Evaluating the Impacts of Combining Embedding Extraction Techniques with Transformer Models for Music Recommendation

Vinicius S. Simm[1], **Matheus H. C. Leme**[1], **Douglas R. Tanno**[1], **Marcos A. Domingues**[1]

[1]Department of Informatics – State Univerity of Maringá – Maringá – PR – Brazil

{pg404811,pg55308,pg404806,madomingues}@uem.br

Abstract. *This paper investigates the impact of different embedding strategies on Transformer-based models for session-based music recommendation. We compared two approaches: a standard Transformer model that learns item embeddings internally, and a second model that uses pre-trained embeddings from Word2Vec (CBOW and Skip-Gram), generating recommendations via cosine similarity. We evaluated on two music datasets, Music4All and Xiami, and our results highlight a trade-off between accuracy and diversity. The Transformer model achieves a significantly higher Hit Rate but shows a stronger popularity bias and lower catalog coverage. On the other hand, the approach using pre-trained embeddings shows greater coverage and diversity but lower accuracy.*

1. Introduction

In recent years, access to music, movies, and books has been increasingly facilitated by the widespread adoption of streaming services. With the vast volume of available items, recommender systems have become essential tools to help users navigate online platforms. In the music domain, this challenge is especially complex due to the wide variety of genres, styles, and contextual factors such as social and geographical influences on user preferences [van den Oord et al. 2013].

Several studies have pointed out that music consumption is highly contextual, and the same user may have different preferences depending on their situation or mood. A common approach to capturing such short-term preferences is to segment user interactions into sessions, defined as sequences of songs accessed by the same user without prolonged interruptions, from which intent can be inferred in real time [Wang et al. 2017]. Sessions may or may not include user identification, leading to two distinct types of problems: session-based recommendation, where the model analyzes anonymous sessions without relying on past history [Jannach et al. 2022]; and session-aware recommendation, which incorporates cross-session data for identified users, enabling more personalized suggestions [Latifi et al. 2021].

Within sessions, items are typically represented using unique identifiers, which fail to capture semantic or structural relationships among them. To address this, items are commonly mapped into continuous vector spaces, known as embeddings. These low-dimensional representations capture latent relationships, allowing recommender systems to reason about item similarity based on co-occurrence patterns and proximity in the embedding space.

Transformer-based models have gained significant attention in the field of recommender systems due to their ability to model long-range dependencies and sequential patterns. In the context of session-based recommendation, they have been used to process sequences of user interactions and generate more accurate predictions by capturing complex temporal dynamics [Vaswani et al. 2017].

Despite their strong performance in accuracy, end-to-end Transformer models are known to suffer from popularity bias, disproportionately recommending a small subset of frequently played items [Abdollahpouri et al. 2019]. This behavior can limit catalog exploration and reduce user satisfaction over time.

In this work, we propose investigating how different modeling strategies within Transformer-based architectures impact recommendation outcomes in a session-based music scenario. Our focus is to evaluate three key aspects: accuracy, popularity bias, and coverage. We compare two Transformer-based approaches: one where the model learns item embeddings internally and produces a probability distribution over all possible items; and another one where fixed, externally computed embeddings are used as input, and the model outputs a predicted embedding vector. In the second case, recommendations are obtained by ranking candidate items based on their cosine similarity with the output vector. By contrasting these two setups, we aim to understand how the choice of item representation and output formulation influences not only the model's hit rate, but also its tendency to favor popular items and its ability to recommend diverse content across the catalog.

The remainder of this paper is organized as follows. Section 2 discusses relevant previous research. Section 3 describes the research methodology adopted in this work, providing detailed information on the datasets, representation capture models and the transformer model, experimental and evaluation settings, and hyperparameter optimization. In Section 4, we present and discuss the main findings obtained with our empirical evaluation. Finally, Section 5 concludes the paper and highlights future directions.

2. Related Work

Recommender systems have been developed to help users navigate large item catalogs, even in situations where they have little or no prior exposure to the available content. One of the central challenges in this domain is striking a balance between user satisfaction and business goals, such as increasing sales, engagement, or customer retention. At the same time, these systems raise concerns about their potential to reinforce unhealthy consumption patterns or limit exposure to diverse content [Jannach and Zanker 2022].

