Towards Playlist Continuation Through Large-Scale Context and Audio-Based Music Representations

Shahrzad Shashaani[1,*], Peter Knees[1]

[1] TU Wien, Faculty of Informatics, Vienna, Austria

Abstract

Music recommendation research faces several challenges when modeling the complex relationships between users, items, and the circumstances under which they interact. In spite of access to commercial catalogs and large customer bases, academic research builds its findings on publicly shared datasets. However, these often only contain selected data modalities, limited catalogs, or temporally restricted snapshots of interaction data. Moreover, they might eventually vanish due to licensing issues. A strategy to overcome some of these limitations could consist in learning multimodal representation learning for playlist continuation. For instance, while metadata and interaction data can be used to learn item representations, content-based data can be used to predict representations for tracks where audio is available but interaction data is lacking.

To address this specific case, in this work, as a first pointer into the overall direction, we explore the integration of deep audio features extracted directly from MP3 files for music playlist completion. We first generate track embeddings using a Convolutional Neural Network (CNN) trained on a subset of MP3 files with the Spotify Million Playlist Dataset (MPSD), using pre-learned Word2Vec embeddings as labels. These embeddings serve as item representations in sequential recommender models such as Bidirectional Encoder Representations from Transformers for Sequential Recommendation (BERT4Rec) and Self-Attentive Sequential Recommendation (SASRec). We evaluate four approaches: (1) training the entire recommender model from scratch, (2) incorporating Word2Vec embeddings as item vector in recommenders, (3) incorporating CNN-predicted embeddings only for the last tracks in a playlist while using Word2Vec embeddings for others, and train the remaining model's parameter, and (4) replacing the CNN with a dilated CNN in the third approach. Our experiments show that audio-based features can enhance playlist continuation, especially in cold-start scenarios, while offering potential for improved explainability over traditional metadata-based methods.

Keywords

Music Recommendation, Representation Learning, Item Embeddings, Playlist Completion.

1. Introduction

Music recommenders have become an inseparable feature of modern streaming platforms, aiming to provide personalized suggestions to enhance users' listening experience and engagement.

Traditional approaches are mainly based on session-based models, collaborative filtering, or content-based filtering. While collaborative filtering leverages user-item interactions, it often suffers from cold-start problems where new items or users lack sufficient historical data [1]. Session-based recommenders treat song sequences as item indexes, modeling sequential dependencies without capturing actual audio characteristics [2]. Content-based recommenders, on the other hand, rely on metadata such as genre, artist, and lyrics, which may overlook the rich audio features embedded within the music itself [3].

Beyond this, (especially academic) music recommendation research faces several challenges by building its findings primarily on publicly shared datasets due to a lack of access to commercial catalogs and large customer bases. Research datasets often only contain selected data modalities, limited item coverage, or temporal restrictions both in their availability and data they cover. A strategy to overcome some of these limitations could consist in learning dataset-transcending and cross-modal music representations. For instance—similar to strategies addressing various cold-start problems—metadata and interaction data can be used to learn item representations, while content-based data can

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

[*] Corresponding author.

shahrzad.shashaani@tuwien.ac.at (S. Shashaani); peter.knees@tuwien.ac.at (P. Knees)
0000-0003-4344-2696 (S. Shashaani); 0000-0003-3906-1292 (P. Knees)

be used to predict representations for tracks where audio is available but interaction data lacking.

Following this idea, in this work, we apply Word2Vec to the Spotify Million Playlist Dataset (MPSD) [4] to obtain track embeddings, which are then used as labels to train a CNN that learns audio-based representations from MP3 files. These audio and co-occurrence-based representations offer a promising foundation for more contextually-aware recommendation tasks, such as playlist continuation, where understanding the relationships between songs is essential. This relates to an early approach for addressing cold-start in collaborative filtering by van den Oord et al. [5]. In this work, among further steps, we explore the method proposed by van den Oord et al. for the purpose of bridging contextual representation learning and audio-based prediction and evaluate it in the context of playlist continuation. It needs to be stated that our primary goal at this point is not to outperform the current state of the art in music recommendation, but rather to investigate the potential of training content-based embeddings using a large-scale dataset, while also enabling generalization to previously unseen items. In addition, we conduct a quantitative evaluation, which is missing in the original work by van den Oord et al. Our study aims to fill this gap and serve as a reproducible reference for this fundamental approach. Specifically, our contributions are as follows:

1. we provide a comprehensive quantitative evaluation of the method by van den Oord et al.[5];
2. we extend the model to a cold-start setting via CNN-based content prediction that is applicable to real-world scenarios; and
3. we publicly release the learned embeddings and prediction models to support reproducibility and future research. This enables representation learning even in cases where access to certain music datasets is limited or revoked.

