Title	Explainable and Accurate Top-N Recommendations in Heterogeneous Information Networks
Authors(s)	Ozsoy, Makbule Gulcin, O'Reilly-Morgan, Diarmuid, Symeonidis, Panagiotis, Tragos, Elias, Hurley, Neil J., Smyth, Barry, Lawlor, Aonghus
Publication date	2020-10-05
Publication information	Ozsoy, Makbule Gulcin, Diarmuid O'Reilly-Morgan, Panagiotis Symeonidis, Elias Tragos, Neil J. Hurley, Barry Smyth, and Aonghus Lawlor. "Explainable and Accurate Top-N Recommendations in Heterogeneous Information Networks." IEEE, October 5, 2020. https://doi.org/10.1109/ACCESS.2020.3028587.
Publisher	IEEE
Item record/more information	http://hdl.handle.net/10197/25697
Publisher's version (DOI)	10.1109/ACCESS.2020.3028587

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 31, 2020, accepted September 21, 2020, date of publication October 5, 2020, date of current version October 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3028587

MP4Rec: Explainable and Accurate Top-N Recommendations in Heterogeneous Information Networks

MAKBULE GULCIN OZSOY, DIARMUID O'REILLY-MORGAN, PANAGIOTIS SYMEONIDIS, ELIAS Z. TRAGOS, NEIL HURLEY, BARRY SMYTH, AND AONGHUS LAWLOR
Insight Centre for Data Analytics, University College Dublin, Dublin 4, D04 V1W8 Ireland

Corresponding author: Makbule Gulcin Ozsoy (makbule.ozsoy@insight-centre.org)

This work was supported in part by the Science Foundation Ireland under Grant SFI/12/RC/2289_P2, and in part by the Samsung Research, Samsung Electronics Company, Seoul, South Korea.

ABSTRACT Neural network-based recommendation algorithms have become the state-of-the-art in recommender systems and can achieve very high predictive accuracy. However, these models are usually considered as black boxes in terms of their interpretability due to the complex structure of their hidden layers. In this research work, we propose MP4Rec, a recommender system using heterogeneous information networks to provide both accurate and explainable recommendations. MP4Rec uses of user-user and item-item similarity matrices and applies a newly proposed pair-wise objective function to make top-N recommendations which are transparent and explainable. The similarity matrices are created from metapaths constructed with the PathSim algorithm, node embeddings with cosine similarity or their combinations. The proposed pair-wise objective function incorporates an additional soft constraint for pushing more explainable items into the top-N recommendations. We have performed several experiments that show the effectiveness of our model by outperforming the state-of-the-art and providing both accurate and explainable recommendations in three well-known datasets.

INDEX TERMS Explainability, heterogeneous information networks, recommender systems, top-n recommendation.

I. INTRODUCTION

Heterogeneous Information Networks (HINs) consist of different node and link (relation) types. Classic local graph structure models [1] and latent factor models [2], [3] may be applied over metapaths in HINs. A metapath is a sequence of relations between node types in a graph and carries a particular semantic meaning. For example, in Fig. 1 an instance of the metapath UMG (user, movie, genre) is:

$$\text{Tom} \xrightarrow{\text{likes}} \text{Avatar} \xrightarrow{\text{belongs to}} \text{Sci-Fi} \xrightarrow{\text{- (belongs to)}} \text{The Terminator} \xrightarrow{\text{- (likes)}} \text{Mary.}$$

Recently, metapaths are started to be used by graph-based neural network methods [4], [5] for recommendation. Reference [4] uses metapaths for top-N recommendations. However, it only captures the similarity between two nodes based on the number of instances of a metapath that connects them, and does not use a metapath based random walk strategy such as Node2Vec [6] or Metapath2Vec [2], to capture the overall relations among the nodes of the graph. Such a strategy could potentially model the rich relational semantics in HINs [7]. Moreover, recent graph-based neural models that use metapaths [4], [5] focus only on providing accurate recommendations and do not build a model which recommends more explainable items, failing to take advantage of the explanatory power of metapaths. Even though their recommendations have potentially good interpretability, there is neither experimental evidence nor any mechanism in their models (such as a constraint in the objective function or an attention mechanism for identifying the most explainable items/features) to leverage the explainability of their recommendations.

In this article, we propose MP4Rec, a new algorithm for item recommendations in HINs, which combines a random walk strategy and an explainability model into a single heterogeneous information-aware neural network. We combine metapaths extracted from a HIN to better explain our predicted item recommendations using hybrid

The associate editor coordinating the review of this manuscript and approving it for publication was Farhana Jabeen.

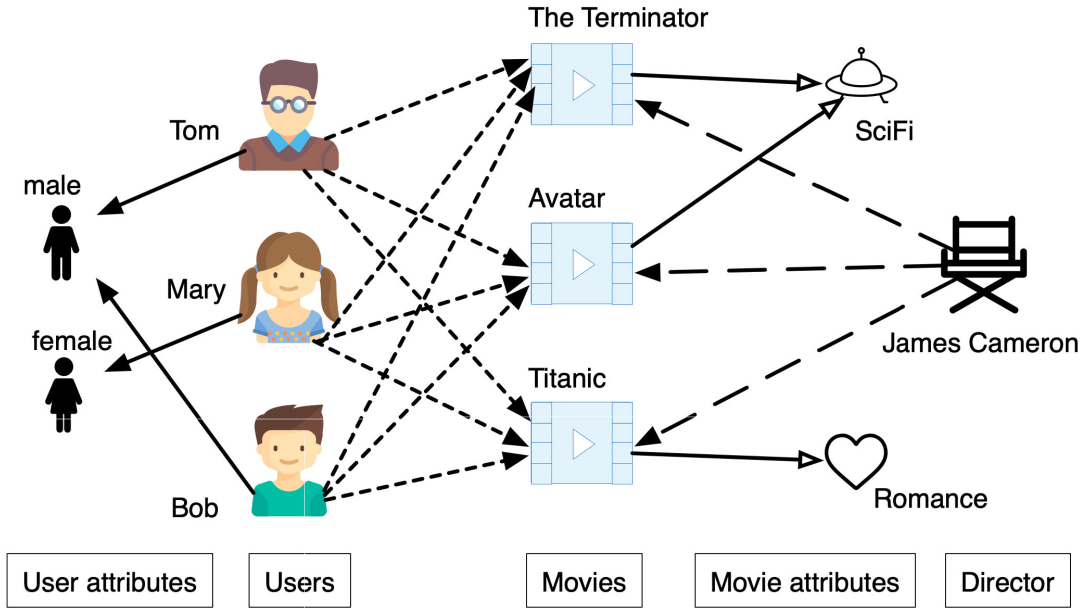

FIGURE 1. Toy example for a movie rental eShop.

metapath-based explanations. We propose a pair-wise objective function for top-N recommendations, which promotes the items that are both explainable and accurate into the top-N set. MP4Rec first extracts different similarity matrices of users and items through either the local/global graph structure or different metapaths and then trains a well-designed deep neural network with these matrices to learn the metapath-level latent factors and their corresponding importance using a metapath-level attention mechanism. The metapath-level attention mechanism is able to find the importance of each metapath. Moreover, in contrast to the related work [4], [5] that has used metapaths in neural networks for increasing only the accuracy of recommendations, we aim in providing both accurate and explainable item recommendations. That is, we combine metapaths extracted from a HIN to better explain our predicted item recommendations using hybrid metapath-based explanations.

Inspired by the work of [8], we learn explanations using association rule mining. In this way, an item recommended to a target user is considered an explainable item if there are one or more items in his/her rating history which support the recommendation with high support, confidence or lift. An example of such an explanation is: "Because you rated the movie Terminator, we recommend Avatar to you", such that the Avatar movie is an explainable item for the target user, because he has previously liked the Terminator movie. The format $\{Explanation\} \Rightarrow \{Recommendation\}$ (e..g., $\{Terminator \Rightarrow Avatar\}$) is well known as the item-style of explanation [9], [10], and depicts the most influential item from the user history profile that is responsible for a recommended item. In the context of metapath-based explanations, we are able to provide the aforementioned item-based style of explanation, by using the metapath MUM (Movie, User, Movie). In addition, we can provide more advanced explanations, i.e. by combining the information from MUM and MGUGM (Movie, Genre, User, Genre, Movie) metapaths as follows: "*Because you showed interest for 3 other Action movies (Avengers, Rambo, Indiana Jones), we recommend you the Terminator movie.*"

The rest of this article is organized as follows: Section II summarises the related work, Section III presents our proposed method, Section IV presents the experimental results, and finally, Section V concludes the paper.

II. RELATED WORK AND PRELIMINARIES

In this section, we explain Heterogeneous Information Networks (HINs), how similarity and latent representation of the entities of HINs are calculated and how the extracted information is used for making recommendations, specifically using neural network techniques.

