Analyzing the Impact of Domain Similarity: A New Perspective in Cross-Domain Recommendation

Ajay Krishna Vajjala, Arun Krishna Vajjala, Ziwei Zhu, and David S. Rosenblum
Department of Computer Science
George Mason University
Fairfax, VA, USA
{akrish, akrishn, zzhu20, dsr}@gmu.edu

***Abstract*—Cross-domain recommendation (CDR) has recently emerged as an effective way to alleviate the cold-start and sparsity issues faced by recommender systems, by transferring information from an auxiliary domain to a target domain to improve recommendations. Studying the similarity between domains is a novel direction in CDR research, potentially opening doors for further exploration. In this context, we introduce a systematic approach to quantify similarity between a pair of domains and explore how current CDR methods perform with both similar and dissimilar domain combinations. We achieve this by presenting two original similarity metrics. Our extensive empirical evaluation on different domain combinations demonstrates that the state-of-the-art CDR algorithms do not perform significantly better when using source domains that are more similar to the target domain, compared to those that are less similar. Importantly, we find that no matter how similarity is measured, it does not correlate with the recommendation performance of the state-of-the-art algorithms.**

***Index Terms*—Domain Similarity, Information Retrieval, Cross-Domain Recommendation, Natural Language Processing**

I. Introduction

In this current day and age, many e-commerce applications rely on recommender systems to recommend items to their customers [1]. The abundance of available information in the digital world, which is growing at an exponential rate, has made it challenging to recommend personalized items to users efficiently [2], [3]. Most recommender systems focus on making recommendations in a single domain (e.g. recommending movies to users based on movie ratings).

Recently, cross-domain recommender systems have emerged as an approach to improve the quality of recommendations. These systems leverage information from a source domain to provide more accurate recommendations in a target domain [4]. Through information transfer, cross-domain recommender systems can provide relevant recommendations to new users, which helps mitigate the cold-start and sparsity issues faced by single-domain recommender systems [1].

Given that cross-domain recommendation (CDR) is a new field of research, a number of models have been proposed and studied [5]. Previous studies have mainly focused on transferring information between domains that are assumed to be related (e.g. books and movies) [6]. However, there is a lack of research examining the correlation between the similarity of two domains and its impact on cross-domain recommendation performance. Intuitively, the greater the similarity between the source domain and target domain, the better the recommendation performance from source to target. For instance, using knowledge about movie preferences should be a good basis for recommending TV shows, but knowledge about movie preferences may be less useful for recommending restaurants. But is that really the case? In this paper we try to answer this question by first defining a set of novel similarity metrics and then presenting results from an extensive set of experiments with three state-of-the-art approaches. Our empirical evaluation demonstrates that the current state-of-the-art CDR models do not perform significantly better with similar domain combinations, and we leave for future work the question of how domain similarity should be exploited in the design of a cross-domain recommendation algorithm. We make our data and code available at https://github.com/ajaykv1/Domain-Similarity-CDR. To summarize, our contribution is as follows:

- We present a novel set of metrics that aim to capture similarity between two distinct domains in the context of cross-domain recommendation.
- We conduct a comprehensive empirical evaluation to investigate the relationship between recommendation performance and domain similarity using three different CDR algorithms on 18 domain combinations.
- We analyze factors such as recommendation algorithm and similarity between domains, which may influence the recommendation performance.

II. Related Work

Cross-domain recommendation has gained significant attention as a way to address the cold-start and sparsity challenges faced by traditional single-domain recommender systems [7]. By leveraging transfer learning techniques, they aim to make better recommendations in the target domain [8]. In recent years, a number of methods have been proposed and shown to improve recommendation quality in cross-domain settings, but the domains selected in these studies lacked a formal criteria [9], [10]. The lack of a systematic way to measure similarity between domains for domain selection, makes it difficult to select the ideal domains for the task of cross-domain recommendation. As a result, we believe researchers are assuming that two domains relate to each other based on human intuition, or are choosing domains based on the limited data available.

Recently, new methods for CDR have been proposed, and they have been labeled as the state-of-the art algorithms for cross-domain recommendation [11], [12]. Hu et al. [13] proposed a transfer learning approach for CDR, known as Collaborative Cross Networks for Cross-Domain Recommendation (CoNet), that uses neural networks as the base model. They selected Books and Movies to be the domains from the Amazon dataset, solely based on the fact that they were the largest domains (contained more products) compared to the others in the dataset. The domains selected for the Mobile dataset were the different genres of news, and this selection was based on the assumption that they were related to each other [13]. Man et al. [14] proposed an embedding and mapping approach for cross-domain recommendation (EMCDR). They used the Movielens-Netflix dataset and the Douban dataset (which contains ratings users gave to books and movies) to evaluate their approach. For the Movielens-Netflix dataset, they picked Movielens as the source domain and Netflix as the target domain. For the Douban dataset, they picked Movies as the source and Books as the target domain [14]. Again, there was no clear reason for the choice of domains.

Recently, a study was conducted by Sahebi et al. [6] which explores domain pairs for cross-domain recommendation. They propose a method that uses Canonical Correlation Analysis (CCA) to find promising source domains for a target domain. To the best of our knowledge, we believe that this is the only study that explores the compatibility of domains, and provides a way to constructively choose a related source domain that will improve recommendations for a target domain. Based on the results for their method (CCA), they concluded that the more overlapping users between the source and target domain, the better the recommendation results are in the target domain. Their study differs from ours in that we develop the similarity metrics based on the item metadata from each domain and investigate the relationship between domain similarity and recommendation performance. In addition, we maintain a 100% user overlap between all the domain combinations to ensure that the results will not be affected by other factors.

