Why Ignore Content? A Guideline for Intrinsic Evaluation of Item Embeddings for Collaborative Filtering

Pedro R. Pires
pedro.pires@dcomp.sor.ufscar.br
Universidade Federal de São Carlos
São Carlos, SP

Bruno B. Rizzi
brunosora@hotmail.com
BTG Pactual
São Paulo, SP

Tiago A. Almeida
talmeida@ufscar.br
Universidade Federal de São Carlos
Sorocaba, SP

ABSTRACT

With the constant growth in available information and the popularization of technology, recommender systems have to deal with an increasing number of users and items. This leads to two problems in representing items: scalability and sparsity. Therefore, many recommender systems aim to generate low-dimensional dense representations of items. Matrix factorization techniques are popular, but models based on neural embeddings have recently been proposed and are gaining ground in the literature. Their main goal is to learn dense representations with intrinsic meaning. However, most studies proposing embeddings for recommender systems ignore this property and focus only on extrinsic evaluations. This study presents a guideline for assessing the intrinsic quality of matrix factorization and neural-based embedding models for collaborative filtering, comparing the results with a traditional extrinsic evaluation. To enrich the evaluation pipeline, we suggest adapting an intrinsic evaluation task commonly employed in the Natural Language Processing literature, and we propose a novel strategy for evaluating the learned representation compared to a content-based scenario. Finally, every mentioned technique is analyzed over established recommender models, and the results show how vector representations that do not yield good recommendations can still be useful in other tasks that demand intrinsic knowledge, highlighting the potential of this perspective of evaluation.

KEYWORDS

embeddings, intrinsic evaluation, qualitative evaluation, recommender systems, similarity tables, intruder detection, autotagging

1 INTRODUCTION

Recommender systems are tools commonly used by companies for enhancing the experience users have when utilizing their services by filtering and recommending particularly relevant information [1]. Among different types of recommender systems, collaborative filtering (CF) is one of the most popular [4].

Pioneer CF recommender systems represented items as sparse vectors of consumption. However, due to the accelerated growth in the number of users and items, this form of representation started facing limitations related to: *(i)* sparsity, since modern recommender systems must deal with a number of possible interactions that follows a power law according to the number of users and items, yielding an often highly sparse interactions matrix; and *(ii)* scalability, given that the vectors used to represent users and items can become quite large, increasing the demand for greater storage and processing capacity.

To circumvent these problems, efforts are being made to represent users and items in a much smaller dimensional space [40]. In this context, two main techniques gained ground with the literature: matrix factorization [27] and neural networks [57]. In the latter, neural embedding models inspired by Natural Language Processing (NLP) have recently gained traction [34]. A significant advantage of embeddings in NLP is their ability to carry intrinsic meaning, i.e., the knowledge encapsulated within the representation beyond the information used for training. In recommender systems, this translates to the potential of leveraging item embeddings for various tasks beyond mere recommendation.

Despite the promising nature of embeddings, most existing research has focused primarily on extrinsic evaluations, neglecting the intrinsic qualities of the learned representations. It is well known that the performance of embeddings in downstream applications does not always correlate with their intrinsic quality [41].

While the primary goal of a recommender system is to provide high-quality recommendations, the vector representations of items and users can also be applied to other tasks such as automatic feature prediction, knowledge discovery, and clustering [21, 31]. These applications can only be significantly improved if the intrinsic quality of the embeddings is adequately assessed. Therefore, evaluating the intrinsic quality of matrix factorization and neural embedding models is crucial for their successful application in diverse contexts.

To address this gap, we present various methods for using item metadata to intrinsically evaluate vector representations of items in a CF recommender system. We introduce a commonly used evaluation technique in recommender system literature and adapt an intrinsic evaluation task from NLP to our context. Given the time-consuming and expertise-dependent nature of subjective analyses, we propose a new quantitative, non-subjective strategy using content-based data to evaluate the intrinsic ranking quality of embeddings. We illustrate our proposed pipeline with an extrinsic evaluation, assessing the quality of vector representations in generating recommendations, and compare these results with our intrinsic evaluation tasks. Our findings demonstrate that embedding models can perform very differently across tasks, with some models that perform poorly in recommendation tasks excelling in intrinsic evaluations, underscoring the necessity of comprehensive analysis when developing new representation models. With these considerations, the primary objectives of this study are:

(1) Introduce new methods for evaluating item embeddings using strategies derived from Natural Language Processing, such as *intruder detection*, and from content-based recommendation, such as *content-based ranking comparison*;
(2) Present a collection of techniques for intrinsically evaluating item embeddings in both subjective and objective manners;

In: Proceedings of the Brazilian Symposium on Multimedia and the Web (WebMedia'2024). Juiz de Fora, Brazil. Porto Alegre: Brazilian Computer Society, 2024.

ISSN 2966-2753

(3) Compare traditional embedding-based recommender models in extrinsic and intrinsic tasks to illustrate the varied performance of embeddings across different applications.

2 RELATED WORK

The earliest collaborative filtering recommender systems employed neighborhood-based methods to compute recommendations. However, as the number of users and items increased, these techniques encountered significant challenges related to sparsity and scalability [25]. Consequently, the use of embeddings for representing users and items gained popularity, i.e., low-dimensional vector representations that carry intrinsic meaning. Initially, matrix factorization models were applied for this task [27], reducing memory consumption and processing demand while generating valuable insights. Over the years, numerous methods have been proposed to generate embeddings, employing diverse strategies such as graph-based algorithms and quantization techniques [58].

Inspired by state-of-the-art Natural Language Processing (NLP) techniques [32, 34], recent methods aim to learn neural embeddings for items and users. The first studies in the area were Item2Vec [2] and Prod2Vec [18], neural networks heavily inspired by the Skip-gram architecture [32], and User2Vec [18], inspired by Paragraph Vector [28]. Subsequent studies have built upon these foundational models, incorporating various techniques to enhance performance, e.g., consuming item metadata [15, 47], leveraging content information to enrich embeddings [19, 51, 56].

Beyond incorporating item content, more complex neural models have been employed, including deep learning [43, 55], recurrent neural networks [22, 51], convolutional neural networks [10, 46], GANs [7, 50], and transformers [26, 54]. Another notable approach is training NLP models on textual data from items, users, or interactions [20, 44], with Large Language Models (LLMs) gaining attention in recent years [16, 30].

A particularly intriguing aspect of neural embeddings is their intrinsic meaning [17], though few studies have thoroughly investigated this property. The most common approach for evaluating item embeddings is using similarity tables [2, 15, 18], but this method heavily relies on subjective opinions, which can be misleading. Other methods, such as genre plotting [2, 15, 44] and sample clustering [26, 51], also depend on human judgment or are difficult to apply across different domains, such as analogy analysis [19].

In NLP, several established methods exist for intrinsic evaluation of word embeddings [3, 17, 35, 53], including: (i) comparing human judgment of word similarity with embedding space similarity; (ii) predicting word analogies through vector arithmetic; (iii) clustering word embeddings and evaluating cluster quality; and (iv) detecting synonyms or “intruders” in groups of similar embeddings. However, many of these methods are challenging to adapt to recommender systems due to their reliance on external semantic datasets.