A. HETEROGENEOUS INFORMATION NETWORKS

Heterogeneous information networks (HINs) are directed graphs using various types of information as nodes that are linked with the edges [11]. Metapaths are defined as a series of paths in the graph connecting nodes using some type of composite operator on node relations [11]. Formally;

1) INFORMATION NETWORK

An information network is defined as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *with a node type mapping function* $\phi : \mathcal{V} \rightarrow \mathcal{Q}$ *and a link type mapping function* $\psi : \mathcal{E} \rightarrow \mathcal{R}$*, where each node* $v \in \mathcal{V}$ *belongs to one particular object type* $\phi(v) \in \mathcal{Q}$*, and each link* $e \in \mathcal{E}$ *belongs to a particular relation* $\psi(e) \in \mathcal{R}$.

For example, in the user-movie graph of Fig. 1, two movies can be connected through the path MGM "movie-genre-movie" (content-based similarity) or MUM "movie-user-movie" (item-based collaborative filtering similarity). Using different paths, different similarities are observed. These paths are called *meta paths* and are formally defined as follows [11]:

2) META PATH

A meta path $\mathcal{P}$ is a path defined on the graph of network schema $T_G = (\mathcal{Q}, \mathcal{R})$, and is denoted in the form of $Q_1 \xrightarrow{R_1} Q_2 \xrightarrow{R_2} \ldots \xrightarrow{R_l} Q_{l+1}$, which defines a composite relation $R = R_1 \circ R_2 \circ \ldots \circ R_l$ between type Q_1 and Q_{l+1}, where $\circ$ denotes the composition operator on relations.

There are many metapaths that can be built from the user-movie graph of Fig. 1. For example, if we start from user nodes, we can build: UMU (user-movie-user), UMGMU (user-movie-genre-movie-user), and UMDMU (user-movie-director-movie-user) to infer user-user similarities, and consequently recommend movies from similar users (i.e., user-based collaborative filtering).

HINs contain rich heterogeneous data, which is used in various problems such as similarity measurement, entity search, clustering, classification, link prediction and recommender systems [5], [11]–[13]. Additional to HINs, knowledge graphs (KG) are also used for representing heterogeneous data. In KGs, the nodes represent entities and edges represent Resource Description Framework (RDF)-like (<subject, property, object>) relations [14]. It is possible to represent a KG as a HIN, where the subject and object types are represented as a node and property as an edge [15]. As a result, KGs can be considered as an instance of HINs [14]. However, KGs have some drawbacks for the computation due to their huge and complex structure. For example, in KGs enumeration of metapaths is impossible or the relations may indicate multiple meanings, i.e., the property of the edges can cause ambiguity [15]. In the literature, various methods, such as [16]–[23], use KGs for making recommendations.

In this work, we focus on recommender systems that use the data provided in HINs, specifically the metapaths. For this purpose, we use metapath similarity measures and embedding (latent-space representation learning) techniques.

B. METAPATH-BASED SIMILARITY MEASURES AND EMBEDDING TECHNIQUES

Metapath-based similarity methods utilise connectivity among the entities [14]. For this purpose, it is possible to use methods based on node neighborhoods or methods using the ensemble of all paths [24]. Node neighborhood measures, such as Common neighbors, Jaccard's coefficient, Adamic/Adar index [25], Preferential attachment [1], are based on the idea that having largely overlapping neighbors indicates similarity between nodes. Methods using ensembles of all paths, such as Katz [26], PageRank [27], SimRank [28], Random Walk with Restart (RWR) [29], PathRank [30], use not only the direct connection but also connections among all nodes.

Learning the network representation through embedding (latent representation learning) techniques is successfully applied in many tasks in data mining, specifically in recommender systems [5]. For example, GraRep [31] learns the representation of the network by capturing the higher-order graph proximity [7]. DeepWalk [32], Node2Vec [6] methods use random walk and Skip-gram techniques, learn network representations and consider each node as an embedding [5]. However, these methods focus on homogeneous networks, ignore the node types and use one metapath, i.e., the rating matrix. Recently, the embedding techniques using the heterogeneous data provided in HINs has gained more attention. [33] categorises the existing heterogeneous network embedding algorithms into three, namely proximity-preserving methods, message-passing methods and relation-learning methods. Among these three categories, the proximity-preserving methods are the most commonly used heterogeneous network embedding technique [33]. They capture the complex interactions among the elements of the HINs and model the rich relational semantics in them [7]. Some example proximity-preserving methods are [2], [3], [34]–[42]. Reference [33] provides further analysis and explanations on heterogeneous network embedding algorithms.

The similarity measures and the latent-space representations (embedding) of the elements of the HINs provide valuable information for making recommendations. Next, we will explain how these are used by the recommender systems methods in the literature.

C. METAPATH-BASED RECOMMENDER TECHNIQUES

HIN-based recommender methods can be grouped into two categories: (i) path-based methods and (ii) embedding-based methods.

The path-based methods utilise the connectivity similarity among users and items to make recommendations [14]. In general, the computed similarity measures are used as an enrichment for the user or item representations or as a regularization term in the objective function. For example, [43] uses item-item similarity, [44] uses item-item, item-user, user-user similarities as the regularization terms, [45] and [12] use metapath similarities for the enrichment of the input user-item interaction matrix [14]. For the computation of similarity, PathSim [11] is commonly used for the metapath similarity and MF techniques are commonly used for the recommendation computation [14]. For example, [12], [13], [43], [45]–[47], employ matrix factorization-based techniques to make recommendations. In addition to positive interactions (e.g., favored items), it is possible to consider negative interactions (e.g., hated items) [14], [48].

There are various recommendation methods employing latent vector representation models (embeddings), such as [4], [5], [7], [49]. These methods do not necessarily utilise HINs. For example, [49] does not use HINs, but

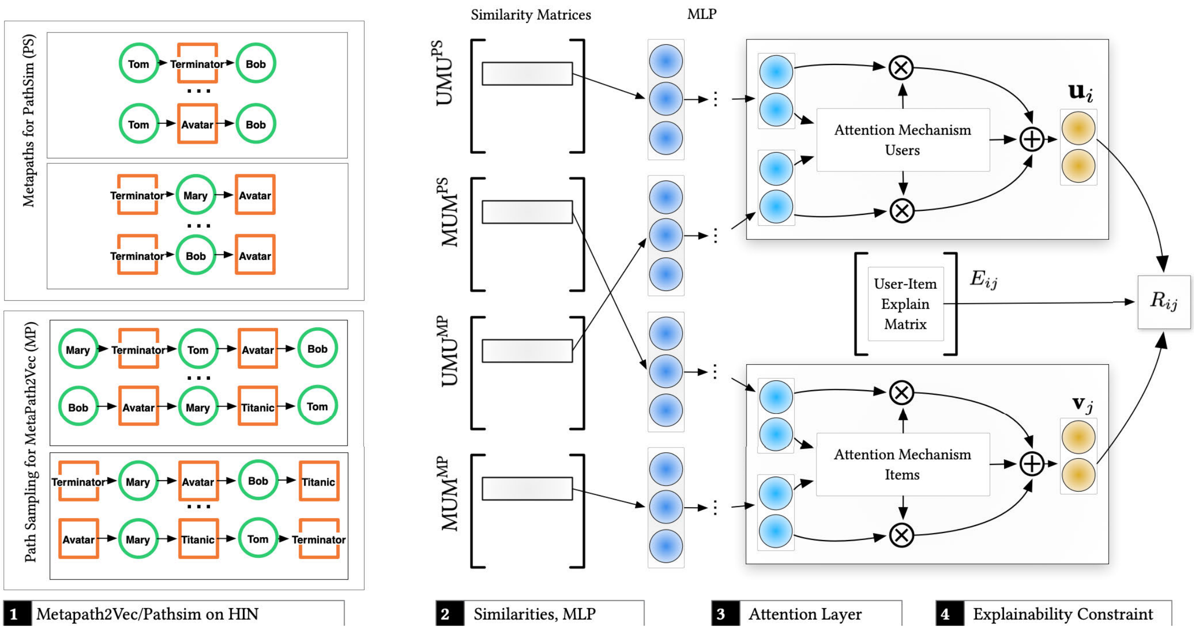

FIGURE 2. **MP4Rec architecture, depicting the four steps described in Sec III.**

uses embeddings in their NeuMF method. In this section, we focus on the recommender methods that utilise HINs, specifically metapaths, to learn the latent vector representations. Embedding-based methods using HINs model the user-item relations by learning the latent vector space representations of metapaths [14]. Reference [5] extends NeuMF in their MCRec method by incorporating a convolution neural layer to capture the context of a user-item interaction, using path sampling over metapaths to build node embeddings. MCRec uses the cross-entropy loss function to predict a user-item interaction. Reference [4] proposes the NeuACF method. It also uses metapath-based similarity matrices as input, exploits an MLP to build the user/item embeddings and a path-level attention layer to identify important metapaths for a user. It uses a point-wise cross-entropy loss function to predict user-item interactions. [7] proposes HERec method, which first learns user or item embeddings from metapath guided random walks and then use the embeddings in a matrix factorization framework to predict the rating scores of the items.