III. Similarity Metrics

To the best of our knowledge, a systematic approach to measuring similarity between domains has not been explored in the field of cross-domain recommendation, making this study one of the initial efforts in doing so. Addressing this issue, we present two novel similarity metrics, which leverage item information from both domains. Given that many datasets for recommender systems associate items with one or more tags, we argue that a domain can be effectively characterized based on its items' tag information. Tags can carry high-level information about items, which can be useful despite the absence of other forms of meta-data. Usually, they are broad terms used as key words to describe items, and their broad nature allows them to describe items across multiple different domains (see Figure 1). Hence, we argue that tags are a strong text-based meta-data for computing similarity. The first similarity metric we present represents each domain as an individual embedding, and quantifies similarity between domains by computing the cosine similarity between their embeddings (see Section III-A). The second similarity metric we present focuses on the item-level similarity across two domains (see Section III-B).

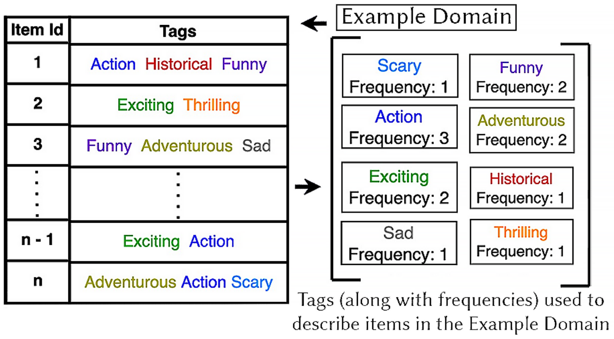

Fig. 1: This figure shows the process of extracting all the individual tags, along with their frequencies, for each domain.

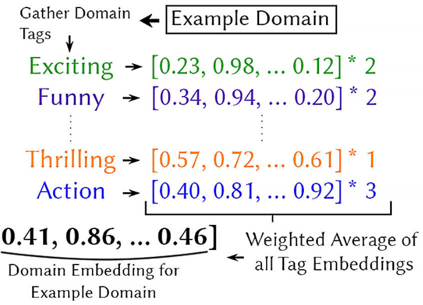
Gather Domain Tags
Example Domain
Exciting → [0.23, 0.98, ... 0.12] * 2
Funny → [0.34, 0.94, ... 0.20] * 2
Thrilling → [0.57, 0.72, ... 0.61] * 1
Action → [0.40, 0.81, ... 0.92] * 3
[0.41, 0.86, ... 0.46]
Weighted Average of all Tag Embeddings
Domain Embedding for Example Domain

Fig. 2: Computing the domain embedding using tags and frequencies within a domain (see Equation 1 for details).

A. Embedding-based Domain Similarity

We use proven NLP techniques to generate a detailed and cohesive embedding for each domain, which we refer to as a *domain embedding*. We leverage item tag information to create the domain embedding, so to ensure the most accurate representation, we generate embeddings for tags using pre-trained GloVe embeddings [15]. Pre-trained GloVe embeddings [15] are vector representations for words, which were generated by running the GloVe algorithm on a large corpus of text data. The various dimensions of the vector represent the underlying meanings of the word based on the context of how it is used. Words with similar meanings will have similar embeddings, while words with dissimilar meaning will have dissimilar embeddings [16].

For this metric, we gather a list of tags, where every tag in the list has been used to describe at least one item within the domain. In addition, we retrieve their frequency to document how many times each tag was used in the domain (see Figure 1). Tags with higher frequencies are important to characterize the domain, since they are used to describe a significant number of items. Essentially, the more often a tag is used, the more likely it is to be used as an indicator to describe the domain as a whole.

For each tag in the domain, we retrieve the corresponding embedding from the GloVe pre-trained embeddings. The GloVe pre-trained embeddings are available in different dimension sizes, which range between 50, 100, 200, and 300 dimensions [15]. We opted for the 300 dimensional embedding to retain as much information as possible from each tag. Tags with higher frequencies describe more items in the domain as opposed to tags with less frequencies, so higher frequency tags are given a higher weight since they carry more information. By providing a higher weight, we are able to preserve the importance of the tags and their value within the domain. Once the embeddings are gathered for all the tags, we compute a weighted average of all the tag embeddings by using their corresponding frequencies. The result of the weighted average is a single embedding, which is a detailed representation of the tag information within the domain (see Figure 2). Let $\boldsymbol{t_1}$ represent the embedding for the first tag in the domain, and $\boldsymbol{t_n}$ represent the embedding for the nth tag in the domain. Let $\boldsymbol{c_1}$ represent the frequency for of the first tag, and $\boldsymbol{c_n}$ represent the frequency of the nth tag. The weighted average of the tag embeddings is computed as follows:

$$t_{emb} = \frac{\sum_{i=1}^{n} t_i * c_i}{\sum_{i=1}^{n} c_i} \tag{1}$$

The tag embeddings are summed together, where each embedding is multiplied by its corresponding frequency, and the final embedding is divided by the sum of all the tag frequencies (see Equation 1). The resulting embedding, $\boldsymbol{t_{emb}}$, contains a deep understanding of items within the domain, and serves as a meaningful representation of a domain, because the knowledge of tags and their importance were preserved through the weighted average.

In this study, we use cosine similarity for the similarity computation between two embeddings. Cosine similarity measures based on orientation rather than magnitude, which makes it a good choice when working with higher dimensional data [17]. We measure the cosine similarity between two domain embeddings to measure how similar two domains are. Let $\boldsymbol{s_{emb}}$ represent the embedding for the source domain, and let $\boldsymbol{t_{emb}}$ represent the embedding for the target domain. The cosine similarity between the two domain embeddings can be computed as follows:

$$sim_{st} = \frac{s_{emb} \cdot t_{emb}}{\|s_{emb}\| \, \|t_{emb}\|} \tag{2}$$

The resulting value $\boldsymbol{sim_{st}}$ represents the similarity between the source ($\boldsymbol{s}$) and target ($\boldsymbol{t}$) domain (see Equation 2). We presented a novel way to represent a domain as an embedding in the context of cross-domain recommendation, and described how to measure similarity between two domains.

