IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 15, 2019, accepted August 4, 2019, date of publication August 12, 2019, date of current version August 22, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2934633

Mining Semantic Knowledge Graphs to Add Explainability to Black Box Recommender Systems

MOHAMMED ALSHAMMARI[1,2], OLFA NASRAOUI[1], AND SCOTT SANDERS[3]

[1]Knowledge Discovery and Web Mining Laboratory, Department of Computer Engineering and Computer Science, University of Louisville, Louisville, KY 40292, USA
[2]Computer Science Department, Northern Border University, Rafha 76313, Saudi Arabia
[3]Department of Communication, University of Louisville, Louisville, KY 40292, USA

Corresponding author: Mohammed Alshammari (msalsh03@louisville.edu)

This work was supported by the Northern Border University, Saudi Arabia through a scholarship.

ABSTRACT Recommender systems are being increasingly used to predict the preferences of users on online platforms and recommend relevant options that help them cope with information overload. In particular, modern model-based collaborative filtering algorithms, such as latent factor models, are considered state-of-the-art in recommendation systems. Unfortunately, these black box systems lack transparency, as they provide little information about the reasoning behind their predictions. White box systems, in contrast, can, by nature, easily generate explanations. However, their predictions are less accurate than sophisticated black box models. Recent research has demonstrated that explanations are an essential component in bringing the powerful predictions of big data and machine learning methods to a mass audience without compromising trust. Explanations can take a variety of formats, depending on the recommendation domain and the machine learning model used to make predictions. The objective of this work is to build a recommender system that can generate both accurate predictions and semantically rich explanations that justify the predictions. We propose a novel approach to build an explanation generation mechanism into a latent factor-based black box recommendation model. The designed model is trained to learn to make predictions that are accompanied by explanations that are automatically mined from the semantic web. Our evaluation experiments, which carefully study the trade-offs between the quality of predictions and explanations, show that our proposed approach succeeds in producing explainable predictions without a significant sacrifice in prediction accuracy.

INDEX TERMS Artificial intelligence, recommender systems, collaborative filtering, matrix factorization, explanations, semantic web.

I. INTRODUCTION

Recommender systems are being increasingly used on online platforms to predict the preferences of users and recommend relevant options to them. In particular, matrix factorization (MF) [1] is a powerful recommendation model that can produce accurate recommendations, but unfortunately lacks transparency. This means that it fails to explain the reasons for its outputs, thus, it is called a black box recommender system. (see Figure 1).

Moreover, users' explicit preferences may not be enough for the model to consider some items in the process of recommending new items. Since users may not have given new items any preferences, these items may be discarded. This is known as the cold-start problem in the field of recommender systems.

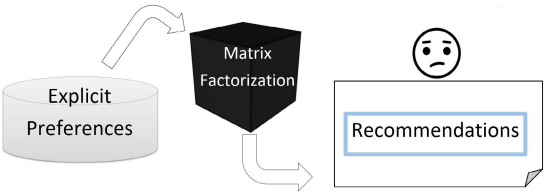


FIGURE 1. Non-Explainable Black Box Matrix Factorization.

Additional information can be used to overcome both the black box and cold-start problems. Information can be found in semantic knowledge graphs (KGs) as defined by Kroetsch and Weikum [2] are ''large networks of entities,

The associate editor coordinating the review of this article and approving it for publication was Ting Li.

their semantic types, properties, and relationships between entities'' built using semantic web technologies. Linked Open Data (LOD) [3] is a platform for linked, structured, and connected data on the web. The goal of LOD is to make information machine processable and semantically linked. For example, in the movie domain, information about movie stars or directors is available in a linked way. If an actor has starred in two movies, those two movies are linked. This can help us infer new facts about movies that eventually lead to the resolution of the cold start and transparency problems mentioned earlier.

Our research question is as follows: can we build semantic knowledge graphs (KGs) about users, items, and semantic attributes to generate explanations for a black box recommender system, while maintaining high prediction accuracy?

This paper's contribution consists of solving the problem of a non-transparent MF recommender system, in addition to constructing semantic KGs about users, items, and semantic attributes for the inference and explanation process (see Figure 2).

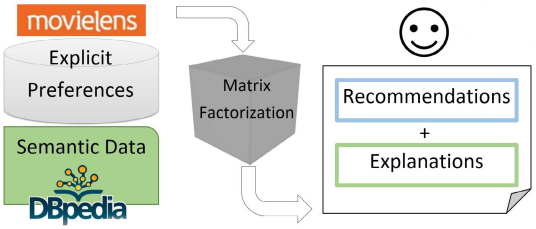


FIGURE 2. Explainable Black Box Matrix Factorization.

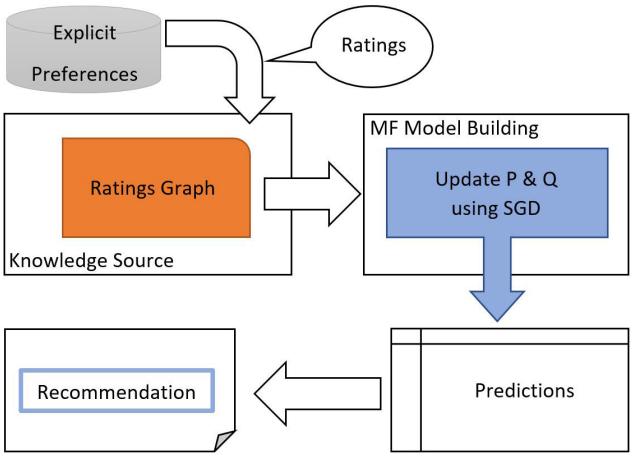


FIGURE 3. Matrix Factorization Flowchart [1].

II. RELATED WORK

A. BLACK BOX RECOMMENDER SYSTEMS

MF is a powerful family of techniques used to build recommender systems [1]. MF aims to learn the latent space vectors p and q for each user and item, respectively. Figure 3 presents a flowchart of the MF method, note that SGD stands for Stochastic Gradient Descent.

The idea is to factorize the rating matrix into lower dimensional spaces using a given number of latent features such that the dot product of the two latent space representations approximates the original ratings in addition to predicting the ratings of unseen items, as shown in 1.

$$\hat{r}_{ij} = p_i q_j^T, \tag{1}$$

where $\hat{r}$ denotes the training set of known ratings, where i and j represent a user and an item, respectively. The latent space vectors, p and q, are found by minimizing the following objective function over known ratings [1]:

$$J = \sum_{i,j \in R} \left(r_{ij} - p_i q_j^T \right)^2 + \beta(\| p_i \|^2 + \| q_j \|^2), \tag{2}$$

which includes a regularization term for each unknown parameter (latent vector) to avoid over-fitting, with β being a regularization coefficient that controls the smoothness of this newly added term. R represents the rating matrix and $R_u i$ denote the rating of item i by user u. J is not convex with respect to all the unknown parameters but is convex with respect to either p or q alone. Therefore, a stochastic gradient descent method [4] is used to find the optimal minimum. The updated rules for the user and item latent factor parameters, p and q, are given by

$$p_u^{(t+1)} \leftarrow p_u^{(t)} + \alpha(2(R_{u,i} - p_u^{(t)}(q_i^{(t)})^T)q_i^{(t)} - \beta p_u^{(t)}) \tag{3}$$

$$q_i^{(t+1)} \leftarrow q_i^{(t)} + \alpha(2(R_{u,i} - p_u^{(t)}(q_i^{(t)})^T)p_u^{(t)} - \beta q_i^{(t)}). \tag{4}$$

The authors in [5] took advantage of Koren's work [1] by incorporating not only known ratings but also side information that comes in two forms: the user side and the item side. Information such as age and gender comes from the user side; however, this information is not always available due to privacy issues. In contrast, genre, size, color, and actors in a film are item side information, and these data are almost always available. Reference [5] took advantage of this extra information to enhance the accuracy of the movie recommendations. They extended the basic MF cost function to a joint MF (JMF) that involves additional terms for item side information. Since the researchers conducted their experiments in the movie domain, they used two types of movie side information: mood and plot keywords. Two movie-by-movie similarity matrices were constructed, using two similarity methods, to be added to the cost function as new terms. According to the authors' comparison, their approach outperformed several other non-context-aware approaches by 10 percent over all matrices.

Kushwaha *et al.* [6] developed the approach of [5] by exploiting the power of the semantic web. In this study, the authors tested the proposed method in two domains: music and movies. They extracted semantic information from the DBpedia [1] dataset, which is a semantic version of Wikipedia, and then they retrieved the artist category information from DBpedia using SPARQL, a semantic web query language, to enrich the model's item side information. Like [5], they

[1] dbpedia.org

constructed a new matrix for the semantic information, added a new term to the JMF cost function, and obtained better results in comparison to JMF and other techniques. Building low dimensional representations of users and items using multiple sources of knowledge was explored by [7]. In their work, they succeeded in building a model to annotate images using the so-called bag-of-features method for image representation and non-negative matrix factorization (NMF) for building the low dimensional latent vector representation. Later, this approach was used in [8] to propose a solution for the item cold-start problem in collaborative filtering using MF. The notion here is to utilize multiple domains in the process of building the model. More specifically, item content features, such as genre, are used to build the items' latent space before learning the user's latent space using another domain, namely the known ratings. Although this approach integrated two sources of data to overcome the cold-start problem, it did not provide explainable recommendations.

B. EXPLANATIONS IN BLACK BOX RECOMMENDER SYSTEMS

Explaining black box recommender systems has been the subject of several studies. RippleNet [9] is an approach that used KGs in collaborative filtering to provide side information for the system in order to overcome sparsity and the cold-start problem. This black box system takes advantage of KGs, which are constructed using Microsoft Satori, to better enhance recommendation accuracy and transparency. The authors simulate the idea of water ripple propagation in understanding user preferences by iteratively considering more side information and propagating the user interests. In the evaluation section, the authors claim that their model is better than state-of-the-art models. The research of [10] focuses on adding explanations to a black box recommender system by using structured knowledge bases. The system takes advantage of historical user preferences to produce accurate recommendations and structured knowledge bases about users and items for generating justifications. After the model recommends items, a soft matching algorithm is used, utilizing the knowledge bases to provide personalized explanations for the recommendations. The authors argue that their model outperforms other baseline methods. Bellini *et al.* [11] focuses on the issue of explaining the output of a black box recommender system. In that work, the SemAuto recommender system is built using the autoencoder neural network technique, which is aware of KGs retrieved from the semantic web. The KGs are adopted for explanation generation. The authors claim that explanations increase the users' satisfaction, loyalty, and trust in the system. In their study, three explanation styles are proposed: popularity-based, pointwise personalized, and pairwise personalized. For evaluation, an A/B test was conducted to measure the transparency of, trust in, satisfaction with, persuasiveness of, and effectiveness of the proposed explanations. The pairwise method was preferred by most users over the pointwise method. Another approach that is explainable and semantic-aware is given in Yang *et al.* [12], where a post-hoc mechanism was proposed to generate explanations. After building the SEP recommender system, a unified heterogeneous information network (HIN) is built to provide justification for the recommended items. Explanation paths between the target user and other system components are established, ranked, and then used to show the explanations. To rank the explanation path candidates, three ranking metrics are used: credibility, readability, and diversity. Abdollahi and Nasraoui [13]–[15] investigate the possibility of generating explanations for the output of a black box system using a neighborhood technique based on cosine similarity. The results show that Explainable Matrix Factorization (EMF) performs better than the baseline approaches in terms of the error rate and the explainability of the recommended items. An example recommendation explanation is shown in Figure 4.