Explainable recommendation is another hot topic both in research and industry [14], [50], [51]. Many of the explainable recommender systems, such as [52]–[56], only make use of the textual data, e.g., reviews, to produce explanations [50], but not the HINs. The provided attributes of users and items and their connections represented in HINs and knowledge graphs enable the recommender systems to reason about and explain the recommendations [57]. Recently, recommendation methods which use HINs or knowledge graphs provide explanations along with the recommendations, such as [23], [57]–[62]. However, these methods have some drawbacks, such as only using item attributes, not using both path similarity and embeddings as the input. Since knowledge graphs are usually huge, extracting all the paths between a user-item pair is computationally prohibitive [51]. Also, making calculations for each user-item pair while making top-n recommendations is computationally intractable.

The proposed MP4Rec method aims to make accurate and explainable top-n recommendations while using both path similarity and node embeddings. It does not introduce any constraint on attributes and allows the researchers/developers to decide which attributes to use. Its neural network architecture is relatively simple, such that it is composed of a single layer MLP component, an attention component and an optimization component which takes the explainability into account. These components are detailed in the next section.

III. THE MP4Rec METHOD

Our proposed method consists of four steps, shown in Fig. 2: (i) We build a HIN based on the participating entities (i.e., user, item, item category, etc.). We run the PathSim/Metapath2Vec algorithms on the HIN and create user-user and item-item similarity matrices based on different metapaths and node sequences. (ii) The similarity matrices are used as an input to a deep neural network. Then, we use a Multi-layer Perceptron (MLP) to capture non-linear relations and build the latent factors of users and items. (iii) The latent factors of users and items are then re-arranged in user-user latent space and item-item latent space, and they are used as input in an attention layer to compute the attention weights for each metapath (i.e., user or item). The final latent factors are fused for user and items, respectively. (iv) Finally, we extend the Bayesian Personalized Ranking (BPR) [63] objective function by adding a soft explainability constraint to push more explainable items in the top-N positions of the recommendation list.

A. SIMILARITY MATRICES CREATION

While using the metapaths, it is possible to use various different similarity techniques, such as SimRank [28], RWR [29], PathSim [11]. We chose PathSim [11] because it is able to find semantically similar peer entities in the network, such that the entities that are not only strongly connected but also have comparable visibility. For example, in an ''author-conference-publication'' network, unlike other metapath-based similarity metrics, PathSim can find similar authors even if they did not attend to the same conferences [11]. Given a symmetric meta path $\mathcal{P}$, the PathSim similarity between two nodes, namely x and y, which are of the same type, can be calculated by equation (1). In this equation, $p_{x \rightsquigarrow y}$ is a path instance between x and y, $p_{x \rightsquigarrow x}$ is that between x and x, and $p_{y \rightsquigarrow y}$ is that between y and y.

$$s(x, y) = \frac{2 \times |p_{x \rightsquigarrow y} : p_{x \rightsquigarrow y} \in \mathcal{P}|}{|p_{x \rightsquigarrow x} : p_{x \rightsquigarrow x} \in \mathcal{P}| + |p_{y \rightsquigarrow y} : p_{y \rightsquigarrow y} \in \mathcal{P}|}, \quad (1)$$

It is also possible to utilise embedding techniques to extract the structural and semantic relations among the nodes. In this work, we use Metapath2Vec [2] as the embedding technique to learn the representation of the elements of the input network. Metapath2Vec [2] extends the DeepWalk [32] and Node2Vec [6] algorithms for heterogeneous graphs, by generating only metapath-based random walks where the walks are constrained on specific metapaths. It creates node embeddings, i.e., learns vector representations, using various parameters, e.g., window size and the number of hops, see [2] for details. Its output embeddings can be used for various tasks, such as classification, clustering and similarity search [2]. For example, [2] uses the output embeddings in an ''author-conference dataset'', and experimentally shows that not only structurally but also semantically similar conferences (e.g., same domain) and authors (e.g., similar prestige levels) can be found.

We compute user-user and item-item similarity matrices based on different metapaths. Having the metapaths as input, we (i) calculate similarity among the metapaths using PathSim, shown as PS in Fig. 2, (ii) calculate node embeddings using Metapath2Vec [2] and then calculate the cosine similarity among them, shown as MP in Fig. 2. We are aware that calculating similarity for all metapaths and/or node embeddings may introduce scalability issues. However, they can be calculated offline, prior to the computations of the recommendation task. Alternatively, the embeddings can be used as the direct input to the system, instead of using their similarities. We consider solutions on scalability related issues as a future work.

B. MLP MECHANISM

For each user/item, we use the similarity matrices from step 1 as the input to a Multi-layer Perceptron (MLP), to learn the metapath-level latent factor as the output. Let's consider the user-user similarity matrix $S^{\mathcal{P}} \in \mathbb{R}^{N \times N}$ over metapath $\mathcal{P}$, where N is the number of user nodes. Thus, for user i, we have an N-dimensional vector $S^{\mathcal{P}}_{i*}$, which is just a row of similarity matrix $S^{\mathcal{P}}$ that keeps the similarities of i with all other users. The MLP projects the initial similarity vector $S^{\mathcal{P}}_{i*}$ of user i to a low-dimensional latent factor. For example, given the initial input vector $S^{\mathcal{P}}_{i*}$, the 1st hidden layer of the MLP maps it to a hidden representation $\boldsymbol{Z}_1 \in \mathbb{R}^K$ through the following mapping function:

$$\boldsymbol{Z}_1 = f\left(\boldsymbol{W}_1^T S^{\mathcal{P}}_{i*} + \boldsymbol{b}_1\right), \quad (2)$$

where $\boldsymbol{W}_1 \in \mathbb{R}^{N \times K}$ is a weight matrix and $\boldsymbol{b}_1 \in \mathbb{R}^K$ is an offset vector. Given an MLP of l layers, in each layer of MLP, the input vector is mapped into another vector in a new (latent) space. The resulting latent representation is then used for constructing (latent) user vectors over metapath $\mathcal{P}$ as follows:

$$\begin{aligned} \boldsymbol{Z}_1 &= f\left(\boldsymbol{W}_1^T S^{\mathcal{P}}_{i*} + \boldsymbol{b}_1\right) \\ \boldsymbol{Z}_2 &= f\left(\boldsymbol{W}_2^T \boldsymbol{Z}_1 + \boldsymbol{b}_2\right) \\ &\dots \\ \boldsymbol{u}_i^{\mathcal{P}} &= \boldsymbol{Z}_l = f\left(\boldsymbol{W}_l^T \boldsymbol{Z}_{l-1} + \boldsymbol{b}_l\right) \end{aligned} \quad (3)$$

Even though any activation function (such as sigmoid, tanh, rectified linear unit - RELU) can be used for the activation function of MLP layers, namely f, [49] states and empirically confirms that RELU is more suitable and works better with sparse datasets, which are common in recommender systems domain. According to these analysis, we chose to use RELU as our activation function.

C. ATTENTION MECHANISM

The output of MLP layer is the latent user (or item) factors over all the input metapaths and they are learned separately for users and items. The next step is to combine these factors and obtain aggregated latent factors, e.g., one for each user i and item j. However, not all the metapaths have the same importance for a user or an item. We use an attention mechanism which assigns weights (normalized so that they sum to 1) for all metapath-based latent factors to identify the most important metapaths as follows:

$$\boldsymbol{u}_i = \sum_{l=1}^{|\mathcal{P}|} Att_i^l \cdot \boldsymbol{u}_i^l, \qquad \boldsymbol{v}_j = \sum_{l=1}^{|\mathcal{P}|} Att_j^l \cdot \boldsymbol{v}_j^l \quad (4)$$

where l is a metapath, Att_i^l (or Att_j^l) is the attention weight, $\boldsymbol{u}_i$ is the user's aggregated latent representation and $\boldsymbol{v}_j$ is the item's aggregated latent representation. For the computation of the attention weights, Att_i^l, we used a layered implementation similar to the attention layer of NeuACF [4]. The attention weights are computed by a two-layer network and the weights are normalized using a softmax function.