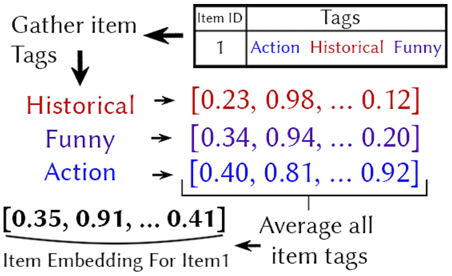

Fig. 3: Computing the embedding for an item using its tags.

B. *Inter-domain Item Similarity*

Domains generally consists of a set of items, and each item is represented by a set of tags. We represent each item as an embedding, using the GloVe pre-trained embeddings (see Section III-A), and find the item-level similarities across domains in order to quantify similarity between two distinct domains [15]. For every item, we collect its associated tags, and retrieve their corresponding embeddings from the GloVe pre-trained embeddings [15]. For this metric, we opted for the 300 dimensional embedding to retain as much information as possible from each tag. Once the pre-trained embeddings are gathered for all tags, we average them together to generate a single item embedding (see Figure 3) . Let $\boldsymbol{t_1}$ represent the embedding for the first tag, and let $\boldsymbol{t_n}$ represent the embedding for the nth tag used to describe the item $\boldsymbol{I}$. The item embedding is computed as follows:

$$I_{emb} = \frac{\sum_{i=1}^{n} t_i}{n} \tag{3}$$

The resulting embedding, $\boldsymbol{I_{emb}}$, is the representation of a single item within a domain, and we generate item embeddings for every item within a domain (using Equation 3). To compute item-level similarity between two domains, we make use of Simple Random Sampling (SRP). SRP is a sampling technique used in statistics, that gives each member an equal chance of being selected from a population sample. This allows for the selected samples to be bias free, and provides for higher generalization ability for the entire population. For this metric, we consider all the items within each domain as the population, and each item as an individual sample from the total population. To compute item-level similarity between two domains, we first randomly select a sample of 100 items from each domain. Next, we create combinations of pairs using the sampled items from both domains. Each combination is composed of one item from the first domain and one item from the second domain. We then calculate the cosine similarity between the embeddings of the two items within each pair. In order to represent the item-level similarity between the two domains, we average the similarity values across all pairs.

By selecting 100 samples from each domain at random (using SRP), we aim to reduce the potential bias in the similarity computation, and generalize the domain using the selected items. In addition, the computational efficiency is improved by using a smaller, random sample size, rather than considering every item in the domain. The item-level similarity across domains not only leverages individual item information but also provides an understanding of the relationship between items in both domains, making it novel in the context of cross-domain recommendation.

C. Construct Validity of Domain Similarity Computation

Given that our study is the first to introduce domain similarity for CDR, we adopt techniques from Measurement Theory to argue that our metrics show construct validity [18]. Fang et al. introduced two main properties that need to be satisfied in order to show that metrics have construct validity: (i) Face Validity; (ii) Content Validity [18].

In order to show face validity of our metrics, we need to confirm that the fundamental aspects of our approach (e.g. model and data) are suitable for the task of computing domain similarity. For our metrics, we use GloVe pre-trained embeddings to retrieve word embeddings for tags. Since we are dealing with tag information, we do not need to account for context, such as sentences, so the choice of GloVe is reasonable in that it captures both the semantic and syntactic meaning of individual words without context of other words. For the task of computing similarity between domains, we leverage tag information, which overlaps between various domains, making it a suitable choice of data. If we were to train on other sources of data (e.g. descriptions, images, etc.) there is no guarantee that the information would be consistent across domains, which would make similarity computation perform poorly. Therefore, we are able to show that our metrics satisfy face validity based on the model and data used to create embeddings.

The content validity of our domain similarity metrics is well-established through a theoretically grounded approach. In the first metric, domain embeddings are created by averaging tag embeddings from GloVe, a method widely recognized for capturing semantic relationships in text data. This approach ensures that each domain's embedding is a comprehensive representation of its key characteristics, as tags frequently used within a domain contribute more significantly to its overall representation. The second metric shows content validity by computing item-level similarities between domains. This is achieved by averaging GloVe embeddings for tags associated with each item, followed by a random sampling technique to ensure unbiased representation and computational efficiency. The use of cosine similarity in both metrics, a standard method for comparing high-dimensional semantic spaces, further solidifies their content validity. These metrics, therefore, offer a robust and theoretically sound framework for assessing domain similarities in cross-domain recommendation, reflecting a deep understanding of domain and item-level characteristics, which shows content-validity.

Fang et al. [18] mentions two other properties to validate construct validity, which includes Convergent and discriminant validity and Predictive validity, where they deal with comparing the similarity metrics in this paper with other established metrics. However, there are no established methods for computing domain similarity, making the additional tests for construct validity not applicable to this situation. Given that our study is the first to introduce domain similarity, we were able to show construct validity based on properties from measurement theory that were applicable in this study [18].

D. Limitations of Computing Similarity

There are some difficulties associated with finding similarity between domains for CDR. Since there is no ground truth about similarity between two domains, we don't know whether the similarity values generated by our metrics make sense in an absolute sense. We introduced two novel similarity measures for domain combinations, and the fact that there does not exist another similarity measure for CDR shows that we may have to rely on human intuition to validate the similarity values. However, in Section III-C, we were able to prove that our similarity metrics had construct validity. In addition, in Section IV-B, we show how the similarity values generated by our metrics contain validity, and how the results make sense based on one's intuition. Given the difficulties and limitations of computing similarity between domains for cross-domain recommendation, the metrics we present in this paper pave way for new research in computing domain similarity. Being able to quantify similarity between two domains systematically, in the context of cross-domain recommendation, is important for the future of CDR.