Your ratings for similar movies

Movie	Your rating (1-5)
Batman	5
Twilight	5
Scream 2	4
Space Jam	3
Dead Man	5

Your neighbor's ratings for this movie

Rating	Number of Neighbors
★	0
★★	0
★★★	6
★★★★	8
★★★★★	7

FIGURE 4. Example of an Explanation of EMF [13]–[15].

LOD has become popular in recent years due to the collaborative efforts of the semantic web community [16]. The structure of this enormous amount of data follows the standards of the resource description framework (RDF). Several studies have exploited LOD in improving recommender systems. Passant [17] is one of the first to use semantic web technologies in this field. The proposed method calculates the similarity between items to produce a list of recommendations. The proposed system takes advantage of the linked data semantic distance (LDSD) algorithm [18], as well as DBpedia, the ontological version of Wikipedia, to retrieve more details about songs' artists for the music recommendation system. Reference [19] used a linked data semantic distance (LDSD) algorithm [18] to build a model that recommends songs. Reference [17] used property values to explain why a certain artist was recommended. Following is an example of their explanation: *Johnny Cash and Elvis Presley share the same value for 'death place': Tennessee.*.

TasteWeights [20] is an interactive hybrid recommender system designed for the music domain.[2] Several sources of information, such as Twitter, Facebook, and Wikipedia, are utilized as a data source for the recommendation process. In addition to generating a visual interactive interface that provides justifications to users, the explanation interface allows users to choose the source of the explanation. If the user chooses to see an explanation based on Facebook data, then users will see their friends who liked the recommended item as an explanation. The same output happens when

[2]A demo video is available at http://bit.ly/TasteWeights

Wikipedia or Twitter is chosen. The system consists of three layers. The first one contains users' liked music gathered from the user's Facebook page. The second layer is the content layer where items' features are listed from all three information sources (i.e., Wikipedia, Facebook, and Twitter). The third layer is the recommendation layer that shows the top recommended items. When retrieving information from Wikipedia, the semantic version of it, DBpedia, is used to perform the task using the query language SPARQL. The authors indicate that as Herlocker [21] and Middleton [22] emphasized previously, an explanation increases the acceptance of a recommendation, and an explanatory interface also helps users understand why certain recommendations are shown for them. It also encourages users to get educated and involved in the recommendation process. Thirty-two real users participated in a user study to evaluate the system's performance and how well the explanation interface helped them understand the recommendation process. The authors concluded that although Wikipedia, when it was the source of the explanation, was more accurate than both Facebook and Twitter, explanations based on Facebook friends was favored by users due to trust in their friends' interests and tastes.

Hu *et al.* [23] emphasized the importance of explanations in recommender systems. In their approach, they relied on HIN [24] to generate semantic and justifiable recommendations. Another study that relied on the HIN technique to build a recommender system is SemRec [25]. In this work, the meta-paths obtained from the HIN are personalized and prioritized to accommodate users' preferences. The cold-start problem is resolved in this work, and they stated that their model outperformed other baseline methods in terms of producing a lower error rate. A study conducted by Musto *et al.* [26] shed light on the significance of natural language explanations in recommender systems and how linked open data can empower them by linking the user's previously preferred items and items' attributes to the new recommendations. The explanation mechanism is based on the notion that descriptive properties that describe the items that the user liked in the past can serve as explanations for the outputs of the recommender system. A user study was conducted to evaluate the system, and the results show that the proposed system succeeded in producing transparent recommendations and explanations. The next study is a Master's thesis written by Ul Haq [27]. The author proposed a hybrid, white box, and explainable approach for recommending movies. In this study, both collaborative and content-based filtering techniques were used and relied on additional information obtained from items and users. The author emphasized that interpretations are crucial in gaining customers' trust and satisfaction. A user study was conducted that included fifty participants to test the system. The results show that most participants preferred to see justifications alongside the recommendations. MoviExplain [3] is a project created by Symeonidis *et al.* [28]). It utilizes the idea of [29] where users are grouped into biclusters, which means each set of users are assigned to a set of movies. One benefit of this technique is that a feature, such as genre, could be extracted from this assignment, leading to explaining the recommendation to users based on this feature. The styles of explanation used in this study were KSE and ISE in addition to a mixed style of the two previous styles, which they called KISE. A user study was conducted in an attempt to justify their assumption that KISE's explanation style is better than the other two styles; they reached the conclusion that their proposed style was preferable by users more than the other two styles using various statistical metrics, such as mean, standard deviation, and Pearson correlation.

Another study [30]) used community tags to explain recommendations. In this study, they categorized explanations into three types: 1) item-based, where explanations were created based on similar items, 2) user-based, where the system relied on similar users to explain its recommendations, and 3) feature-based, where various features, such as genre, were used to justify the output. The authors of this work used the KSE explanation style. An example of an explanation could look like "*This movie is being recommended to you because it is tagged with* ***mystery***, *which exists in movies you've liked previously*".

III. PROPOSED METHODS

A. SEMANTIC KNOWLEDGE GRAPHS (KGS)

The web is abundant with information that is being harvested and structured into Knowledge Graphs (KGs). KGs are extensive networks of objects, along with their properties, their semantic types, and the relationships between objects representing factual information in a specific domain [31]. Examples of KGs are DBpedia [32], Freebase [33], Wikidata [34], YAGO [35], NELL [36], and the Google Knowledge Graph [37]. In this study, DBpedia is used to build the desired KGs about users, items, and semantic properties. In contrast with [38], where only one semantic attribute (actors) was considered in building the KG and, hence, the model, more influential semantic attributes (subject(s), actor(s), director(s), producer(s), and writer(s)) are included in the current paper to compute the similarity between items. The LDSD algorithm [18] is used to weigh the similarity between items. Then, Matrix Factorization (MF), [1] with an added regularization term for Joint MF (JMF) [5], is used for building the model.

1) LINKED DATA SEMANTIC DISTANCE (LDSD)

Passant [19] proposed a method to build a music recommender system using Semantic Web resources. The proposed algorithm captures both in-going and out-going as well as direct and indirect links between entities. Figure 5 shows a generic example of a semantic KG containing entities and links. The symbol r_i in Figure 5 represents a resource (e.g. movie, actor, etc). Whereas l_j is a link or property

[3] http://delab.csd.auth.gr/MoviExplain

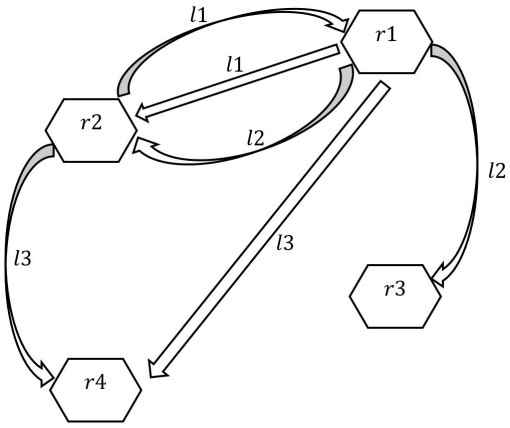


FIGURE 5. A Generic Semantic Knowledge Graph.

(e.g. starring, directedBy, etc). Out of r_i and l_j we can extract six RDF triples that exist in this graph and they are:

$$E = \{(l1 : r1 : r2),\ (l1 : r2 : r1),\ (l2 : r1 : r2),\ (l2 : r1 : r3),\ (l3 : r1 : r4),\ (l3 : r2 : r4)\}.$$

As mentioned earlier, there are in-going and out-going, as well as direct and indirect relationships between resources, which in total, represent the Linked Open Data (LOD). We explore using three semantic similarities previously used in mining knowledge graphs [19]. These are direct, indirect, and combined similarities, as described below:

a: DIRECT SIMILARITY

If there exists a property (l_x) that directly links two resources (r_y and r_z), then the direct similarity value $C^{(d)}_{l_x,r_y,r_z}$ is 1, otherwise 0:

$$C^{(d)}_{l_x,r_y,r_z} = \begin{cases} 1 & if\ exists\ (l_x : r_y : r_z) \\ 0 & otherwise \end{cases} \tag{5}$$

$C^{(d)}$ denotes a triple of semantic data, where superscript $^{(d)}$ means direct. Looking back to Figure 5, there exist six direct relationships between the four resources. Therefore, using 5, we have the following $C^{(d)}$ values:

$$C^{(d)}_{l1:r1:r2} = C^{(d)}_{l1:r2:r1} = C^{(d)}_{l2:r1:r2} = 1$$

Similarly, we can aggregate similarities over many properties as shown in 6,

$$C^{(d)}_{n,r_y,r_z} = \begin{cases} \sum\limits_{l_x} C_{l_x,r_y,r_z} & if\ exists\ (l_x : r_y : r_z) \\ 0 & otherwise \end{cases} \tag{6}$$

For example, $C^{(d)}_{n:r1:r2} = 2$.

Also, we can aggregate similarities over many target resources as in 7

$$C^{(d)}_{l_x,r_y,n} = \begin{cases} \sum\limits_{r_z} C_{l_x,r_y,r_z} & if\ exists\ (l_x : r_y : r_z) \\ 0 & otherwise \end{cases} \tag{7}$$

Thus for example, $C^{(d)}_{l2:r1:n} = 2$ The first similarity function is thus obtained as follows:

$$LDSD^{(d)}(r_y, r_z) = \frac{1}{1 + C^{(d)}(n, r_y, r_z) + C^{(d)}(n, r_z, r_y)}, \tag{8}$$

where n represents the total number of links between resources r_y and r_z. A weighted version of this function is introduced using the weighting methodology in [39], where the value of $C^{(d)}$ is normalized by the *log* of the total number of n resources directly linked to r_y or r_z by l_x:

$$LDSD^{(wd)}(r_y, r_z) = 1/1 + \sum_x \frac{C^{(d)}(l_x, r_y, r_z)}{1 + log(C^{(d)}(l_x, r_y, n))} + \sum_x \frac{C^{(d)}(l_x, r_z, r_y)}{1 + log(C^{(d)}(l_x, r_z, n))} \tag{9}$$

b: INDIRECT IN AND OUT SIMILARITY

Another LDSD algorithm is designed to handle the indirect, in and out, RDF triples. Looking at the following formulas:

$$C^{(ii)}_{l_x,r_y,r_z} = \begin{cases} 1 & if\ exists\ \mathbf{n}\ in\ (l_x : n : r_y)\ and\ (l_x : n : r_z) \\ 0 & otherwise, \end{cases} \tag{10}$$

and

$$C^{(io)}_{l_x,r_y,r_z} = \begin{cases} 1 & if\ exists\ \mathbf{n}\ in\ (l_x : r_y : n)\ and\ (l_x : r_z : n) \\ 0 & otherwise; \end{cases} \tag{11}$$

we can compute the following indirect in and out similarities values respectively:

$$C^{(ii)}_{l_2,r_2,r_3} = 1 \tag{12}$$

$$C^{(io)}_{l_3,r_1,r_2} = 1 \tag{13}$$

The idea basically is that if there exists a resource that is in both triples with the same property, then the value is 1, otherwise 0. Using this assumption, we can infer the following relationships from Figure 5: (note that Superscript $^{(i)}$ indicates indirect, $^{(ii)}$ indicates indirect in, and $^{(io)}$ indicates indirect out).