D. ADJUSTING THE OBJECTIVE FUNCTION FOR BOOSTING MORE EXPLAINABLE ITEMS

For the top-N recommendation task, most of the existing methods use point-wise recovery loss or pair-wise ranking loss [64]. According to [64], while the point-wise loss focuses on entry-wise consistency (of the recommendation and original input), pair-wise loss focuses on structure consistency and preserves the pair-wise preference structure [64]. As explained in [64], pair-wise ranking methods treat data as the set of triplets where each triplet is composed of user i and two items, j and k. The triplet indicates the preference of the user of the first item, item j, over the second item, item k. In the recommender systems settings, the observed interactions, i.e., positive items, are preferred to unobserved interactions, i.e., negative items. As a result of pair-wise learning, the positive items are ranked higher than the negative ones in the output recommendation [49]

In this work, we used a pair-wise ranking loss, namely the Bayesian Personalized Ranking (BPR) [63] loss: Let $\mathcal{D}_u$ be a set of item pairs (j, k), where user u has interacted with item j but not with item k. A predicted score for the relevance of item j to user u, is given by $\mathbf{u}_u^T \mathbf{v}_j$. Assuming that user u should be more interested in item j than item k, the BPR [63] algorithm seeks latent vectors $\mathbf{u}_u$ for all users u and $\mathbf{v}_j$ for all items j to minimise the pair-wise ranking loss (Equation (5) without the term inside the dashed-box). In order to recommend more explainable items, we included an additional term in the classic BPR loss formula, the term inside the dashed-box, in the following equation:

$$\sum_u \sum_{(j,k)\in\mathcal{D}_u} -\log\sigma\Big((\mathbf{u}_u^T\mathbf{v}_j - \mathbf{u}_u^T\mathbf{v}_k) \boxed{-\beta(\|\mathbf{u}_u-\mathbf{v}_j\|_2^2\,E_{uj} - \|\mathbf{u}_u-\mathbf{v}_k\|_2^2\,E_{uk})}\Big) + \lambda_U\|\mathbf{u}_u\|_2^2 + \lambda_V(\|\mathbf{v}_j\|_2^2 + \|\mathbf{v}_k\|_2^2) \quad (5)$$

where σ is the sigmoid function, λ_U and λ_V are scalars used for tuning the effect of regularization terms, β is a scalar used for tuning the effect of explainability and E_{uj} indicates the explainability of an item j for the user u. In equation (5), we use both the explainability of the already interacted item j, and the non-interacted item k. Similar to the original equation of BPR [63] we use the pair-wise comparison and we assume user u finds the already interacted item more explainable than the non-interacted item. The expression $\|\mathbf{u}_u - \mathbf{v}_j\|_2^2$ constrains the distance of the user/item vectors in the latent space, in such a way that they are close to each other (i.e., their difference is close to zero). One disadvantage of introducing an explanation constraint on the loss function is that it can degrade the performance of the recommendation algorithm in terms of novelty and diversity metrics. We want to focus on these aspects of the recommendations in a future work.

Inspired by the work of [8], the explainability of an item for a user, namely E_{uj}, is learned by association rule mining. A recommended $item_i$ is considered as an explainable item for the $user_u$, if there are one or more interactions in the $user_u$'s history, which implies that the $item_i$ is relevant to the $user_u$ with high support (*supp*), confidence (*conf*) and lift measures. A rule $\{X \Rightarrow Y\}$ holds with *supp*, if *supp*% of the transactions contain $X \cup Y$. It is a frequency measure of how often a pattern is observed. The rule holds with *conf*, if *conf*% of transactions that contain X also contain Y as defined by the ratio of their supports $supp(X \cup Y)/supp(X)$. The rule's *lift* measures the positive correlation between two items defined as the combined support of the items, $supp(X \cup Y)$, divided by the support if the items were independent, $supp(X) \times supp(Y)$.

TABLE 1. Parameters used by association rule mining.

A_j	A_j are the items that are associated with item j
M_{ij}	M_{ij} is explainability matrix. $M_{ij} = 1$ if the rule $(i \Rightarrow j)$ holds.
E_{uj}	E_{uj} is the explainability of the item j for the user u

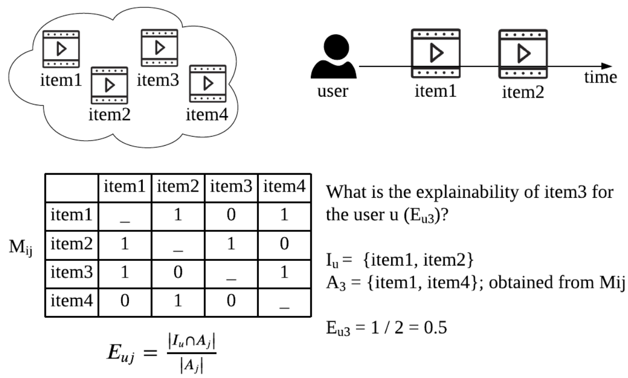

FIGURE 3. An example computation of the explainability of an item for a user.

The explainability of an item for a user is calculated by $E_{uj} = \frac{|I_u \cap A_j|}{|A_j|}$, where I_u is the set of items that the $user_u$ has previously interacted with, e.g., clicked, purchased items, and A_j is the set of items that are associated with $item_j$. For the computation of A_j, a binary item-item explainability matrix M_{ij} is utilised, where $A_j \Leftrightarrow M_{ij} = 1$ returns the items associated with $item_j$ according to the association rules. For example, assume there are four distinct items in the item space (from $item_1$ to $item_4$), and the row for the $item_3$ is $M_{3j} = [1, 0, \_, 1]$, as in the Fig. 3. Then the A_3 contains $item_1$ and $item_4$, but not $item_2$. $M_{ij} = 1$ when the rule $\{i \Rightarrow j\}$ has been found; i.e. the *supp*, *conf* and *lift* are all above the pre-set thresholds. By setting user-defined thresholds for *supp*, *conf*, and *lift*, we identify all rules of size 2 (i.e., $\{X \Rightarrow Y\}$) from the user-item matrix.

An example of an association rule mining-based explanation is: "Because you rated the Terminator movie, we recommend you the Avatar movie", indicating that the Avatar is an explainable item for the target user, since the user has previously liked the Terminator. The format $\{Explanation\} \Rightarrow \{Recommendation\}$ (e..g., $\{Terminator \Rightarrow Avatar\}$) is well known as the item-style of explanation [9], [10], and depicts the most influential item from the user history profile that is responsible for a recommended item. Even though this kind of explanation does not fully utilise the information obtained from metapaths, we consider it as one step forward for explaining a neural networks-based recommendation method.

TABLE 2. Statistics of the used datasets.

Datasets	Relations (A-B)	#A	#B	#A-B
MovieLens	User(U)-Movie(M)	943	1,682	100,000
	User-Sex(S)	943	2	943
	User-Occupation(O)	943	21	943
	Movie-Genre(G)	1,682	19	2,893
Amazon	User(U)-Item(I)	6,170	2,753	195,791
	Item-Brand(B)	2,753	334	2,753
	Item-Category(Ca)	2,753	22	5,508
Yelp	User(U)-Business(B)	16,239	14,284	198,397
	User-User	16,239	16,239	158,590
	Business-City(Ci)	14,267	47	14,267
	Business-Category(Ca)	14,180	511	40,009

TABLE 3. The selected metapaths used in each dataset.

Dataset	Metapaths
MovieLens	UMU, USU, UOU, UMGMU, MGM, MUM, MUOUM, MUSUM
Amazon	UIU, UIBIU, UICIU, IUI, ICI, IBI
Yelp	UBU, UUU, UBCiBU, UBCaBU, BCaB, BCiB, BUB, BUUUB

IV. EXPERIMENTAL EVALUATION

In this section, the datasets, the evaluation protocol, experimental settings and evaluation results are explained.

A. DATASETS

We use three widely used datasets from different domains, namely MovieLens[1] movie dataset, Amazon[2] purchases dataset and Yelp[3] business dataset. For Amazon, we select only the items of the Electronics category. For Yelp we use the 20-core setting so that users have at least 20 interactions. The detailed descriptions of the datasets are shown in Table 2. From each dataset, various combinations of metapaths can be produced. Here, we selected only a subset containing short metapaths reported in Table 3, since long metapaths are likely to introduce noisy semantics [11].

B. EVALUATION PROTOCOL, METRICS AND SETTINGS

We follow the evaluation protocol used in [5]. We split the entire dataset into training and test splits by 80%-20% split ratios. Moreover, we used 10% of the training dataset as the validation set for parameter tuning. For the computation of overall evaluation metrics, the metrics are first computed per user and then they are averaged over all users. For stability, similar to [5], we perform multiple runs using different random splits to report the evaluation results.

In the training step, we optimized our model with Adam optimizer, and set the batch size to 512. For each dataset (MovieLens, Amazon and Yelp), we set the learning rates to 10^{-4}, 10^{-4} and 5×10^{-5} and the hidden size of a 2-layer MLP to 64, 512 and 64, respectively. Also, for all three datasets, we set the node embedding size to 64, the number of random walks for each node to 100, and the number of negative samples used by Metapath2Vec to 10 negatives for each positive. Moreover, we calculated E_{uj} using the thresholds 0.2, 0.2, 0.2 for MovieLens and 0.005, 0.005, 0.005 for Amazon and Yelp, for *supp*, *conf* and *lift* parameters.