IV. Experiments and Results

We conduct thorough experiments to answer the following research questions: **RQ1:** How effective is the similarity computation between domains? **RQ2:** How does domain similarity correlate with the recommendation performance for the state-of-the-art CDR approaches? **RQ3:** Are GloVe pre-trained embeddings a reasonable choice for quantifying similarity between domains?

A. Experimental Setup

Domain	**User #**	**Item #**	**Inter. #**	**Sparsity(%)**
Comedy	2217	4977	35645	99.67%
Action	2217	2927	20960	99.67%
Adventure	2217	1070	7663	99.67%

TABLE I: Statistics of Movielens Domains

Domain	**User #**	**Item #**	**Inter. #**	**Sparsity(%)**
Romance	11878	1437	79525	99.53%
Historical	11878	465	25733	99.53%
Nonfiction	11878	547	30271	99.53%

TABLE II: Statistics of Books Domains

Domain	**User #**	**Item #**	**Inter. #**	**Sparsity(%)**
Music Instr.	8808	12406	28731	99.97%
Video Games	8808	15152	35100	99.97%
Software	8808	4456	10323	99.97%

TABLE III: Statistics of Amazon product Domains

*1) **Datasets**:* We used three real-world datasets to conduct the experiments. The first dataset, **Movielens-25M** [19], contains rating information provided by users from the Movielens website. It contains 5,000,095 ratings, provided by 162,541 users, and 1,093,36 tags across 62,423 movies [19]. We use

Domain Similarities with Domain Embeddings					
Movielens Dataset		**Books-genome Dataset**		**Amazon products Dataset**	
Domain Combinations	**Similarity**	**Domain Combinations**	**Similarity**	**Domain Combinations**	**Similarity**
Action & Adventure	0.96098	Romance & Historical	0.86616	Music Instr. & Software	0.49735
Action & Comedy	0.93134	Nonfiction & Historical	0.93392	Software & Video Games	0.76430
Comedy & Adventure	0.93051	Romance & Nonfiction	0.81063	Music Instr. & Video Games	0.54254

TABLE IV: Similarities between domain combinations across three different datasets, using the domain embedding method.

Item-Level Domain Similarities					
Movielens Dataset		**Books-genome Dataset**		**Amazon products Dataset**	
Domain Combinations	**Similarity**	**Domain Combinations**	**Similarity**	**Domain Combinations**	**Similarity**
Action & Adventure	0.35438	Romance & Historical	0.73957	Music Instr. & Software	0.33179
Action & Comedy	0.32491	Nonfiction & Historical	0.78319	Software & Video Games	0.45020
Comedy & Adventure	0.32860	Romance & Nonfiction	0.66449	Music Instr. & Video Games	0.34883

TABLE V: Similarities between the domain combinations across datasets using inter-domain item similarity approach.

movie genres as the domains, and we select Comedy movies, Action movies and Adventure movies to be the domains from this dataset. We maintain a 100% user overlap across the three domains, and regulate the sparsity to be the same for each domain (see Table I). The sparsity is regulated by removing ratings, at random, from the domains until the sparsity levels are equal to the most sparse domain. This allows for a controlled experiment that measures how domain similarity affects recommendation performance without any other factors affecting the results.

The second dataset, **Books-genome** [20], contains rating information provided by users from the Good Reads website. This dataset contains rating information for 350,332 users on 9,374 items, along with 727 tags to represent the items [20]. Similar to the Movielens dataset, we use genres of books to be the domains, and selected Romance books, Historical books, and Nonfiction books as the domains for this dataset. We maintain a 100% user overlap and regulate the sparsity to be similar across all the domains (see Table II).

The third dataset, **Amazon products** [21], contains rating information provided by users on various Amazon products. This dataset contains a total of 34,686,770 ratings, provided by 6,643,669 users across 2,441,053 items [13]. We select the Music Instrument products, Video Game products, and Software products as the three domains in this dataset. Similar to the previous datasets, we maintain a 100% user overlap between the domains, and regulate the sparsity to be equal across domains (see Table III).

These datasets are considered to be strong datasets for recommender systems research [13], [20], [22]–[24]. We believe there can be added complexity and challenges in drawing clear conclusion from the results if we increase the number of domains. As a result, we intentionally selected three domains from each dataset, where each domain contains a significant amount of data.

*2) **Baseline Cross-Domain Recommendation Models**:* We selected three prominent baselines based on the recent research in CDR. The first model is **CoNet** (Collaborative Cross Networks for Cross-Domain Recommendation) [13]. CoNet transfers knowledge across domains through cross-connections between two base neural networks, and learns complex user-item relationships by using deep transfer learning. The second baseline CDR model we used is **EMCDR** (Embedding and Mapping Approach for Cross Domain Recommendation) [14]. This model uses matrix factorization to learn latent factors for both domains, and employs a MLP network to map the user latent features from the source to the target domain. The third baseline model we used is **SSCDR** (Semi-Supervised Learning for Cross-Domain Recommendation) [25]. This model is an extension to EMCDR, which includes the item information from the source domain into the training process to learn a better representation of the data that will be transferred in the base MLP network. Many recent CDR methods have compared their algorithm performances against the baselines we selected for this study [7], [26]–[29]. In addition, majority of the recent CDR methods are extensions of CoNet and EMCDR, in that they add more domain specific information and complex neural architectures to show improvement [30]–[34]. The baselines we have chosen for this study are relatively recent, and have proven to be powerful competitors, making them deserving methods to represent the state-of-the-art for CDR [11], [35]. We used **Recbole-CDR** [36], an open source recommender systems library, for the implementations of these CDR baselines.

