Sound and Music Recommendation with Knowledge Graphs

Sergio Oramas*, Pompeu Fabra University
Vito Claudio Ostuni*, Pandora Media Inc.
Tommaso Di Noia, Polytechnic University of Bari
Xavier Serra, Pompeu Fabra University
Eugenio Di Sciascio, Polytechnic University of Bari

The Web has moved, slowly but steady, from a collection of documents towards a collection of structured data. Knowledge graphs have then emerged as a way of representing the knowledge encoded in such data as well as a tool to reason on them in order to extract new and implicit information. Knowledge graphs are currently used, e.g., to explain search results, to explore knowledge spaces, to semantically enrich textual documents or to feed knowledge intensive applications such as recommender systems. In this work we describe how to create and exploit a knowledge graph to supply a hybrid recommendation engine with information that builds on top of a collections of documents describing musical and sound items. Tags and textual descriptions are exploited to extract and link entities to external graphs such as WordNet and DBpedia which are in turn used to semantically enrich the initial data. By means of the knowledge graph we build, recommendations are computed using a feature combination hybrid approach. Two explicit graph feature mappings are formulated to obtain meaningful item feature representations able to catch the knowledge embedded in the graph. Those content features are further combined with additional collaborative information deriving from implicit user feedback. An extensive evaluation on historical data is performed over two different datasets. A dataset of sounds composed by tags, textual descriptions and user's download information gathered from Freesound.org, and a dataset of songs that mixes song textual descriptions with tags and user's listening habits extracted from Songfacts.com and Last.fm respectively. Results show significant improvements with respect to state of the art collaborative algorithms in both datasets. In addition, we show how the semantic expansion of the initial descriptions helps in achieving much better recommendation quality in terms of aggregated diversity and novelty.

CCS Concepts:•**Information systems** → **Recommender systems;** *Information extraction;* Ontologies;

Additional Key Words and Phrases: Knowledge graphs, recommender systems, music, entity linking, novelty, diversity

ACM Reference Format:
Sergio Oramas, Vito Claudio Ostuni, Tommaso Di Noia, Xavier Serra, and Eugenio Di Sciascio 2015. Sound and Music Recommendation with Knowledge Graphs. *ACM Trans. Intell. Syst. Technol.* 9, 4, Article 1 (October 2015), 21 pages.
DOI: 0000001.0000001

1. INTRODUCTION

Information overload in modern Web applications challenges users in their decision-making tasks. Recommender systems have emerged in the last years as fundamental tools in assisting users to find, in a personalized manner, what is relevant for

* These authors contributed equally to this work

Author's addresses: S. Oramas and X. Serra, Music Technology Group, University Pompeu Fabra, Barcelona, Spain; emails: {sergio.oramas, xavier.serra}@upf.edu; V. C. Ostuni, Pandora Media Inc., Oakland, CA, USA; email: {vostuni@pandora.com}; T. Di Noia and E. Di Sciascio, SisInf Lab, Polytechnic University of Bari, Bari, Italy; emails: {tommaso.dinoia, eugenio.disciascio}@poliba.it;

DOI: 0000001.0000001

them in overflowing knowledge spaces. Within the recommender systems arena, there are two main approaches for computing recommendations: collaborative filtering and contend-based ones. The most popular is collaborative filtering which provides recommendations to a user by considering the preferences of other users with similar tastes. Content-based systems recommend items sharing similar features to those a user has preferred in past[1]. Both approaches can be combined to build hybrid systems [Burke 2002]. When available, the usage of side information about items has proven to boost the performances of pure collaborative-filtering techniques [Ning and Karypis 2012]. Furthermore, many works in the past [Mobasher et al. 2004; Cantador et al. 2008; Semeraro et al. 2009; Anand et al. 2007] have raised the problem of introducing semantics in content item representations, usually using ontologies, to mitigate the issues of pure keyword- and attribute-based representations.

In this work we tackle the problem of computing sound and music recommendations leveraging both semantic content features extracted from textual descriptions and collaborative features from implicit user feedback. The approach we propose to recommend musical items consists mainly of two parts: (i) the enrichment of original data attached to items and linkage to knowledge graphs, (ii) the effective exploitation of the graph-based nature of such data for computing the recommendations.