In many areas of recommender systems, the accuracy of the recommendation is prioritized over other characteristics such as quality [52], and although item embeddings are often measured in downstream applications such as recommendations, this does not guarantee intrinsic quality [41]. High-quality embeddings can boost various tasks in a recommendation scenario, such as automatic feature prediction, user and item clustering, and knowledge discovery [21, 31], or enhance semantic-based recommendation engines, that use intrinsic knowledge as the main strategy for filtering items [24]. Embeddings with strong intrinsic value can even assist in uncovering item metadata for systems relying on categorical features [6, 49], which are often incomplete or inaccurate, and enhance traditional collaborative filtering models [33].

To the best of our knowledge, no work on the recommender system literature has focused on studying the intrinsic aspects of item embeddings. Studies proposing novel embedding-based models often perform simple forms of intrinsic evaluation, normally based on subjective approaches that can add human bias and problems to the conclusions. Neglecting or improperly conducting intrinsic evaluations can result in the loss of valuable information that could support related tasks and improve representation models.

In this context, we introduce alternative methods for intrinsically evaluating recommender system embeddings. Besides presenting commonly used evaluation techniques, we adapt an NLP evaluation task to the recommender context and propose a novel quantitative metric for intrinsic quality using a content-based ranking comparison approach. We conduct both extrinsic and intrinsic evaluations, comparing results across different tasks to illustrate the varied performance of embedding models.

3 SUBJECTIVE METHODS FOR INTRINSIC EVALUATION

The intrinsic evaluation of embeddings in both NLP and recommender systems often relies on subjective approaches, where the quality of the representation is assessed based on human opinions. In this section, we present two subjective tasks: similarity tables and intruder detection.

3.1 Similarity Tables

The similarity table evaluation strategy involves training different models on the same datasets and selecting a known item as a seed. The distances between this seed item and all other items are calculated for each representation using a measure of angular similarity. The top-N nearest items for each representation are then displayed side-by-side in a table. Human evaluators can subjectively assess each group of nearest items to determine how well they match the target item. This task can be repeated with multiple known items, providing a broader perspective on the representations' behavior.

Similarity tables are the most straightforward and intuitive way to check for intrinsic meaning in the evaluated representations. This ease of use makes it one of the most commonly employed schemes [2, 15, 18]. However, it heavily depends on human interpretation based on subjective analysis [17].

3.2 Intruder Detection

In NLP, intruder detection, also known as outlier detection, involves identifying "intruder" words in a set [41]. We propose adapting this task for the recommender system context by treating items as words. To do this, we first select a few target items (which can be randomly drawn or, preferably, human-selected) and, using the embedding-vector space, find the top-N nearest items to each target. A random item from the representation space is then added to the neighboring items, and Human evaluators must then determine the “intruder” items (i.e., the randomly selected ones). The accuracy of these guesses can be computed and used to compare the methods.

Although the task's outcome is quantitative, subjectivity is reduced but not eliminated. It still relies on human interpretation and can be time- and cost-intensive, requiring one or more individuals to analyze each chosen example. However, the need for multiple evaluators can be mitigated using crowdsourcing platforms, such as Amazon Mechanical Turk[1] and Prolific[2], which allows researchers to hire participants from diverse demographics.

3.3 Shortcomings of subjective evaluation

The weak point of using the aforementioned subjective strategies resides in their high dependence on human opinions, which may be heavily biased or too shallow for a particular domain [12]. Human opinions can unfairly penalize embeddings when the grouping criteria selected by the algorithm differs from the one preferred by the evaluator [59].

To avoid human bias, the experiments must be conducted with a vast range of people, which is more time-demanding and will most likely result in cost increases. To overcome subjectivity in the existing evaluation schemes, it is recommended to use objective approaches, i.e., the ones that consume additional data to quantify the intrinsic quality and are calculated automatically.

4 OBJECTIVE METHODS FOR INTRINSIC EVALUATION

To avoid bias problems that subjective evaluation suffers, here we suggest the use of two different metrics: automatic feature prediction and a novel metric of content-based ranking comparison.

4.1 Automatic Feature Prediction

Automatically discovering item features involves predicting a set of unknown tags for a target item i based on its most similar items. Also known as auto-tagging, it is a specific problem in the fields of recommender systems and knowledge discovery, with many methods proposed exclusively for this [8, 29, 45].

We can employ a neighborhood-based automatic feature prediction approach to assess the intrinsic value of a vector representation [2, 15]. For each item i in the entire catalog, its k nearest items are selected, considering the embedding-vector space. Next, the attributes of the target item are discovered through some voting process, such as a simple majority vote.We can then compare the original item's attributes with the ones predicted by the neighbors and quantify the intrinsic quality of the representation using traditional classification metrics, such as precision, recall, or F1-score.

4.2 Content-based Ranking Comparison

As an additional objective and automatic method for intrinsically evaluating vector representations of items, we propose a novel metric that compares the neighborhood generated by the embeddings in the vector space with a neighborhood constructed using content-based information about the items. The quality of the comparison can then be quantified using a ranking comparison metric.

To properly evaluate the spatial distribution generated by the learned embeddings, we first assume that there is a correct order for the neighborhood of a given item when filtering its most similar items on the embeddings vector space. In this study, we constructed the target ordering using the similarities between the high-level features of the items: the item's category, genre, or tags. As we aim to approximate items with similar features, we can compare this neighborhood with the one generated by the embeddings. Moreover, it is possible to use more complex and domain-specific techniques for ranking, e.g., low-level visual features for movies [13] and context and metadata graph embeddings for music [48].

Using those general features to describe the items, we can represent an item i through a bag-of-words encoding, i.e., an array of attributes $\vec{i}$. With this representation, we can build a similarity matrix C of dimensions $|I| \times |I|$, in which $|I|$ represents the number of items in the catalog. Thus, for a given pair of items i and j, $C_{i,j}$ stores their similarity when using the content-based representation, i.e., vectors $\vec{i}$ and $\vec{j}$, calculated using a metric such as cosine similarity. Similarly, we construct the similarity matrix $\mathcal{E}$, which stores the similarity of the items' dense embeddings.

Afterward, for each item $i \in I$, we can construct two neighborhoods, $\mathcal{N}_i^C$ and $\mathcal{N}_i^{\mathcal{E}}$. The former corresponds to the subset of items most similar to i considering the similarity matrix C, i.e., the items that share the most related content-based features. The latter represents the same neighborhood concept but uses the similarity values stored in matrix $\mathcal{E}$. Both $\mathcal{N}_i^C$ and $\mathcal{N}_i^{\mathcal{E}}$ may be limited to a restricted number of neighbors, defined by a hyperparameter k, to reduce memory consumption when calculating the ranking.

With both similarity matrices constructed, we can compare them using different metrics and approaches, e.g., traditional or utility-based ranking measures, and sample sets' similarity metrics. Regardless of the adopted strategy, all of them are maximized according to the same ordering, i.e., the one created using the content-based representation, with the differences being in how the neighborhoods are used to calculate the final score. In the following, we offer metrics for each one of those approaches.

4.2.1 Rank correlation metrics.

When considering the order of the items, we can calculate metrics designed to compare the correlation of rankings when you have a reference ranking, such as Spearman's rank correlation coefficient ρ, Kendall's τ coefficient, or the Normalized Distance-based Performance Measure (NDPM).