*3) **Evaluation Metrics**:* We choose hit ratio (HR), mean reciprocal rank (MRR), and normalized discounted cumulative gain (NDCG) to be the ranking metrics in this study. We evaluate the top 10 items. HR measures to see whether a test item is present within the top-N recommended items:

$$HR = \frac{1}{|U_{test}|} \sum_{u \in U_{test}} \delta(p_u \leq topN) \tag{4}$$

where $\boldsymbol{U_{test}}$ is the set of test users, $\boldsymbol{p_u}$ is the position of the test item for the user ($\boldsymbol{u}$), and $\boldsymbol{\delta(\cdot)}$ is the indicator function. MRR and NDCG are ranking metrics that assign higher scores to the items that appear higher in the top-N list of recommendations, and they are defined as follows:

$$NDCG = \frac{1}{|U_{test}|} \sum_{u \in U_{test}} \frac{log2}{log(p_u + 1)}, \quad MRR \;= \frac{1}{|U_{test}|} \sum_{u \in U_{test}} \frac{1}{p_u} \tag{5}$$

Movielens Dataset

EMCDR

T: **Adven.**	**MRR@10**	**NDCG@10**	**HR@10**
Action	0.0250	0.0361	0.0853
Comedy	0.0291**	0.0408**	0.0921**
Paired t-test	$p<0.4467$	$p<0.4595$	$p<0.3899$

T: **Action**	**MRR@10**	**NDCG@10**	**HR@10**
Adventure	0.0168	0.0209	0.0543
Comedy	0.0255**	0.0307**	0.0777**
Paired t-test	$p<0.1132$	$p<0.2047$	$p<0.1081$

T: **Comedy**	**MRR@10**	**NDCG@10**	**HR@10**
Action	0.0616**	0.0665**	0.1520**
Adventure	0.0533	0.0604	0.1510
Paired t-test	$p<0.6079$	$p<0.4058$	$p<0.7079$

SSCDR

T: **Adven.**	**MRR@10**	**NDCG@10**	**HR@10**
Action	0.0199	0.0289	0.0727
Comedy	0.0296**	0.0406**	0.0940**
Paired t-test	$p<0.8356$	$p<0.8096$	$p<0.6984$

T: **Action**	**MRR@10**	**NDCG@10**	**HR@10**
Adventure	0.0068	0.0085	0.0234
Comedy	0.0176**	0.0232**	0.0580**
Paired t-test	$p<0.6218$	$p<0.5569$	$p<0.9267$

T: **Comedy**	**MRR@10**	**NDCG@10**	**HR@10**
Action	0.0203**	0.0203**	0.0534**
Adventure	0.0141	0.0172	0.0473
Paired t-test	$p<0.1547$	$p<0.1741$	$p<0.2583$

CoNet

T: **Adven.**	**MRR@10**	**NDCG@10**	**HR@10**
Action	0.0391**	0.0531**	0.1105
Comedy	0.0363	0.0522	0.1143**
Paired t-test	$p<0.1359$	$p<0.1092$	$p<0.1469$

T: **Action**	**MRR@10**	**NDCG@10**	**HR@10**
Adventure	0.0295**	0.0354	0.0777
Comedy	0.0285	0.0354	0.0795**
Paired t-test	$p<0.3385$	$p<0.6454$	$p<0.7928$

T: **Comedy**	**MRR@10**	**NDCG@10**	**HR@10**
Action	0.0662**	0.0670**	0.1428**
Adventure	0.0626	0.0639	0.1382
Paired t-test	$p<0.2131$	$p<0.2317$	$p< 0.2443$

Books-Genome Dataset

EMCDR

T: **Romance**	**MRR@10**	**NDCG@10**	**HR@10**
Historical	0.1764	0.2053	0.3515
Nonfiction	0.1815**	0.2123**	0.2661**
Paired t-test	$p<0.6428$	$p<0.5658$	$p<0.6420$

T: **Nonfict.**	**MRR@10**	**NDCG@10**	**HR@10**
Historical	0.0474**	0.0658**	0.1296**
Romance	0.0442	0.0629	0.1288
Paired t-test	$p<0.6055$	$p<0.5978$	$p<0.5709$

T: **Historical**	**MRR@10**	**NDCG@10**	**HR@10**
Nonfiction	0.0646	0.0860	0.1584
Romance	0.0749**	0.0995**	0.1839**
Paired t-test	$p<0.7848$	$p<0.9377$	$p<0.4131$

SSCDR

T: **Romance**	**MRR@10**	**NDCG@10**	**HR@10**
Historical	0.0308	0.0434	0.1005
Nonfiction	0.1341**	0.1713**	0.3309**
Paired t-test	$p<2e$-5*	$p<2e$-8*	$p<3e$-9*

T: **Nonfict.**	**MRR@10**	**NDCG@10**	**HR@10**
Historical	0.0286	0.0435	0.0964
Romance	0.0365**	0.0547**	0.1183**
Paired t-test	$p<0.4446$	$p<0.3464$	$p<0.1971$

T: **Historical**	**MRR@10**	**NDCG@10**	**HR@10**
Nonfiction	0.0465	0.0677	0.1410
Romance	0.0890**	0.1178**	0.2146**
Paired t-test	$p<2e$-7*	$p<4e$-8*	$p<8e$-8*

CoNet

T: **Romance**	**MRR@10**	**NDCG@10**	**HR@10**
Historical	0.0323	0.0465	0.1085
Nonfiction	0.1610**	0.1940**	0.3433**
Paired t-test	$p<1e$-5*	$p<4e$-6*	$p<7e$-8*

T: **Nonfict.**	**MRR@10**	**NDCG@10**	**HR@10**
Historical	0.0634**	0.0895**	0.1799**
Romance	0.0469	0.0661	0.1316
Paired t-test	$p<0.2721$	$p<0.3880$	$p<0.5512$