The enrichment of data consists in using entity linking techniques for extracting semantic entities from item textual descriptions and linking them to external knowledge graphs such as WordNet [Miller 1995] and DBpedia [Bizer et al. 2009] for gathering additional knowledge. All those different information are eventually merged together and represented by means of a new knowledge graph. This latter is thus exploited together with collaborative information from implicit feedback for computing the recommendations. Two graph feature mappings are defined to leverage the new knowledge graph and obtain expressive feature representations. All different features are combined together in a feature combination hybrid schema [Burke 2002] and used to feed a content-based recommender. An extensive experimental evaluation was carried out on two different datasets –the one related to sounds, the other to songs– to evaluate the recommendation quality in terms of accuracy, novelty and aggregate diversity.

In this work, we deal with two slightly different problems in the music ecosystem. We address the songs recommendation problem and that of recommending sounds to users in online sound sharing platforms. The two tasks addresses two separate categories of users in the music domain: on the one hand, we have music consumers (songs and artists recommendation); on the other hand, we have music producers (sounds recommendation).

Music recommendation has received a lot of attention in the last decade [Knees and Schedl 2013]. As a matter of fact, the discovery of new songs and artists is a task that the music consumers of a Web radio or of a music store are naturally led to perform daily. Hence, helping them by recommending the best choices results in immediate impact also in industrial and commercial scenarios.

Differently from the previous case, recommendation of sounds has received scant attention even though it may be of interest in many scenarios of music creation. As an example, we may consider producers of electronic music that typically downloads and use sound samples. They might be interested in the recommendation of relevant sounds downloaded by users with similar tastes or similar (not equal) to those they previously used in their musical compositions. Likely, they are also looking not just for popular sounds as they want their production to be unique. To this end, we first cen-

[1] In this paper, when we mention "content" we mainly refer to textual and semantically enriched information associated to sounds and songs.

tered our study in Freesound[2], one of the most popular sites on the Web for sharing audio clips, accounting more than 4 million registered users and about 300,000 uploaded sounds, which are described in terms of textual descriptions and tags. In Freesound, different kind of users may be observed [Font et al. 2012] (e.g. music producers, composers, sound designers, soundscape enthusiasts), and also different types of sounds (e.g. sound samples, field recordings, soundscapes, loops). We have the intuition that collaborative features may help in the personalization of the recommendations, whilst the introduction of semantic features may lead to a better exploitation of less popular items. To evaluate this hypothesis, a dataset composed of sound descriptions and historical data about user's download behavior was collected.

To demonstrate the suitability of the proposed methodology for both types of musical users (producers and consumers), a music recommendation experiment was also performed. To this end, a dataset of songs which combines tags and textual descriptions with users' implicit feedback was created by aggregating information gathered from Songfacts[3] and Last.fm[4]. Songfacts is an online database that collects, stores and provides facts, stories and trivia about songs, whilst in Last.fm a detailed profile of each user's musical taste is built by recording details of the tracks the user listens to.

The evaluation performed on both datasets showed that the semantic expansion of the original data combined with user collaborative features allows the system to enhance recommendation quality especially in terms of aggregated diversity and novelty while keeping high performance in terms of accuracy.

Our main contributions in this work are summarized as follows:

— We define a novel method to enrich the description of musical and sound items with semantic information.
— We propose two different graph-embedding approaches to encode knowledge graph information into a linear feature representation.
— We present a methodology to recommend musical items combining semantic and collaborative features, that turns out in a high level of personalization of recommendations, in terms of prediction accuracy, catalog coverage and long tail recommendations.
— We tackle for the first time the problem of sound recommendation.

The reminder of the paper is structured as follows. The next section introduces the basic technologies used to build the knowledge graph at the basis of our recommendation system. Section 3 describes the problem and the semantic expansion applied to the initial data. Then, Section 4 defines the adopted recommendation approach while in Section 5 we explain the experimental evaluation and discusses the obtained results. Finally, Section 7 concludes the paper and points out for future lines of work.

2. BACKGROUND TECHNOLOGIES

In order to make the paper self-consistent, in this section we overview the basic notions and technologies we adopted to build the knowledge graph we exploit in our recommender system.