For comparing the rankings using Spearman's ρ, we can calculate the correlation coefficient for each item and average the results, as shown in Equation 1. To calculate ρ_i, we use Equation 2, in which d corresponds to the difference among ranking positions when sorting the items according to $\mathcal{N}_i^C$ and $\mathcal{N}_i^{\mathcal{E}}$.

$$\rho = \frac{\sum_{i \in I} \rho_i}{|I|} \quad (1) \qquad \rho_i = 1 - \frac{6 \sum_{j \in I} d_j^2}{|I|(|I|^2 - 1)} \quad (2)$$

4.2.2 Set similarity metrics.

If we limit the neighborhoods to the top-K similar items, we can treat them both as sample sets and calculate metrics designed to compare the similarity and diversity of sets, such as the Jaccard Index or the Sørensen–Dice coefficient.

For the Jaccard Index J, we must first calculate the Jaccard Index J_i for each item i, and then average the results, as shown in Equation 3. For J_i, we must build two sets, S_i^C and $S_i^{\mathcal{E}}$, consisting of the top-K most similar items from both $\mathcal{N}_i^C$ and $\mathcal{N}_i^{\mathcal{E}}$, respectively, and then calculate the set similarity using Equation 4:

$$J = \frac{\sum_{i \in I} J_i}{|I|} \quad (3) \qquad J_i = \frac{|S_i^C \cap S_i^{\mathcal{E}}|}{|S_i^C \cup S_i^{\mathcal{E}}|} \quad (4)$$

[1] Amazon Web Services. *Amazon Mechanical Turk*. Available at: https://www.mturk.com/.

[2] Prolific. *Prolific: Definitive human data to deliver world-leading research and AI.* Available at: https://www.prolific.com/.

4.2.3 Utility-based ranking scores. Finally, we can also adapt utility-based ranking metrics, using the content-based similarities to quantify the utility of the embeddings ranking.

Here, we explain how the Normalized Discounted Cumulative Gain (NDCG) can be adapted for this type of evaluation. First, as it is commonly calculated for the metric, we define the overall NDCG as the average of the NDCG for each item i (NDCG$_i$), as shown in Equation 5. NDCG$_i$, on the other hand, is defined as the Discounted Cumulative Gain of the item (DCG$_i$) divided by its Ideal Discounted Cumulative Gain (CB-IDCG$_i$), normalizing the final value to a 0–1 range, as shown in Equation 6.

$$\text{NDCG} = \frac{1}{|I|}\sum_{i \in I}(\text{NDCG}_i) \quad (5) \qquad \text{NDCG}_i = \frac{\text{DCG}_i}{\text{IDCG}_i} \quad (6)$$

The main differences arise when calculating the DCG$_i$ and the IDCG$_i$. For both scores, we consider that the "gain" of a given item j in the neighborhood of i is given by the values of matrix C, i.e., the content-based similarity matrix. The ideal gain, IDCG$_i$, is retrieved using the top-k items of the content-based neighborhood, $\mathcal{N}_i^{\mathcal{C}}$, while the obtained gain, DCG$_i$ uses the embedding-based neighborhood, $\mathcal{N}_i^{\mathcal{E}}$, as presented in Equations 7 and 8, respectively.

$$\text{IDCG}_i = \sum_{n=1}^{k} \frac{C_{i,\mathcal{N}_{in}^{\mathcal{C}}}}{\log_2(n+1)} \quad (7) \qquad \text{DCG}_i = \sum_{n=1}^{k} \frac{C_{i,\mathcal{N}_{in}^{\mathcal{E}}}}{\log_2(n+1)} \quad (8)$$

For the ideal score (Equation 7), we defined the gain for the n_{th} item in the neighborhood as $C_{i,\mathcal{N}_{in}^{\mathcal{C}}}$, which corresponds to the content-based similarity stored in C between the target item i and the n_{th} item of i's neighborhood in $\mathcal{N}^{\mathcal{C}}$. For the generated score (Equation 8), the gain is calculated similarly. It is represented by $C_{i,\mathcal{N}_{in}^{\mathcal{E}}}$, corresponding to the similarity stored in C, but between the target item i and the n_{th} item of i's neighborhood in $\mathcal{N}^{\mathcal{E}}$.

5 EXPERIMENTAL SETUP

This section details the experimental setup. First, we present the datasets used in the experiments, then we describe the benchmark algorithms and the fine-tuning phase.

5.1 Datasets and Data Preprocessing

Table 1 presents the datasets used in the experiments. They are publicly available, used in past research or challenges, and provide item metadata. The features describing the items can be of two types: *(i)* categories, attributes inherent to the item, informed by the system owner; or *(ii)* tags, values informed by users without moderation. Since user-informed tags are liable to noise and inconsistency, we opt to use only the top-100 most recurring tags by dataset, as performed in studies of tag-based recommender systems [11, 14].

5.2 Embedding-based Algorithms

We have implemented two well-known methods for matrix factorization, Alternating Least Squares (ALS) [23] and Bayesian Personalized Ranking (BPR) [37], using the `implicit` [8] library. Moreover, considering that a good intrinsic meaning is commonly achieved by context-window models [32], we have also implemented two contextual neural embeddings-based recommenders, Item2Vec (I2V) [2] and User2Vec (U2V) [18], using the `gensim` [36] library. All models were selected based on their popularity in recent studies [38, 39], ease of replication, and the fact that they do not rely on item metadata, showing the power those methods can have for figuring out knowledge about the items without consuming this information.

We fine-tuned the methods through a grid search holdout maximizing NDCG@15 [42], with a rate of 8:1:1 for training, validation, and test sets, respectively. For ALS and BPR, we tested hidden factors of sizes f = {50, 100, 300}, regularization factor λ = {0.01, 0.1, 1} and learning rate α = {0.0025, 0.025, 0.25}, using 100 epochs for training. For the embedding models, we varied the number of epochs n = {50, 100, 200}, sub-sampling rate of frequent items $t = \{10^{-5}, 10^{-4}, 10^{-3}\}$, and exponent to shape the negative sampling distribution α = {-1.0, -0.5, 0.5, 1.0}, as recommended by Caselles-Duprés et al. [5]. For any unmentioned parameter, we used the library default values.

6 EXTRINSIC RESULTS

To assess the representation models' ability to provide good recommendations, i.e., to evaluate the models extrinsically, we conducted a top-N ranking task, calculating the NDCG for multiple values of N. Results are shown in Table 2, in which darker-toned cells correspond to better results, with the best result for each combination of dataset and the value for N highlighted in **bold**.

The results indicate a similar performance between ALS, BPR, and Item2Vec. On the contrary, User2Vec was the worst, obtaining low results for every dataset and threshold. ALS and BPR tend to present better results for small values of N, with their worst results being achieved when $N = 20$. The same is not true for Item2Vec and User2Vec, which benefit from bigger values of N.

To properly analyze the models, we conducted a non-parametric Friedman test to verify if there is a statistically significant difference between them [9], using a ranking constructed with the results. The test indicates that the models differ, with 99% confidence (X_F^2 = 29.96). We then conducted a Nemenyi test to compare them to each other [9]. With 95% confidence and considering a critical difference of 1.21, there is no statistical evidence of superiority among ALS, BPR, and Item2Vec. Additionally, we have that the three models are statistically superior to User2Vec, considering that the differences in the average rankings were superior to the critical difference [9].

Results of the extrinsic experiment show that ALS, BPR, and Item2Vec achieve very similar results and that User2Vec is the most unsuitable method for this task compared to the others. However, following insights for the NLP area [41], this behavior may not necessarily repeat when we apply the same representation vectors to intrinsic tasks. Therefore, we conducted different intrinsic evaluation strategies to assess this particularity.