For example, $C^{(io)}_{l_3,r_1,r_2} = 1$ means that both r_1 and r_2 are indirectly linked by outgoing link l_3 from both resources to r_4. For indirect incoming relationship, an example from Figure 5 is $C^{(ii)}_{l_2,r_2,r_3} = 1$, where we can see that the link l_2 is ingoing into both resources r_2 and r_3 from one resource r_1. Finally the equation for the LDSD similarity is given by combining both ingoing and outgoing similarities:

$$LDSD^{(i)}(r_y, r_z) = \frac{1}{1 + C^{(io)}(n, r_y, r_z) + C^{(ii)}(n, r_y, r_z)} \tag{14}$$

n indicates the total number of indirect in-going or out-going links between resources r_y and r_z. The weighted version is given by:

$$LDSD^{(wi)}(r_y, r_z) = 1/1 + \sum_x \frac{C^{(ii)}(l_x, r_y, r_z)}{1 + log(C^{(ii)}(l_x, r_y, n))} + \sum_x \frac{C^{(io)}(l_x, r_y, r_z)}{1 + log(C^{(io)}(l_x, n, r_z))}. \tag{15}$$

The values of $C^{(ii)}$ and $C^{(io)}$ are normalized by the *log* of the total number of n resources indirectly linked (in-going or out-going) to r_y by l_x.

c: COMBINED SIMILARITY

Lastly, a final combined and weighted version of LDSD is formulated as follows:

$$LDSD^{(wc)}(r_y, r_z) = 1/1 + \sum_x \frac{C^{(d)}(l_x, r_y, r_z)}{1+log(C^{(d)}(l_x, r_y, n))} + \sum_x \frac{C^{(d)}(l_x, r_z, r_y)}{1+log(C^{(d)}(l_x, r_z, n))} + \sum_x \frac{C^{(ii)}(l_x, r_y, r_z)}{1+log(C^{(ii)}(l_x, r_y, n))} + \sum_x \frac{C^{(io)}(l_x, r_y, r_z)}{1+log(C^{(io)}(l_x, n, r_z))}. \quad (16)$$

It combines both weighted, direct and indirect, LDSD equations mentioned earlier.

To sum up, the similarity measures allow us to construct an item by item semantic knowledge graph using semantic data. We can then use this graph to add an explanation regularization term in the rating reconstruction loss function. Because we worked on the movie and book item domains, we focused on the indirect, in-going and out-going, relationships. The reason is that there are almost no direct links between items. However, actors, as an example in the movie domain, can indirectly, both in and out, link different items to each other. 15 allows us to construct a semantic KG that captures direct and indirect semantic relationships between items.

B. LINKED DATA SEMANTIC DISTANCE MATRIX FACTORIZATION (LDSDMF)

The proposed loss function is designed by combining and extending rating reconstruction terms from both [1] for pure rating-based reconstruction and [5] for taking into account item similarity, which in our case will be built using the built semantic knowledge graphs. This loss function is defined as follows:

$$J = \sum_{u,i \in R} (R_{u,i} - p_u q_i^T)^2 + \frac{\gamma}{2} \sum_{i,j \in S^{ldsd}} (S_{i,j}^{ldsd} - q_i q_j^T)^2 + \frac{\beta}{2}(\| p_u \|^2 + \| q_i \|^2). \quad (17)$$

$R_{u,i}$ represents the rating for item i by user u. p_u and q_i represent the low dimensional latent factor vectors of users and items, respectively. S^{ldsd} is the semantic KG constructed using 15. q_i and q_j indicate two items in the KG, S^{ldsd}, and γ is a coefficient that weighs the contribution of the new term, S^{ldsd}. Stochastic gradient descent [4] is employed to update p and q iteratively until J converges.

The complete flowchart of the model is depicted in Figure 6.

The gradient of J with respect to p_u is given by:

$$\frac{\partial J}{\partial p_u} = -2\,(R_{u,i} - p_u q_i^T) q_i + \beta p_u. \quad (18)$$

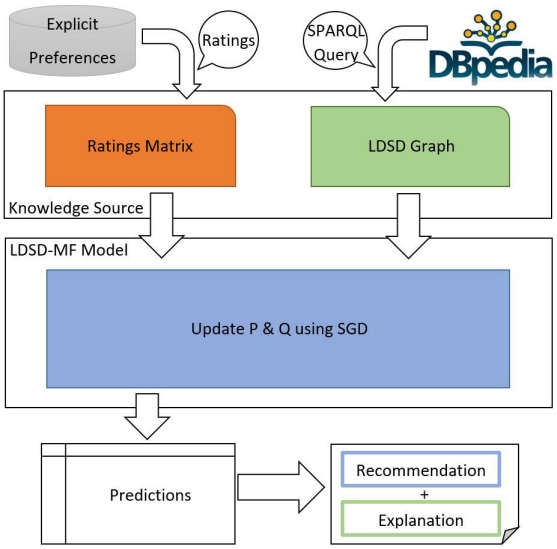


FIGURE 6. LDSDMF Flowchart.

The gradient of J with respect to q_i is given by

$$\frac{\partial J}{\partial q_i} = -2(R_{u,i} - p_u q_i^T) p_u + 2\gamma (S_{i,j} - q_i q_j^T) q_j + \beta q_i. \quad (19)$$

The gradient descent updating rules are thus given by:

$$p_u^{(t+1)} \leftarrow p_u^{(t)} + \alpha(2(R_{u,i} - p_u^{(t)}(q_i^{(t)})^T) q_i^{(t)} - \beta p_u^{(t)}) \quad (20)$$

$$q_i^{(t+1)} \leftarrow q_i^{(t)} + \alpha(2(R_{u,i} - p_u^{(t)}(q_i^{(t)})^T) p_u^{(t)} + 2\gamma(S_{i,j}^{ldsd} - q_i^{(t)}(q_j^{(t)})^T) q_j^{(t)} - \beta q_i^{(t)}). \quad (21)$$

The semantic explanation KGs are constructed using the approach described in section III-A for all semantic attributes, and hence explanations. In addition to the known ratings used to update q_i, the semantic explanation KGs also contribute to the final predicted rating of item i by user u.

C. INFERRED FACT STYLE EXPLANATION

We propose a new explanation style that utilizes the previously constructed KGs on users and semantic attributes. In this style, the uncertainty degree of the users' preferences for semantic attributes is employed to justify a recommendation. Inference rules are used to derive new knowledge from known facts [40] and thus augment the semantic knowledge graphs with new information or facts. For example, if A is of type B and B is of type C, then A must be of type C. The previous example is a situation with complete certainty; however, some cases do not enjoy full certainty in inference, and for those cases, we obtained the uncertainty degree from the constructed user by semantic attribute matrix based on the work of [38]. For example, if a user, u, watched, interacted with, or rated a certain item, i, and this item is linked to a certain semantic attribute, a, a new inferred fact can be derived, namely "user u likes semantic attribute a to a certain degree". The likability degree depends on the number of times the user interacts with items that are linked to that specific semantic attribute. Figure 7 illustrates an example

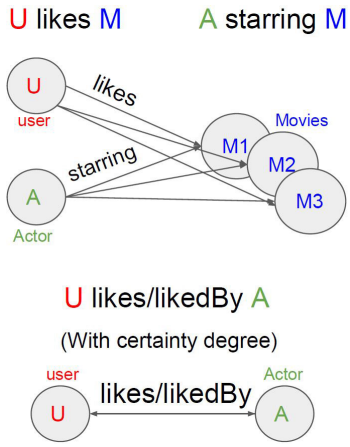


FIGURE 7. Inferred Facts: Movie Example.

of an inferred fact in the movie domain. In Section IV-A.5, we will show an example of this explanation style when we describe the real user study that we conducted.

IV. EXPERIMENTAL EVALUATION

We perform two types of experiments. The first set of experiments (subsection IV-A) are offline, meaning that they use a benchmark data set to validate the recommendation accuracy and explanations using offline objective metrics on held-out test data. The second set of experiments (subsection IV-B) are based on a real user study that will evaluate the semantic explanations.

A. OFFLINE EXPERIMENTS

1) EXPERIMENTAL SETTING

Our experiments are based on the MovieLens 100K benchmark dataset. The total number of users is 943, and that of movies is 1,862. SPARQL, a semantic web query language, is used for the mapping process between MovieLens and the DBpedia KG, and movie titles are used for the mapping. The results indicate that 1,012 movies intersected in the two datasets. The reasons for this reduction are either absent movies in DBpedia or different spellings. The mapping also resulted in a decrease in the total number of ratings to 60K. All ratings are normalized to 1, and the hyper-parameters are set to $\alpha = 0.01$, $\beta = 0.1$, and $\gamma = 0.9$, after being tuned using cross-validation. 90% of the ratings are used for training the model, and 10% are used for testing the model. Since our method randomly initializes the user and item latent spaces, an average of 10 experiments is reported.

Five different properties are extracted from the semantic KG DBpedia: subject, actor, director, producer, and writer. The total number of unique subjects is shown in the second column of Table 1. The third column in Table 1 shows the total number of previously existing triples of movies and semantic attributes in DBpedia. An example could be "Mel Gibson is starring in Braveheart." The fourth column in Table 1 describes the size of the constructed semantic KG with the total number of triples in each KG. For example, "User 581 likes the actor Ben Kingsley to a certain degree."

TABLE 1. Numeric values of selected semantic attributes in the experiment, with unique IDs in the second column, the total number of triples for movies in the third column, and the total number of triples for users in the fourth column.

semantic Attribute	Unique ID	Triple (movies)	Triple (users)
Subject	4996	19983	818784
Actor	4165	6770	332484
Director	1193	1577	92008
Producer	1154	1868	103943
Writer	1491	1944	110692

Five baseline methods are used for comparison: MF [1], Probabilistic Matrix Factorization (PMF) [41], Asymmetric Matrix Factorization (AMF) [7], EMF [13]–[15], and Asymmetric Semantic Explainable Matrix Factorization (ASEMF_UIB) [38].

Our hypothesis for the significance test is that our model is better than baseline approaches using all metrics. The null hypothesis that we are trying to reject is that the mean of all metrics for all models are equal by conducting a t-test experiments. The models are ran 10 times while randomly initializing the user and item latent factors, then we calculated all metrics and did the significance tests which are reported in this paper.

2) RECOMMENDER SYSTEM EVALUATION

Two kinds of metrics are used to evaluate the recommender system. The first one calculates the error rate, see (22), while the second one calculates the mean absolute precision at cutoff n of the top N recommended items, see (23).

a: ROOT MEAN SQUARE ERROR (RMSE)

$$RMSE = \sqrt{\frac{1}{|T|}\sum_{(u,i)\in T}\left(r'_{ui} - r_{ui}\right)^2}. \tag{22}$$

T is set of user-item pairs for which ratings are available, r'_{ui} represents the predicted rating for item i by user u, and r_{ui} is the actual rating on item i by user u.

b: MEAN ABSOLUTE PRECISION AT CUTOFF n (MAP@N)

$$MAP@N = \frac{1}{|U|}\sum_{u=1}\frac{1}{|I|}\sum_{k=1}^{N} P_u(k).rel_u(k) \tag{23}$$

MAP@N measures the relevance of the recommended items in each position in the list. Those in a higher position weigh more than those at the end of the list. U and I denote the total number of users and items, respectively. N represents the length of the recommendation list, and k is the current position of a recommended item at the calculation of the precision value. P is the precision value, and $rel(k)$ indicates whether an item is relevant or not.