T: **Historical**	**MRR@10**	**NDCG@10**	**HR@10**
Nonfiction	0.0948**	0.1202	0.2051
Romance	0.0926	0.1220**	0.2208**
Paired t-test	$p<0.5227$	$p<0.4746$	$p<0.3918$

Amazon Products Dataset

EMCDR

T: **Software**	**MRR@10**	**NDCG@10**	**HR@10**
Video Game	0.0032**	0.0044**	0.0084**
Music Instr.	0.0019	0.0024	0.0042
Paired t-test	$p<0.5089$	$p<0.7377$	$p<0.6128$

T: **Music Ins.**	**MRR@10**	**NDCG@10**	**HR@10**
Video Game	0.0208	0.0231	0.0381
Software	0.0268**	0.0285**	0.0399**
Paired t-test	$p<0.4787$	$p<0.5775$	$p<0.8759$

T: **Vid. Game**	**MRR@10**	**NDCG@10**	**HR@10**
Software	0.0054**	0.0069**	0.0132**
Music Instr.	0.0030	0.0042	0.0083
Paired t-test	$p<0.8103$	$p<0.9462$	$p<0.8283$

SSCDR

T: **Software**	**MRR@10**	**NDCG@10**	**HR@10**
Video Game	0.0017	0.0035	0.0098**
Music Instr.	0.0026**	0.0039**	0.0084
Paired t-test	$p<0.7429$	$p<0.6647$	$p<0.4814$

T: **Music Ins.**	**MRR@10**	**NDCG@10**	**HR@10**
Video Game	0.0046	0.0062	0.0142
Software	0.0051**	0.0069**	0.0149**
Paired t-test	$p<0.1756$	$p<0.1189$	$p<0.1737$

T: **Vid. Game**	**MRR@10**	**NDCG@10**	**HR@10**
Music Instr.	0.0056**	0.0071**	0.0141**
Software	0.0044	0.0058	0.0125
Paired t-test	$p<0.7198$	$p<0.6402$	$p<0.3438$

CoNet

T: **Software**	**MRR@10**	**NDCG@10**	**HR@10**
Video Game	0.0025	0.0056	0.0167**
Music Instr.	0.0050**	0.0067**	0.0126
Paired t-test	$p<0.2287$	$p<0.2533$	$p<0.2766$

T: **Music Ins.**	**MRR@10**	**NDCG@10**	**HR@10**
Video Game	0.0098	0.0125	0.0205
Software	0.0104**	0.0139**	0.0298***
Paired t-test	$p<0.7001$	$p<0.5820$	$p<0.4429$

T: **Vid. Game**	**MRR@10**	**NDCG@10**	**HR@10**
Music Instr.	0.0048	0.0066	0.0145**
Software	0.0052**	0.0066	0.0125
Paired t-test	$p<0.1533$	$p<0.1727$	$p<0.1275$

TABLE VI: Cross-domain recommendation results across three datasets for 18 domain combinations. The results for the Movielens dataset are on the left, the Books-genome dataset in the middle, and the Amazon products dataset on the right. Results show performance in target domain (T) when leveraging information from different source domains. For each target domain, the best result for each algorithm is marked with two stars($**$), and the most similar source domain is *italicised*. Two-tailed paired t-test shows significance of results in the target domain when using different source domains to transfer information. P-values less than 0.05 are marked with an asterisk ($*$) to show that the results in the target domain are significantly better when using one source domain compared to the other.

B. ***RQ1:*** *Domain Similarity Results*

Our similarity metrics, as shown in Table IV and Table V, align with our intuitive understanding of domain relationships. For instance, in the Movielens dataset, Action and Adventure domains are closely related, while Comedy is equidistant to both. A study done by Matthews et al. [37] gathered the topic compositions within each genre and plotted them in a high dimensional space, with similar genres close together and dissimilar genres being further apart. We can see that our similarity metrics mirror the findings of Matthews et al.'s study [37]. Similarly, in the Books-genome dataset, our metrics capture the similarity between domains accurately. Nonfiction and Historical are most similar, in line with Matthews et al.'s findings [37], which reinforces the effectiveness of our metrics. In the Amazon products dataset, our metrics reflect the natural relationships between domains. Software and Video Games, being closely related digital products, have the highest similarity, while Music Instruments and Video Games share some commonality due to music usage in games. Software and Music Instruments, being fundamentally different, exhibit the least similarity. The results for the similarity values between distinct domains, computed using our novel metrics, between the genres were consistent with the study done by Matthews et al. [37], and the similarity values for the domain combinations within the Amazon products dataset aligned with the regular intuition. As a result, we conclude that our similarity metrics captures the similarity between two domains effectively.

C. ***RQ2:*** *Recommendation Performance*

We evaluate recommendation performance of 3 baseline CDR models on 18 different domain combinations (see Table VI). We conduct a two-tailed paired t-test for each target domain to see if using a similar source domain produces statistically significant results compared to using a less similar source domain.

1) ***Movielens Dataset Results****:* In the Adventure domain, EMCDR and SSCDR outperformed other models when Com-

Domain Similarities with Domain Embeddings (Bert)					
Movielens Dataset		**Books-genome Dataset**		**Amazon products Dataset**	
Domain Combinations	**Similarity**	**Domain Combinations**	**Similarity**	**Domain Combinations**	**Similarity**
Action & Adventure	0.99763	Romance & Historical	0.97664	Music Instr. & Software	0.92729
Action & Comedy	0.99597	Nonfiction & Historical	0.99051	Software & Video Games	0.97908
Comedy & Adventure	0.99682	Romance & Nonfiction	0.96269	Music Instr. & Video Games	0.92182