2. Related Work

Recent advances in deep learning have significantly influenced music recommendation. Convolutional Neural Networks (CNNs) are effective at learning audio representations from spectrograms, enabling improved content-based recommendations [6]. The work of van den Oord et al. [5] introduced a deep content-based system that extracts audio embeddings via a CNN, which inspired our approach. While incorporating negative signals has been shown to enhance sequential recommendation by distinguishing relevant from irrelevant items [7], we remain curious about the potential of CNNs in leveraging user-collected playlists for music recommendation, where explicit negative signals are not available. Since CNNs can efficiently extract meaningful audio features directly from raw audio, we believe they may offer valuable insights, especially in cold-start scenarios where new or less popular items are involved. Another approach is to use word embedding techniques like Word2Vec to learn meaningful song representations from playlist co-occurrence patterns [8]. These methods treat playlists as sentences and songs as words, generating vector embeddings that capture contextual relationships.

Predicting and recommending tracks that appropriately fit into an existing playlist, or playlist completion, is a key task in music recommendation. The RecSys 2018 challenge focused on this, aiming to build systems that recommend missing songs for incomplete playlists [4]. Monti et al. [9] combine Recurrent Neural Networks (RNNs) with pre-trained embeddings to model playlist dynamics and semantic relationships. Volkovs et al. [10] present a scalable two-stage pipeline that retrieves and re-ranks candidate songs. Gatzioura et al. [11] propose a hybrid system that combines Latent Dirichlet Allocation with Case-Based Reasoning to recommend songs based on past playlist similarity. Bendada et al. [12] introduce a flexible, scalable represent-then-aggregate strategy that can incorporate Transformers and other techniques. Yang et al. [13] address cold-start and popularity bias with a two-part model combining autoencoders and character-level CNNs using playlist titles.

Although various approaches have addressed playlist continuation, the cold-start problem remains an ongoing challenge. For playlist recommendation in such settings, Chen et al. [14] proposed a multitask learning framework to improve generalization on new playlists and items. More recently, Yurekli et al. [15] introduced a multistage retrieval system that clusters playlists using user-generated titles and applies latent semantic indexing to uncover relationships between tracks and titles. For a new playlist, their model retrieves similar clusters and ranks tracks accordingly. In our work, we extend this line of research by leveraging embeddings learned directly from audio to improve playlist continuation. By integrating deep content-based representations into sequential models, we explore whether audio-derived embeddings provide a more meaningful structure for generating contextually relevant recommendations.

3. Methodology

In this section, we describe the methodology used to integrate deep audio embeddings into sequential recommender models for playlist completion. Our approach consists of several key components: data preprocessing, Word2Vec embeddings generation, audio embedding extraction using CNNs, and the integration of these embeddings into sequential recommenders for playlist continuation. The complete pipeline is shown in Figure 1.

3.1. Dataset and Preprocessing

We used MPSD, which includes track metadata and playlists of varying lengths. Since MPSD does not provide audio information, we utilize a subset of MP3 files collected using the Spotify API in Fall 2022. More details about the datasets can be found in Data Preperation and Statistics. The preprocessing pipeline is as follows:

- Using MPSD playlists as input to train a Word2Vec model, which provides semantic track embeddings for use in both CNN (as labels) and recommender models (as item vectors).
- Identifying overlapping tracks between MPSD and our collected audio dataset. These tracks are then split into training and testing sets, with 1,200,455 samples used for training and 187,507 samples reserved for testing.

3.2. Word2Vec: Create Track Embeddings

In this step, we apply the Word2Vec algorithm to the MPSD dataset to generate track embeddings. The dataset consists of numerous playlists, each containing a sequence of tracks. We treat each playlist as a ”sentence” and each track index (unique track ID) as a ”word”. Given this input, Word2Vec learns an embedding representation for each track based on its co-occurrence with other tracks in playlists.

Word2Vec captures meaningful relationships between tracks, as songs frequently appearing together in playlists are mapped to similar embedding spaces. This property makes the learned embeddings useful as labels for training our CNN model, enabling the CNN to predict track representations based on their audio features. Also, these embeddings have meaningful information to be used as item representations in recommender models.

3.3. CNN: Predicting Item Representations for the Recommender System

Inspired by van den Oord et al. [5], we employ a CNN to predict audio-based track embeddings. The overlapping tracks between MPSD and our collected audio dataset serve as input data, and by applying Word2Vec on MPSD, we have embeddings for all the corresponding audio data. Rather than using raw audio directly, we first convert MP3 files into spectrograms, which represent the time-frequency distribution of audio signals.

Unlike [5], which randomly sampled 3-second windows from tracks, we use the full 30-second audio clips, ensuring that the model captures a broader range of musical characteristics. Additionally, we experiment with dilated CNNs, which expand the receptive field without increasing the number of parameters. Dilated CNNs allow the network to capture both local and long-range dependencies in audio signals, potentially improving representation learning by preserving hierarchical structures.