3) EXPLAINABILITY EVALUATION

Three metrics are used to measure the explainability of the recommended items [13]–[15]. Let U represent the total

TABLE 2. RMSE, varying the number of features, K.

RMSE						
K	MF	PMF	AMF	EMF	$ASEMF_{UIB}$	LDSDMF
10	0.205	0.698	0.236	0.205	0.205	**0.204**
20	0.212	0.698	0.27	0.211	**0.204**	**0.204**
30	0.214	0.698	0.309	0.215	**0.204**	**0.204**
40	0.216	0.7	0.344	0.217	**0.203**	0.205
50	0.217	0.7	0.374	0.217	**0.203**	0.206

TABLE 3. RMSE significance test results in the movie domain (K = 10).

Model 1	Model 2	p-value
MF	**LDSDMF**	2.3e-07
PMF	**LDSDMF**	4.04e-54
AMF	**LDSDMF**	6.6e-22
EMF	**LDSDMF**	4.8e-08
ASEMF_UIB	**LDSDMF**	1.3e-07

number of users, $\mathcal{R}$ the set of recommended items, and W the set of explainable items.

a: MEAN EXPLAINABILITY PRECISION (MEP)

The first metric, MEP, computes the ratio of the number of simultaneously recommended and explainable items to the total number of recommended items over all users.

$$MEP = \frac{1}{|U|} \sum_{u \in U} \frac{|\mathcal{R} \cap W|}{|\mathcal{R}|} \tag{24}$$

b: MEAN EXPLAINABILITY RECALL (MER)

The second metric, MER, calculates the ratio of the number of simultaneously recommended and explainable items to the total number of explainable items, again, over all users.

$$MER = \frac{1}{|U|} \sum_{u \in U} \frac{|\mathcal{R} \cap W|}{|W|} \tag{25}$$

c: EXPLAINABILITY F-SCORE (xF-SCORE)

The xF-score combines MEP and MER using the harmonic mean.

$$xF - score = 2 * \frac{MEP * MER}{MEP + MER} \tag{26}$$

4) DISCUSSION

Table 2 shows the error rates of all the methods. The best values are in bold (the lower the value, the better). When $K = 10$, LDSDMF significantly outperforms all the other methods with a small p-value as shown in Table 3; however, it competes with $ASEMF_{UIB}$ as the number of hidden features increases.

Figure 8 illustrates how each model performs when considering the MAP@N against the number of features. LDSDMF significantly outperforms all baseline approaches when $K = 10$ as shown in Table 4. As K increases, PMF and AMF were the winners.

In Figures 9 and 10, there are six graphs showing the performance of all models while varying θ^s and θ^n. θ^s is a threshold for items to be considered semantically explainable or not,

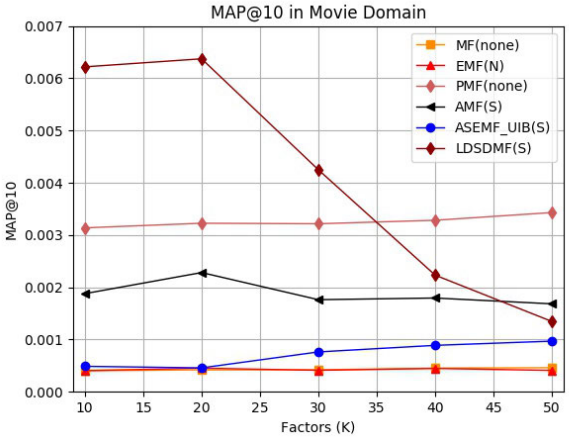


FIGURE 8. MAP@10 Performance While Varying the Number of Features, K.

TABLE 4. MAP@10 significance test results in the movie domain (K = 10).

Model 1	Model 2	p-value
MF	**LDSDMF**	1.6e-15
PMF	**LDSDMF**	7.3e-09
AMF	**LDSDMF**	6.5e-11
EMF	**LDSDMF**	1.6e-15
ASEMF_UIB	**LDSDMF**	1.3e-12

and θ^n is a threshold for items to be explainable based on the neighborhood technique used in the baseline EMF [13]–[15]. The formula for generating the neighborhood-based explainability matrix is

$$W_{ui} = \begin{cases} \frac{|N'(u)|}{|N_k(u)|} & if \ \frac{|N'(u)|}{|N_k(u)|} > \theta^n \\ 0 & otherwise, \end{cases} \tag{27}$$

where $N'(u)$ denotes the set of neighbors of user u who rated item i, and $N_k(u)$ depicts the list of the k nearest neighbors of u.

The three graphs in Figure 9 illustrate that when θ^s is set to 0, which means that all items (even those with a small explainability value) are considered explainable, the baseline PMF is the winner. However, when adding more restrictions to items to be considered semantically explainable, the proposed method, LDSDMF, significantly outperformed the other methods for all θ^s values by all metrics (MEP, MER, and $xF - score$). Tables 5, 6, and 7 present the significance test results.

Figure 10 presents the models' performance when measuring the explainability of the recommended items based on the neighborhood technique. Our model, LDSDMF, significantly exceeded all baseline methods in all three metrics (see Tables 8, 9, and 10 for significance test results). This observation shows that our proposed method recommends more accurate explainable items, based on semantic KGs and neighborhood based techniques, than all the baseline methods.

From the information presented in all figures above we can reach a conclusion that a large K would not make the system

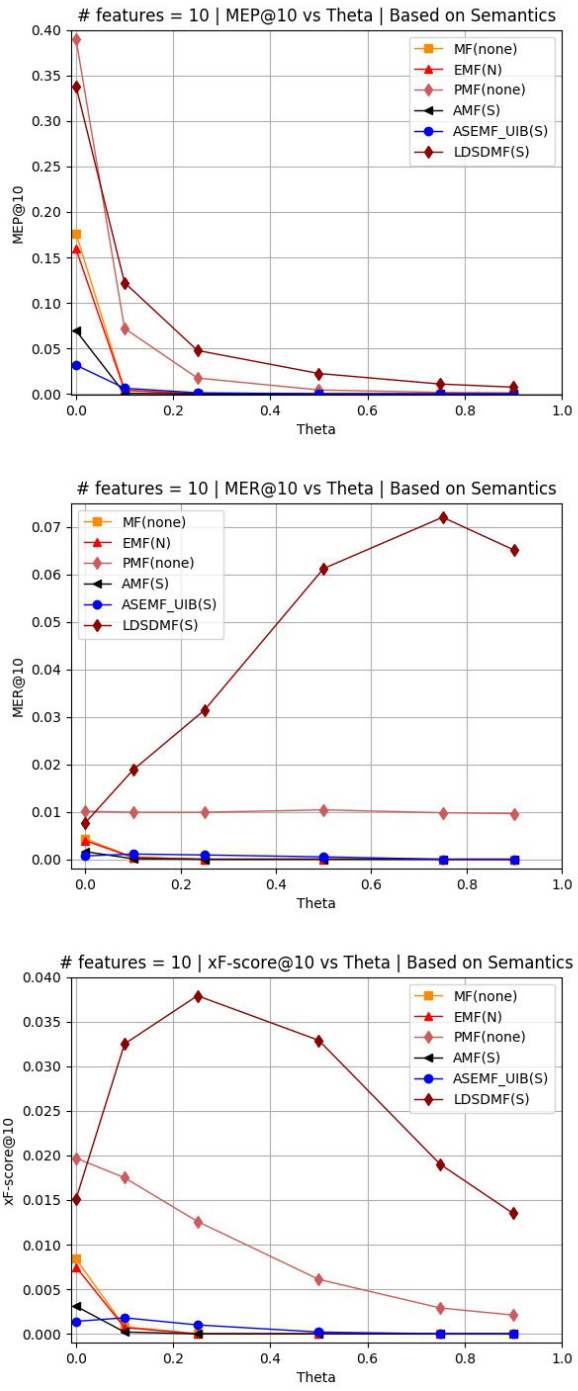


FIGURE 9. The upper graph shows the results of MEP@10 for all methods, while the middle one shows MER@10 for all methods, and the lower graph illustrates the results of all methods using the xF-score metric, which utilizes semantic KGs against K.

perform better; As Shi *et al.* [5] has stated, increasing the number of features will not guarantee better results, but it will increase the complexity of the system and exhaust machine resources.

5) CASE STUDY

We selected a sample user from our real data set as an example to show how the model captures the user's preferences and recommends new items accordingly with an explanation. User 586 in the MovieLens dataset rated 94 movies, including *Twister (1996)* and *Tombstone (1993)* with 4-star ratings and *Apollo 13 (1995)* with a 3-star rating. All three movies are starred by Bill Paxton. *Titanic (1997)* includes the same actor

TABLE 5. MEP@10 significance test results ($K = 10$ and $\theta^S = 0.25$) using semantic KGs.

Model 1	Model 2	p-value
MF	**LDSDMF**	8.06e-23
PMF	**LDSDMF**	3.05e-17
AMF	**LDSDMF**	8.06e-23
EMF	**LDSDMF**	8.1e-23
ASEMF_UIB	**LDSDMF**	2.6e-20

TABLE 6. MER@10 significance test results ($K = 10$ and $\theta^S = 0.25$) using semantic KGs.

Model 1	Model 2	p-value
MF	**LDSDMF**	6.2e-21
PMF	**LDSDMF**	2.1e-15
AMF	**LDSDMF**	6.2e-21
EMF	**LDSDMF**	6.3e-21
ASEMF_UIB	**LDSDMF**	1.3e-19

TABLE 7. xF-score@10 significance test results ($K = 10$ and $\theta^S = 0.25$) using semantic KGs.

Model 1	Model 2	p-value
MF	**LDSDMF**	1.1e-21
PMF	**LDSDMF**	5.1e-16
AMF	**LDSDMF**	1.1e-21
EMF	**LDSDMF**	1.1e-21
ASEMF_UIB	**LDSDMF**	5.6e-20

TABLE 8. MEP@10 significance test results ($K = 10$ and $\theta^n = 0.25$) using neighborhood technique.

Model 1	Model 2	p-value
MF	**LDSDMF**	1.9e-21
PMF	**LDSDMF**	3.9e-17
AMF	**LDSDMF**	1.2e-13
EMF	**LDSDMF**	1.9e-21
ASEMF_UIB	**LDSDMF**	9.9e-19

TABLE 9. MER@10 significance test results ($K = 10$ and $\theta^n = 0.25$) using neighborhood technique.

Model 1	Model 2	p-value
MF	**LDSDMF**	1.2e-21
PMF	**LDSDMF**	1.4e-15
AMF	**LDSDMF**	5.3e-15
EMF	**LDSDMF**	1.2e-21
ASEMF_UIB	**LDSDMF**	5.9e-19

TABLE 10. xF-score@10 significance test results ($K = 10$ and $\theta^n = 0.25$) using neighborhood technique.