These systems are widely adopted across e-commerce platforms, where they suggest products like books, clothing, or electronics, as well as in streaming services, where they recommend music, movies, and TV shows. Music recommendation, however, presents unique characteristics compared to other domains: songs are typically shorter than items such as films or travel packages; the item catalog is extraordinarily large; poor recommendations tend to have less negative impact; rich content-based features can be extracted directly from the songs; repeated recommendations are often acceptable or even desirable; and consumption tends to occur sequentially and often passively [Schedl 2019].

To enable recommendation models to understand better and compare items, it is common to represent each item as a dense vector in a low-dimensional continuous space,

known as an embedding. These embeddings are constructed so that items with similar usage patterns are positioned close to each other in the vector space. This representation allows the model to capture latent semantic relationships among items, derived from co-occurrence rather than explicit metadata or user profiles [Koren et al. 2009].

Recently, Transformer-based models have gained significant attention in the field of recommender systems due to their ability to capture complex dependencies and contextual relationships within sequential data. By using self-attention mechanisms, Transformers can model interactions among all elements in a session, regardless of their relative positions. This makes them particularly effective in session-based recommendation tasks, where the order and co-occurrence of items encode valuable behavioral signals [Vaswani et al. 2017].

A common characteristic of many Transformer-based recommendation models is the use of a classification-based output layer, where the model predicts the next item by computing a probability distribution over the entire item vocabulary. This formulation may be straightforward and effective for optimizing ranking metrics, such as hit rate, but it tends to amplify popularity bias and reduce the coverage percentage [Abdollahpouri et al. 2019].

An alternative to classification-based outputs is to formulate recommendations as a retrieval problem in the embedding space. Instead of outputting a probability distribution over item classes, the model generates a target embedding vector intended to be close to the embedding of the next relevant item. This approach shifts the focus from item frequency to geometric proximity, which can reduce the influence of item popularity and promote greater coverage of the item space. Cosine similarity, in particular, has been widely adopted in embedding-based retrieval tasks due to its ability to measure angular closeness independent of vector magnitude, making it well-suited for applications in recommendation and natural language processing [Mikolov et al. 2013a].

Another study compared session-average, LSTM, and Transformer models for session-based music recommendation, all using pre-trained Word2Vec embeddings and cosine similarity for item retrieval [Simm et al. 2025]. Their results confirmed the Transformer's superiority in Hit Rate, but at the cost of reduced catalog coverage. Thus, the authors identified a direct comparison with architectures that predict items via classification layers as a future direction, which the present work directly addresses.

In this work, we aim to explore the capabilities of Transformer models trained end-to-end, as well as with separately computed embeddings and cosine distance-based recommendation. We evaluate their performance in terms of accuracy and coverage, among other metrics, in order to shed light on the potential existence of an intrinsic bias toward more popular songs.

3. Methodology

This section details the datasets used in this research, as well as the methodology adopted to conduct the experiments, to support the reproducibility and validity of the results.

3.1. Datasets

For this work, we used the Music4All and Xiami datasets, both from the music domain. The first, Music4All [Santana et al. 2020], was recently developed by collecting play-

back history from the Last.fm platform[1]. This dataset includes over 15,000 users and their respective histories, totaling more than 100,000 music tracks. The Xiami dataset [Wang et al. 2016], on the other hand, was derived from the Xiami Music online service[2] and contains approximately 360,000 tracks and 4,200 users.

These two datasets exhibit distinct characteristics that allow us to evaluate the performance of the proposed models in different scenarios within the same application domain. Music4All has a significantly larger number of users, although the average number of tracks per user is lower—around 361 tracks, of which approximately 184 are unique. In contrast, in the Xiami dataset, each user consumes an average of about 1,000 tracks, with approximately 370 of them being unique.

These structural differences can significantly impact the performance of the models. In the case of Xiami, although users exhibit a higher volume of interactions, there is a tendency to listen to the same tracks repeatedly. On the other hand, Music4All offers a broader user base with more diverse music consumption profiles, which may pose an additional challenge for recommender systems.