The network consists of multiple convolutional layers followed by fully connected layers, with the final output being a low-dimensional audio embedding. The architecture is designed to learn

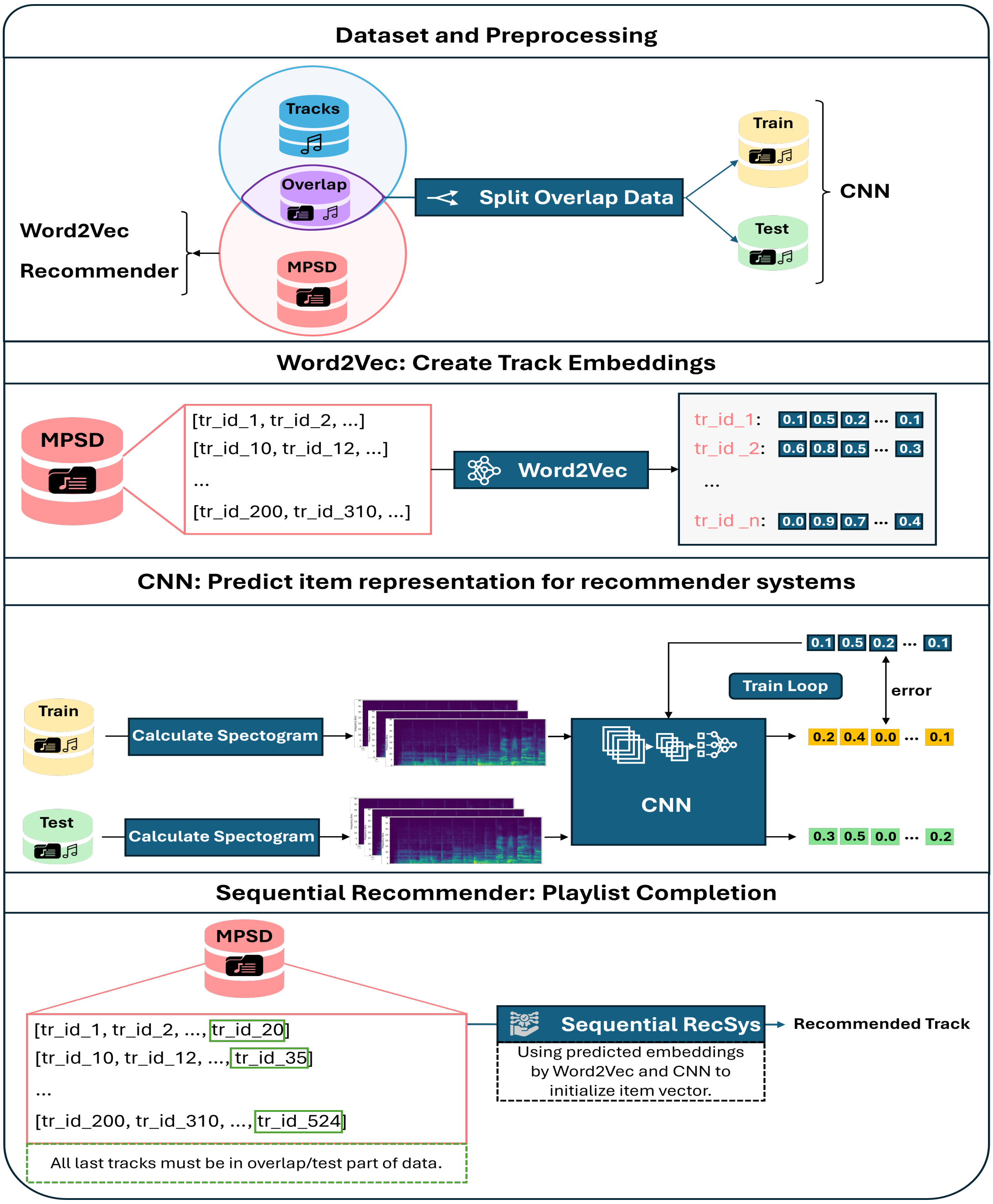
Dataset and Preprocessing
Tracks
Overlap
MPSD
Split Overlap Data
Train
Test
CNN
Word2Vec
Recommender
Word2Vec: Create Track Embeddings
MPSD
[tr_id_1, tr_id_2, ...]
[tr_id_10, tr_id_12, ...]
...
[tr_id_200, tr_id_310, ...]
Word2Vec
tr_id_1: 0.1 0.5 0.2 ... 0.1
tr_id _2: 0.6 0.8 0.5 ... 0.3
...
tr_id _n: 0.0 0.9 0.7 ... 0.4
CNN: Predict item representation for recommender systems
Train
Calculate Spectogram
Test
Calculate Spectogram
CNN
Train Loop
0.1 0.5 0.2 ... 0.1
error
0.2 0.4 0.0 ... 0.1
0.3 0.5 0.0 ... 0.2
Sequential Recommender: Playlist Completion
MPSD
[tr_id_1, tr_id_2, ..., tr_id_20]
[tr_id_10, tr_id_12, ..., tr_id_35]
...
[tr_id_200, tr_id_310, ..., tr_id_524]
All last tracks must be in overlap/test part of data.
Sequential RecSys
Recommended Track
Using predicted embeddings by Word2Vec and CNN to initialize item vector.