While executing the evaluation, similar to the evaluation setting described in [5], for each positive item associated with a target user in the test set we randomly sample 50 negative items that have no interaction records with the user. Then, we use MP4Rec to rank these positive and negative items. We report Precision and Recall at rank N (i.e., Prec@N, Recall@N) as the accuracy-based evaluation metrics. To measure the explainability, similar to [8], [65], we use the Explainability Precision (EP), as shown in equation (6). EP is the proportion of explainable items in the top-N recommendation list relative to the number of recommended items for each user.

$$EP = \frac{|\text{explainable items} \cap \text{recommended items}|}{|\text{recommended items}|} \quad (6)$$

The final results are first averaged over all the test items of a user and then averaged over all users, to obtain the Mean *EP* (MEP), precision and recall.

C. EVALUATION RESULTS

In this section, we analyse the impact of various parameters, evaluate the performance of MP4Rec in terms of precision, recall and MEP and present qualitative analysis on explainability of the recommendations.

1) PARAMETER TUNING

We explore how the performance of MP4Rec is affected by the λ and β parameters which are used in equation (5), as well as by the number of hops and window size parameters, which are used when creating node embeddings. This step is executed using the validation split. First, we explore the effect of the λ parameter which tunes the regularization terms. For the experiments, we didn't differentiate λ_U and λ_V and set both to the same value. To decide on the best performing λ, we kept $\beta = 0.25$ and evaluated MP4Rec using the values: $[0.001, 0.010, 0.050, 0.100, 0.250, 0.500]$. In all the datasets, we observed that the precision and explainability scores increase until a certain λ value (not necessarily the same value), and then they start to decrease. According to our observations, we set λ to 0.100, for all the three datasets. Next, we explore the effect of the β parameter that controls explainability in equation (5). We experimented on a range of β values from $[0.00, 0.25, 0.50, 0.75, 1.00]$, on all three datasets. Experimental results show that increasing the β values increases the explainability score, but decreases the precision scores. According to the evaluation results, we set β to 0.25, for all datasets, in the rest of the experiments. Last, we analysed the window size(ws) and number of hops(n_hops); which are used by Metapath2Vec for creating node embeddings. For the analysis, we used [UMU, MUM] as the input paths and assigned ws to [2, 5, 7] and n_hops to [3, 10, 25]. The best performance is obtained when ws is

[1]https://grouplens.org/datasets/movielens/100k/
[2]http://jmcauley.ucsd.edu/data/amazon/
[3]http://www.yelp.com/dataset-challenge

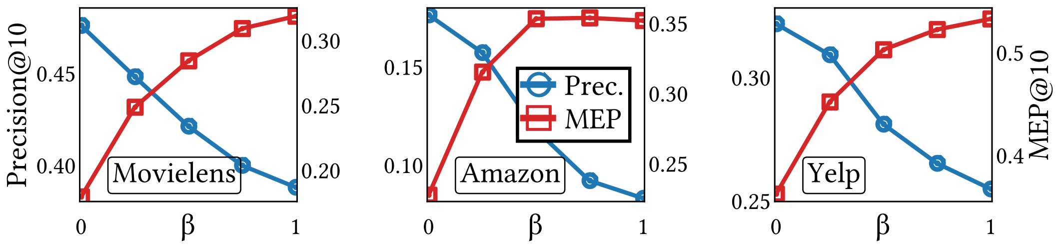

FIGURE 4. Trade off between precision and explainability.

5 and n_hops is 10; these values are used in the rest of the experiments.

2) CHOOSING THE RIGHT PATH

Considering that there is a large number of metapaths available in HINs and their combination introduces even more variations, it is necessary to decide which metapaths will be used. In this section, using the validation split, the metapaths to be used are chosen. In the literature, the knowledge of experts [2], heuristic approaches [7] or recently, algorithmic approaches [36], [61] are used for this purpose. Here, we consider the user-item paths as the building blocks and used UMU and MUM for MovieLens, UIU and IUI for Amazon, UBU, BUB for Yelp as the base metapaths.

We used the findings of [5] to choose the paths to calculate the PS similarity. Reference [5] also used MovieLens and Yelp datasets and found that the paths UMGMU and UBCaBU, for MovieLens and Yelp respectively, gained higher attention. For the Amazon dataset, after experimenting on available metapaths and observing that their accuracy and explainability results are similar, we decided to use IBI relation in addition to the base paths. As a result, to calculate the PS similarities, we used [UMU, UMGMU, MUM, MGM] for MovieLens, [UIU, UIBIU, IUI, IBI] for Amazon and [UBU, UBCaBU, BUB, BCaB] for Yelp.

In order to choose the paths to calculate the MP similarity, we experimented on various combinations of paths. For MovieLens, the results show that the embeddings, even the ones using the baseline paths, e.g., [UMU, MUM], are able to produce high precision and recall. Even though additional metapaths do not increase accuracy, they help to improve the explainability. As a result, we decided to use [UMU, MUM, MUOUM] for MovieLens when calculating the MP similarity. For the Amazon and Yelp datasets, the results show that all the embeddings using any metapath perform equally well. Observing a slight increase in explainability scores, around 0.01, we decided to use [UIU, UICIU, IUI] and [UBU, UBCaBU, BUB], for Amazon and Yelp datasets, respectively.

3) IMPACT OF USING PS AND/OR MP SIMILARITY

The experiments on the impact of using PS and/or MP similarity reveal that using either PS or MP similarity perform equally well, reported in Table 4. PS similarity, using PathSim, captures similarity from the connected metapaths, whereas MP similarity, using node embeddings, learns the network representation from multiple random walks. We observe from Table 4 that combining PS and MP similarity increases explainability while keeping precision performance. We also observe that the results using either only PS or only MP similarity are very close, so we conclude that either is good enough as long as the right metapaths for each approach are chosen (as explained in the previous paragraph).

TABLE 4. Impact of using PS and/or MP similarity.

Model	MovieLens		Amazon		Yelp	
	Prec.	MEP	Prec.	MEP	Prec	MEP
PS sim.	0.448	0.248	0.157	**0.315**	0.309	0.451
MP sim.	0.452	0.296	0.150	0.310	0.291	0.443
Both	**0.456**	**0.301**	**0.159**	0.313	**0.310**	**0.456**

4) IMPACT OF THE EXPLAINABILITY CONSTRAINT

We present the effect of the β parameter that controls explainability in equation (5). In order to present the overall effect of the β parameter on the precision and explainability metrics and to allow comparisons with other analysis described in the next sections, the exploration is executed using the test set. We used PS similarity and the same range of the β values used in parameter tuning for the analysis. Fig. 4 presents how the performance of MP4Rec is affected by the β value in terms of providing explainable and accurate recommendations. The results of the analysis revealed that when the explainability constraint is used ($\beta = 0.25$), the precision decreases slightly, whereas the explainability score increases sharply compared to when the explainability constraint is not used ($\beta = 0.0$), for all three datasets.

We further analyse the effect of the distance term ($\|\mathbf{u}_u - \mathbf{v}_j\|_2^2$) in accuracy and explainability of the recommended items, i.e., by setting the distance terms to 1.0. The results (Fig. 5) show that when the distance terms are not used ($\beta = 0.25$ w/out dist.), the precision remains nearly equal to its value when explainability is not used ($\beta = 0.0$) and the increase in explainability is relatively small. For example, in MovieLens, the precision values are 0.476, 0.448, 0.473 and explainability scores are 0.179, 0.248 and 0.192 for $\beta = 0.0$, $\beta = 0.25$ (with dist.) and $\beta = 0.25$ (w/out dist.). According to these observations, using the distance

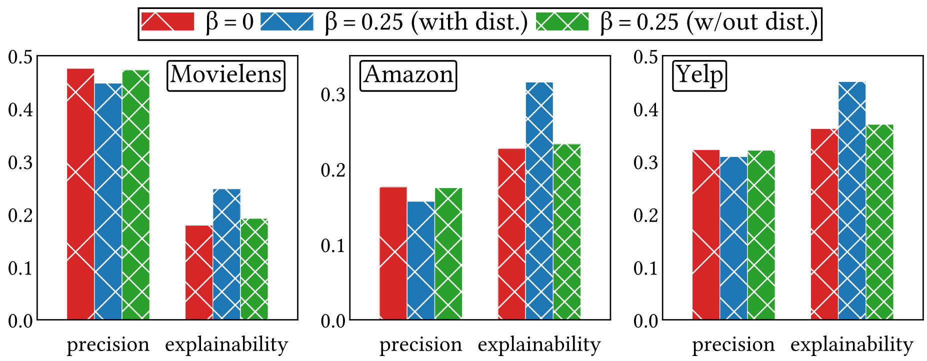

FIGURE 5. Impact of explainability constraint: When explainability constraint is not used ($\beta = 0.0$) or when it is used with ($\beta = 0.25$ with dist.) or without distance term ($\beta = 0.25$ w/out dist.).

TABLE 5. Comparison of algorithms on three datasets in terms of Precision@10, Recall@10 and MEP@10. "*" marks the best performance from the baselines and "#" marks the statistically significant improvements of MP4Rec over them (two-tailed t-test at 0.05).