Using the timestamp records of user-song interactions available in both datasets, we processed the data to identify individual consumption sessions for each user. To accomplish this, a 30-minute interval was adopted to partition users' interaction sequences, as suggested by [Wang et al. 2021]. This segmentation aims to ensure contextual independence between consecutive sessions. Additionally, to ensure the quality and relevance of the analyzed sessions and to focus the evaluation on users with more consistent behavior patterns, sessions containing fewer than three tracks were excluded. Finally, the training and test sets were derived from the processed data using the following strategy: the last (n-th) track interacted with in each session for all users was reserved for the test set, while the previous $(n-1)$ tracks were used to train the models [Ludewig et al. 2021].

3.2. Embedding Generation Models

To represent songs as feature vectors in a continuous space with traditional approaches, or embeddings, this work adopts the Continuous Bag-of-Words (CBOW) and Skip-Gram models from the framework presented by [Mikolov et al. 2013a]. These models were originally developed for the field of Natural Language Processing. However, they have proven effective for learning co-occurrence-based representations in various applications [Mikolov et al. 2013a]. In parallel with their original purpose, user-interacted music sessions are treated as sentences, and each song plays the role of a word. The objective, therefore, is to learn embeddings that capture the contextual relationships between songs, such as those that tend to be listened to together.

Originally, the goal of the CBOW model is to predict a target word (item) based on its surrounding context. As described by [Mikolov et al. 2013b], this model computes the average of the context item vectors, which is then fed into a simple neural network trained to predict the target item. In our implementation, for a given target song within a session, CBOW considers the songs played before and after it as the context. The embeddings of these context songs are combined, and the model is optimized to maximize the probability of predicting the target song. This approach results in embeddings that

[1]https://www.last.fm
[2]http://www.xiami.com

place songs frequently surrounded by similar contexts in close proximity in the vector space, thereby capturing a notion of contextual similarity.

The Skip-Gram model operates in the inverse manner to CBOW, that is, a given word (item) is used to predict its surrounding context. Given an input word, the model is trained to predict the words appearing in its vicinity within the same sentence. This architecture is promising for capturing semantic representations of items, being particularly effective on large datasets and in producing high-quality embeddings for less frequent items [Levy and Goldberg 2014]. In our context, music sessions are used to train the Skip-Gram model, aiming to optimize embeddings so that a song can accurately predict which songs users listen to before and after it. In this way, we construct embeddings that can share similar contexts.

3.3. Transformer

The Transformer model, introduced by [Vaswani et al. 2017], marked a paradigm shift in sequence modeling by eliminating the need for recurrence and convolution. Instead, it relies entirely on an attention mechanism to capture global dependencies between elements of the input and output. A key innovation is the self-attention mechanism, which enables the model to dynamically assign importance weights to all positions in a sequence when processing each individual element.

Unlike traditional sequential models such as Recurrent Neural Networks, which may struggle to capture long-term dependencies, the Transformer processes the entire session as a whole [Kang and McAuley 2018]. In our session-based music recommendation context, the self-attention mechanism computes the attention weight for each pair of songs within a session, enabling the model to consider the influence of all previously listened-to songs when predicting the next song, regardless of their position in the sequence.

Transformer models can be used in two distinct ways. In the first end-to-end approach, the model learns its own embeddings during training and uses them in a final classification layer to predict the next item from all available items. In the second approach, the Transformer receives pre-trained, static embeddings generated by another technique and uses them to produce embeddings as output, rather than directly predicting the next item [Vasile et al. 2016].

3.4. Experimental Setup

To investigate how different architectural choices affect the performance of session-based recommendation models, we designed an experiment comparing two distinct strategies for generating recommendations using Transformer-based architectures. Both approaches use the same model backbone but differ in two critical aspects: how item embeddings are obtained and used during training, and how recommendations are generated at inference time. The goal of this comparison is to understand how the embedding strategy and output formulation influence key performance indicators, particularly in terms of accuracy, popularity bias, and item coverage.