Model 1	Model 2	p-value
MF	**LDSDMF**	1.1e-21
PMF	**LDSDMF**	9.2e-16
AMF	**LDSDMF**	6.4e-15
EMF	**LDSDMF**	1.1e-21
ASEMF_UIB	**LDSDMF**	5.9e-19

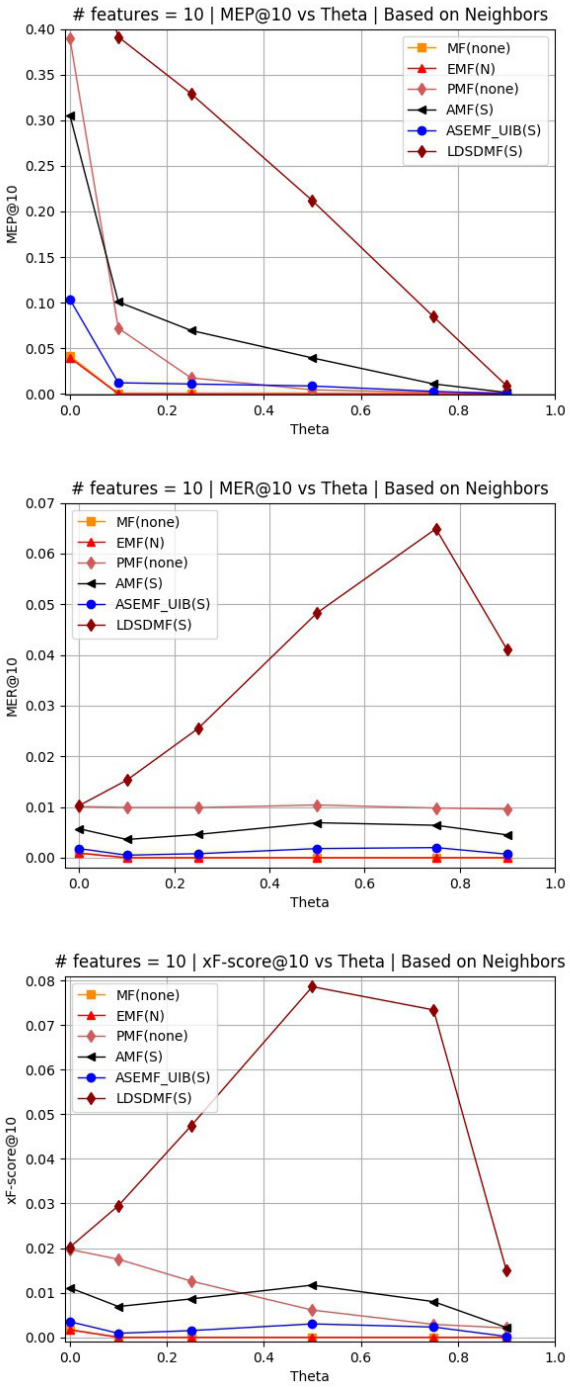


FIGURE 10. The upper graph shows the results of MEP@10 for all methods, while the middle one shows the MER@10 results for all methods, and the lower graph illustrates the results of all methods using the neighborhood explainability graph against K.

in the starring actors list, and the model recommended this movie among the top 10 recommended items. Using the semantic KGs on users and semantic attributes that were built by the model, our model succeeds in capturing the user's semantic attribute preferences and recommends new items accordingly. Figure 11 depicts a projected example of what an explanation would look like for user 586.

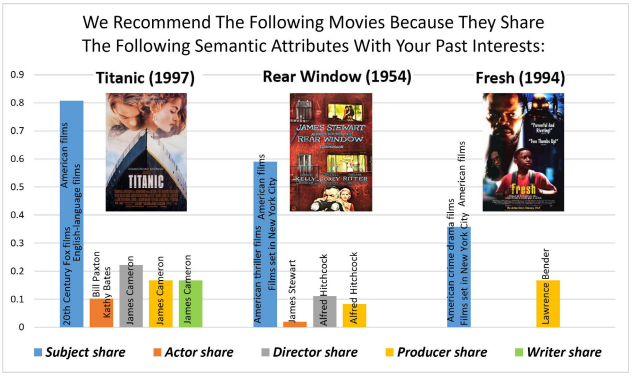


FIGURE 11. Example of Inferred Fact Style Explanation.

6) SUMMARY OF OFFLINE EXPERIMENTAL RESULTS

The experimental results showed that adding semantics resulted in improved recommendation and explainability. Although one would expect explainability to decrease recommendation accuracy, it is important to note that our proposed methods utilize more semantic data than mere user ratings. This additional data compensates for lack of rating and sparseness of data, thus improving accuracy.

B. REAL USER STUDY

In this section, we validate the explainability of the model, proposed in Sections III-C and III-A, by conducting a user study experiment in the movie domain. Our research questions are as follows:

- RQ1: Does the number of semantic attributes used in the explanation, whether it is low or high, impact user satisfaction? Satisfaction is defined as the ease of system usability and the enjoyment of use [42].
- RQ2: Does an explanation that uses a higher number of semantic attributes increase perceived transparency? Transparency is providing information to the user so he or she can comprehend how the system works and the justification behind the recommendation [42].
- RQ3: Does the number of semantic attributes used in the explanation (Low (1 semantic attribute), Medium (3 semantic attributes), or High (5 semantic attributes)) impact the perceived effectiveness? Effectiveness is defined as the ability of the explanation to help users make good decisions [42].

1) HYPOTHESIS

Suppose that a recommender system recommends two items i_1 and i_2 alongside their explanations. Given the explanation definition in Sections III-A and 7, if i_1 uses more semantic attributes in the explanations than i_2 (Figure 12), does recommending i_1 result in a better satisfaction than recommending i_2 from the user perspective? Our hypothesis can be summarized as follows: Recommending an item with an explanation that shows more semantic attributes will lead to higher user satisfaction.

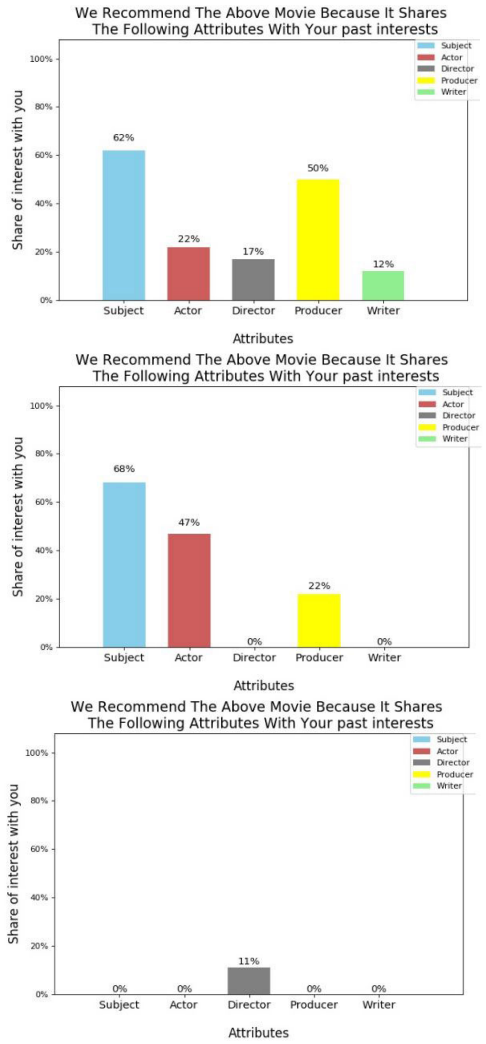


FIGURE 12. A comparison of three explanations for the three groups, group High explanation on the upper side, group Medium explanation on the meddle, and group Low explanation on the lower side. The explanation according to which group the user was randomly assigned to will be exposed to the user alongside the recommendation during the experiment. The explanation on the upper side shows more semantic attributes than the other two explanations.

2) METHODS

A web app platform, similar to commercial movie recommender engines used by Netflix, Amazon Video, and Hulu, was designed to conduct the study. The application used the MovieLens benchmark data set.[4]

The explanations are divided into three groups based on the number of semantic attributes randomly chosen to explain the recommended movie as follows:

- Low: Up to one semantic attribute used for explanation.
- Medium: Up to three semantic attributes used for explanation.
- High: Up to five semantic attributes used for explanation.

[4]https://grouplens.org/datasets/movielens/

3) SUBJECT RECRUITMENT

The Institutional Review Board at University of Louisville reviewed and authorized our study. Participants were students in a large urban, southern university and were recruited to participate in the study via personal and email invitations. A Surface Pro laptop and a desktop were provided to the participants to use for this experiment. Google forms was used to construct and host the questions and the results were securely stored on Google drive.

4) SAMPLE SIZE ESTIMATION

To estimate the sample size, we performed a statistical power analysis. The effect size in this study is large using Cohen's [43] criteria. When α is set to 0.05, and power is set to 0.8, the sample size needed is approximately 10.

The 34 participants were randomly assigned to either the *low*, *medium*, or *high* group representing the number of semantic attributes used in explanation. The number of people in each group are as follows:

- Low = 11
- Medium = 12
- High = 11

5) PROCEDURES

The experiment proceeded as follows:

1) The participant was asked to rate, on a 1 to 5 scale, at least 10 movies they have previously watched from a selection of movies.
2) A recommendation alongside an explanation, based on the participant's assigned group, was provided to the user. The recommendation and explanation were selected from a pool of recommendations that were calculated using the method proposed in Sections 7 and III-A, such that the correct number of semantic attributes used in the explanation were displayed to the user depending on the experimental group to which the participant was assigned (i.e. "Low (1)", "Medium (3)", or "High (5)").
3) The participant was asked to fill out a Likert Scale questionnaire. Table 11 shows the questions used in this study.
4) Demographic information was collected from the participant including age, gender, major of study, weekly hours watching movies, and favorite movie semantic attributes. Table 12 presents the questions used in this experiment. This information was requested to study potential confounding factors on the participants' satisfaction with the explanations.

A snapshot of the application is shown in Figures 13 and 14. The duration of the experiment was around 30 minutes.

6) ANALYSIS OF USER STUDY RESULTS

In this study, participants were asked to answer five questions regarding their experience after using the model.

TABLE 11. Likert scale survey questions.

Question 1	"Based on the share of semantic attributes between the recommended movie and your interest in these semantic attributes, this is a good recommendation."
Question 2	"This explanation helps me understand why this movie was recommended."
Question 3	"Based on the share of semantic attributes between the recommended movie and my interest in these semantic attributes, I will watch this movie."
Question 4	"Based on the share of semantic attributes between the recommended movie and my interest in these semantic attributes, I can determine how well I will like this movie."
Question 5	"This explanation helps me understand how the recommender system works."

TABLE 12. Demographic questions.

Question 1	"What is your gender"
Question 2	"What is your age?"
Question 3	"What is your major of study?"
Question 4	"How many hours per week do you watch movies on average?"
Question 5	"What are the most influential semantic attributes that encourage you to watch a movie?"
Question 6	"How familiar are you with automated recommender systems?"
Question 7	"Check all the online entertainment services that you have used in the past."

