooling.

2.3 DistilBERT

Compared to sparse representation and word embedding, BERT [5] has the advantage of capturing the context of words through bidirectional training, allowing it to understand meaning based on surrounding text. We use DistilBERT [16], a smaller version of BERT, to reduce the size of the model and increase its speed. It reduces the size of the model by 40% while retaining 97% of its language understanding capabilities and being 60% faster.

The texts are lowercased and tokenized using the WordPiece tokenizer [21]. After tokenization, the token embeddings are processed through 6 transformer blocks to extract 768-dimensional features. For pooling, we use the output from the first position (SOS token) of the feature embeddings as the global sentence feature. It is then utilized for intent classification through MLP layers. We compare a probing model that freezes the DistilBERT and only trains the classifier with a fine-tuning model that adjusts all parameters of the DistilBERT.

2.4 Llama

Llama [19] is a generative model that excels at text generation and performs effectively on large-scale language tasks. We measured the zero-shot and few-shot performance of Llama-3.2 (1B-Instruct and 3B-Instruct) and Llama-3.1 (8B-Instruct) [7], which are small-scale models suitable for real-time conversational scenarios, to compare how pre-trained general-purpose LLMs perform in the music domain. The input to the Llama model consists of a list of selectable labels and an instruction for classification, and the output is a list of classified labels. In zero-shot tasks, the model makes predictions without any task-specific examples, while few-shot tasks provide 5 examples to guide the model's predictions. Below is an example of references for user intents and musical attributes.

Input: *"I want to create a playlist of classical music"*
Output: `['initial_query', 'add_filter']`

Input: *"Hello, can I get some pop music for a hangout later today?"*
Output: `['genre', 'theme']`

2.5 Dataset

We utilized user intent and musical attribute annotations proposed by Doh et al. [6] [1]. For the music discovery conversation taxonomy, Doh et al. [6] employed a grounded theory approach [9] to analyze the existing human-to-human music dialogue dataset (CPCD [3]). As a result, they proposed a taxonomy of 8 user intents and 15 musical attributes, with three annotators annotating 888 dialogues in a multi-label format for user intents and musical attributes.

[1] We attach the dataset in the supplementary materials

Tag	Llama						Sparse Representation		Word Embedding	DistilBERT	
	1B_0	1B_5	3B_0	3B_5	8B_0	8B_5	TF-IDF	BoW	Word2Vec	Probing	Finetune
Initial Query	0.00	0.04	0.41	0.79	0.46	0.41	0.87	0.91	0.93	0.90	0.97
Greeting	0.10	0.17	0.32	0.73	0.72	0.62	0.89	0.96	0.80	0.95	0.99
Add Filter	0.85	0.72	0.49	0.80	0.75	0.91	0.93	0.92	0.90	0.94	0.96
Remove Filter	0.14	0.00	0.30	0.13	0.39	0.61	0.00	0.50	0.36	0.38	0.76
Continue	0.12	0.00	0.18	0.15	0.21	0.24	0.61	0.46	0.32	0.62	0.80
Accept Response	0.62	0.02	0.60	0.58	0.72	0.82	0.87	0.88	0.81	0.88	0.95
Reject Response	0.08	0.00	0.00	0.17	0.36	0.62	0.17	0.36	0.50	0.65	0.82
Macro Avg	0.27	0.14	0.33	0.48	0.52	0.61	0.62	0.71	0.66	0.76	0.89

Table 1: F1 scores for user intents: "0" indicates zero-shot, and "5" refers to few-shot (5-shot) performance.