[3] Anime Recommendations dataset. Available at: www.kaggle.com/datasets/CooperUnion/anime-recommendations-database

[4] Data Mining Hackathon on Big Data (7GB). Available at: www.kaggle.com/c/acm-sf-chapter-hackathon-big

[5] DeliciousBookmarks. Available at: www.grouplens.org/datasets/hetrec-2011/

[6] Last.FM. Available at: www.grouplens.org/datasets/hetrec-2011/

[7] MovieLens 25M. Available at: www.grouplens.org/datasets/movielens/

[8] Ben Frederickson. 2017. *Implicit: Fast Python Collaborative Filtering for Implicit Datasets.* Available at: https://github.com/benfred/implicit

Dataset	*Users*	*Items*	*Interactions*	*Sparsity*	*Categories*	*Tags*
Anime[3]	73,514	11,200	7,813,733	99.05%	43	N/A
BestBuy[4]	1,268,702	69,858	1,865,269	99.99%	1,540	N/A
Delicious[5]	1,867	69,223	104,799	99.92%	N/A	14,346
Last.FM[6]	1,892	17,632	92,834	99.72%	N/A	9,718
MovieLens[7]	162,541	59,047	25,000,095	99.74%	20	65,464

Table 1: Description of each dataset used in the experiments. N/A is used for datasets without categories or tags.

Dataset	*N*	**Representation Model**			
		ALS	**BPR**	**I2V**	**U2V**
Anime	**10**	**0.2548**	0.1826	0.1196	0.0080
	15	**0.2374**	0.1736	0.1275	0.0084
	20	**0.2302**	0.1700	0.1348	0.0090
BestBuy	**10**	0.0633	**0.0746**	0.0485	0.0135
	15	0.0633	**0.0746**	0.0557	0.0160
	20	0.0633	**0.0746**	0.0611	0.0176
Delicious	**10**	0.0548	0.0467	**0.0855**	0.0177
	15	0.0548	0.0467	**0.0969**	0.0235
	20	0.0548	0.0467	**0.1045**	0.0389
Last.FM	**10**	**0.1865**	0.1598	0.1729	0.0173
	15	0.1864	0.1597	**0.1894**	0.0229
	20	0.1864	0.1597	**0.1999**	0.0306
MovieLens	**10**	**0.3067**	0.1870	0.1132	0.0004
	15	**0.2727**	0.1683	0.1211	0.0004
	20	**0.2568**	0.1599	0.1291	0.0005

Table 2: NDCG achieved by each algorithm in each dataset in a top-N recommendation task with different values of N.

7 INTRINSIC RESULTS

To intrinsically evaluate the vector representation, we performed the tasks presented in Sections 3 and 4, discussing the main differences of each experiment. The results show how the same representation model can perform differently according to the task, especially when comparing subjective to objective strategies.

7.1 Similarity Table

We built two similarity tables using popular items from datasets of widely known domains: Last.FM (Table 3) and MovieLens (Table 4), along with their top-3 neighbors in each representation.

In Table 3, all methods found a neighborhood with similar items to the target. In most cases, the bands and artists have completely varied for each representation model, with only a few exceptions such as *Jay-Z*, *30 Seconds to Mars*, and *Beyoncé*, that were present on three of the algorithms. Even with the selection of different artists, the music genres were normally very related to the target. Some methods behave in a more conservative manner, such as BPR returning only hip-hop and rap artists for *Eminem*, while others returned relevant items, but deviating from the tags, such as *Ke$ha* and *P!nk* in the I2V neighborhood for *Eminem*. Even so, we can say that every method achieved some pertinent neighborhood. ALS was the worst due to certain tag contradictions in its results, such as the learned neighborhood for *The Beatles*, which instead of other rock or 60s bands, contains Arabic and baroque artists.

We can not say the same for Table 4. BPR and User2Vec, especially, found some very related items, such as the sequels for *X-Men*. However, they also found some odd neighbors, such as *Jack-O* or *Men in Black*, a horror and sci-fi movie, respectively, for *Toy Story*, a children's animation. For some movies, such as *Titanic*, all methods performed poorly when comparing the genres between target and neighbor items. On the other hand, all neighboring movies are considered classic films, implying the representations discovered a pattern. Due to these conflicting results, it is hard to select a superior representation without relying on human personal opinions.

As mentioned, this evaluation method is heavily influenced by human subjectivity. For instance, in Table 3, among all neighborhoods of *Shakira*, the ones composed by *Rihanna*, *Britney Spears*, *Katy Perry*, *Mariah Carey* and *Beyoncé* would be the most similar if we consider that they are all world-famous pop artists. However, *Thalía*, *Fanny Lu* and *Juanes* are all Latin pop artists; hence, they are strongly connected with the target, *Shakira*. Therefore, deciding what is more similar is complex, and our opinions and backgrounds can strongly skew our guesses.

7.2 Intruder Detection

We conducted the task using Anime, MovieLens, and Last.FM datasets, as they are from well-known domains and contain well-curated additional information, e.g., genre and release year. For each dataset, we selected 15 items to use as seeds, of which 10 were popular items, and 5 were completely random. We then built five questionnaires, each with 15 items, alternating between the representation models. Finally, we asked a group of 10 individuals to discover the intruder. Each representation model received 30 votes, and the accuracy for each model is shown in Table 5. Results are shown in three different views: the general accuracy, considering all of the 15 items, the accuracy for only the 10 popular items, and the accuracy for the 5 random (and probably unknown) items.

BPR and User2Vec performed better at building a good quality neighborhood, as they usually presented the highest number of correct answers per dataset. Item2Vec also constructed a satisfactory neighborhood, being a close second in almost every case.

BPR generally achieved the best results in the scenario considering all items, being the best model for MovieLens and Last.FM, and the second-best for Anime. User2Vec presented promising results for the Anime dataset and reached 100% accuracy in the scenario where items were randomly selected, showing that it could generate a relevant neighborhood even in cases where there is little knowledge about the item. This result is exciting, considering the scores obtained by the model in the extrinsic evaluation (Section 6).

Target item	Representation Model			
	ALS	**BPR**	**I2V**	**U2V**
The Beatles *classic rock*	Ricky Nelson *rock*	Beach Boys *60s*	David Bowie *rock*	The Kinks *60s*
	Souad Massi *female, arabic*	John Lennon *classic rock*	Radiohead *alternative*	Rolling Stones *classic rock*
	Andrés Segovia *baroque*	Ringo Starr *classic rock*	Led Zeppelin *hard rock, rock*	Velvet Underground *psychedelic*
Eminem *hip-hop, rap*	Ice Cube *hip-hop, rap*	Jay-Z *hip-hop, rap*	Ke$ha *pop, dance*	Akon *hip-hop, rap*
	Bizarre *hip-hop, rap*	50cent *hip-hop, rap*	P!nk *pop, female*	Nelly *hip-hop, rap*
	Xzibit *hip-hop, rap*	Kanye West *hip-hop, rap*	Jay-Z *hip-hop, rap*	Jason Derulo *pop, rnb*
Shakira *female, pop*	Juanes *latin, pop*	Beyoncé *rnb, pop*	Rihanna *pop, rnb*	Katy Perry *pop, female*
	Fanny Lu *latin, pop*	Marilyn Monroe *jazz, female*	Beyoncé *rnb, pop*	Mariah Carey *rnb, pop*
	Thalía *latin, pop*	Rihanna *pop, rnb*	Britney Spears *pop, dance*	Beyoncé *rnb, pop*