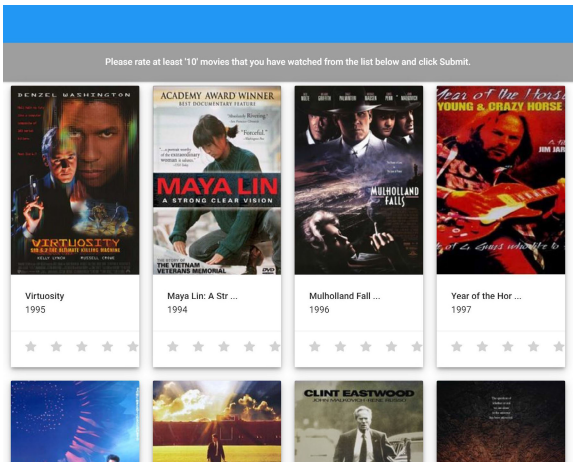


FIGURE 13. A snapshot of the recommender system app showing a list of movies for the user to rate.

Figure 15 shows a vertical bar chart for all participants' answers to all of Table 11's five questions. The most repeated answer was "Somewhat Agree" across all questions, followed by the "Neutral" answer option, then by "Strongly Agree" respectively. The answers "Somewhat Disagree" and "Strongly Disagree" were the least chosen answers by all participants for all questions. Figure 16 depicts the answers for participants in the group High. People were assigned randomly to each group. The answers "Strongly Agree" and "Somewhat Agree" were the most popular answers to all questions. Only four participants were neither agreeing nor disagreeing to question one, and only one participant

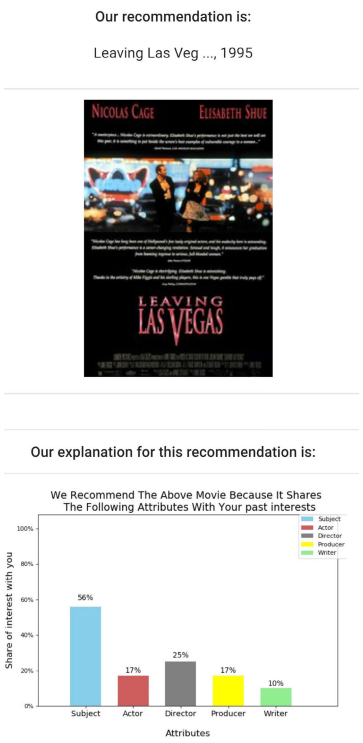


FIGURE 14. A snapshot of a recommendation and its explanation presented to a user.

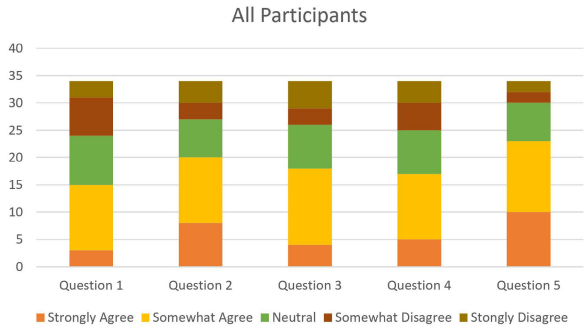


FIGURE 15. A Vertical bar chart of the answers to the questions in Table 11 for all participants.

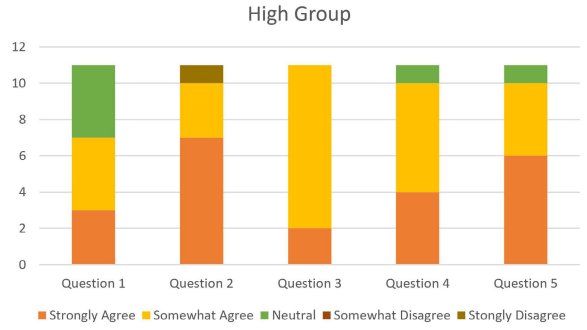


FIGURE 16. A Vertical bar chart of the answers to the questions in Table 11 for participants in the group "High".

disagreed in questions four and five. It is worth noting that more than half of the participants strongly agreed to question two, which is about how the explanation helped them

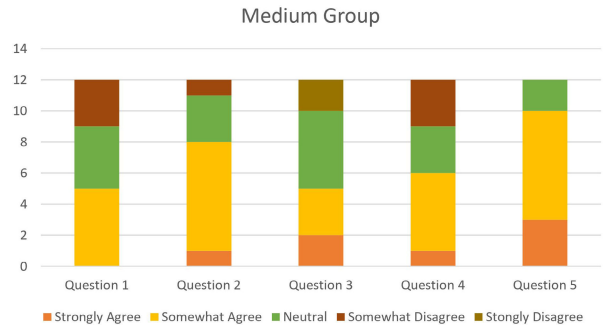


FIGURE 17. A Vertical bar chart of the answers to the questions in Table 11 for participants in the group "Medium".

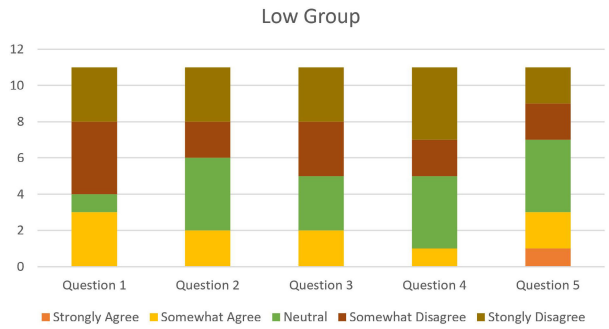


FIGURE 18. A Vertical bar chart of the answers to the questions in Table 11 for participants in the group "Low".

understand the recommendation. Figure 17 shows the answers of participants in the group Medium. "Somewhat Agree" and "Neutral" responses were the most chosen responses by people in this group. Followed by "Strongly Agree" and "Somewhat Disagree". Two participants preferred "Strongly Disagree" as their answer to question three. In the group Low, as shown in Figure 18, more than half of the participants chose "Strongly Disagree" and "Somewhat Disagree" as their answers to all questions. The next response in line was the Natural response choice, followed by "Somewhat Agree", and only one participant gave a "Strongly Agree" answer to question five in this group.

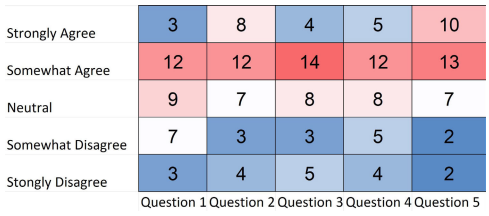


FIGURE 19. A Heat-map plot of the answers to the questions in Table 11 for all participants.

Figure 19 depicts a Heat map plot showing the distribution of all answers to all questions by all participants. The most popular answer is "Somewhat Agree" followed by "Neutral" as the second most popular. "Strongly agree" is next in line, then "Somewhat Disagree" and " Strongly Disagree" answers were the least preferred answers by participants in the Medium group.

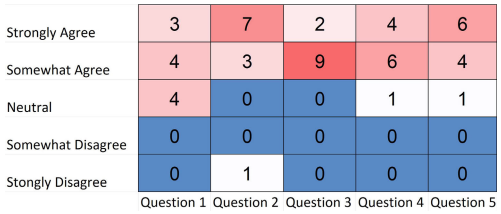


FIGURE 20. A Heat-map plot of the answers to the questions in Table 11 for participants in the group "High ".

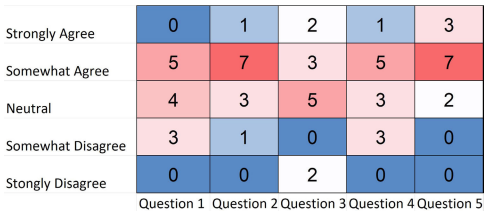


FIGURE 21. A Heat-map plot of the answers to the questions in Table 11 for participants in the group "Medium".

Figure 20 shows the responses from participants in the group High. The figure shows a clear tendency to the Agree than to the Disagree answers. In contrast, responses from participants in the group Low, as illustrated in Figure 22, tend to the Disagree side more than the Agree side. Lastly, the heat map in Figure 21 is scattered over all responses to all question from participants in the group Medium.

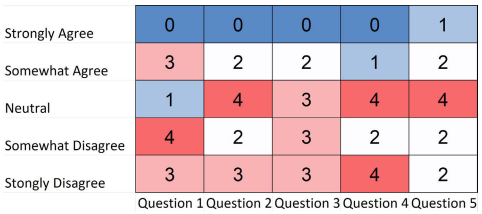


FIGURE 22. A Heat-map plot of the answers to the questions in Table 11 for participants in the group "Low".

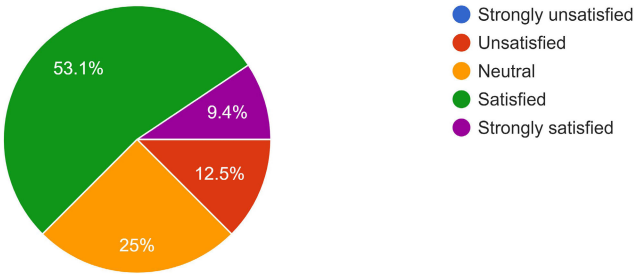


FIGURE 23. Distribution of the participants' satisfaction with the explanation.

Figure 23 indicates the satisfaction level with the explanation for all participants in this study. More than half of them were satisfied, whereas around 10% were strongly satisfied. 25% of the participants were neither satisfied nor unsatisfied, and 12.5% were not satisfied. No participant responded with the strongly unsatisfied answer option.

Figures 24, 25, 26, 27, 28, 29, and 30, show the responses of all participants to the demographic questions in Table 12. The answers for these questions were optional.

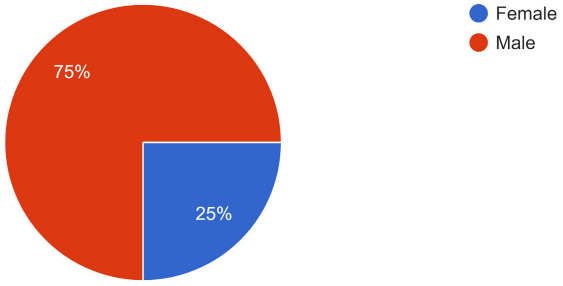


FIGURE 24. Distribution of the participants' gender.

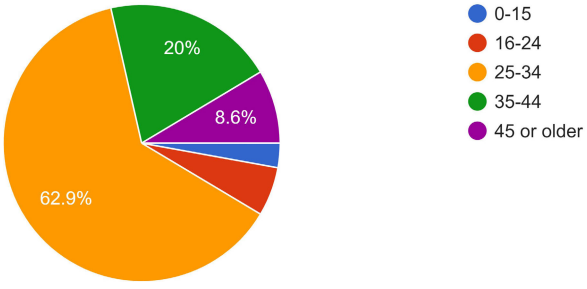


FIGURE 25. Distribution of the participants' age.

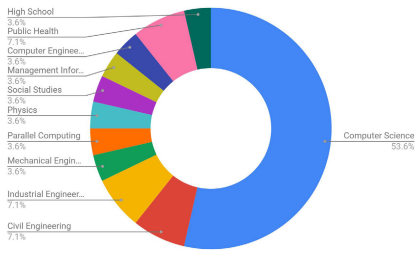


FIGURE 26. Distribution of the participants' major of study.