TABLE VII: Domain embedding similarity using Bert

Item-Level Domain Similarities (Bert)					
Movielens Dataset		**Books-genome Dataset**		**Amazon products Dataset**	
Domain Combinations	**Similarity**	**Domain Combinations**	**Similarity**	**Domain Combinations**	**Similarity**
Action & Adventure	0.83287	Romance & Historical	0.95192	Music Instr. & Software	0.80179
Action & Comedy	0.80413	Nonfiction & Historical	0.95590	Software & Video Games	0.82267
Comedy & Adventure	0.80691	Romance & Nonfiction	0.93111	Music Instr. & Video Games	0.81660

TABLE VIII: Inter-domain item similarity using Bert

edy was the source domain, while CoNet performed better with Action as the source. Surprisingly, as shown by a two-tailed paired t-test, using Comedy as the source domain showed no statistically significant difference compared to Action, despite Adventure's higher similarity to Action. The same pattern held for the Action domain. In the Comedy domain, baseline CDR models performed better when Action was the source domain, but again, the results were statistically insignificant. Despite domain similarities, source domains didn't significantly impact results (see Tables IV and V). Based on the results from Tables IV and V, we can see that the similarity between Comedy and Action versus Comedy and Adventure is relatively equal, but the similarity does not matter due to insignifican results.

*2) **Books-genome Dataset Results**:* In the Romance target domain, Nonfiction as the source domain led to better results across all metrics compared to Historical. EMCDR and SSCDR showed significant improvements with Nonfiction as the source, but not CoNet. Surprisingly, despite Romance being more similar to Historical than Nonfiction, the latter resulted in significant improvements. This suggests a counter-intuitive negative correlation between similarity and recommendation performance. When Nonfiction was the target, EMCDR, and CoNet performed better with Historical as the source, while SSCDR did better with Romance as the source. Despite Historical's higher similarity to Nonfiction, no significant differences were observed as per the two-tailed paired t-test. In the Historical target domain, EMCDR and SSCDR outperformed others, and CoNet performed well with Romance as the source. However, only SSCDR showed statistically significant results when transferring from Romance, despite Romance being the most similar source to Historical. These inconsistencies suggest that current state-of-the-art baselines have difficulty consistently achieving significant results (see Tables IV and V for similarity details).

*3) **Amazon products dataset results**:* In the Software target domain, EMCDR performed well when using Video Games as the source, while CoNet and SSCDR performed well with Music Instruments as the source. Despite Software being more similar to Video Games, results varied across baseline models. However, the two-tailed paired t-test found no statistical significance, regardless of the source domain. When Music Instruments was the target domain, all baseline CDR models performed better with Software as the source. Surprisingly, even though Music Instruments are more similar to Video Games than Software, the results remained statistically insignificant. In the Video Games target domain, EMCDR outperformed with Software as the source, SSCDR performed well with Music Instruments, and CoNet showed no preference between the two. Yet, the paired t-test revealed no statistical significance, irrespective of the source domain.

*4) **Explanation for the Lack of Significant Results for the State-of-the-Art**:* Current state-of-the-art CDR methods focus on transferring user behavior across domains by constructing user profiles for users in the target domain, based on their interaction history in the source domain [13], [14], [25]. This approach relies on the assumption that user preferences will remain the same across different domains, no matter how similar or dissimilar the domains are. As a result, each domain is usually represented based on the interactions of users and items, rather than the shared characteristics of items across different domains. For example, they do not consider how items from one domain are connected to items from another domain, and how that can improve the creation of user profiles and recommendation results in a target domain. Based on the results in Table VI, we believe the insignificant results are a result of current CDR methods not effectively leveraging similarities between domains, by not exploiting other information outside of interaction patterns. The insignificant improvement when using more similar source domains does not necessarily imply the ineffectiveness of our similarity metrics, in fact, it can act as a sign that the current algorithms are not effectively leveraging similarities between domains. We argue that by leveraging all available similarities between domains (explicit and implicit), the recommendation performance in the target domain has a chance to improve significantly.

*D. **RQ3:** GloVe Embeddings for Similarity Computation*

We chose to use GloVe embeddings for word representation in this study. There have been recent advances in NLP that use transformer based architectures to create embeddings for text, for example, Bidirectional Encoder Representations from Transformers (BERT) has shown to be very effective in representing text as embeddings [38]. However, BERT is known for representing text sequences that contain multiple words as a single embedding, and it has proved to be very powerful [38]. In order to ensure a comprehensive exploration of options for word embeddings, we conducted additional experiments that use BERT embeddings for our similarity metrics (Table VII and Table VIII). The results showed the similarity values between domain combinations is less distinct, and compared to GloVe, the similarity scores were much higher between domains. Given that BERT embeddings consist of high dimensions (786), and the embeddings are normalized, the results are not surprising [38]. Considering that GloVe was provided a clear differentiation in similarity values across domain combinations, along with the computational efficiency, we used GloVe for our study. However, we encourage exploration of other embedding methods for future work.

V. Conclusion

We presented two novel similarity metrics to quantify similarity systematically between two domains in the context of cross-domain recommendation. We demonstrated that our metrics possess construct validity, and showed they can effectively produce similarity values that match with an existing study. We conducted an extensive evaluation across different datasets to see if the current state-of-the-art algorithms made significantly better recommendations in the target domain with more similar source domains compared to less similar source domains. From the experiments, we found that the CDR models do not perform significantly better when using more similar source domains compared to less similar source domains, and no matter how similarity between domains is measured, the results of the recommendation performance do not correlate with the similarity values.

References

[1] R. Véras, D. and C. Ferraz, “Cd-cars: Cross-domain context-aware recommender systems,” *Expert Systems with Applications*, 2019.

[2] I. Fernández-Tobías, I. Cantador, M. Kaminskas, and F. Ricci, “Cross-domain recommender systems: A survey of the state of the art,” in *Spanish conference on information retrieval*, 2012.

[3] F. Ricci, L. Rokach, and B. Shapira, “Recommender systems: introduction and challenges,” *Recommender systems handbook*, 2015.