Table 3: Similarity table of five popular artists from the Last.FM dataset

Target item	Representation Model			
	ALS	**BPR**	**I2V**	**U2V**
Friday the 13th *horror, thriller*	A View to Kill *action*	Nigthmare in Elm Street *horror*	Gremlins 2 *comedy, horror*	Friday the 13th 2 *horror*
	Child's Play *horror, thriller*	Friday the 13th 3 *horror*	Texas Chainsaw Massacre *horror*	Halloween II *horror*
	Pet Sematary *horror*	Children of the Corn *horror*	Halloween *horror*	Child's Play *horror, thriller*
Titanic *drama*	Groundhog Day *comedy*	Truman Show *comedy*	Gd. Will Hnt. *drama*	Jurassic Park *action, sci-fi*
	Truman Show *comedy*	Catch Me If You Can *crime, drama*	Men in Black *action, sci-fi*	Truman Show *comedy*
	Christmas Do-Over *comedy*	My Best Friend's Wedding *comedy*	Saving Private Ryan *drama, war*	Men in Black *action, sci-fi*
Toy Story *children*	Average Italian *comedy*	Muppet Treasure Island *children*	Braveheart *drama, war*	Lion King *children*
	The Pride & The Passion *war, action*	Babe *children*	12 Monkeys *sci-fi, thriller*	Toy Story 2 *children*
	Barbie *animation*	Jack-O *horror*	The Usual Suspects *crime*	Men in Black *action, sci-fi*

Table 4: Similarity table of five popular movies from the MovieLens dataset

It is important to highlight that the conducted experiment does not have strong statistical rigor and may not represent an accurate evaluation of the embedding models. The interview was conducted with only ten participants without concern about selecting persons with different backgrounds and belonging to contrasting demographic groups. Nonetheless, this is one of the main drawbacks of this evaluation scheme. Conducting a proper intruder detection task is very time and resource-demanding. Even so, the obtained results can provide some valuable insights about the models.

7.3 Automatic Feature Prediction

In every evaluated dataset, each item is described with a single feature related to the domain of the problem, such as genres for movies and styles for music artists. For each item in the datasets, we predicted their features using the most recurrent features of other k nearest items, with k ranging between $10, 15$ and 20. For each prediction, we checked if the selected feature was correct, and using the average multiclass precision and recall, we computed the F1-score for each model, selecting the value for k that resulted in the best F1-score. Table 6 shows the results.

When comparing the results, it is challenging to indicate a superior model. However, User2Vec performed best on BestBuy and for

Seed items	Dataset	Representation Model			
		ALS	BPR	I2V	U2V
All	**Anime**	43.3%	63.3%	60.0%	**90.0%**
	MovieLens	33.3%	**66.7%**	46.7%	43.3%
	Last.FM	66.7%	**90.0%**	86.7%	66.7%
Popular	**Anime**	44.4%	66.7%	55.6%	**83.3%**
	MovieLens	38.9%	**61.1%**	44.4%	44.4%
	Last.FM	72.2%	**94.4%**	88.9%	77.8%
Random	**Anime**	41.7%	58.3%	66.7%	**100.0%**
	MovieLens	25.0%	**75.0%**	50.0%	41.7%
	Last.FM	58.3%	**83.3%**	**83.3%**	50.0%

Table 5: Accuracy for the intruder detection task

some values of k for the Delicious dataset, also being a close second on Last.FM. This indicates that the neural model discovered the most about the intrinsic content of all evaluated methods for these specific datasets. This outcome is interesting when we compare the results of the feature prediction task with those of the extrinsic experiment. In the latter, User2Vec was the worst representation model for every value of N and dataset. For the BestBuy dataset, the NDCG when $N = 10$ was more than five times worse than that of BPR. This behavior highlights how the results of extrinsic and intrinsic tasks can drastically differ.

Dataset	k	Representation Model			
		ALS	BPR	I2V	U2V
Anime	**10**	0.4717	**0.5251**	0.4931	0.4771
	15	0.4049	**0.4635**	0.4342	0.4003
	20	0.4002	**0.4661**	0.4422	0.3998
BestBuy	**10**	0.0835	0.1208	0.3006	**0.3361**
	15	0.0658	0.0999	0.2542	**0.2854**
	20	0.0657	0.1006	0.2514	**0.2797**
Delicious	**10**	**0.1362**	0.1236	0.1318	0.1355
	15	0.0951	0.0840	0.0893	**0.1003**
	20	0.0979	0.0871	0.0917	**0.1036**
Last.FM	**10**	**0.4649**	0.3753	0.4046	0.4579
	15	**0.4268**	0.3274	0.3418	0.4249
	20	**0.4365**	0.3350	0.3485	0.4334
MovieLens	**10**	0.5009	**0.5158**	0.4597	0.4083
	15	0.4572	**0.4722**	0.4052	0.3438
	20	0.4730	**0.4879**	0.4181	0.3576

Table 6: F1-score in an automatic feature prediction task with different values of k.

In addition, we can see those methods that reached better scores in Table 6 may differ from those of the intruder detection task. In the former, ALS was the most accurate model for datasets Last.FM and Delicious when $k = 10$, while the latter was the least accurate for every dataset, including the aforementioned ones. This shows how even different intrinsic metrics can achieve varying results, especially when comparing subjective approaches to objective ones since the former is more prone to human bias.

7.4 Content-based Ranking Comparison

Lastly, we have calculated the three metrics for assessing the item embeddings' intrinsic quality using a content-based ranking comparison, as detailed in Section 4.2. Results for the Spearman correlation coefficient are shown in Table 7, for the Jaccard Index in Table 8, and NDCG in Table 9. For both the Jaccard Index and the NDCG, we used a neighborhood size k ranging from $\{10, 15, 20\}$.

Like the automatic feature prediction task, the performance varied widely according to the metric and dataset, with each model scoring higher in a specific case. For Spearman's ρ, Item2Vec was the best model for Anime and BestBuy datasets, contrary to what happened on the intruder detection and automatic feature prediction, in which the model was surpassed by BPR and User2Vec, depending on the dataset and task. User2Vec achieved the best results for Delicious and Last.FM, which is also impressive since its behavior on the intruder detection task for dataset Last.FM was poorly, being the worst or second-worst model.

When limiting the observed neighborhood to a subset of items, as it is performed on the Jaccard Index and NDCG, the scores were vastly different from Spearman's ρ. For the Jaccard Index, Item2Vec achieved the worst results for the Anime dataset and the best for Delicious and Last.FM. User2Vec presented the highest scores for BestBuy and competitive performance in the Anime dataset. Although ALS achieved some promising results for the Last.FM dataset, it was surpassed by every other model for every dataset. Finally, for the NDCG, the results were similar to the ones obtained in the automatic feature prediction (Table 6), which is expected given that both metrics limit the observed neighborhood and weigh their scores according to the content information.

The experiment shows how different metrics of ranking comparison can achieve different results for the same representation model and dataset. The use of a specific metric can vary according to the analysis's objective and the content-based representation's characteristics. When we want to evaluate the entire ranking, considering only the relative position of items, rank correlation metrics, such as Spearman's ρ, are more well-suited. In cases where only the quality of the neighborhood is important, disregarding the intensity of the item's similarity, metrics of set similarity are more recommended, such as the Jaccard Index. Lastly, when we are only interested in the neighborhood's items but want to weigh the results according to their similarity scores and rank positions, we can calculate utility-based metrics such as NDCG.