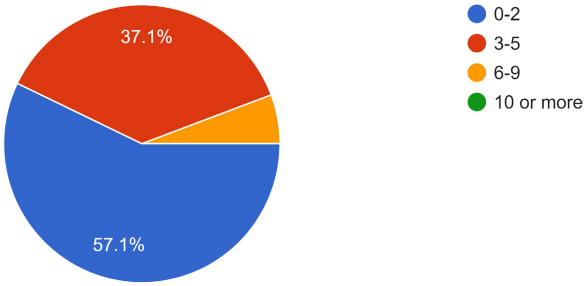


FIGURE 27. Distribution of the participants' weekly hours watching movies.

Three-quarters of the participants were males and the rest were females as shown in Figure 24. Figure 25 represents the age distribution of all participants. More than 60% are between the age of 25 and 34 years, followed by 20% participants aged between 35 and 44 years. The rest of the participants' ages are distributed in the other groups.

The most common major of study for participants was Computer science followed by other majors as shown in Figure 26. Most of the participants watch movies for around 0 to 5 hours a week as reported in Figure 27. Figure 29 shows half of the volunteers were either moderately or somewhat familiar with the automated recommender systems, whereas 32.4% are slightly familiar. 14.7% were

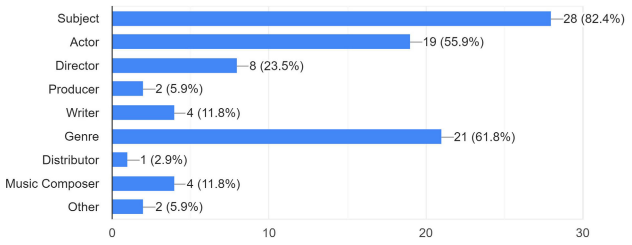


FIGURE 28. Distribution of the participants' favorite movies' semantic attributes.

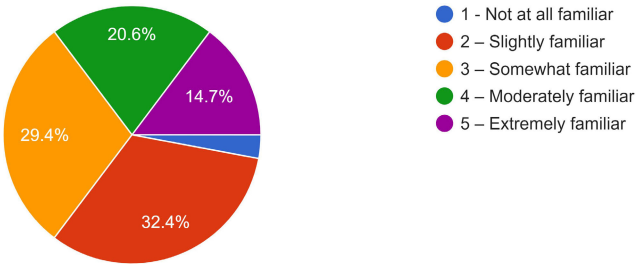


FIGURE 29. Distribution of the participants' familiarity with recommender systems.

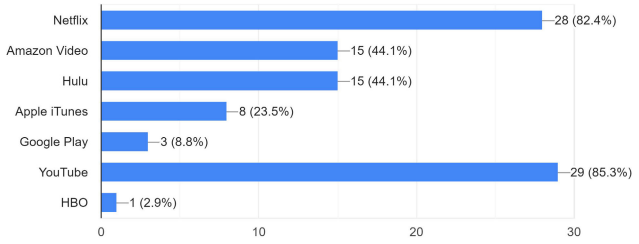


FIGURE 30. Distribution of the participants' most used online entertainment services.

extremely familiar and a small portion of the participants were not familiar at all.

Figure 28 denotes the distribution of the participants regarding the most influential semantic attributes that encourage them to watch a movie. It indicates that subject, genre and actor were the most influential ones while the producer and the music-composer were the least influential. Figure 30 represents the online entertainment services that the volunteers have used in the past. YouTube and Netflix are the most popular services followed by Amazon Video and Hulu. Google Play and HBO were the least used services by participants.

7) HYPOTHESIS TESTING

In the previous subsection IV-B.6, we showed how the responses of the participants varied according to the designated groups (High, Medium, and Low) where participants were assigned randomly. The plots suggest that people in the group "High" tend to give more positive responses than others in the other groups.

In this section, analytical testing is conducted to determine the significance of the those findings of this study. First, it is essential to evaluate the reliability of the Likert scale questionnaire by calculating Cronbach's Alpha [44]. The correlation of the survey questions and the 34 participants

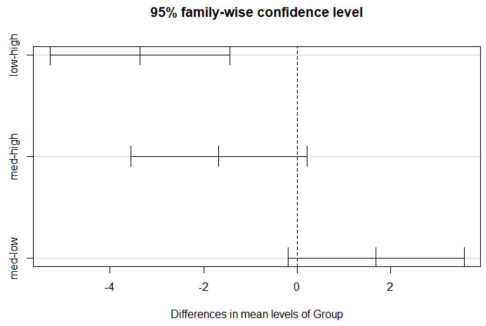


FIGURE 31. Visualization of differences of mean levels of pairs of groups for satisfaction.

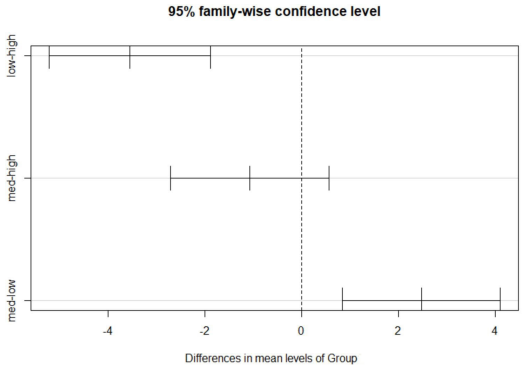


FIGURE 32. Visualization of differences of mean levels of pairs of groups for transparency.

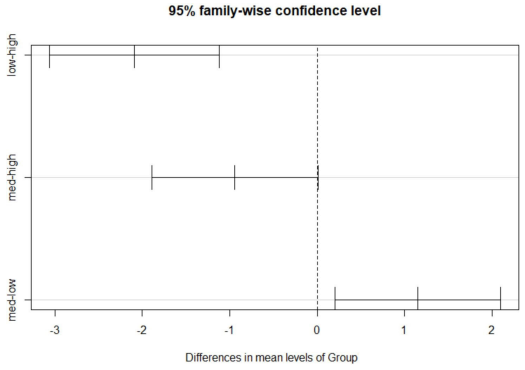


FIGURE 33. Visualization of differences of mean levels of pairs of groups for effectiveness.

TABLE 13. Mean and standard deviation for all groups for regarding all three explanation aspects.

Explanation Aspect	Groups					
	High		Medium		Low	
	Mean	STD	Mean	STD	Mean	STD
Satisfaction	8.09	0.94	6.41	2.02	4.72	2.24
Transparency	8.81	1.32	7.75	1.35	5.27	2
Effectiveness	4.27	0.64	3.33	0.98	2.18	2.1

was 0.86, which is above the threshold of 0.7 for an acceptable level of reliability.

Table 14 presents the relationship between the explanation aspects, satisfaction, transparency, and effectiveness, and the questions in the survey listed in Table 11.

An Analysis of Variance (ANOVA) test was conducted to study the effect of the explainability variable on the designated aspects in Table 11. The null hypothesis of an ANOVA test is that the mean of the three groups, High, Medium, and Low, are equal.

- Satisfaction:
  RQ1 asked whether the number of semantic attributes used in an explanation impact user's satisfaction with a recommender system. A composite measure of user satisfaction with the explanation was created using questions 1 and 3. The results show that there is a significant difference between the three groups indicating that the number of semantic attributes do influence user satisfaction with an explanation, $f\,(2{,}31) = 9.27$, $p < .001$, $\eta^2 = .37$. We conducted a Tukey's HSD (Honestly Significant Difference) post-hoc test to determine which pair of groups were significantly different from each other. The family-wise significance interval was at 95%, and Table 15 summarizes the results. From this table, it is clear that there is a significant difference between group High and group Low with a very small, statistically significant p-value. However, there is no significant difference between group High and group Medium nor between group Medium and group Low. Figure 31 shows a visualization of the differences between the means of the three groups.
- Transparency
  RQ2 asked if the number of semantic attributes used an in explanation predicted the perceived transparency of the recommender system. Questions 2 and 5 were used to create a composite measure of transparency. The results found that there was a significant difference between how the High, Medium, and Low group perceived the transparency of the recommender system, $f\,(2, 31) = 14.49$, $p < .001$, $\eta^2 = .48$. With a family-wise significance interval at 95%, we conducted a Tukey's HSD post-hoc test to determine which pairs of groups significantly differed from each other. Table 16 presents the outcome. As shown in the table, the very small adjusted p-values indicate a significant difference between the groups High and Low, as well as between the groups Medium and Low. However, there is no significant difference between the groups Medium and High. Figure 32 displays a visualization of the mean differences between the groups.
- Effectiveness
  RQ3 asked if the number of semantic attributes used in an explanation influenced the perceived effectiveness of the recommender system. The results found significant differences between the grouprs, $f\,(2{,}31) = 14.12$, $p < .001$, $\eta^2 = .48$, demonstrating the number of semantic attributes did impact perceived effectiveness. A Tukey's HSD post-hoc test was conducted to determine which pairs of groups had significant signed difference. Table 17 indicates that there is a significant signed difference between groups High and Low as well as between groups Medium and Low. The p-values are below the threshold of 0.05 allowing for the rejection of the null hypothesis. Meanwhile, there is not a significant

TABLE 14. Categorization of the survey questions from Table 11 according to the research questions.

Explanation Aspect	Question
Satisfaction	1 and 3
Transparency	2 and 5
Effectiveness	4

TABLE 15. Tukey multiple comparisons of means at 95% family-wise confidence interval for satisfaction.

Group pairs	Difference	Adjusted p-value	reject
High-Low	-3.3636	0.0004	True
High-Medium	-1.6742	0.0888	False
Medium-Low	1.6893	0.0852	False

TABLE 16. Tukey multiple comparisons of means at 95% family-wise confidence interval for transparency.

Group pairs	Difference	Adjusted p-value	reject
High-Low	-3.5454	0.0000	True
High-Medium	-1.0681	0.2555	False
Medium-Low	2.4772	0.0021	True

TABLE 17. Tukey multiple comparisons of means at 95% family-wise confidence interval for effectiveness.

Group pairs	Difference	Adjusted p-value	reject
High-Low	-2.0909	0.0000	True
High-Medium	-0.9393	0.0529	False
Medium-Low	1.1515	0.0147	True

difference between the High and Medium groups. Figure 33 presents the differences in means between the three designated groups.

Table 13 shows the mean and standard deviation for all groups regarding the tested explanation styles, satisfaction, transparency, and effectiveness.

C. SUMMARY OF EXPERIMENTAL EVALUATION

In this section, we presented the results of an offline and online evaluation of the methods proposed in Section III. In offline evaluation, we used objective metrics to measure the recommendation accuracy of the proposed methods as well as the explainability. The overall results indicate that our model succeeded in increasing the transparency of the system while keeping the error rate at a low level.

In the online evaluation, the final results indicate that the participants had a good perception of the explanation capability, especially when including more item properties in the explanation generation process.

V. CONCLUSION

As recommendation systems become an essential component of big data and artificial intelligence (AI) systems, and as these systems embrace more and more sectors of society, it is becoming ever more critical to build trust and transparency into machine learning algorithms without significant loss of prediction power. Our research harnesses the power of AI, such as KGs and semantic inference, to help build explainability into accurate black box predictive systems in a way that is modular and extensible to a variety of prediction tasks within and beyond recommender systems.

In this study, we concentrated on collaborative filtering (CF) techniques, as they excel in handling the big data with which the web is abundant and tend to outperform content-based filtering techniques. More specifically, we focused on matrix factorization, a state-of-the-art CF technique that builds low-dimensional spaces for hidden features to predict unseen items' ratings and efficiently deals with sparse data. Nevertheless, the lack of transparency significantly reduces user satisfaction and trust in the system. The cold start problem is another issue from which CF techniques suffer.