The Figure 1 presents the two strategies investigated in our research. In the first approach, illustrated by the red path and arrows, we adopted a traditional end-to-end training strategy. The model receives input sequences composed of song IDs representing user sessions. These IDs are mapped to learnable embedding vectors, which are optimized

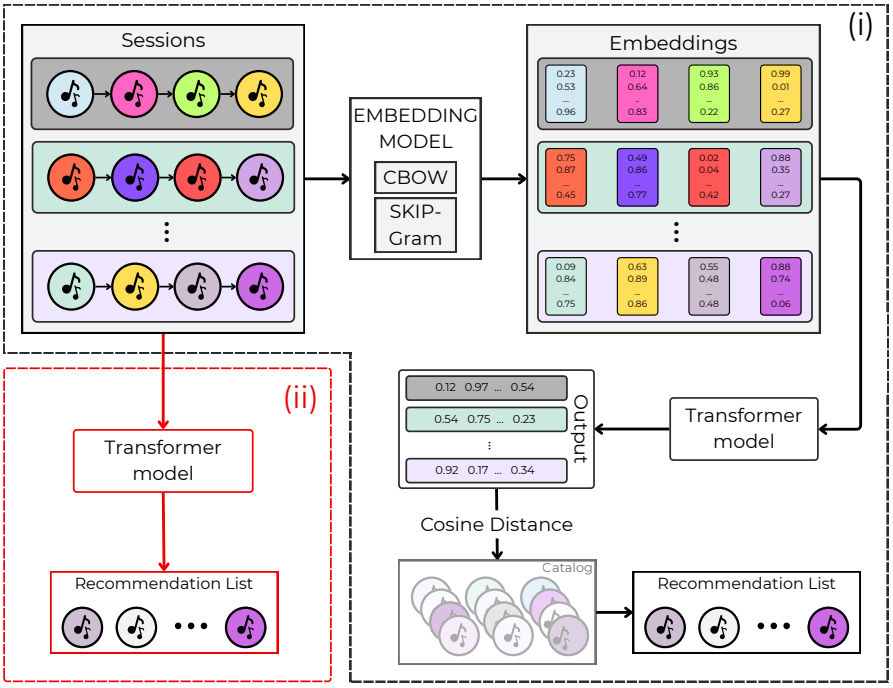

Figura 1. Flowchart of the data processing pipeline, illustrating (i) the models with separate Embedding and recommendation components and (ii) the model with integrated Embedding within the Transformer.

jointly with the Transformer during training. The model outputs a probability distribution over the item vocabulary using a dense classification layer, and the item with the highest predicted probability is selected as the recommendation.

In the second approach, corresponding to the black path and arrows in the Figure 1, we decouple the embedding learning from the recommendation model. First, we train two Word2Vec models, with both CBOW and Skip-Gram variations, on the sequence of sessions to learn static embeddings for each item, based solely on their co-occurrence patterns. Then, the original sessions, represented as sequences of item IDs, are transformed into sequences of corresponding Word2Vec embeddings. These sequences are used as input to a Transformer model, trained to predict the embedding vector of the next item in the session. Instead of using a classification layer, the model outputs a dense vector in the same embedding space as Word2Vec. Recommendations are generated by computing the cosine similarity between the predicted output vector and all item embeddings learned by Word2Vec. The most similar items are selected as top recommendations. To efficiently perform similarity search across the embedding space, we use the FAISS library[3].

The metrics Hit Rate (HR), Precision, Recall, F-Score, Mean Reciprocal Rank (MRR), Normalized Discounted Cumulative Gain (NDCG), Mean Average Precision (MAP), Coverage, and Popularity Bias were used to quantify the quality of the top-10 recommended tracks. HR indicates the proportion of times a relevant item appears among the top-k recommended items; Precision represents the proportion of relevant items within the top-k recommendations; Recall measures the proportion of relevant items retrieved

[3]https://faiss.ai

out of all relevant items; F-Score is the harmonic mean of Precision and Recall. MRR calculates the average reciprocal rank of the first relevant item in the list; NDCG evaluates the quality of recommendations by considering both relevance and position of items, giving higher weight to earlier ranks; MAP is the mean of precision at positions of relevant items; Coverage expresses the proportion of unique items recommended relative to the total available in the catalog; and Popularity Bias refers to the tendency of recommendation algorithms to favor frequently interacted or popular items, often at the expense of less popular but potentially more relevant content. This can improve accuracy metrics but typically reduces diversity and limits the exposure of niche items.