In 2012, Google announced its Knowledge Graph[5] as a new tool to improve the identification and retrieval of entities in return to a search query. Most of the knowledge encoded in Google Knowledge Graph actually came from Freebase. This was a crowdsourced effort to create a base of facts in all possible knowledge domains. Actually,

[2] http://freesound.org
[3] http://songfacts.com
[4] http://last.fm
[5] https://googleblog.blogspot.it/2012/05/introducing-knowledge-graph-things-not.html

over the years, we have assisted to the creation of knowledge graphs aimed at different targets such as WordNet[6] containing lexical information for the English language. In WordNet, words are grouped by *synsets*, which are sets of cognitive synonyms.

Alongside with the development of the above mentioned initiatives, inspired by the seminal paper by Tim Berners-Lee et al. [Berners-Lee et al. 2001], in the last decade, a set of technologies for the creation of the so called Semantic Web were developed. These technologies are the cornerstone in the development of the Linked Data initiative[7]: an effort to create, interconnect and publish semantic datasets. Among them, the most important is for sure DBpedia[8]. This encodes an important amount of the information available in Wikipedia as RDF triples and is freely available on the Web. If we think that each Wikipedia page corresponds to a DBpedia entity, it is easy to see how rich the knowledge available in the DBpedia graph is.

Knowledge graphs have become a powerful tool to represent knowledge in the form of a labelled directed graph and to give meaning (semantics) to textual information. One of the tasks in Natural Language Processing is indeed that of identifying relevant entities within a text. In case these entities are available as nodes in a knowledge graph this task is usually referred as Entity Linking (EL). The main task of an EL system is to disambiguate the mention of an entity belonging to a knowledge base within its textual context and eventually map entities to each mention within the analyzed text. Entity Linking tools usually execute a pipe of three main procedures [Shen et al. 2015]: *candidate entity generation*, *candidate entity ranking*, *unlinkable mention prediction*. Within the Linked Data technological stack, there are many tools available to perform entity linking from text [Oramas et al. 2016] which refer, e.g., to DBpedia or Freebase entities and most of them reach very good results in terms of number of entities identified within a text.

3. KNOWLEDGE ENRICHMENT VIA ENTITY LINKING

In order to add more semantics to the description of musical items, we exploit contextual information, i.e., tags and text descriptions, and then use this information to create a knowledge graph. Several approaches have been developed to enrich tags with semantics [Garcia-Silva et al. 2012]. We follow an ontology-based approach, enriching both tags and keywords extracted from textual descriptions by associating them with relevant entities defined in online semantic datasets. The first step in this direction is to link and disambiguate tags and keywords to Linked Data resources. For this purpose we adopted Babelfy, a state of the art tool for Entity Linking and Word Sense Disambiguation [Moro et al. 2014]. Babelfy maps words from a given text to entities in the BabelNet[9] knowledge base. BabelNet is a multilingual encyclopedic dictionary that mixes knowledge from WordNet and Wikipedia. Thus, for every mapped and disambiguated text fragment, Babelfy returns the related WordNet *synset*, and/or the related Wikipedia page (and its equivalent DBpedia resource).

To build our semantically enriched graph, the entity linking tool is firstly run on both tags and keywords of every item. Identified named entities are linked to DBpedia resources, whilst disambiguated words are linked to WordNet *synsets*. Every musical item is added to the graph, and connected with the words taken from its context that are identified as entities by Babelfy. Words are in turn connected with their corresponding URIs, whether they are a DBpedia resource or a WordNet *synset*. Subsequently, we use both WordNet and DBpedia to semantically expand the entities added

[6] http://wordnet.princeton.edu
[7] http://linkeddata.org
[8] http://dbpedia.org
[9] http://babelnet.org

to the graph after the entity linking phase. Each synset obained from the linking is further expanded considering other concepts in the WordNet hierarchy of sysnsets by following the hypernymy[10] relations. From the WordNet hierarchy we extract up to 2-hop hypernyms starting from the mapped synset. We empirically selected the maximum distance of two hops because we wanted to avoid too broad generalization of the original concept. For the same reason we discard those hypernyms farther less than six hops away from the root of the WordNet hierarchy. Regarding DBpedia, Babelfy returns directly the URI of the linked entity and a set of related Wikipedia categories. In Wikipedia, categories are used to organize the resources, and they help users to group articles of the same subject. This reflects in DBpedia as resources are related to categories through the property `dcterms:subject`[11]. Those categories are in turn organized in a taxonomy. In particular, more specific categories are related to more generic ones by means of the `skos:broader`[12] property. Thus, for each category found by Babelfy, all the direct broader categories were gathered and added to our knowledge graph. Similarly to what we did with WordNet, only one level of broader categories were considered to avoid too broad or unrelated categories.