To tackle these issues, we proposed to use semantic knowledge graphs (KG) that correlate the user with the item's semantic attributes based on the number of interactions between them in the user's history. Item properties are retrieved by SPARQL, the SQL-like semantic web query language, from semantic web databases such as DBpedia. The semantic KGs are used in the latent spaces to build the final model and to generate justifications for the recommendations. They also work as a warm-up solution for the cold start problem.

We conducted an offline evaluation to measure the error rate, recommendability, and the explainability of the recommended items. We also evaluated the explainability of all models, using neighborhood based explainability measures, in the movie domain.

An online evaluation was conducted with a user study of 34 individuals. The results clearly show that the proposed explanation style increased the user perception of system transparency, while being more effective in encouraging the user to accept the recommendation, leading to higher user satisfaction.

REFERENCES

[1] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.
[2] M. Kroetsch and G. Weikum, "Special issue on knowledge graphs," *J. Web Semantics*, Mar. 2015. Accessed: Oct. 2, 2018. [Online]. Available: https://iccl.inf.tu-dresden.de/web/JWS_special_issue_on_Knowledge_Graphs
[3] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data—The story so far," *Int. J. Semantic Web Inf. Syst.*, vol. 5, no. 3, pp. 1–22, Jul. 2009.
[4] S. Funk. (2006). *Netflix Update: Try This at Home*. Accessed: Feb. 9, 2017. [Online]. Available: https://sifter.org/~simon/journal/20061211.html
[5] Y. Shi, M. Larson, and A. Hanjalic, "Mining contextual movie similarity with matrix factorization for context-aware recommendation," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 1, pp. 16:1–16:19, Feb. 2013.
[6] N. Kushwaha, S. Mehrotra, R. Kalia, D. Kumar, and O. P. Vyas, "Inclusion of semantic and time-variant information using matrix factorization approach for implicit rating of Last.Fm dataset," *Arabian J. Sci. Eng.*, vol. 41, no. 12, pp. 5077–5092, Dec. 2016.
[7] J. BenAbdallah, C. Juan Caicedo, A. Fabio Gonzalez, and O. Nasraoui, "Multimodal image annotation using non-negative matrix factorization," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol. (WI-IAT)*, vol. 1, Aug. 2010, pp. 128–135.
[8] B. Abdollahi and O. Nasraoui, "A cross-modal warm-up solution for the cold-start problem in collaborative filtering recommender systems," in *Proc. ACM Conf. Web Sci. (WebSci)*, 2014, pp. 257–258.
[9] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo, "RippleNet: Propagating user preferences on the knowledge graph for recommender systems," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2018, pp. 417–426.
[10] Q. Ai, V. Azizi, X. Chen, and Y. Zhang, "Learning heterogeneous knowledge base embeddings for explainable recommendation," *Algorithms*, vol. 11, no. 9, p. 137, 2018.

[11] V. Bellini, A. Schiavone, T. Di Noia, A. Ragone, and E. Di Sciascio, "Knowledge-aware autoencoders for explainable recommender systems," in *Proc. 3rd Workshop Deep Learn. Recommender Syst. (DLRS)*, 2018, pp. 24–31.

[12] F. Yang, N. Liu, S. Wang, and X. Hu, "Towards interpretation of recommender systems with sorted explanation paths," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 667–676.

[13] B. Abdollahi and O. Nasraoui, "Explainable matrix factorization for collaborative filtering," in *Proc. 25th Int. Conf. Companion World Wide Web*, 2016, pp. 1–6.

[14] B. Abdollahi and O. Nasraoui, "Using explainability for constrained matrix factorization," in *Proc. 11th ACM Conf. Recommender Syst.*, Como, Italy, 2017, pp. 79–83. ACM.

[15] B. Abdollahi, "Accurate and justifiable: New algorithms for explainable recommendations," Ph.D. dissertation, Univ. Louisville, Louisville, Kentucky, 2017.

[16] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "DBpedia—A crystallization point for the Web of data," *J. Web Semantics*, vol. 7, no. 3, pp. 154–165, 2009.

[17] A. Passant and S. Decker, "Hey! Ho! Let's go! Explanatory music recommendations with dbrec," in *Proc. 7th Extended Semantic Web Conf. (ESWC)*, Heraklion, Crete, Greece. Berlin, Germany: Springer, 2010, pp. 411–415.

[18] A. Passant, "Measuring semantic distance on linking data and using it for resources recommendations," in *Proc. AAAI Spring Symp., Linked Data Meets Artif. Intell.*, vol. 77, 2010, p. 123.

[19] A. Passant, "Dbrec: Music recommendations using dbpedia," in *Proc. 9th Int. Semantic Web Conf.*, Shanghai, China. New York, NY, USA: Springer-Verlag, 2010, pp. 209–224.

[20] S. Bostandjiev, J. O'Donovan, and T. Höllerer, "TasteWeights: A visual interactive hybrid recommender system," in *Proc. 6th ACM Conf. Recommender Syst. (RecSys)*, 2012, pp. 35–42.

[21] J. L. Herlocker, J. A. Konstan, and J. Riedl, "Explaining collaborative filtering recommendations," in *Proc. ACM Conf. Comput. Supported Cooperat. Work*, Philadelphia, PA, USA, 2000, pp. 241–250.

[22] S. E. Middleton, H. Alani, and D. C. D. Roure, "Exploiting synergy between ontologies and recommender systems," *CoRR*, vol. 55, pp. 41–50, Apr. 2002.

[23] J. Hu, Z. Zhang, J. Liu, C. Shi, P. S. Yu, and B. Wang, "RecExp: A semantic recommender system with explanation based on heterogeneous information network," in *Proc. 10th ACM Conf. Recommender Syst. (RecSys)*, 2016, pp. 401–402.

[24] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu, "RankClus: Integrating clustering with ranking for heterogeneous information network analysis," in *Proc. 12th Int. Conf. Extending Database Technol., Adv. Database Technol. (EDBT)*, 2009, pp. 565–576.

[25] C. Shi, Z. Zhang, Y. Ji, W. Wang, P. S. Yu, and Z. Shi, "SemRec: A personalized semantic recommendation method based on weighted heterogeneous information networks," *World Wide Web*, vol. 22, no. 1, pp. 153–184, 2018.

[26] C. Musto, F. Narducci, P. Lops, M. de Gemmis, and G. Semeraro, "Linked open data-based explanations for transparent recommender systems," *Int. J. Hum.-Comput. Stud.*, vol. 121, pp. 93–107, Jan. 2019.

[27] R. U. Haq, "Hybrid recommender system towards user satisfaction," M.S. thesis, Univ. Ottawa, Ottawa, ON, Canada, 2013.

[28] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, "MoviExplain: A recommender system with explanations," in *Proc. 3rd ACM Conf. Recommender Syst. (RecSys)*, 2009, pp. 317–320.

[29] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, "Justified recommendations based on content and rating data," in *Proc. WebKDD Workshop Web Mining Web Usage Anal.*, 2008, pp. 1–14.

[30] J. Vig, S. Sen, and J. Riedl, "Tagsplanations: Explaining recommendations using tags," in *Proc. 14th Int. Conf. Intell. User Interfaces (IUI)*, 2009, pp. 47–56.

[31] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," *Proc. IEEE*, vol. 104, no. 1, pp. 11–33, Jan. 2016.

[32] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: A nucleus for a Web of open data," in *The Semantic Web*. Berlin, Germany: Springer, 2007, pp. 722–735.

[33] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Vancouver, BC, Canada, 2008, pp. 1247–1250.

[34] D. Vrandečić, "Wikidata: A new platform for collaborative data collection," in *Proc. 21st Int. Conf. Companion World Wide Web (WWW)*, 2012, pp. 1063–1064.

[35] M. Fabian Suchanek, G. Kasneci, and G. Weikum, "Yago: A core of semantic knowledge," in *Proc. 16th Int. Conf. World Wide Web (WWW)*, 2007, pp. 697–706.

[36] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell, "Toward an architecture for never-ending language learning," in *Proc. 24th AAAI Conf. Artif. Intell. (AAAI)*, 2010, pp. 1306–1313.

[37] A. Singhal. (2012). *Introducing the Knowledge Graph: Things, not Strings*. [Online]. Available: https://www.blog.google/products/search/introducing-knowledge-graph-things-not/

[38] M. Alshammari, O. Nasraoui, and B. Abdollahi, "A semantically aware explainable recommender system using asymmetric matrix factorization," in *Proc. 10th Int. Joint Conf. Knowl. Discovery, Knowl. Eng. Knowl. Manage.*, 2018, pp. 1–6.

[39] J. Zhong, H. Zhu, J. Li, and Y. Yu, "Conceptual graph matching for semantic search," in *Conceptual Structures: Integration and Interfaces*. Berlin, Germany: Springer, 2002, pp. 92–106.

[40] J. P. McGlothlin and L. R. Khan, "Materializing and persisting inferred and uncertain knowledge in RDF datasets," in *Proc. 24th AAAI Conf. Artif. Intell. (AAAI)*, Atlanta, GA, USA, Jul. 2010, pp. 1–8.

[41] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. 20th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2007, pp. 1257–1264.

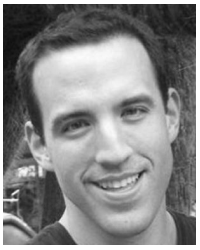
[42] N. Tintarev and J. Masthoff, "A survey of explanations in recommender systems," in *Proc. IEEE 23rd Int. Conf. Data Eng. Workshop (ICDEW)*, Washington, DC, USA, Apr. 2007, pp. 801–810.

[43] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*. Hoboken, NJ, USA: Lawrence Erlbaum Associates, 1988.

[44] L. J. Cronbach, "Coefficient alpha and the internal structure of tests," *Psychometrika*, vol. 16, no. 3, pp. 297–334, Sep. 1951.

MOHAMMED ALSHAMMARI received the Bachelor of Education degree in computer from the University of Hail, Saudi Arabia, in 2008, and the M.Sc. degree in computer science from the University of Leicester, U.K., in 2011. He is currently pursuing the Ph.D. degree with the Knowledge Discovery and Web Mining Laboratory, Computer Engineering and Computer Science Department, University of Louisville, KY, USA. From 2012, he is a Lecturer with the Northern Border University, Saudi Arabia.

OLFA NASRAOUI received the Ph.D. degree in computer engineering and computer science from the University of Missouri, Columbia, in 1999. She is currently the endowed Chair of e-commerce and the Founding Director of the Knowledge Discovery and Web Mining Laboratory, University of Louisville, where she is also a Professor of computer engineering and computer science. She was a recipient of the National Science Foundation CAREER Award, two Best Paper Awards for theoretical contributions in computational intelligence at the ANNIE 2001 Conference, and a recent one at the Knowledge Discovery and Information Retrieval, KDIR 2018 Conference.



SCOTT SANDERS received the M.A. degree in communication from Purdue University, in 2007, and the Ph.D. degree in communication from the University of Southern California, in 2012. He is currently an Assistant Professor of communication with the University of Louisville, where he teaches courses on communication technologies, social media, and research methods. As a Multi-Disciplinary Researcher drawing from the fields of marketing, communication, computer science, and psychology, his primary research interest includes brand–consumer communication via social media.