To evaluate the recommendations and measure the performance, we adopted the Next Item Prediction approach [Ludewig and Jannach 2018]. In this setup, the last track of each session is withheld. The model is trained on all remaining tracks in each session. For testing, an entire session is provided as input, allowing the model to predict the next item. The predictions are then compared against the withheld track, and the evaluation metrics are computed based on the relevance and accuracy of the recommendations.

3.5. Hyperparameters Optimization

To ensure fair comparison and optimal performance across different model configurations, we employed Optuna[4], a state-of-the-art hyperparameter optimization framework designed for efficient exploration of complex search spaces. Optuna leverages Bayesian optimization with pruning strategies to intelligently sample promising configurations, using a user-defined objective function to guide the search process [Akiba et al. 2019].

In our work, hyperparameter tuning was handled differently for each modeling approach due to differences in architecture and computational constraints. For the end-to-end Transformer model, where embeddings are learned jointly with the model, we defined a search space over three key parameters: embedding dimension, number of attention heads, and feedforward network size. The objective was to maximize the HR@10, using the model's softmax-based output probabilities to evaluate recommendation accuracy.

On the other hand, for the embedding-separated architecture, hyperparameter optimization was conducted in two distinct stages to reduce the dimensionality of the search space. In the first stage, we optimized the parameters of the Word2Vec model, which was responsible for generating static embeddings for each item. The tuned parameters included vector size, context window, learning rate, minimum learning rate, number of negative samples, subsampling threshold, and the choice between hierarchical softmax and negative sampling.

To evaluate Word2Vec configurations without involving the full Transformer training, we followed a strategy inspired by prior work [Santana 2020]: a simple baseline model was used, in which session vectors were constructed by averaging the embeddings of all items in the session, and the next item was recommended based on cosine similarity. The HR@10 achieved by this baseline was used as the optimization objective for the Word2Vec stage.

Once the best Word2Vec model was selected, we proceeded to the second stage, in which the Transformer model was trained using fixed input embeddings. As in the joint-

[4]https://github.com/pfnet/optuna

training scenario, the Transformer hyperparameters were optimized to maximize HR@10, but now using cosine similarity between the model's output vector and the Word2Vec embedding space to generate recommendations.

This two-phase approach enabled us to balance optimization quality and computational feasibility, while also facilitating a more interpretable comparison between the modeling pipelines.

4. Results and Discussions

The comparative analysis of the models evaluated in our work clearly highlights the trade-off between accuracy and diversity in recommendations. As summarized in Tables 1 and 2, the end-to-end Transformer model achieved significantly higher accuracy metrics (HR, MRR, and NDCG) across both datasets. However, we observe that this improved accuracy came at the expense of a drastic reduction in catalog coverage and an increased popularity bias. In contrast, the other two models, which employed pre-trained embeddings using Word2Vec (CBOW and Skip-Gram), demonstrated greater flexibility and diversity in their recommendations, although their accuracy metrics were consistently lower. This clearly demonstrates the Transformer's strong tendency to focus on the most popular tracks, thereby reducing the diversity of recommendations. As illustrated by Figures from 2 to 5, the two datasets used inherently favor this behavior by the Transformer model.

Tabela 1. Comparing the recommendation models for the top-10 recommendations in the Xiami dataset.

Embeddings	HR	Precision	Recall	F-Score	MRR	NDCG	MAP	Coverage	Popularity bias
CBOW	0,310	0,031	0,310	0,056	0,195	0,223	0,195	0,640	0,003
Skip-Gram	0,337	0,034	0,337	0,061	0,233	0,258	0,233	0,658	0,003
Transformer	0,811	0,081	0,811	0,148	0,608	0,657	0,608	0,162	0,011

Tabela 2. Comparing the recommendation models for the top-10 recommendations in the Music4All dataset.