Figure 1: Pipeline of the method proposed for bridging contextual representation learning and audio-based prediction, evaluated in the context of playlist continuation.

representations that capture meaningful audio patterns. The model is trained using the pre-learned Word2Vec embeddings as labels. This ensures that the learned audio-based embeddings align with the semantic information captured in playlist co-occurrence data.

3.4. Sequential Recommendation: Playlist Completion

In the final step, we incorporate deep audio embeddings into sequential recommender models to predict the next track in a playlist. Our primary goal is to evaluate whether content-based embeddings improve recommendations in cold-start scenarios, where tracks lack interaction history. To achieve this, we initialize item embeddings in the recommender system using the Word2Vec representations, except for the last track in each playlist. The last track's item embedding is replaced with the CNN-predicted representation. To prevent data leakage, we ensure that all last tracks belong to the test set and have never been used in CNN training. For comparison, we also evaluate the performance of models when their item vector is initialized randomly (similar to the original models' training method), and when it is completely replaced with Word2Vec representations. We evaluate two sequential recommenders:

1. **Self-Attentive Sequential Recommendation (SASRec) [16]:** a self-attentive model capturing sequence dependencies. We first train the model from scratch as the baseline and then replace item embeddings with Word2Vec embeddings and also with CNN-predicted deep audio embeddings, only for the last tracks, to examine their impact.
2. **Bidirectional Encoder Representations from Transformers (BERT4Rec) [17]:** a transformer-based sequential recommender that treats playlists as sequences and predicts missing tracks. We use a similar training process as SASRec.

We used the basic implementations of the recommender models provided by [18], and made modifications based on our proposed algorithm.

4. Experiments

4.1. Data Preperation and Statistics

MPSD is a large-scale collection of user-generated playlists from 2010 to 2017, comprising 2,262,292 tracks and 1,000,000 playlists, providing valuable sequence data for playlist modeling, however lacking audio features. To bridge this gap, we built an audio dataset using a previously collected set of MP3 files using the Spotify API. Our audio subset represents about 61% overlap (1,375,634 tracks and 988,585 playlists) with MPSD, which is a significant portion of the MPSD track list. This subset enables us to learn deep content-based embeddings from raw audio and integrate them into recommender models. To assess the coverage and distribution of our audio subset, Figure 2 presents a t-SNE plot of the Word2Vec embeddings trained on MPSD, showing that the audio subset is well-distributed across the embedding space. This indicates that our subset is representative of the broader playlist structure.

To extract meaningful representations from raw audio, we converted MP3 files into log-compressed mel-spectrograms using the Librosa library [19]. The spectrograms were generated with a sampling rate of 22,050 Hz, 128 mel components, an FFT window size of 1,024, and a hop length of 512. These parameters were chosen to balance computational efficiency with sufficient frequency resolution, ensuring the spectrogram captures both low and high-frequency musical characteristics.

4.2. Models and Training

4.2.1. Word2Vec

To learn track embeddings from playlists, we trained a Word2Vec model on the full MPSD dataset. We tested several embedding sizes (40, 50, 100, 400), observing that the overall embedding space structure remained almost stable in t-SNE visualization. However, we selected 400 for the final experiments, since larger embedding dimensions allow the model to capture more nuanced relationships between tracks, especially in high-dimensional feature spaces like music. Although smaller dimensions can still represent Word2Vec track similarities effectively, they may lose the important details necessary for CNNs and recommenders.

Figure 3 shows the t-SNE result after applying Word2Vec with an embedding size of 400. With a closer look, we can identify different possible clusters and specify frequent artists in each of the highlighted regions. We only plot track embeddings of the most frequent artist names within each cluster in the smaller plots. Interestingly, some regions demonstrate natural closeness: for example, the Green (Reggae) and Red (Latin, Reggae) regions are adjacent, reflecting the stylistic and cultural influences between these genres. Similarly, Yellow (Jazz, Musicals) and Purple (Classical) are positioned closely, emphasizing their shared instrumental and compositional characteristics. This visualization also captures meaningful relationships in the data: since Word2Vec was trained on track IDs, the