Tag	Llama						Sparse Representation		Word Embedding	DistilBERT	
	1B_0	1B_5	3B_0	3B_5	8B_0	8B_5	TF-IDF	BoW	Word2Vec	Probing	Finetune
Track	0.13	0.16	0.07	0.16	0.19	0.19	0.10	0.35	0.31	0.39	0.73
Artist	0.52	0.52	0.62	0.57	0.69	0.76	0.82	0.82	0.72	0.89	0.94
Year	0.09	0.10	0.11	0.19	0.17	0.32	0.44	0.67	0.31	0.75	0.89
Popularity	0.01	0.02	0.02	0.03	0.02	0.04	0.00	0.29	0.05	0.10	0.62
Culture	0.02	0.02	0.02	0.03	0.02	0.05	0.00	0.00	0.00	0.36	0.40
Similar Track	0.02	0.01	0.01	0.01	0.08	0.04	0.00	0.33	0.04	0.33	0.80
Similar Artist	0.05	0.05	0.05	0.10	0.20	0.15	0.14	0.34	0.25	0.52	0.74
User	0.00	0.03	0.04	0.03	0.01	0.04	0.00	0.47	0.20	0.25	0.71
Theme	0.15	0.18	0.19	0.30	0.23	0.35	0.70	0.76	0.63	0.74	0.89
Mood	0.06	0.11	0.09	0.17	0.11	0.20	0.24	0.57	0.55	0.47	0.75
Genre	0.21	0.26	0.24	0.36	0.32	0.47	0.71	0.83	0.67	0.80	0.93
Instrument	0.00	0.01	0.01	0.02	0.04	0.04	0.00	0.44	0.00	0.22	0.71
Vocal	0.00	0.02	0.02	0.04	0.02	0.05	0.00	0.40	0.13	0.25	0.35
Tempo	0.02	0.02	0.02	0.05	0.03	0.06	0.00	0.17	0.10	0.20	0.63
Macro Avg	0.09	0.11	0.11	0.15	0.15	0.20	0.23	0.46	0.28	0.45	0.72

Table 2: F1 scores for musical attributes: "0" indicates zero-shot, and "5" refers to few-shot (5-shot) performance.

2.6 Evaluation Metric

For our multi-label classification task, we used the macro-averaged F1 score to give equal weight to each label, ensuring a comprehensive evaluation across all labels. The best performance threshold for each class label was determined on the validation set (ranging from 0 to 1 with 0.01 increments) and applied to the test set. Values above the threshold were labeled as 1, and those below as 0.

2.7 Training Setup Details

The dataset was split into train, validation, and test sets in an 8:1:1 ratio, preserving the proportion of each label. We used the following hyperparameter settings: a batch size of 64, 15 epochs, the Adam optimizer [10], and a step learning rate scheduler that decreases the learning rate of the optimizer by a factor of 0.9 at the end of each epoch. The learning rate was set to 2e-4. For the concatenated setting, model was fine-tuned using concatenated sentence inputs.

2.8 Concatenate Previous Dialogue History

The music discovery conversation dataset is characterized as a multi-turn chat dataset between the recommender and the music seeker. Through turn-taking, we can perform intent classification that considers the previous context. In the movie intent classification, Cai et al. [2] reported that the classification performance of user intent and satisfaction significantly improves by incorporating context features into the classification model. Following previous study [2], we also compare the performance of intent classification using concatenated text sentences from previous dialogue turns with the case where only the current query is used.

3 Result

3.1 Performance Comparison

Tables 1 and 2 show the performances of intent classification frameworks. The fine-tuned DistilBERT outperforms all the other models for both user intent and musical attribute classification. While other models had difficulty understanding musi-

cal attributes compared to user intents, fine-tuned model handled both tasks effectively. The significant improvement in musical attribute classification (0.46→0.72) shows that our model has effectively acquired musical knowledge.

Sparse representation, word embedding, and probing models struggled with classifying less frequent labels, such as *remove_filter*, *continue*, and *reject_response* for user intent, and *popularity*, *culture*, *similar_track*, *instrument*, *vocal*, and *tempo* for musical attributes. The fine-tuned model significantly improved performance on these labels, demonstrating that it can achieve high performance even with fine-tuning on a small amount of data.

Also, musical attributes exhibited different results depending on the complexity of the words used. For instance, *popularity* saw a significant performance improvement after fine-tuning (0.10→0.62) due to the high repitition of words like *'popular'* and *'hits'*. In contrast, *culture* faced more difficulty (0.36→0.40) as it includes a wide range of words related to different cultures and countries.

The general-purpose Llama models demonstrated lower performance compared to our model fine-tuned on music-domain-specific data, indicating that they lacks sufficient knowledge of the music domain. The Llama models demonstrated improved performance as its size increased; however, using very large models such as 70B and 405B presents challenges in conversational situations that require real-time feedback due to the computational constraints. While Llama-3.1-8B-Instruct model achieved a few-shot performance of 0.61 for user intents, it only showed a few-shot performance of 0.20 for musical attributes, highlighting its difficulty in understanding musical attributes.