Embeddings	HR	Precision	Recall	F-Score	MRR	NDCG	MAP	Coverage	Popularity bias
CBOW	0,362	0,036	0,362	0,066	0,205	0,242	0,205	0,786	0,026
Skip-Gram	0,420	0,042	0,420	0,076	0,198	0,250	0,198	0,933	0,030
Transformer	0,703	0,070	0,703	0,128	0,506	0,553	0,506	0,524	0,039

Figures 2 and 3 present histograms showing the average play frequency by popularity percentile. These figures clearly illustrate the long-tail phenomenon, where a small fraction (between 0 and 0.1%) of tracks concentrates a very large number of plays. The average play counts for these top percentiles reached approximately 6,500 and 1,300 plays for the Music4All and Xiami datasets, respectively. There is also a drastic drop in play frequency for less popular tracks starting from the next percentile range (0.1–0.25%). Both graphics highlight the significant imbalance in consumption patterns in the data.

Additionally, Figures 4 and 5 show Lorenz curves [Lorenz 1905] for the Music4All and Xiami datasets, respectively. They compare the actual distribution of plays to an expected uniform distribution. The dashed line represents an ideal scenario of perfect

equality, while the blue curve reflects the real data distribution. Both datasets exhibit similar behavior, where approximately 20% of the most popular tracks account for 80% of all plays.

These figures suggest that any model trained to optimize next-track prediction tasks is inherently biased toward recommending a small subset of the most popular songs. This would help explain the observed tendency toward popularity bias in the end-to-end Transformer approach.

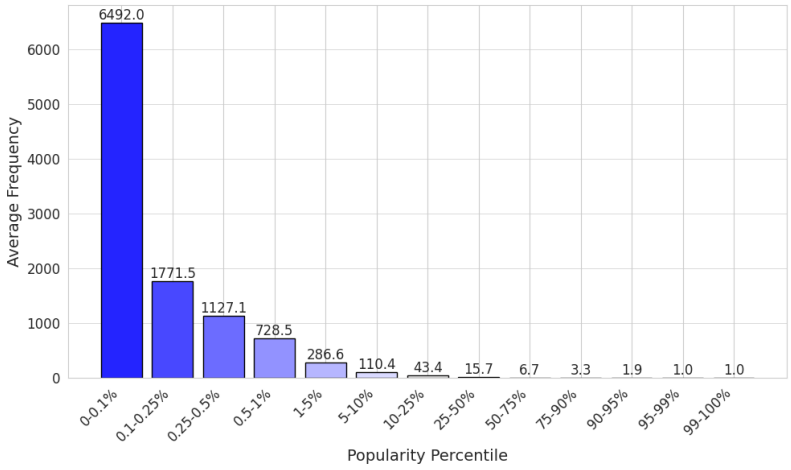

Figura 2. Distribution of average item frequency across popularity percentiles in the Music4All dataset.

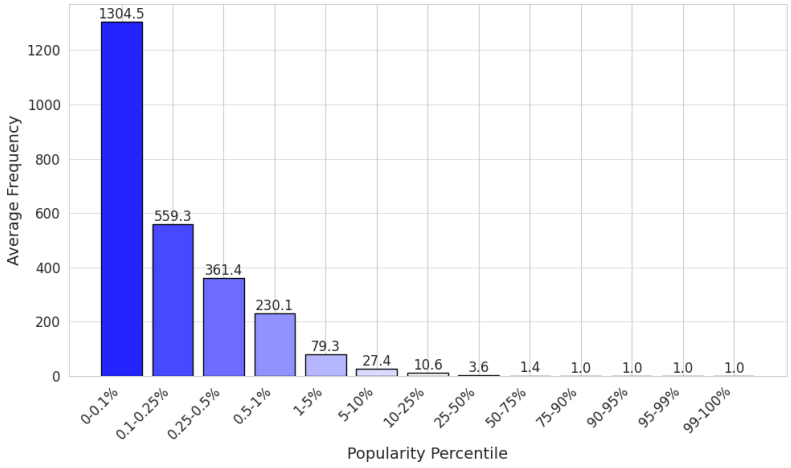

Figura 3. Distribution of average item frequency across popularity percentiles in the Xiami dataset.