[4] A. Krishna Vajjala, D. Meher, S. Pothagoni, Z. Zhu, and D. Rosenblum, “Vietoris-rips complex: A new direction for cross-domain cold-start recommendation,” 2024.

[5] A. Sahu and P. Dwivedi, “User profile as a bridge in cross-domain recommender systems for sparsity reduction,” *Applied Intelligence*, 2019.

[6] S. Sahebi and P. Brusilovsky, “It takes two to tango: An exploration of domain pairs for cross-domain collaborative filtering,” in *ACM RecSys*, 2015.

[7] H. Wang, Y. Zuo, H. Li, and J. Wu, “Cross-domain recommendation with user personality,” *Knowledge-Based Systems*, 2021.

[8] Z. Xu, F. Zhang, W. Wang, H. Liu, and X. Kong, “Exploiting trust and usage context for cross-domain recommendation,” *IEEE Access*, 2016.

[9] F. Zhu, C. Chen, Y. Wang, G. Liu, and X. Zheng, “Dtcdr: A framework for dual-target cross-domain recommendation,” in *CIKM*, 2019.

[10] T. Anwar and V. Uma, “Cd-spm: Cross-domain book recommendation using sequential pattern mining and rule mining,” *Journal of King Saud University-Computer and Information Sciences*, 2019.

[11] T. Zang, Y. Zhu, H. Liu, R. Zhang, and J. Yu, “A survey on cross-domain recommendation: taxonomies, methods, and future directions,” *ACM Transactions on Information Systems*, 2022.

[12] C. Zhao, C. Li, R. Xiao, H. Deng, and A. Sun, “Catn: Cross-domain recommendation for cold-start users via aspect transfer network,” in *ACM SIGIR*, 2020.

[13] G. Hu, Y. Zhang, and Q. Yang, “Conet: Collaborative cross networks for cross-domain recommendation,” in *CIKM*, 2018.

[14] T. Man, H. Shen, X. Jin, and X. Cheng, “Cross-domain recommendation: An embedding and mapping approach.” in *IJCAI*, 2017.

[15] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *EMNLP*, 2014.

[16] A. Kurdija, P. Afric, L. Sikic, B. Plejic, M. Silic, G. Delac, K. Vladimir, and S. Srbljic, “Building vector representations for candidates and projects in a cv recommender system,” in *AIMS*, 2020.

[17] F. Rahutomo, T. Kitasuka, and M. Aritsugi, “Semantic cosine similarity,” in *The 7th international student conference on advanced science and technology ICAST*, 2012.

[18] Q. Fang, D. Nguyen, and D. Oberski, “Evaluating the construct validity of text embeddings with application to survey questions,” *EPJ Data Science*, 2022.

[19] M. Harper and J. Konstan, “The movielens datasets: History and context,” *TiiS*, 2015.

[20] D. Kotkov, A. Medlar, A. Maslov, U. Satyal, M. Neovius, and D. Glowacka, “The tag genome dataset for books,” in *CHIIR*, 2022.

[21] J. Ni, J. Li, and J. McAuley, “Justifying recommendations using distantly-labeled reviews and fined-grained aspects,” in *EMNLP*, 2019.

[22] G. Behera and N. Nain, “Collaborative filtering with temporal features for movie recommendation system,” *Procedia Computer Science*, 2023.

[23] S. Shekhar, A. Singh, and A. Gupta, “A deep neural network (dnn) approach for recommendation systems,” in *CICT*, 2022.

[24] N. Joorabloo, M. Jalili, and Y. Ren, “Improved recommender systems by denoising ratings in highly sparse datasets through individual rating confidence,” *Information Sciences*, 2022.

[25] S. Kang, J. Hwang, D. Lee, and H. Yu, “Semi-supervised learning for cross-domain recommendation to cold-start users,” in *CIKM*, 2019.

[26] L. Zhang, Y. Ge, J. Ma, J. Ni, and H. Lu, “Knowledge-aware neural collective matrix factorization for cross-domain recommendation,” *arXiv*, 2022.

[27] R. Wang, Z. Xie, G. Qi, and P. Li, “Naui: Neural attentive user interest model for cross-domain ctr prediction,” in *Journal of Physics: Conference Series*, 2021.

[28] H. Li, L. Yu, Y. Leng, and Q. Du, “Co-capsule networks based knowledge transfer for cross-domain recommendation,” in *ICASSP*, 2021.

[29] A. Gkillas and D. Kosmopoulos, “A cross-domain recommender system using deep coupled autoencoders,” *arXiv*, 2021.

[30] P. Li and A. Tuzhilin, “Ddtcdr: Deep dual transfer cross domain recommendation,” in *WSDM*, 2020.

[31] W. Liu, X. Zheng, M. Hu, and C. Chen, “Collaborative filtering with attribution alignment for review-based non-overlapped cross domain recommendation,” in *WWW*, 2022.

[32] H. Kuang, W. Xia, X. Ma, and X. Liu, “Deep matrix factorization for cross-domain recommendation,” in *IAEAC*, 2021.

[33] P. Li, “Leveraging multi-faceted user preferences for improving click-through rate predictions,” in *ACM RecSys*, 2021.

[34] H. Yan, P. Zhao, F. Zhuang, D. Wang, Y. Liu, and V. Sheng, “Cross-domain recommendation with adversarial examples,” in *DASFAA*, 2020, pp. 573–589.

[35] Y. Zhu, Z. Tang, Y. Liu, F. Zhuang, R. Xie, X. Zhang, L. Lin, and Q. He, “Personalized transfer of user preferences for cross-domain recommendation,” in *WSDM*, 2022.

[36] W. X. et al, “Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms,” in *CIKM*, 2021.

[37] P. Matthews and K. Glitre, “Genre analysis of movies using a topic model of plot summaries,” *JASIST*, 2021.

[38] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *HLT-NAACL*, 2019.