3.2 Concatenating Previous Dialogue History

Figure 2 shows the F1 score by varying the consideration of the previous dialogue turns, using fine-tuned model. For user intents, the best performance was achieved when considering only the previous query, which is an X-value of 0.5. Including more context beyond this point decreases performance. User intents exhibit local characteristics, where considering only the previous query is often sufficient. For instance, *initial_query* and *greeting* can be determined based on the current query alone, while *add_filter* and *continue* can be inferred by reviewing only the previous query.

For musical attributes, incorporating context led to worse performance compared to using only the current query (X-value of 0). This is because musical attributes were primarily determined by the presence of musical terms in the current query, regardless of the prior context.

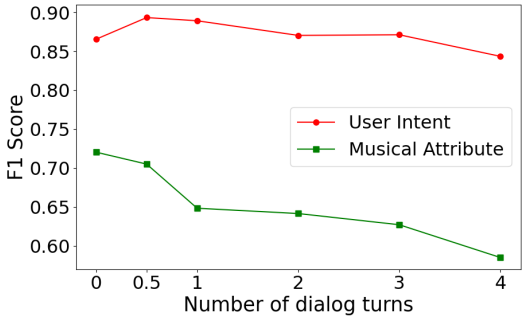

Figure 2: F1 score comparison by varying context length. The X-axis value 0 represents considering only the current query, 0.5 represents considering the previous query, and 1 to 4 represents the number of considered previous turns.

4 Conclusion

We proposed a user intent and musical attribute classification model for music discovery conversations. By fine-tuning a pre-trained language model, our model shows significantly enhanced performance in both user intent and musical attribute classification, especially for labels with a small amount of data. This suggests potential applications in fields where well-annotated large-scale data are not available, such as intent classification tasks in music discovery conversations and the development of conversational music recommendation systems.

We also introduced a method of concatenating the previous context, which helps the model better understand user intents throughout the conversation. The context length is crucial for effectively capturing relevant information, and we found that for user intent, considering only the most recent query provides the best results.

Additionally, we evaluated the zero-shot and few-shot classification performance of the Llama 3 models, which performed lower than the domain-specific fine-tuned DistilBERT model. This suggests that general-purpose LLMs like Llama lack sufficient music domain knowledge, making them less effective for intent classification.

5 Limitations

5.1 Failure Cases

In this section, we will discuss some representative failure cases, which can serve as reference points for future fine-tuning or further developments.

The model usually predicted the correct label but often made unnecessary additional predictions. The concatenated previous query is in blue. The correctly predicted intent is in green, and the incorrectly predicted intent is in red.

Query: *"[MUSIC]That's a classic. I have added a few more rock anthems for your choice. Any other song, or artist? Great hits too! Could we move away from the rock genre? Give me some hits that are classics that everyone knows and would love to hear!"*
Ground Truth: ['add_filter', 'remove_filter', 'accept_response']
Prediction: ['add_filter', 'remove_filter', 'continue', 'accept_response', 'reject_response']

Sometimes, the model incorrectly predicted a label or failed to include a necessary label.

Query: *"[MUSIC]These are the results that populated. Nothing directly by Handel. Are these okay? please can you add a little bit of Katherin jenkins"*
Ground Truth: ['add_filter', 'accept_response']
Prediction: ['add_filter', 'continue']

For musical attributes, the model also predicted many unnecessary labels, similar to what was observed with user intent.

Query: *"Is it the Corssroads Album?"*
Ground Truth: ['track']
Prediction: ['track', 'artist', 'year', 'genre', 'instrument']

It also occasionally failed to include the necessary musical attributes, as shown below.

Query: *"I would like to create a 90s Hip Hop/Rap playlist to listen while I clean my house."*
Ground Truth: ['year', 'genre', 'theme']
Prediction: ['genre', 'theme', 'mood', 'instrument']

5.2 Wrong Inference

For user intent, 3 wrong inferences occurred across the entire test dataset, and in each case, the predicted intents were present in the query but had not been provided. Below is one example.

Query: *"can you add some 80 'reggae"*
Prediction: ['reggae', ...]

For musical attributes, 6 wrong inferences occurred, where the model predicted *gender*, which was not a given. In some cases, there was no content related to gender in the provided query.

Query: *"Best Friend is my favorite"*
Prediction: ['gender', ...]