To show an example of entity linking performed by Babelfy we use the sound `prac-snare2.wav`[13] from Freesound. The description associated to this sound is *"standard snare sample. lower/mid tuning on the head"* and tags are *drums*, *percussion*, *snare*. Babelfy was able to detect and link most of the entities. Just to describe a few of them, the word *sample* from the description was linked to the DBpedia entity `Sampling_(music)`, the tag *percussion* was mapped to the DBpedia entity `Rythm_section`, the tag *snare* was linked to the WordNet concept `snare_drum.n.01` and DBpedia entity `Snare_drum`. As shown in Figure 1, DBpedia entities and WordNet synsets are then further enriched with their related categories and hypernyms. Following the Linked Data principles[14], we reused classes and properties from external vocabularies. The final knowledge graph after the entity linking and expansion process contains four main classes: `wordnet:Synset`, `Entity`, `Tag` and `skos:Concept` and seven relations: `hasTag`, `hasKeyword`, `wordnet:synset_member`, `dcterms:relation`, `dcterms:subject`, `skos:broader` and `wordnet:hypernym`[15]. In particular, for the sounds recommendation dataset based on Freesound we further enriched the ontology originally developed in [Font and Oramas 2014] as also shown in the left hand side of Figure 1.

4. RECOMMENDATION APPROACH

As aforementioned, we adopted a hybrid recommendation approach to leverage both collaborative information coming from the user's community and content information coming from the knowledge graph. Following the taxonomy of hybrid recommender systems presented in [Burke 2002] we developed a hybrid feature combination recommender system. The particularity of such schema is that hybridization is not based on the combination of different recommendation components but instead on the combination of different data sources. Specifically, collaborative information is treated as additional features of the content feature space and a content-based technique is used over this augmented space. Therefore, we build feature item representations by considering the item graph-based descriptions represented in the knowledge graph and enrich

[10] Hypernymy models generalization relations between synsets.

[11] http://dublincore.org/documents/dcmi-terms/#elements-subject

[12] http://www.w3.org/2004/02/skos/core#broader

[13] http://www.freesound.org/people/TicTacShutUp/sounds/439/

[14] http://www.w3.org/DesignIssues/LinkedData.html

[15] All the prefix we use here are the ones available via the http://prefix.cc service