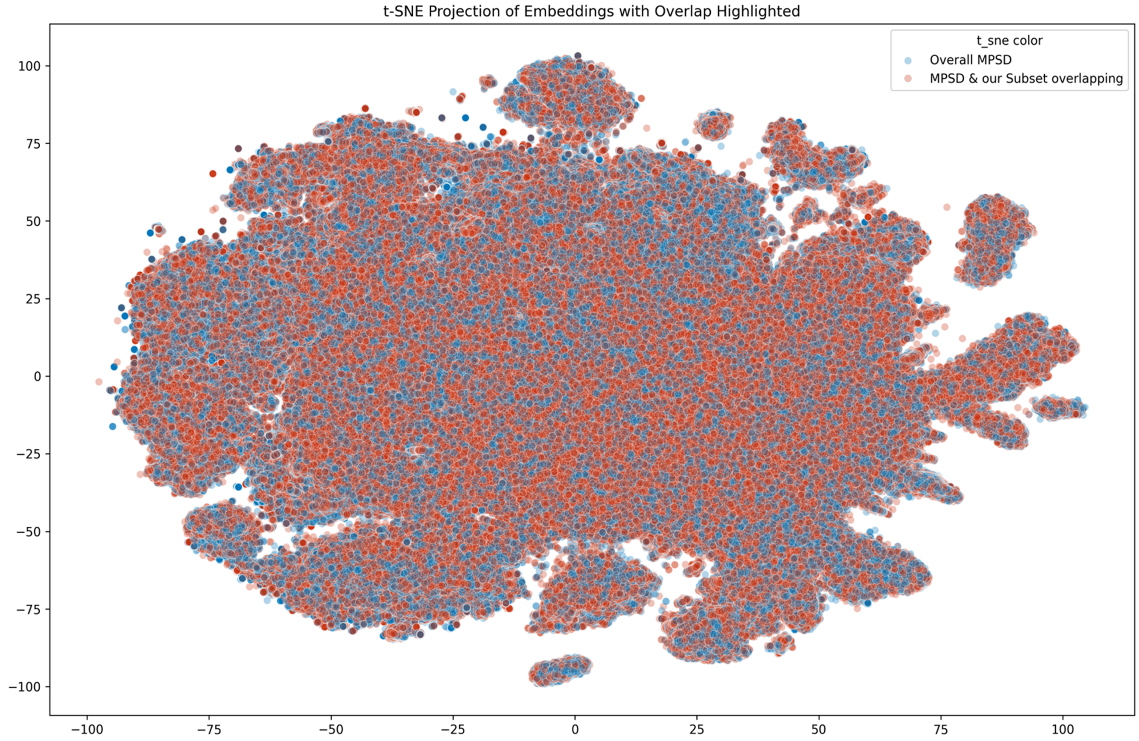

Figure 2: t-SNE visualization of MPSD track embeddings generated with Word2Vec. The overlapping tracks with our audio subset are highlighted in red.

resulting clusters reflect artist associations. As a result, these regions may reveal possible dominant genre tendencies, offering insight into how Word2Vec embeddings encode playlist information—useful for recommendation explainability.

4.2.2. CNN

Our CNN model is designed to learn deep content-based track representations from mel-spectrograms. The architecture consists of four convolutional layers (128->256->256->512->512 channels), each followed by non-overlapping max pooling layers to progressively reduce temporal resolution while preserving essential features. All convolutions use a stride of 1 to capture fine-grained patterns, and in the dilated variant, the last two convolutional layers employ dilation rates of 2 and 4, respectively, to expand the receptive field without increasing kernel size. Inspired by [5], we apply a global temporal pooling layer after the final convolution to aggregate temporal information using mean, max, and L2-norm pooling functions. This pooled representation is passed through three fully connected layers, reducing the dimensionality to a 400-dimensional track embedding.

4.2.3. Recommender Systems

For the SASRec model, we set the hidden units to 400 to match the dimension of the learned embeddings. The model consists of two transformer blocks and uses one attention head. A dropout rate of 0.1 was applied to regularize the model. In the BERT4Rec model, the model's hidden size was also set to 400 to match the learned embeddings, with two hidden layers and two attention heads, due to resource limitations. Both models were trained using the Adam optimizer, with these hyperparameters chosen to balance model capacity and training efficiency.