5.3 Model

The DistilBERT is not a state-of-the-art model. Better performance could be achieved by fine-tuning larger models (e.g., RoBERTa, Llama) on more advanced computing resources.

5.4 Dataset

The CPCD dataset has an imbalanced distribution of user intents and musical attributes. For instance, user intents such as *remove_filter* and *reject_response*, as well as musical attributes like *culture*, *popularity*, and *vocal*, are underrepresented in the dataset, making fine-tuning difficult. Therefore, a larger dataset with more balanced labels is needed. It is expected that training on an additional dataset, such as the LP-MusicDialog [6] dataset, could mitigate this issue to some extent. Moreover, using generative LLMs for data augmentation could be a good approach to address the data imbalance.

5.5 Concatenating Previous Dialogue History

User intents and musical attributes in our dataset exhibit local characteristics, meaning they require very little previous dialogue history for classification. If a dataset with more organically connected queries within the dialogue were used, future research could explore utilizing dialogue history to improve classification performance.

6 Ethical Considerations

While user intent and musical attribute classification tasks themselves may not directly introduce ethical risks, the training data used can contain biases that affect outcomes. If the training data reflects demographic biases, the model may reinforce these biases, leading to unfair or harmful recommendations. Therefore, it is crucial to implement ethical oversight throughout the model development process, including regular audits of the training data for biases and considerations of the implications of the model's outputs on different user groups. Continuous monitoring and adjustments are necessary to mitigate potential risks.

References

[1] Anmol Bhasin, Bharatram Natarajan, Gaurav Mathur, and Himanshu Mangla. 2020. Parallel intent and slot prediction using mlb fusion. In *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*.

[2] Wanling Cai and Li Chen. 2020. Predicting user intents and satisfaction with dialogue-based conversational recommendations. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, pages 33–42.

[3] Arun Tejasvi Chaganty, Megan Leszczynski, Shu Zhang, Ravi Ganti, Krisztian Balog, and Filip Radlinski. 2023. Beyond single items: Exploring user preferences in item sets with the conversational playlist curation dataset. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2754–2764.

[4] Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[6] Seungheon Doh, Keunwoo Choi, Daeyong Kwon, Taesu Kim, and Juhan Nam. 2024. Music discovery dialogue generation using human intent analysis and large language models. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*.

[7] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

[8] Soyeon Caren Han, Siqu Long, Huichun Li, Henry Weld, and Josiah Poon. 2022. Bi-directional joint neural networks for intent classification and slot filling. *arXiv preprint arXiv:2202.13079*.

[9] Shahid N Khan. 2014. Qualitative research method: Grounded theory. *International journal of business and management*, 9(11):224–233.

[10] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations (ICLR)*.

[11] Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. 2019. An evaluation dataset for intent classification and out-of-scope prediction. *arXiv preprint arXiv:1909.02027*.

[12] Lefteris Loukas, Ilias Stogiannidis, Odysseas Diamantopoulos, Prodromos Malakasiotis, and Stavros Vassos. 2023. Making llms worth every penny: Resource-limited text classification in banking. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pages 392–400.

[13] François Mairesse, Milica Gasic, Filip Jurcicek, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. 2009. Spoken language understanding from unaligned data using discriminative classification models. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4749–4752. IEEE.

[14] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[15] Lingfeng Pan, Yi Zhang, Feiliang Ren, Yining Hou, Yan Li, Xiaobo Liang, and Yongkang Liu. 2018. A multiple utterances based neural network model for joint intent detection and slot filling. In *CCKS Tasks*.

[16] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS*.

[17] Jetze Schuurmans and Flavius Frasincar. 2019. Intent classification for dialogue utterances. *IEEE Intelligent Systems*, 35(1):82–88.

[18] Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*.

[19] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

[20] Henry Weld, Xiaoqi Huang, Siqu Long, Josiah Poon, and Soyeon Caren Han. 2022. A survey of joint intent detection and slot filling models in natural language understanding. *ACM Computing Surveys*.

[21] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

A Appendix

A.1 Data Vocabulary Size

As seen in Tables 1 and 2, the Bag of Words model outperformed both TF-IDF and word embeddings. Notably, in the context of musical attribute classification, it even surpassed the performance of DistilBERT's probing model. This can likely be attributed to a significant overlap in vocabulary between the training and test datasets.