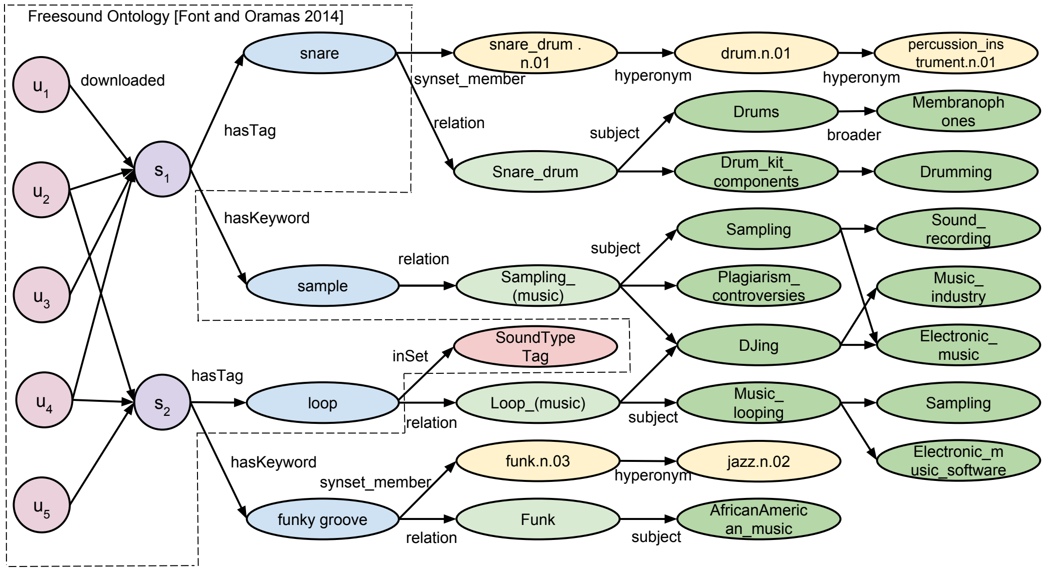

Fig. 1. Portion of the final knowledge graph enriched with WordNet and DBpedia

such feature vectors with collaborative features. Subsequently, we use such data to feed a content-based recommendation engine.

A common way of computing content-based recommendation is learning a function that, for each item in the system, predicts the relevance of such item for the user. The application of Machine Learning techniques is a typical way to accomplish such task. A *top-N* item recommendation problem in a standard content-based setting is mainly split into two different tasks: (i) given a collection of items for which past user's preferences are available, learn a regression or classification model to predict the relevance associated to unknown items; (ii) eventually, according to such scores, recommend the most relevant items to the user. Past user's preferences can be obtained from either explicit or implicit feedback. As for Freesound, we considered as an implicit positive feedback the "download data". The rationale behind our choice is that if a user downloads a sound it is reasonable to assume that she likes it even without an explicit rating, as the system lets users listen to sounds before downloading. Also the Last.fm dataset used in the experimental evaluation contains user song listening actions, which is another form of implicit feedback. Thus, in the following we will refer to the problem of computing recommendations from implicit feedback data. Following the notation introduced by [Rendle et al. 2009] for implicit feedback scenarios, let S be the matrix of implicit feedback, where $s_{ui} = 1$ if item i was downloaded from user u, 0 otherwise. Starting from S we define $I_u^+ = \{i \in I | s_{ui} = 1\}$ as the set of relevant items for u. The main problem with implicit feedback is that they reflect only positive user preferences. On the contrary, the system cannot infer anything about what the user dislikes. The unobserved data are a mixture of actually negative and missing values [Rendle et al. 2009], but the system does not have any information for discriminating between them. Then, learning a predictive model from such unary data becomes infeasible because there are no negative examples. To overcome this issue for each user we select a portion of unobserved items $I_u^- \subset (I \setminus I_u^+)$ to be used as negative data points in the training of the model. In [Ostuni et al. 2013], the authors show that choosing $|I_u^-| = 2 \cdot |I_u^+|$ does not affect accuracy results. The unobserved items are exactly the items that have to be ranked. The ultimate goal of the system is to rank in the *top-N* positions items likely to be relevant for the user.

Given the generic user u, let T_u be the training set for u defined as:

$$T_u = \{\langle x_i, s_{ui}\rangle | i \in (I_u^+ \cup I_u^-)\}$$

where $x_i \in \mathbb{R}^D$ is the feature vector associated to the item i and let TS_u be the test set defined as:

$$TS_u = \{\langle x_i, s_{ui}^*\rangle | i \in (I \setminus I_u^+)\}$$

The two tasks for the *top-N* recommendation problem, in our setting, consist then of: (i) learning a function $f_u : \mathbb{R}^D \to \mathbb{R}$ from the training data T_u which assigns a relevance score to the items in I; (ii) using such function to predict the unknown score s_{ui}^* in the test set TS_u, to rank them and recommend the *top-N*.

Given that items are represented as entities in a knowledge graph we are particularly interested in those machine learning methods that are appropriate for dealing with objects structured as graphs. There are two main ways of learning with structured objects. The first is to use *Kernel Methods* [Shawe-Taylor and Cristianini 2004]. Given two input objects i and j, defined in an input domain space D, the basic idea behind Kernel Methods is to construct a kernel function $k : D \times D \to \mathbb{R}$, that can be informally seen as a similarity measure between i and j. This function must satisfy $k(i, j) = \langle \phi(i), \phi(j)\rangle$ for all $i, j \in D$, where $\phi : D \to F$ is a mapping function to a inner product feature space F. Then, the classification or regression task involves linear convex methods based exclusively on inner products computed using the kernel in the embedding feature space. The alternative way is to explicitly compute the *explicit feature mapping* $\phi(i)$ and to directly use linear methods in the related space. By transforming the graph domain into a vector domain any traditional learning algorithm working on feature vectors can be applied.