Model	MovieLens			Amazon			Yelp		
	Prec.	Recall	MEP	Prec.	Recall	MEP	Prec	Recall	MEP
UserKNN [66]	0.456	0.215	0.197	0.135	0.207	0.188	0.263	0.239	0.339
PR [67]	0.400	0.188	0.144	0.114	0.175	0.173	0.169	0.153	0.356
BPR [63]	0.447	0.211	0.163	0.128	0.197	0.191	0.228	0.207	0.345
NeuMF [49]	0.441	0.206	0.162	0.147	0.227	0.189	0.212	0.193	0.239
MCRec [5]	0.301	0.139	0.167	0.059	0.081	0.087	0.250	0.228	0.353
PathSim [11]	0.453	0.214	0.210*	0.137	0.210	0.198	0.267	0.242	0.341
NeuACF [4]	0.475	0.227	0.153	0.153	0.236	0.183	0.301	0.276	0.307
KGAT [19]	**0.488***	**0.228***	0.193	0.162*	0.254*	0.210*	0.310*	0.284*	0.390*
MP4Rec (without expl., $\beta = 0$)	0.483	**0.228**	0.197	**0.176**	**0.269**	0.227#	**0.321#**	**0.292#**	0.364
MP4Rec, (with expl., $\beta = 0.25$)	0.456	0.216	**0.301#**	0.159	0.244	**0.313#**	0.310	0.282	**0.456#**

constraint is effective when explainability is considered to be as important as accuracy. However, for different objectives, the importance of explainability can be relaxed and the distance constraint can be removed from the equation. In our evaluation section, unless stated otherwise, we keep the distance term in the loss function following equation (5).

5) COMPARISON WITH OTHER METHODS

We compare MP4Rec with three algorithmic families: (i) the collaborative filtering (CF) based methods, which utilise only the user-item interaction information: UserKNN [66], Pairwise Ranking (PR) [67], BPR [63], and NeuMF [49],[4] (ii) HIN-based methods, which utilise additional information (categories, brands, cities, etc.): MCRec [5], PathSim [11],[5] and NeuACF [4] and (iii) a Knowledge-graph based method, namely KGAT [19] which models the knowledge-graph in an end-to-end fashion and uses embedding and attention mechanisms to make recommendations. When the same datasets are used as the comparison partners, the parameters reported in the original papers are used, otherwise, they are tuned to get the best results for these methods.

The experimental results of MP4Rec and the comparison partners on three datasets are reported in Table 5. All reported results are tested using two-tailed t-test at the 0.05 level. Among the CF-based methods, the UserKNN is very robust in both providing accurate and explainable recommendations. Among the HIN-based methods, NeuACF and PathSim algorithms outperform most of the CF methods in terms of accuracy, because they exploit the additional heterogeneous information context such as metapaths. We attribute the poor performance of MCRec to reproducibility problems we had; i.e. its public code does not include code for the node embedding creation, as also noted in [68]. KGAT [19], which is a knowledge graph-based method, outperforms all the other baseline methods. Our proposed method, MP4Rec, is consistently better than or equal to all the comparison partners on all three datasets. MP4Rec without explainability ($\beta = 0$) has the best precision/recall performance, except precision on MovieLens (which is only 0.005 less than the top-performing baseline method). This confirms the importance of choosing the right metapaths and the usage of attention mechanism. MP4Rec ($\beta = 0.25$) is particularly good at making explainable recommendations, such that the explainability increases substantially, i.e., around 0.10, while keeping the precision and recall performance. We attribute this behavior to the incorporation of explainability in the objective function. This shows that our proposed algorithm can outperform the state-of-the-art in both accuracy and explainability, being also flexible so that it can optimize for either target by properly choosing the right parameter.

[4]We run NeuMF without pre-training the GMF component

[5]We adjust the PathSim to top-N recommendation task by using the acquired similarity matrix in UserKNN

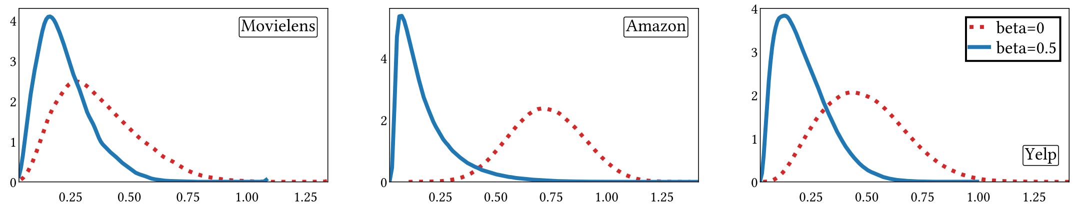

FIGURE 6. Explainability effect in the three datasets when using the user/item distance in the loss function. With explainability (beta=0.5) the distance between the user and item latent factors is being significantly reduced in all datasets.

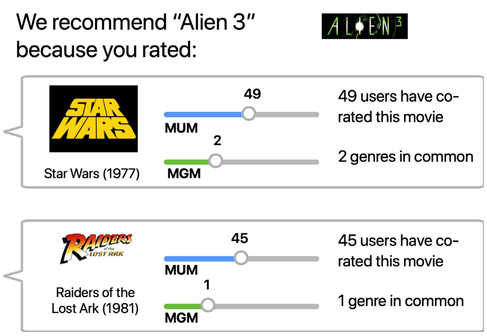

FIGURE 7. MUM, MGM based explanations for u5 in MovieLens. The sliders rank movies based on the number of metapaths that support the explanation.

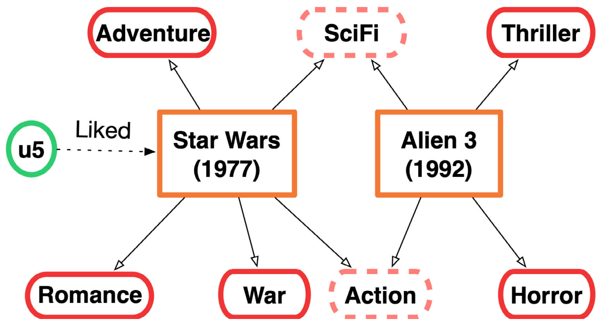

FIGURE 8. MGM-based explanation for user 5 (u5).

6) QUALITATIVE ANALYSIS ON EXPLAINABILITY

Fig. 6 shows the explainability effect on all three datasets in terms of the distance between the user and item latent factors. We trained MP4Rec without explainability ($\beta = 0$) and with explainability ($\beta = 0.5$) and we measured the cosine distance between the user and the item latent factors, i.e., $cosine\_distance = 1 - cosine\_similarity$. Cosine similarity lies between $[-1, 1]$, thus cosine distance lies within $[0, 2]$, with 0 showing similarity (smallest distance). As shown in Fig. 6, for all three datasets, when $\beta = 0.5$ (with explainability) the cosine distance between user and item latent factors moves very close to zero, which proves that the items are more explainable. This difference is more evident in Amazon (Fig. 6c), where initially the items are far away from the user, while in MovieLens even without explainability the cosine distance is not large. This can be justified by the density of MovieLens compared to the sparsity of Amazon.

We visualize the metapath based explanations that we provide to a sample user 5(u5), for top-1 movie recommendation, in Fig. 7. As shown, the recommended movie is Alien 3 (1992) because of two other movies: 1. Star Wars (1977), 2. Raiders of the Lost Arc (1981). In Fig. 7, we see item-style and feature-style explanations. With the item-style explanation we show movies that support our recommendation, and with the feature-style explanation, we depict why the movies that are used for explaining a recommendation are ranked in such a way. Thus, Star Wars is ranked as the most influential movie because out of the 100 users that rated Alien3, 49 users have co-rated it with the recommended one (i.e. we found 49 instances of metapath MUM) and both movies have 2 genres in common (i.e, we found 2 instances of metapath MGM) which are *sci-fi* and *action* of the 7 different combined genres of both movies, as shown in Fig. 8. Our explanation style extends the one proposed in MCRec [5], by providing an additional explicit ranking among movies used for an explanation based on the number of instances of each metapath. MP4Rec can be considered as one step forward for explaining a neural network-based recommendation method.

V. CONCLUSION

We propose MP4Rec, an algorithm which provides both accurate and explainable item recommendations using HINs. Experiments on three well-known datasets show that MP4Rec is consistently better than the state-of-the-art. We attribute the accuracy and explainability performance to the right choices on metapaths, the usage of attention mechanism and to the incorporation of explainability in the objective function. We consider MP4Rec as one step forward for explaining a neural networks-based recommendation method.

In the future, we want to use a node-level attention mechanism to better exploit the node attributes of a graph. We are aware that calculating similarity for all metapaths and/or node embeddings may introduce scalability issues, although there are many well-known techniques to deal with this problem efficiently [69]. In the future, we also want to use embeddings as the direct input to the system, instead of only using their similarities. Also, considering the fact that there are a large number of metapaths and their possible combinations, we are aware of the importance of finding an efficient way of choosing the right path. We conjecture that joint learning approaches can be used for this purpose. In the future, we also want to work in this direction.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for the helpful feedback.