A likely explanation for the reduced diversity in the Transformers output lies in its architectural design. Since the model treats the recommendation task as a multi-class classification problem, with a dense layer with one output neuron per item, it is inherently sensitive to class imbalance. This is especially problematic in the music domain, where the distribution of item interactions tends to be highly skewed toward a few popular songs.

On the other hand, the CBOW and Skip-Gram variants showed considerably higher Coverage, as their cosine similarity-based recommendation strategy is not directly

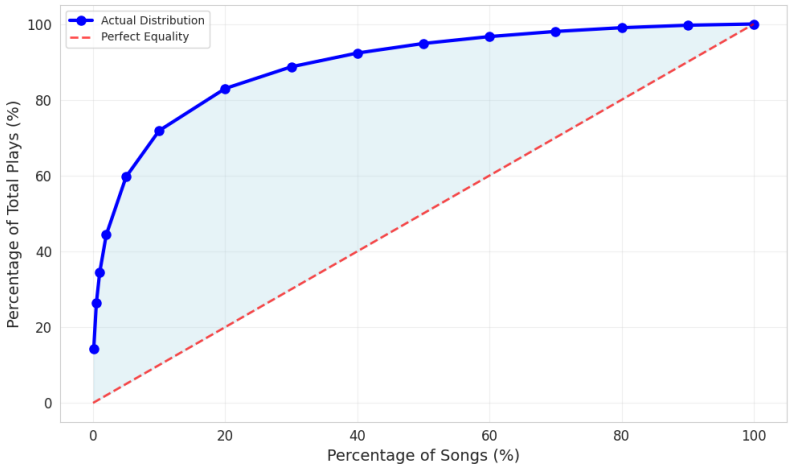

Figura 4. Cumulative play distribution in Music4All dataset.

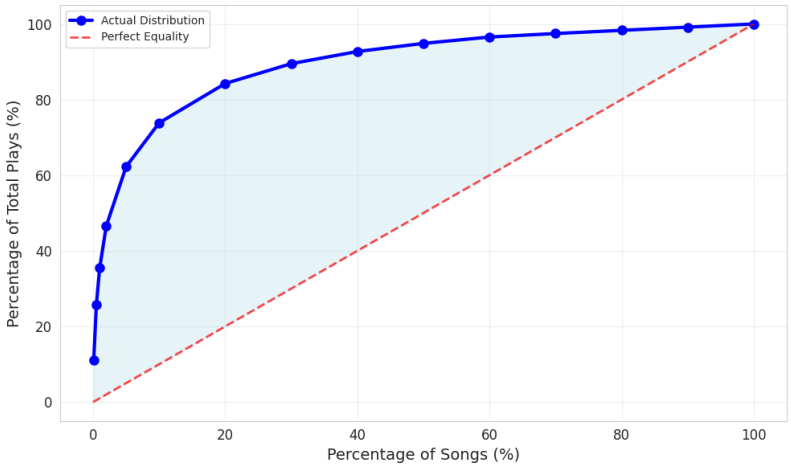

Figura 5. Cumulative play distribution in Xiami dataset.

influenced by item frequency during training and is less prone to popularity bias. Despite these advantages, the alternative models struggled to match the Transformer's Hit Rate. Several factors may explain this performance gap. First, training the Transformer to regress the next embedding vector using cosine similarity may not have been sufficient to learn precise item-level predictions, especially in high-dimensional spaces. Second, the decoupled two-stage training approach may have limited the model's ability to fully capture session-specific behavioral patterns. Lastly, while cosine similarity provides a principled way to mitigate popularity bias and encourage diversity, it may not always yield the most accurate next item predictions in practice.

In summary, the choice of architecture determines the recommender's behavior. While the end-to-end Transformer is a specialist optimized to predict the next interaction accurately, the Transformer combined with embeddings obtained through Word2Vec models is a semantic generalist that explores broader contextual relationships. Thus, the decision on which approach to adopt in practice, therefore, depends on the practitioner's objectives: to maximize immediate engagement (precision) or to promote catalog discovery and exploration (diversity and novelty).