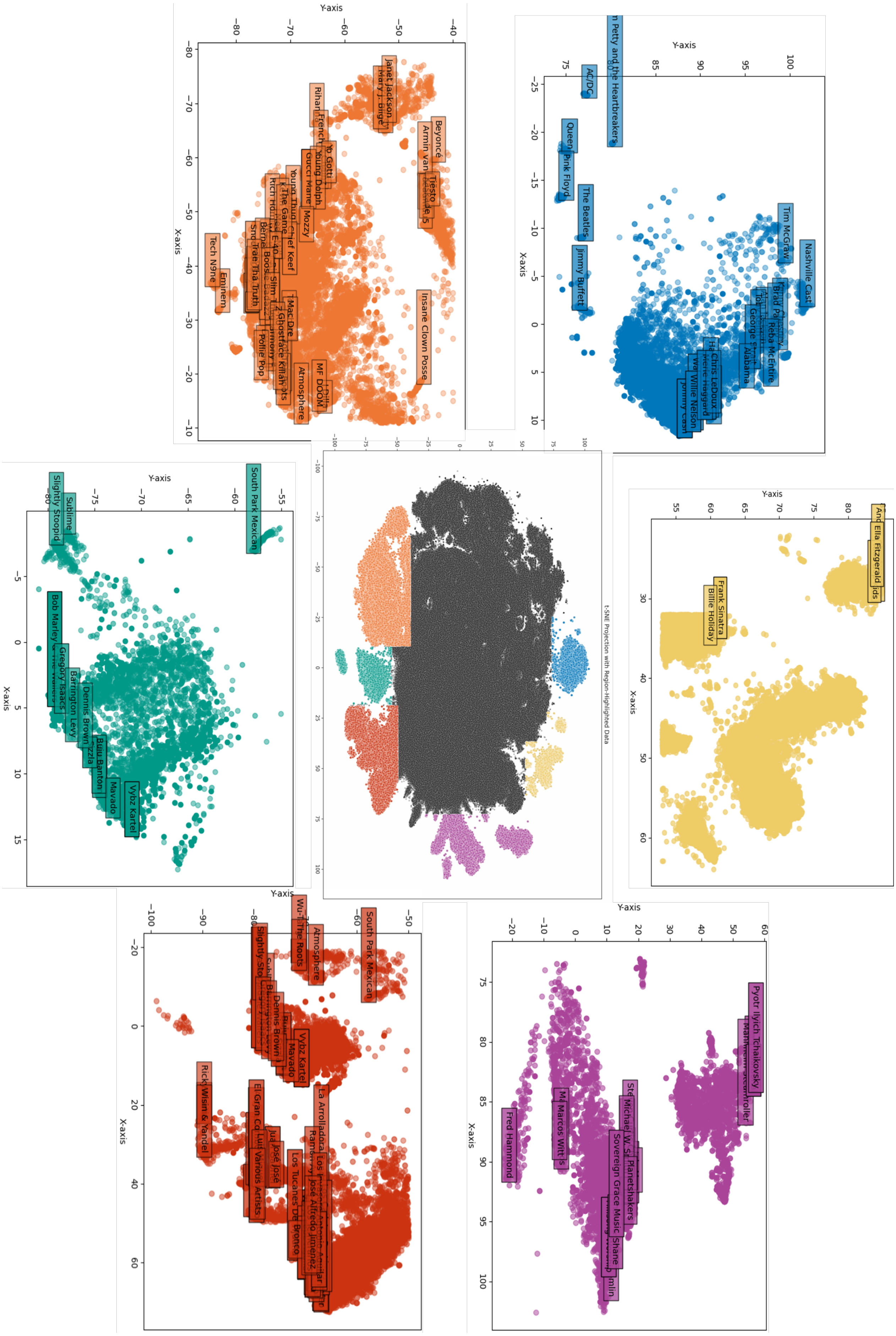

Figure 3: t-SNE visualization of Word2Vec embeddings. We observe genre-based groupings, such as: Blue (Country, Rock), Orange (Hip-Hop, Rap, R&B), Green (Reggae), Red (Latin, Reggae), Purple (Classical), and Yellow (Jazz, Musicals), where the most frequent artists are shown in those regions.

5. Results and Discussion

We evaluate our playlist continuation approach using three standard metrics: NDCG@K, which measures ranking quality by considering the position of relevant tracks; HR@K, which checks if at least one ground truth track appears in the top-K recommendations; and MAP@K, which computes the mean precision of relevant tracks at different cut-offs. Table 1 presents the performance of SASRec and BERT4Rec models under different initialization strategies. Since no prior work has evaluated SASRec or BERT4Rec on MPSD, we trained both models from scratch as baselines for fair comparison. Additionally, to ensure a valid evaluation, we cut the playlists—only when necessary—so that the final track in each sequence belongs to the CNN test set. This ensures that the CNN never sees these track embeddings during training and prevents data leakage. This adjustment may also lead to slightly different results compared to using the original, unmodified dataset.

Table 1
Performance comparison of different models using NDCG@k, HR@k, and MAP@k metrics.