REFERENCES

[1] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *Proc. 12th Int. Conf. Inf. Knowl. Manage.*, 2003, pp. 1019–1031.

[2] Y. Dong, N. V. Chawla, and A. Swami, "Metapath2vec: Scalable representation learning for heterogeneous networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 135–144.

[3] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, "Network embedding as matrix factorization: Unifying DeepWalk, LINE, PTE, and node2vec," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, 2018, pp. 459–467.

[4] X. Han, C. Shi, S. Wang, P. S. Yu, and L. Song, "Aspect-level deep collaborative filtering via heterogeneous information networks," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3393–3399.

[5] B. Hu, C. Shi, W. X. Zhao, and P. S. Yu, "Leveraging meta-path based context for Top- n recommendation with a neural co-attention model," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1531–1540.

[6] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.

[7] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, "Heterogeneous information network embedding for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 357–370, Feb. 2019.

[8] G. Peake and J. Wang, "Explanation mining: Post hoc interpretability of latent factor models for recommendation systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2060–2069.

[9] M. Bilgic and R. J. Mooney, "Explaining recommendations: Satisfaction vs. Promotion," *Beyond Personalization*, vol. 5, p. 153, Dec. 2005.

[10] J. L. Herlocker, J. A. Konstan, and J. Riedl, "Explaining collaborative filtering recommendations," in *Proc. ACM Conf. Comput. supported Cooperat. work - CSCW*, 2000, pp. 241–250.

[11] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "PathSim: Meta path-based top-K similarity search in heterogeneous information networks," *Proc. VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, Aug. 2011.

[12] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han, "Personalized entity recommendation: A heterogeneous information network approach," in *Proc. 7th ACM Int. Conf. Web Search Data Mining*, 2014, pp. 283–292.

[13] H. Zhao, Q. Yao, J. Li, Y. Song, and D. L. Lee, "Meta-graph based recommendation fusion over heterogeneous information networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 635–644.

[14] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, and Q. He, "A survey on knowledge graph-based recommender systems," 2020, *arXiv:2003.00911*. [Online]. Available: http://arxiv.org/abs/2003.00911

[15] X. Cao, C. Shi, Y. Zheng, J. Ding, X. Li, and B. Wu, "A heterogeneous information network method for entity set expansion in knowledge graph," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2018, pp. 288–299.

[16] J. Huang, W. X. Zhao, H. Dou, J.-R. Wen, and E. Y. Chang, "Improving sequential recommendation with knowledge-enhanced memory networks," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 505–514.

[17] Z. Sun, J. Yang, J. Zhang, A. Bozzon, L.-K. Huang, and C. Xu, "Recurrent knowledge graph embedding for effective recommendation," in *Proc. 12th ACM Conf. Recommender Syst.*, Sep. 2018, pp. 297–305.

[18] Y. Cao, X. Wang, X. He, Z. Hu, and T.-S. Chua, "Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences," in *Proc. World Wide Web Conf.*, May 2019, pp. 151–161.

[19] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge graph attention network for recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 950–958.

[20] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *Proc. World Wide Web Conf.*, 2019, p. 3307, doi: 10.1145/3308558.3313417.

[21] X. Tang, T. Wang, H. Yang, and H. Song, "AKUPM: Attention-enhanced knowledge-aware user preference model for recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1891–1899.

[22] 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.*, Oct. 2018, pp. 417–426.

[23] Q. Zhu, X. Zhou, J. Wu, J. Tan, and L. Guo, "A knowledge-aware attentional reasoning network for recommendation," in *Proc. Conf. Artif. Intell. (AAAI)*, 2020, pp. 6999–7006.

[24] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 58, no. 7, pp. 1019–1031, 2007.

[25] L. Adamic and E. Adar, "How to search a social network," *Social Netw.*, vol. 27, no. 3, pp. 187–203, Jul. 2005.

[26] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, Mar. 1953.

[27] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Comput. Netw. ISDN Syst.*, vol. 30, nos. 1–7, pp. 107–117, Apr. 1998.

[28] G. Jeh and J. Widom, "SimRank: A measure of structural-context similarity," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2002, pp. 538–543.

[29] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu, "Automatic multimedia cross-modal correlation discovery," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2004, pp. 653–658.

[30] S. Lee, S. Park, M. Kahng, and S.-G. Lee, "PathRank: Ranking nodes on a heterogeneous graph for flexible hybrid recommender systems," *Expert Syst. Appl.*, vol. 40, no. 2, pp. 684–697, Feb. 2013.

[31] S. Cao, W. Lu, and Q. Xu, "GraRep: Learning graph representations with global structural information," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 891–900.

[32] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 701–710.

[33] C. Yang, Y. Xiao, Y. Zhang, Y. Sun, and J. Han, "Heterogeneous network representation learning: Survey, benchmark, evaluation, and beyond," 2020, *arXiv:2004.00216*. [Online]. Available: http://arxiv.org/abs/2004.00216

[34] S. Chang, W. Han, J. Tang, G.-J. Qi, C. C. Aggarwal, and T. S. Huang, "Heterogeneous network embedding via deep architectures," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 119–128.

[35] H. Gui, J. Liu, F. Tao, M. Jiang, B. Norick, and J. Han, "Large-scale embedding learning in heterogeneous event data," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 907–912.

[36] T.-Y. Fu, W.-C. Lee, and Z. Lei, "HIN2 Vec: Explore meta-paths in heterogeneous information networks for representation learning," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 1797–1806.

[37] Y. Shi, Q. Zhu, F. Guo, C. Zhang, and J. Han, "Easing embedding learning by comprehensive transcription of heterogeneous information networks," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2190–2199.

[38] C. Zhang, A. Swami, and N. V. Chawla, "SHNE: Representation learning for semantic-associated heterogeneous networks," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, Jan. 2019, pp. 690–698.

[39] X. Wang, Y. Zhang, and C. Shi, "Hyperbolic heterogeneous information network embedding," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 5337–5344.

[40] L. Xu, X. Wei, J. Cao, and P. S. Yu, "Embedding of embedding (EOE): Joint embedding for coupled heterogeneous networks," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, 2017, pp. 741–749.

[41] P. Pham and P. Do, "W-MetaPath2 Vec: The topic-driven meta-path-based model for large-scaled content-based heterogeneous information network representation learning," *Expert Syst. Appl.*, vol. 123, pp. 328–344, Jun. 2019.

[42] C. Liu, J. Yu, Y. Liu, M. Yu, R. Yu, X. Li, M. Zhao, T. Xu, H. Liu, and L. Xu, "RL4HIN: Representation learning for heterogeneous information networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[43] X. Yu, X. Ren, Q. Gu, Y. Sun, and J. Han, "Collaborative filtering with entity similarity regularization in heterogeneous information networks," in *Proc. IJCAI*, vol. 27, 2013, pp. 1–5.