When comparing the content-based ranking metrics and the achieved results for the extrinsic evaluation, it is clear how representation models not useful for generating recommendations may still have value when used in intrinsic tasks. User2Vec, statistically proven as the worst method for the top-N recommendation problem, presented excellent results for some datasets on the content-based analysis, especially when calculating the NDCG. Additionally, the differences in the obtained results with the intruder detection's accuracy show how subjective approaches can lead to contrasting conclusions about the model's quality. This entire comparison demonstrates how a thoroughly performed analysis can lead to more knowledge about the behavior of representation models.

Dataset	Representation Model			
	ALS	BPR	I2V	U2V
Anime	0.1440	0.2504	**0.2801**	0.1860
BestBuy	0.4729	0.4648	**0.4839**	0.4798
Delicious	0.2922	0.2809	0.2773	**0.2927**
Last.FM	0.2325	0.3239	0.2567	**0.3551**
MovieLens	**0.2566**	0.2543	0.2270	0.2139

Table 7: Spearman's rank correlation coefficient ρ of the content-based ranking comparison.

Dataset	k	Representation Model			
		ALS	BPR	I2V	U2V
Anime	**10**	0.0368	0.0404	0.0320	**0.0416**
	15	0.0345	**0.0382**	0.0310	0.0379
	20	0.0327	**0.0368**	0.0302	0.0350
BestBuy	**10**	0.0040	0.0060	0.0220	**0.0263**
	15	0.0050	0.0072	0.0266	**0.0321**
	20	0.0057	0.0082	0.0308	**0.0368**
Delicious	**10**	0.0024	0.0023	**0.0026**	0.0026
	15	0.0032	0.0031	**0.0034**	0.0033
	20	0.0039	0.0037	0.0040	**0.0041**
Last.FM	**10**	0.0217	0.0176	**0.0249**	0.0187
	15	0.0271	0.0221	**0.0299**	0.0229
	20	0.0314	0.0258	**0.0349**	0.0266
MovieLens	**10**	0.0015	**0.0017**	0.0012	0.0014
	15	0.0019	**0.0021**	0.0015	0.0017
	20	0.0022	**0.0025**	0.0018	0.0019

Table 8: Jaccard Index@k of the content-based ranking comparison, with different values of k.

8 CONCLUSION

Embeddings with strong intrinsic meaning can benefit many tasks beyond recommendation. While intrinsic evaluation has gained attention in NLP, it is poised to become a focal point in recommender systems. However, intrinsic evaluations of vector representations for recommender systems are rarely conducted, with most studies focusing solely on extrinsic assessments. Even when intrinsic evaluations are performed, they often rely on human interaction, which can be time-consuming and susceptible to human bias.

This study presented approaches to assess the intrinsic quality of item embeddings. We first detailed a well-known evaluation method and adapted an NLP evaluation task for recommender systems. Since both methods are subjective and rely on human opinions, we also introduced two evaluation schemes based on objective metrics: a feature prediction task and a novel strategy for obtaining a quantitative score through content-based ranking comparison. For the latter, we provided various metrics to assess ranking quality. We compared four models that learn item embeddings across these evaluation approaches and conducted an extrinsic evaluation via traditional top-N ranking recommendation.

Dataset	k	Representation Model			
		ALS	BPR	I2V	U2V
Anime	**10**	0.4690	**0.4999**	0.4677	0.4594
	15	0.4563	**0.4882**	0.4579	0.4412
	20	0.4475	**0.4807**	0.4513	0.4290
BestBuy	**10**	0.1151	0.1486	0.3210	**0.3535**
	15	0.1107	0.1434	0.3086	**0.3407**
	20	0.1075	0.1400	0.3000	**0.3317**
Delicious	**10**	0.1850	0.1776	0.1847	**0.1862**
	15	0.1847	0.1775	0.1840	**0.1868**
	20	0.1845	0.1774	0.1829	**0.1870**
Last.FM	**10**	**0.4357**	0.3850	0.4139	0.4232
	15	**0.4351**	0.3850	0.4120	0.4226
	20	**0.4350**	0.3849	0.4107	0.4220
MovieLens	**10**	0.4333	**0.4439**	0.4023	0.3636
	15	0.4263	**0.4376**	0.3949	0.3560
	20	0.4210	**0.4330**	0.3893	0.3502

Table 9: NDCG@k of the content-based ranking comparison, with different values of k.

The extrinsic evaluation revealed similarities between the two matrix factorization methods and Item2Vec. Each model excelled on specific datasets, making it difficult to declare a superior model. Conversely, User2Vec performed poorly across all datasets in the extrinsic evaluation, emerging as the worst method. Intriguingly, the intrinsic tasks were the opposite. User2Vec excelled in generating representations with intrinsic value, ranking first or second for most datasets for both subjective and objective approaches. Our findings highlight the necessity of careful intrinsic evaluation to avoid misleading impressions of a model's capabilities.

Considering the presented metrics assume that metadata accurately describes the item, which may only sometimes be the case, for future research, we recommend a detailed study of content-based representation and item-sorting methods to improve the proposed strategy's quality. Utilizing more domain-specific datasets with extended feature sets and descriptive attributes can enrich the results. We also plan to analyze why each algorithm performed better in each task and dataset. Finding which characteristics favor specific evaluation scenarios can help future researchers develop task-specific recommenders. Additionally, a similar study focusing on user embeddings is suggested, using demographic information to overcome the problem of data scarcity users commonly have.

Ultimately, future studies on item embeddings should heed their intrinsic quality. An in-depth analysis can offer a comprehensive view of the models, useful for tasks beyond recommendation and potentially accelerating the development of new methods.

ACKNOWLEDGMENTS

This study was financed by the Brazilian Coordination for the Improvement of Higher Education Personnel (CAPES), Finance Code 88882.426978/2019-01; the Brazilian National Council for Scientific and Technological Development (CNPq); and The São Paulo Research Foundation (FAPESP), grants #2019/15393-4 and #2021/14591-7.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, 6 (2005), 734–749. https://doi.org/10.1109/TKDE.2005.99

[2] Oren Barkan and Noam Koenigstein. 2016. Item2Vec: Neural Item Embedding For Collaborative Filtering. In *IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP 2016)*. IEEE, Vietri sul Mare, Italy, 1–6. https://doi.org/10.1109/MLSP.2016.7738886

[3] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL '14)*. Association for Computational Linguistics, Baltimore, MD, USA, 238–247. https://doi.org/10.3115/v1/P14-1023

[4] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. 2013. Recommender systems survey. *Knowledge-Based Systems* 46 (2013), 109–132. https://doi.org/10.1016/j.knosys.2013.03.012

[5] Hugo Caselles-Duprés, Florian Lesaint, and Jimena Royo-Letelier. 2018. Word2vec applied to recommendation: hyperparameters matter. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. Association for Computing Machinery, Vancouver, Canada, 352–356. https://doi.org/10.1145/3240323.3240377

[6] Chao Chang, Junming Zhou, Yu Weng, Xiangwei Zeng, Zhengyang Wu, Chang-Dong Wang, and Yong Tang. 2023. KGTN: Knowledge Graph Transformer Network for explainable multi-category item recommendation. *Knowledge-Based Systems* 278 (2023), 110854. https://doi.org/10.1016/j.knosys.2023.110854