Model	Metric	@1	@5	@10	@20	@50	@100	@200
SASRec (Training from Scratch)	NDCG	0.003	0.008	0.011	0.015	0.021	0.027	0.034
	HR	0.003	0.013	0.022	0.036	0.069	0.105	0.155
	MAP	0.003	0.006	0.008	0.009	0.009	0.010	0.010
SASRec (Word2Vec)	NDCG	**0.028**	**0.054**	**0.066**	**0.078**	**0.093**	**0.105**	**0.116**
	HR	**0.029**	**0.079**	**0.115**	**0.161**	**0.241**	**0.313**	**0.394**
	MAP	**0.028**	**0.046**	**0.051**	**0.054**	**0.057**	**0.058**	**0.058**
SASRec (Word2Vec + CNN)	NDCG	0.022	0.042	0.053	0.064	0.078	0.089	0.101
	HR	0.023	0.063	0.096	0.137	0.208	0.275	0.351
	MAP	0.022	0.035	0.042	0.046	0.049	0.051	0.051
SASRec (Word2Vec + Dilated-CNN)	NDCG	0.013	0.035	0.047	0.058	0.071	0.083	0.095
	HR	0.014	0.054	0.085	0.124	0.196	0.262	0.336
	MAP	0.013	0.030	0.038	0.042	0.046	0.048	0.049
BERT4Rec (Training from Scratch)	NDCG	0.007	0.018	0.024	0.031	0.043	0.053	0.064
	HR	0.007	0.028	0.048	0.077	0.137	0.198	0.273
	MAP	0.007	0.014	0.017	0.019	0.021	0.022	0.022
BERT4Rec (Word2Vec)	NDCG	0.009	0.013	0.022	0.027	0.035	0.047	0.057
	HR	0.010	0.020	0.037	0.051	0.082	0.141	0.202
	MAP	0.009	0.011	0.013	0.015	0.019	0.020	0.022
BERT4Rec (Word2Vec + CNN)	NDCG	0.010	0.012	0.021	0.027	0.033	0.048	0.059
	HR	0.010	0.020	0.035	0.052	0.080	0.142	0.204
	MAP	0.010	0.011	0.012	0.015	0.018	0.020	0.023
BERT4Rec (Word2Vec + Dilated-CNN)	NDCG	0.008	0.011	0.021	0.025	0.031	0.044	0.053
	HR	0.008	0.020	0.034	0.050	0.077	0.138	0.193
	MAP	0.008	0.010	0.012	0.014	0.016	0.019	0.021

For SASRec, incorporating Word2Vec embeddings significantly improves performance over the baseline. For example, SASRec (Word2Vec) gets the highest NDGC@1 while the baseline only achieves the same result in NDGC@100. While augmenting Word2Vec with CNN or Dilated-CNN features, shows slightly lower performance than Word2Vec alone, these models still outperform the baseline across all metrics, confirming that content-aware representations enhance sequential recommendation quality. In contrast, BERT4Rec shows more mixed results. The baseline generally outperforms the others, except at @1, suggesting that BERT4Rec benefits less from external item initializations. This may be because the pre-trained Word2Vec/CNN embeddings may not align well with the transformer-based architecture of BERT4Rec. Besides, increasing the number of hidden layers and attention heads can increase its performance.

Importantly, initializing item vectors with Word2Vec or CNN-based features not only improves performance in most cases, but also accelerates training convergence. This highlights a practical benefit of leveraging pre-trained representations in deep sequential recommender systems.

6. Conclusion

In this study, we experimentally explored using content-based embeddings, extracted either from Word2Vec or CNNs, to enhance sequential playlist recommendation. Our results show that SASRec significantly benefits from these embeddings, with all proposed variants outperforming the baseline model. While Word2Vec embeddings yield the best overall performance, CNN-based features remain essential in cold-start settings, where new tracks lack listening history but have available audio content. In contrast, BERT4Rec showed limited gains from content-based initialization, highlighting that its self-attention mechanisms may already capture sufficient contextual information without external embeddings initialization. Nevertheless, its performance could potentially be improved by tuning architectural parameters, such as the number of hidden layers and attention heads.

Importantly, our work revisits and builds upon the approach introduced by van den Oord et al.[5], applying their method to a new, large-scale playlist dataset and extending it with a quantitative evaluation, which was missing in the original paper. Our goal is not to introduce a novel task, but to assess the potential of deep content-based embeddings in different recommendation scenarios—particularly in addressing the cold-start problem—and to provide learned item representations that can serve as a reference for future work.

A key strength of our approach lies in its ability to address the cold-start problem. By representing unseen tracks using CNN-predicted embeddings, without requiring interaction data, we can generate recommendations even for new or unpopular songs. This highlights the value of content-aware embeddings in playlist completion, offering a practical solution not only in cold-start scenarios, but a strong baseline for future work on explainability.

For future work, several directions can be explored to further enhance performance and generalizability. First, experimenting with alternative sequential models such as GRU4Rec may yield deeper insight into how different architectures interact with content-based embeddings. Second, improving the CNN architecture could enhance embedding quality. Third, comparing our approach with other recent baselines, such as [20], would help measure its effectiveness better. Finally, leveraging the learned embeddings in new recommendation contexts may reveal their potential for explainability in music recommendation by linking predictions to interpretable audio features. Beyond these directions, future work could also investigate dataset-transcending music representations to resolve misalignments across entities, modalities, and semantic concepts. As this paper indicates, multi-modal and multi-source representation learning can serve as a viable vehicle for overcoming some of the research challenges resulting from a fragmented dataset landscape.