[44] C. Luo, W. Pang, Z. Wang, and C. Lin, "Hete-CF: Social-based collaborative filtering recommendation using heterogeneous relations," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2014, pp. 917–922.
[45] X. Yu, X. Ren, Y. Sun, B. Sturt, U. Khandelwal, Q. Gu, B. Norick, and J. Han, "Recommendation in heterogeneous information networks with implicit user feedback," in *Proc. 7th ACM Conf. Recommender Syst.*, 2013, pp. 347–350.
[46] Y. Wang, Y. Xia, S. Tang, F. Wu, and Y. Zhuang, "Flickr group recommendation with auxiliary information in heterogeneous information networks," *Multimedia Syst.*, vol. 23, no. 6, pp. 703–712, 2017.
[47] J. Zheng, J. Liu, C. Shi, F. Zhuang, J. Li, and B. Wu, "Recommendation in heterogeneous information network via dual similarity regularization," *Int. J. Data Sci. Analytics*, vol. 3, no. 1, pp. 35–48, Feb. 2017.
[48] C. Shi, Z. Zhang, P. Luo, P. S. Yu, Y. Yue, and B. Wu, "Semantic path based personalized recommendation on weighted heterogeneous information networks," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 453–462.
[49] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.
[50] X. Chen, Y. Zhang, and Z. Qin, "Dynamic explainable recommendation based on neural attentive models," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 53–60.
[51] Y. Zhang and X. Chen, "Explainable recommendation: A survey and new perspectives," *Found. Trends Inf. Retr.*, vol. 14, no. 1, pp. 1–101, 2020.
[52] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2014, pp. 83–92.
[53] X. Chen, Z. Qin, Y. Zhang, and T. Xu, "Learning to rank features for recommendation over multiple categories," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2016, pp. 305–314.
[54] S. Seo, J. Huang, H. Yang, and Y. Liu, "Interpretable convolutional neural networks with dual local and global attention for review rating prediction," in *Proc. 11th ACM Conf. Recommender Syst.*, Aug. 2017, pp. 297–305.
[55] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *Proc. World Wide Web Conf. World Wide Web*, 2018, pp. 1583–1592.
[56] Y. Tay, A. T. Luu, and S. C. Hui, "Multi-pointer co-attention networks for recommendation," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2309–2318.
[57] X. Wang, D. Wang, and C. Xu, "Explainable reasoning over knowledge graphs for recommendation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Aug. 2019, pp. 5329–5336.
[58] Q. Ai, V. Azizi, X. Chen, and Y. Zhang, "Learning heterogeneous knowledge base embeddings for explainable recommendation," *Algorithms*, vol. 11, no. 9, p. 137, Sep. 2018.
[59] X. Wang, X. He, F. Feng, L. Nie, and T.-S. Chua, "TEM: Tree-enhanced embedding model for explainable recommendation," in *Proc. World Wide Web Conf.*, 2018, pp. 1543–1552.
[60] W. Ma, M. Zhang, Y. Cao, W. Jin, C. Wang, Y. Liu, S. Ma, and X. Ren, "Jointly learning explainable rules for recommendation with knowledge graph," in *Proc. World Wide Web Conf.*, 2019, pp. 1210–1221.
[61] X. Huang, Q. Fang, S. Qian, J. Sang, Y. Li, and C. Xu, "Explainable interaction-driven user modeling over knowledge graph for sequential recommendation," in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 548–556.
[62] Y. Zhang, X. Xu, H. Zhou, and Y. Zhang, "Distilling structured knowledge into embeddings for explainable and accurate recommendation," in *Proc. 13th Int. Conf. Web Search Data Mining*, Jan. 2020, pp. 735–743.
[63] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proc. twenty-fifth Conf. Uncertainty Artif. Intell.*, 2009, pp. 1–5.
[64] F. Zhao and Y. Guo, "Improving top-n recommendation with heterogeneous loss," in *Proc. IJCAI*, 2016, pp. 2378–2384.
[65] B. Abdollahi and O. Nasraoui, "Using explainability for constrained matrix factorization," in *Proc. 11th ACM Conf. Recommender Syst.*, 2017, pp. 79–83.
[66] J. Herlocker, J. A. Konstan, and J. Riedl, "," *Inf. Retr.*, vol. 5, no. 4, pp. 287–310, 2002, doi: 10.1023/A:1020443909834.
[67] M. Jahrer and A. Töscher, "Collaborative filtering ensemble for ranking," in *Proc. 2011 Int. Conf.*, vol. 18, 2011, pp. 153–167.
[68] M. F. Dacrema, P. Cremonesi, and D. Jannach, "Are we really making much progress? A worrying analysis of recent neural recommendation approaches," in *Proc. 13th ACM Conf. Recommender Syst.*, Sep. 2019, pp. 101–109.
[69] S. Kim, H. Kim, and J.-K. Min, "An efficient parallel similarity matrix construction on MapReduce for collaborative filtering," *J. Supercomput.*, vol. 75, no. 1, pp. 123–141, Jan. 2019.

MAKBULE GULCIN OZSOY received the B.Sc., M.Sc., and Ph.D. degrees from the Department of Computer Engineering, Middle East Technical University (METU), Ankara, Turkey, in 2008, 2011, and 2016, respectively. She was a Visiting Student with the University of Calgary, Calgary, AB, Canada, between October 2014 and August 2015. She is currently a Postdoctoral Researcher with the Insight Centre for Data Analytics, University College Dublin (UCD), Dublin, Ireland. Her research interests include recommender systems, machine learning, and information retrieval.

DIARMUID O'REILLY-MORGAN received the B.A. degree in English and film studies from Trinity College Dublin and the M.Sc. degree in computer science from University College Dublin (UCD). Since 2019, he has been working as a Research Assistant/Engineer with the Insight Centre for Data Analytics, UCD. His research interests include the applicability of graph neural networks and federated learning to recommender systems.

PANAGIOTIS SYMEONIDIS received the B.Sc. degree in applied informatics and the M.Sc. degree in information systems from the University of Macedonia, Thessaloniki, in 1996 and 2004, respectively, and the Ph.D. degree in web mining and information retrieval for personalization from the Department of Informatics, Aristotle University of Thessaloniki, Greece, in 2008. He has been an Assistant Professor with the Faculty of Computer Science (scientific sector INF/01), Free University of Bozen-Bolzano, Italy, since November 2016. Before moving to Bolzano, he worked for eight years as an Adjunct Assistant Professor with the Department of Informatics, Aristotle University of Thessaloniki. He is a coauthor of three international books, one Greek book, six book chapters, 25 journal publications, and 38 conference/workshop publications. His published articles have received more than 2850 citations according to google scholar. Recently, he recognized from AMiner among the Most Influential Researchers in the last decade to the field of recommender systems. His research interests include web mining (usage mining, content mining, and graph mining), information retrieval, collaborative filtering, recommender systems, social media in web 2.0, and location-based social networks.

ELIAS Z. TRAGOS received the M.B.A. degree in techno-economics and the Ph.D. degree in wireless communications. He is currently a Research Project Manager with the Insight Centre for Data Analytics, UCD, Ireland. He has been actively involved in many EU and national research projects as a Researcher, the Technical Manager, and a Project Coordinator. He has been the Technical Coordinator of the SmartCities project RERUM. He has been extensively involved with the IERC, leading the Activity Chain 05 for Trusted IoT and co-leading the AC 03 for IoT Results Exploitation, contributing every year in drafting the IoT strategic roadmap in the IERC Cluster Book. His research interests include machine learning, distributed artificial intelligence, wireless and mobile communications, the Internet of Things, cognitive radios, network architectures, fog computing, security, and privacy.

NEIL HURLEY received the M.Sc. degree in mathematical science from University College Dublin (UCD), in 1988, and the Ph.D. degree, in 1995. In 1989, he joined the Hitachi Dublin Laboratory (a computer science research laboratory), Trinity College Dublin. He initially worked on knowledge-based environments for engineering design. A patent for the system was granted in U.S. and Europe. He became the Group Leader of the Parallel Computing Group, HDL, in 1995. During this period, he worked on the development of parallel simulation and optimization software, such as parallel finite element analysis and dynamic load balancing. In 1998, he was appointed as a Manager of HDL. He became a Lecturer with the Department of Computer Science, UCD, in 1999. He has established the Information Hiding Laboratory, in 2001, a research laboratory focused on technology for embedding information in digital content. His research interest includes parallel and distributed computing. He has won over €1 million in research funding from the Enterprise Ireland, the Science Foundation Ireland, EU, and industrial partners.

BARRY SMYTH received the D.Tech. degree (Hons.) from Robert Gordon University, U.K., in 2014. He holds the Digital Chair of Computer Science at University College Dublin (UCD). He was the Director of the Clarity Centre for Sensor Web Technologies, from 2008 to 2013, and has previously held the position of Head at the School of Computer Science and Informatics, UCD. He is currently the Director of the Insight Centre for Data Analytics. Since 1992, he has published over 400 peer-reviewed articles. His research has attracted more than 13 000 citations. He has an H-index of 58 and received more than 20 best paper awards for his work. His research interests include artificial intelligence and case-based reasoning, machine learning, recommender systems, user modeling, and personalization. His research interests extend beyond the laboratory and over years he has established a track-record for successfully translating his research into commercial opportunities and received numerous awards for his commercialization endeavour. He has been a Fellow of the European Coordinating Committee on Artificial Intelligence (ECCAI), since 2003, and has been a member of the Royal Irish Academy, since 2011. He is also a member of the Irish Times Trust. In 2014, he was named as the SFI Researcher of the Year. In 2006, he was a Finalist in the Ernst and Young Entrepreneur of the Year Award. He received the Irish Software Association's (ISA) Inaugural Award for his Outstanding Academic Achievement, in 2012. In 1999, he co-founded ChangingWorlds based on his personalization research and helped to grow the company to more than 150 employees before it was acquired by Amdocs Inc., in 2008. In 2008, he also co-founded HeyStaks based on his social search research. He has helped HeyStaks to raise more than '3.5m in venture capital funding to date and continues to advise the company on its technology and market strategy. He is actively involved in the Irish startup scene as an Advisor and an Investor and he serves on the boards of a number of local startups.

AONGHUS LAWLOR received the M.Sc. and Ph.D. degrees from University College Dublin (UCD). He is currently an Assistant Professor with UCD and a Principal Investigator with the Insight Centre for Data Analytics. He has long experience in the areas of machine learning, artificial intelligence, explainable AI, natural language processing, and recommender systems. He has been actively involved in several EU and national research projects in the above mentioned areas and has been the Co-Principal Investigator of very large industrial projects. He has also received several funded grants from the Enterprise Ireland and the Science Foundation Ireland, leading to patent applications.