[7] Hao Chen, Zefan Wang, Feiran Huang, Xiao Huang, Yue Xu, Yishi Lin, Peng He, and Zhoujun Li. 2022. Generative Adversarial Framework for Cold-Start Item Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. Association for Computing Machinery, Anchorage, AK, USA, 2565–2571. https://doi.org/10.1145/3477495.3531897

[8] Gabriel de Souza P. Moreira, Dietmar Jannach, and Adilson Marques da Cunha. 2019. On the Importance of News Content Representation in Hybrid Neural Session-based Recommender Systems. *IEEE Access* 7 (2019), 169185–169203. https://doi.org/10.1109/ACCESS.2019.2954957

[9] Janez Demšar. 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. *The Journal of Machine Learning Research* 7 (2006), 1–30. https://doi.org/10.5555/1248547.1248548

[10] Chengxin Ding, Zhongying Zhao, Chao Li, Yanwei Yu, and Qingtian Zeng. 2023. Session-based recommendation with hypergraph convolutional networks and sequential information embeddings. *Expert Systems with Applications* 223, 119875 (2023), 1–11. https://doi.org/10.1016/j.eswa.2023.119875

[11] Douglas Eck, Paul Lamere, Thierry Bertin-Mahieux, and Stephen Green. 2007. Automatic generation of social tags for music recommendation. In *Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS 2007)*. Curran Associates Inc., Vancouver, Canada, 385–392. https://doi.org/10.5555/2981562.2981611

[12] Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems With Evaluation of Word Embeddings Using Word Similarity Tasks. In *Proceedings of the 1st Workshop on Evaluating Vector Space Representations for NLP*. Association for Computational Linguistics, Berlin, Germany, 30–35. https://doi.org/10.18653/v1/W16-2506

[13] Ralph José Rassweiler Filho, Jônatas Wehrmann, and Rodrigo C. Barros. 2017. Leveraging Deep Visual Features for Content-based Movie Recommender Systems. In *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN 2017)*. IEEE, Anchorage, AK, USA, 604–611. https://doi.org/10.1109/IJCNN.2017.7965908

[14] Claudiu S. Firan, Wolfgang Nejdl, and Raluca Paiu. 2007. The Benefit of Using Tag-Based Profiles. In *Proceedings of the 5th Latin American Web Conference (LA-WEB '07)*. IEEE Computer Society, Santiago, Chile, 32–41. https://doi.org/10.1109/LA-WEB.2007.24

[15] Peng FU, Jiang hua LV, Shi long MA, and Bing jie LI. 2017. Attr2vec: A Neural Network Based Item Embedding Method. In *Proceedings of the 2nd International Conference on Computer, Mechatronics and Electronic Engineering (CMEE 2017)*. DEStech Publications, Xiamen, China, 300–307. https://doi.org/10.12783/dtcse/cmee2017/19993

[16] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-REC: Towards Interactive and Explainable LLMs-Augmented Recommender System. *arXiv*: 2303.14524 (2023), 1–17. https://doi.org/10.48550/arXiv.2303.14524

[17] Anna Gladkova and Aleksandr Drozd. 2016. Intrinsic Evaluations of Word Embeddings: What Can We Do Better?. In *Proceedings of the 1st Workshop on Evaluating Vector Space Representations for NLP*. Association for Computational Linguistics, Berlin, Germany, 36–42. https://doi.org/10.18653/v1/W16-2507

[18] Mihajlo Grbovic, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikit Savla, Varun Bhagwan, and Doug Sharp. 2015. E-commerce in Your Inbox: Product Recommendations at Scale. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15)*. Association for Computing Machinery, Sydney, Australia, 1809–1818. https://doi.org/10.1145/2783258.2788627

[19] Asnat Greenstein-Messica, Lior Rokach, and Michael Friedman. 2017. Session-Based Recommendations Using Item Embedding. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces (IUI '17)*. Association for Computing Machinery, Limassol, Cyprus, 629–633. https://doi.org/10.1145/3025171.3025197

[20] S. Hasanzadeh, S. M. Fakhrahmad, and M. Taheri. 2020. Review-Based Recommender Systems: A Proposed Rating Prediction Scheme Using Word Embedding Representation of Reviews. *Comput. J.* bxaa044, ; (2020), 1–10. https://doi.org/10.1093/comjnl/bxaa044

[21] Antonio Hernando, JesÚs Bobadilla, and Fernando Ortega. 2016. A Non Negative Matrix Factorization for Collaborative Filtering Recommender Systems Based on a Bayesian Probabilistic Model. *Knowledge-Based Systems* 97, C (2016), 188—-202. https://doi.org/10.1016/j.knosys.2015.12.018

[22] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-Based Recommendations with Recurrent Neural Networks. In *Proceedings of the International Conference on Learning Representations (ICLR 2016)*. OpenReview, San Juan, Puerto Rico, 1–10.

[23] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM '08)*. IEEE Computer Society, Pisa, Italy, 263–272. https://doi.org/10.1109/ICDM.2008.22

[24] Salmo M.S. Júnior and Marcelo G. Manzato. 2015. Collaborative Filtering Based on Semantic Distance Among Items. In *Proceedings of the 21st Brazilian Symposium on Multimedia and the Web (WebMedia '15)*. Association for Computing Machinery, Manaus, Brazil, 53–56. https://doi.org/10.1145/2820426.2820466

[25] Shan Khsuro, Zafar Ali, and Irfan Ullah. 2016. Recommender Systems: Issues, Challenges, and Research Opportunities. In *Proceedings of the 7th International Conference on Information Science and Applications (ICISA 2016)*. Springer Science+Business Media, Ho Chi Minh, Vietnam, 1179–1189. https://doi.org/10.1007/978-981-10-0557-2_112

[26] Jooeun Kim, Jinri Kim, Kwangeun Yeo, Eungi Kim, Kyoung-Woon On, Jonghwan Mun, and Joonseok Lee. 2024. General Item Representation Learning for Cold-start Content Recommendations. *arXiv*: 2404.13808 (2024), 1–14. https://doi.org/10.48550/arXiv.2404.13808

[27] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques For Recommender Systems. *Computer* 42, 8 (2009), 30–37. https://doi.org/10.1109/MC.2009.263

[28] Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML 2014)*. JMLR.org, Beijing, China, 1188–1196. https://doi.org/10.5555/3044805.3045025

[29] Pasquale Lisena, Albert Meroño-Peñuela, and Raphaëla Troncy. 2022. MIDI2vec: Learning MIDI embeddings for reliable prediction of symbolic music metadata. *Semantic Web* 13, 3 (2022), 357–377. https://doi.org/10.3233/SW-210446

[30] Junling Liu, Chao Liu, Peilin Zhou, Qichen Ye, Dading Chong, Kang Zhou, Yueqi Xie, Yuwei Cao, Shoujin Wang, Chenyu You, and Philip S.Yu. 2023. LLM-Rec: Benchmarking Large Language Models on Recommendation Task. *arXiv*: 2308.12241 (2023), 1–13. https://doi.org/10.48550/arXiv.2308.12241

[31] Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, and Guangquan Zhang. 2015. Recommender system application developments: A survey. *Decision Support Systems* 74 (2015), 12–32. https://doi.org/10.1016/j.dss.2015.03.008

[32] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Conrado, and Jeffrey Dan. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS 2013)*. Curran Associates Inc., Stateline, NV, USA, 3111–3119. https://doi.org/10.5555/2999792.2999959