In the bigger picture, the goal of this and the intended follow-up work is primarily to establish baseline results of existing and often referenced pipelines and variations thereof in specific application tasks and evaluation settings. A central goal is to leverage the (unfortunately vanishing) resources available to music recommendation research and to identify—ideally—general representations, that can be reused and built upon in future recommendation tasks by publicly sharing them with the research community. Learned representations might be a way to overcome the limitations the community is facing and help to sustain the research area of music RecSys.

Acknowledgments

This research was funded in whole or in part by the Vienna Science and Technology Fund (WWTF) [Grant ID: 10.47379/DCDH001]. For open access purposes, the author has applied a CC BY public copyright license to any author-accepted manuscript version arising from this submission.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (2009) 30–37.

[2] B. Hidasi, A. Karatzoglou, L. Baltrunas, D. Tikk, Session-based recommendations with recurrent neural networks, arXiv preprint arXiv:1511.06939 (2015).

[3] M. Schedl, H. Zamani, C.-W. Chen, Y. Deldjoo, M. Elahi, Current challenges and visions in music recommender systems research, International Journal of Multimedia Information Retrieval 7 (2018) 95–116.

[4] C.-W. Chen, P. Lamere, M. Schedl, H. Zamani, Recsys challenge 2018: Automatic music playlist continuation, in: Proceedings of the 12th ACM Conference on Recommender Systems, 2018, pp. 527–528.

[5] A. Van den Oord, S. Dieleman, B. Schrauwen, Deep content-based music recommendation, Advances in neural information processing systems 26 (2013).

[6] K. Choi, G. Fazekas, M. Sandler, K. Cho, Convolutional recurrent neural networks for music classification, in: 2017 IEEE International conference on acoustics, speech and signal processing (ICASSP), IEEE, 2017, pp. 2392–2396.

[7] P. Seshadri, S. Shashaani, P. Knees, Enhancing sequential music recommendation with negative feedback-informed contrastive learning, in: Proceedings of the 18th ACM Conference on Recommender Systems, 2024, pp. 1028–1032.

[8] O. Barkan, N. Koenigstein, Item2vec: neural item embedding for collaborative filtering, in: 2016 IEEE 26th international workshop on machine learning for signal processing (MLSP), IEEE, 2016, pp. 1–6.

[9] D. Monti, E. Palumbo, G. Rizzo, P. Lisena, R. Troncy, M. Fell, E. Cabrio, M. Morisio, An ensemble approach of recurrent neural networks using pre-trained embeddings for playlist completion, in: Proceedings of the ACM Recommender Systems Challenge 2018, 2018, pp. 1–6.

[10] M. Volkovs, H. Rai, Z. Cheng, G. Wu, Y. Lu, S. Sanner, Two-stage model for automatic playlist continuation at scale, in: Proceedings of the ACM Recommender Systems Challenge 2018, 2018, pp. 1–6.

[11] A. Gatzioura, J. Vinagre, A. M. Jorge, M. Sanchez-Marre, A hybrid recommender system for improving automatic playlist continuation, IEEE Transactions on Knowledge and Data Engineering 33 (2019) 1819–1830.

[12] W. Bendada, G. Salha-Galvan, T. Bouabça, T. Cazenave, A scalable framework for automatic playlist continuation on music streaming services, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023, pp. 464–474.

[13] H. Yang, Y. Jeong, M. Choi, J. Lee, Mmcf: Multimodal collaborative filtering for automatic playlist continuation, in: Proceedings of the ACM Recommender Systems Challenge 2018, 2018, pp. 1–6.

[14] D. Chen, C. S. Ong, A. K. Menon, Cold-start playlist recommendation with multitask learning, arXiv preprint arXiv:1901.06125 (2019).

[15] A. Yürekli, C. Kaleli, A. Bilge, Alleviating the cold-start playlist continuation in music recommendation using latent semantic indexing, International Journal of Multimedia Information Retrieval 10 (2021) 185–198.

[16] W.-C. Kang, J. McAuley, Self-attentive sequential recommendation, in: 2018 IEEE international conference on data mining (ICDM), IEEE, 2018, pp. 197–206.

[17] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, P. Jiang, Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer, in: Proceedings of the 28th ACM international conference on information and knowledge management, 2019, pp. 1441–1450.

[18] A. Klenitskiy, A. Vasilev, Turning dross into gold loss: is bert4rec really better than sasrec?, in: Proceedings of the 17th ACM Conference on Recommender Systems, 2023, pp. 1120–1125.

[19] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, O. Nieto, librosa: Audio and music signal analysis in python., SciPy 2015 (2015) 18–24.

[20] M. C. McCallum, F. Korzeniowski, S. Oramas, F. Gouyon, A. F. Ehmann, Supervised and unsuper-

vised learning of audio representations for music understanding, arXiv preprint arXiv:2210.03799 (2022).