Train	Test	Overlap (Ratio)
4,835	1,184	908 (76.7%)

Table 3: Vocabulary size of Train and Test datasets, with the ratio of Test vocabulary overlapping with Train.

A.2 Uset Intent, Musical Attribute Frequency

Overall, the models struggled to classify less frequent labels. To facilitate understanding of the results for each label, we present the average number of each user intent and musical attribute per dialogue.

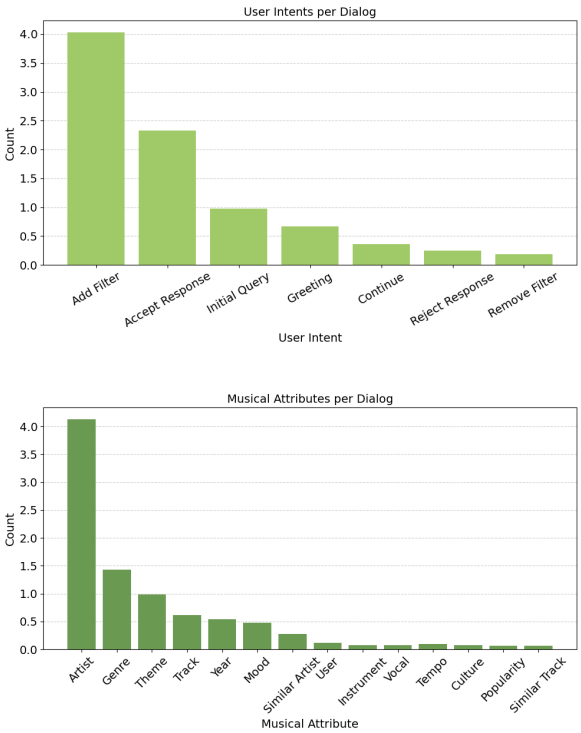

Figure 3: Comparison of Average Counts per Dialog for User Intents and Musical Attributes

A.3 Prompts

A.3.1 Zero-shot Prompt for User Intent

"From the following list of user intents: [initial_query, greeting, add_filter, remove_filter, continue, accept_response, reject_response]. Return only the intents that directly and accurately describe the input text. Ignore any loosely related or vaguely connected intents. Provide the result strictly in a list format. Do not generate any additional text or explanation.

Input: "input text" Output: ["

A.3.2 Few-shot Prompt for User Intent

"From the following list of user intents: [initial_query, greeting, add_filter, remove_filter, continue, accept_response, reject_response]. Return only the intents that directly and accurately describe the input text. Ignore any loosely related or vaguely connected intents. Provide the result strictly in a list format. Do not generate any additional text or explanation.

Example:

Input: "I want to listen recent famous songs."
Output: [add_filter]

Input: "Hello, can you suggest calm music to listen while sleeping?"
Output: [initial_query, greeting, add_filter]

Input: "Wow, I love the vide of these songs!"
Output: [accept_response]

Input: "I think these songs are too fast and loud for me."
Output: [remove_filter, reject_response]

Input: "Can you suggest more like these?"
Output: [continue]

Input: "input text" Output: ["

A.3.3 Zero-shot Prompt for Musical Attribute

"From the following list of musical attributes: [track, artist, year, popularity, culture, similar_track, similar_artist, user, theme, mood, genre, instrument, vocal, tempo]. Return only the attributes that directly and accurately describe the input text. Ignore any loosely related or vaguely connected attributes. Provide the result strictly in a list format. Do not generate any additional text or explanation.

Input: "input text" Output: ["

A.3.4 Few-shot Prompt for Musical Attribute

"From the following list of musical attributes: [track, artist, year, popularity, culture, similar_track, similar_artist, user, theme, mood, genre, instrument, vocal, tempo]. Return only the attributes that directly and accurately describe the input text. Ignore any loosely related or vaguely connected attributes. Provide the result strictly in a list format. Do not generate any additional text or explanation.

Example:

Input: "I want to listen recent famous songs."
Output: [year, popularity]

Input: "Show me faster songs than Ed Sheeran - Shape of You."
Output: [tempo, artist, track]

Input: "Please recommend me some female artists like Rihanna."
Output: [similar_artist, vocal]

Input: "I need exciting hiphop playlist to listen while I exercise."
Output: [mood, genre, theme]

Input: "African songs to listen with my friends."
Output: [culture, user]

Input: "input text" Output: ["