5. Conclusion and Future Works

This work compared the effectiveness of Transformer-based recommendation models in session-based music scenarios across two distinct approaches: a traditional architecture with an internal embedding layer; and an alternative pipeline where item embeddings are computed externally, sequences of vectors are passed through a Transformer encoder, and recommendations are generated based on cosine similarity.

The results demonstrated that the cosine similarity-based approach achieved higher coverage, indicating a broader diversity in the recommended items and a reduced bias toward popular tracks. On the other hand, the traditional Transformer consistently outperformed the other variants in terms of Hit Rate, showing a stronger tendency to recommend frequently played songs.

To further improve performance and deepen understanding of the observed trade-offs, two main research directions emerge. The first is to integrate specialized embedding models, as Word2Vec, directly within the embedding layer of deep learning frameworks, allowing their semantic structure to influence the training process more effectively in an end-to-end manner. The second is to mitigate the bias introduced by class imbalance in the original Transformer model, which is particularly challenging in the music recommendation domain due to the long-tail effect, where a small number of songs dominate user interactions while the majority of tracks receive relatively few plays, or tailor a specific optimization function for that. Addressing these imbalances could enhance both diversity and fairness in recommendations without significantly compromising accuracy.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Referências

Abdollahpouri, H., Mansoury, M., Burke, R., and Mobasher, B. (2019). The unfairness of popularity bias in recommendation. *CoRR*, abs/1907.13286.

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD'19)*, page 2623–2631.

Jannach, D., Quadrana, M., and Cremonesi, P. (2022). Session-based recommender systems. In *Recommender Systems Handbook*, pages 301–334. Springer.

Jannach, D. and Zanker, M. (2022). *Value and Impact of Recommender Systems*, pages 519–546. Springer US, New York, NY.

Kang, W.-C. and McAuley, J. (2018). Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE.

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Latifi, S., Mauro, N., and Jannach, D. (2021). Session-aware recommendation: A surprising quest for the state-of-the-art. *Information Sciences*, 573:291–315.

Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems*, 27.

Lorenz, M. O. (1905). Methods of measuring the concentration of wealth. *Publications of the American statistical association*, 9(70):209–219.

Ludewig, M. and Jannach, D. (2018). Evaluation of session-based recommendation algorithms. *CoRR*, abs/1803.09587.

Ludewig, M., Mauro, N., Latifi, S., and Jannach, D. (2021). Empirical analysis of session-based recommendation algorithms. *User Modeling and User-Adapted Interaction*, 31(1):149–181.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. arXiv preprint, arXiv:1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Santana, I. A. P. (2020). Exploração de redes neurais recorrentes na recomendação sensível ao contexto de músicas. Dissertação (mestrado em ciência da computação), Universidade Estadual de Maringá, Maringá, PR.

Santana, I. A. P., Pinhelli, F., Donini, J., Catharin, L., Mangolin, R. B., Feltrim, V. D., Domingues, M. A., et al. (2020). Music4all: A new music database and its applications. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 399–404.

Schedl, M. (2019). Deep learning in music recommendation systems. *Frontiers in Applied Mathematics and Statistics*, 5.

Simm, V. S., Tanno, D. R., and Domingues, M. (2025). Exploring transformers in embedding-based music recommendation. In *Anais do XVII Congresso Brasileiro de Inteligência Computacional (CBIC 2025)*, pages 1–8.

van den Oord, A., Dieleman, S., and Schrauwen, B. (2013). Deep content-based music recommendation. In *Advances in Neural Information Processing Systems*, volume 26.

Vasile, F., Smirnova, E., and Conneau, A. (2016). Meta-prod2vec: Product embeddings using side-information for recommendation. In *Proceedings of the 10th ACM conference on recommender systems*, pages 225–232.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, D., Deng, S., and Xu, G. (2017). Sequence-based context-aware music recommendation. *Information Retrieval Journal*, 21(2-3):230–252.

Wang, D., Deng, S., Zhang, X., and Xu, G. (2016). Learning music embedding with metadata for context aware recommendation. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 249–253.

Wang, S., Cao, L., Wang, Y., Sheng, Q. Z., Orgun, M. A., and Lian, D. (2021). A survey on session-based recommender systems. *ACM Computing Surveys*, 54(7):1–38.