[33] Cataldo Musto, Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. 2017. Semantics-aware Recommender Systems exploiting Linked Open Data and graph-based features. *Knowledge-Based Systems* 136 (2017), 1–14. https://doi.org/10.1016/j.knosys.2017.08.015

[34] Makbule Gulcin Ozsoy. 2016. From Word Embeddings to Item Recommendation. *arXiv*: 1601.01356 (2016), 1–8. https://doi.org/10.48550/arXiv.1601.01356

[35] Yuanyuan Qiu, Hongzheng Li, Shen Li, Yingdi Jiang, Renfen Hu, and Lijiao Yang. 2018. Revisiting Correlations between Intrinsic and Extrinsic Evaluations of Word Embeddings. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data (CCL 2018)*. Springer, Changsha, China, 209–221. https://doi.org/10.1007/978-3-030-01716-3_18

[36] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks (LREC 2010)*. European Language Resources Association (ELRA), Valletta, Malta, 45–50. https://doi.org/10.13140/2.1.2393.1847

[37] Stefen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings*

of the 25th Conference on Uncertainty in Artificial Intelligence (UAI '09). AUAI Press, Montreal, Canada, 452–461. https://doi.org/10.5555/1795114.1795167

[38] Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. 2020. Neural Collaborative Filtering vs. Matrix Factorization Revisited. In *Proceedings of the 14th ACM Conference on Recommender Systems (RecSys '20)*. Association for Computing Machinery, Virtual Event, Brazil, 240–248. https://doi.org/10.1145/3383313.3412488

[39] Steffen Rendle, Walid Krichene, Li Zhang, and Yehuda Koren. 2022. Revisiting the Performance of iALS on Item Recommendation Benchmarks. In *Proceedings of the 16th ACM Conference on Recommender Systems (RecSys '22)*. Association for Computing Machinery, Seattle, WA, USA, 427–435. https://doi.org/10.1145/3523227.3548486

[40] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John T. Riedl. 2000. Application of Dimensionality Reduction in Recommender System - A Case Study. In *Proceedings of the 9th WebKDD Workshop on Web Mining for e-commerce (WebKDD '00)*. Association for Computing Machinery, Boston, Massachusetts, USA, 1–12. https://doi.org/10.21236/ada439541

[41] Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*. Association for Computational Linguistics, Lisbon, Portugal, 298–307. https://doi.org/10.18653/v1/D15-1036

[42] Guy Shani and Asela Gunawardana. 2011. Evaluating Recommendation Systems. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). Springer US, New York, NY, USA, Chapter 8, 257–259. https://doi.org/10.1007/978-0-387-85820-3

[43] Sumit Sidana, Mikhail Trofimov, Oleh Horodnytskyi, Charlotte Laclau, Yury Maximov, and Massih-Reza Amini. 2021. User preference and embedding learning with implicit feedback for recommender systems. *Data Mining and Knowledge Discovery* 35 (2021), 568–592. https://doi.org/10.1007/s10618-020-00730-8

[44] Abe Vallerian Siswanto, Lilian Tjong, and Yordan Saputra. 2018. Simple Vector Representations of E-commerce Products. In *2018 International Conference on Asian Language Processing (IALP 2018)*. IEEE, Bandung, Indonesia, 368–372. https://doi.org/10.1109/IALP.2018.8629245

[45] Yang Song, Lu Zhang, and Clyde Lee Giles. 2011. Automatic tag recommendation algorithms for social recommender systems. *ACM Transactions on the Web* 4, 1 (2011), 4:1–4:31. https://doi.org/10.1145/1921591.1921595

[46] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining (WSDM '18)*. Association for Computing Machinery, Marina Del Rey, CA, USA, 565–573. https://doi.org/10.1145/2939672.2939673

[47] Flavian Vasile, Elena Smirnova, and Alexis Conneau. 2016. Meta-Prod2Vec: Product Embeddings Using Side-Information for Recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. Association for Computing Machinery, Boston, Massachusetts, USA, 225–232. https://doi.org/10.1145/2959100.2959160

[48] Dongjing Wang, Guandong Xu, and Shuiguang Deng. 2017. Music recommendation via heterogeneous information graph embedding. In *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN 2017)*. IEEE, Anchorage, AK, USA, 596–603. https://doi.org/10.1109/IJCNN.2017.7965907

[49] Jiaqi Wang and Jing Lv. 2020. Tag-informed collaborative topic modeling for cross domain recommendations. *Knowledge-Based Systems* 203 (2020), 106119. https://doi.org/10.1016/j.knosys.2020.106119

[50] Qinyong Wang, Hongzhi Yin, Hao Wang, Quoc Viet Hung Nguyen, Zi Huang, and Lizhen Cui. 2019. Enhancing Collaborative Filtering with Generative Augmentation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '19)*. Association for Computing Machinery, Anchorage, AK, USA, 548–556. https://doi.org/10.1145/3292500.3330873

[51] Tian Wang, Yuri M. Brovman, and Sriganesh Madhvanath. 2021. Personalized Embedding-based e-Commerce Recommendations at eBay. *arXiv:* 2102.06156 (2021), 1–9. https://doi.org/10.48550/arXiv.2102.06156

[52] Heitor Werneck, Nícollas Silva, Matheus Carvalho Viana, Fernando Mour ao, Adriano C. M. Pereira, and Leonardo Rocha. 2020. A Survey on Point-of-Interest Recommendation in Location-based Social Networks. In *Proceedings of the Brazilian Symposium on Multimedia and the Web (WebMedia '20)*. Association for Computing Machinery, São Luís, Brazil, 185–192. https://doi.org/10.1145/3428658.3430970

[53] Dongqiang Yang, Ning Li, Li Zou, and Hongwei Ma. 2022. Lexical semantics enhanced neural word embeddings. *Knowledge-Based Systems* 252 (2022), 109298. https://doi.org/10.1016/j.knosys.2022.109298

[54] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang. 2024. Self-Supervised Learning for Recommender Systems: A Survey. *IEEE Transactions on Knowledge and Data Engineering* 36 (2024), 335–355. https://doi.org/10.1109/TKDE.2023.3282907

[55] Hafed Zarzour, Ziad A. Al-Sharif, and Yaser Jararweh. 2019. RecDNNing: a recommender system using deep neural network with user and item embeddings. In *Proceedings of the 10th International Conference on Information and Communication Systems (ICICS 2019)*. IEEE, Irbid, Jordan, 99–103. https://doi.org/10.1109/IACS.2019.8809156

[56] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative Knowledge Base Embedding for Recommender Systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. Association for Computing Machinery, San Francisco, CA, USA, 353–362. https://doi.org/10.1145/2939672.2939673

[57] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* 52, 1 (2019), 5:1–5:35. https://doi.org/10.1145/3285029

[58] Xiangyu Zhao, Maolin Wang, Xinjian Zhao, Jiansheng Li, Shucheng Zhou, Dawei Yin, Qing Li, Jiliang Tang, and Ruocheng Guo. 2023. Embedding in Recommender Systems: A Survey. *arXiv:* 2310.18608 (2023), 1–42. https://doi.org/10.48550/arXiv.2310.18608

[59] Lütfi Kerem Şenel, İhsan Utlu, Veysel Yücesoy, Aykut Koç, and Tolga Çukur. 2018. Semantic Structure and Interpretability of Word Embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26, 10 (2018), 1769–1779. https://doi.org/10.1109/TASLP.2018.2837384