While kernel methods have been widely applied to solve different tasks, their usage becomes prohibitive when dealing with large datasets. In addition, when the input data lie in a high-dimensional space, linear kernels have performances comparable to more complex non linear ones. Due to the high volume of users we deal with in our Freesound dataset (see Section 5), we focused on learning methods which are computationally efficient. For this reason we adopted the approach of computing the explicit feature mapping of the item graphs and use linear methods to learn the user model. Specifically, we use the Linear Support Vector Regression [Ho and Lin 2012] algorithm. Regarding the explicit feature mapping computation we define two sparse high-dimensional feature maps: the one based on entities, the other on paths that we call *entity-based item neighborhood mapping* and *path-based item neighborhood mapping*, respectively. In the following we formalize the computation of such graph embeddings.

4.1. Explicit feature mappings for graph-based Item Representations

Let us formally define the knowledge graph as a multi-relational graph $G = \{t \mid t \in E \times R \times E\}$, where E denotes the set of entities and R indicates the set of properties or relations, namely the edge labels. Moreover, we have $I \subseteq E$ since we consider items as a particular type of entities.

With E_i^h we denote the set of entities reachable in *at most h* hops from i according to the shortest path in G. For a generic item i we then define its h-hop neighborhood graph $G_i^h = \{t = (e_i, r_j, e_k) \mid t \in E_i^h \times R \times E_i^h\}$ that is the subgraph of G induced by the set of triples involving entities in E_i^h.

Figure 2 shows an example of 3-hop item neighborhood graph for item i, namely G_i^3. We see that, if we consider the shortest path, all the entities are no more than 3 hops

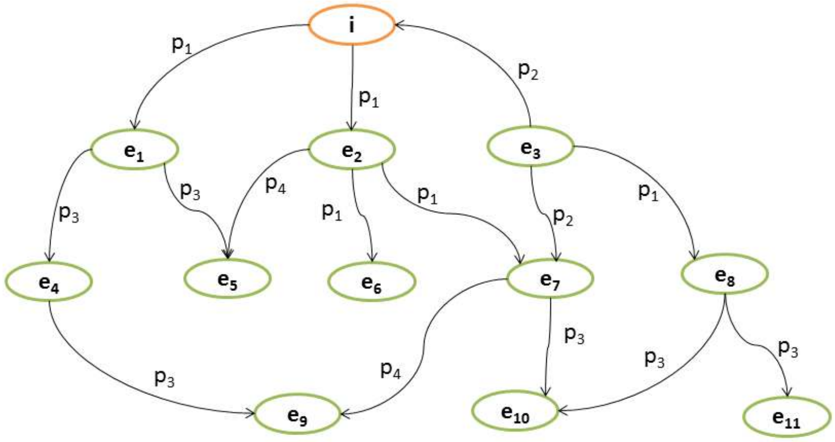

Fig. 2. An example of 3-hop item neighborhood graph for the item i.

distant from i. To clarify the definition and computation of G_i^h and E_i^h for item i, we show their computation with reference to the example shown in Figure 2:
$G_i^1 = \{(i, p_1, e_1), (i, p_1, e_2), (e_3, p_2, i)\}$
$G_i^2 = G_i^1 \bigcup \{(e_1, p_3, e_4), (e_1, p_3, e_5), (e_2, p_4, e_5), (e_2, p_1, e_6), (e_2, p_1, e_7), (e_3, p_2, e_7), (e_3, p_1, e_8)\}$
$G_i^3 = G_i^2 \bigcup \{(e_4, p_3, e_9), (e_7, p_4, e_9), (e_7, p_3, e_{10}), (e_8, p_3, e_{10}), (e_8, p_3, e_{11})\}$
$E_i^1 = \{e_1, e_2, e_3\}$
$E_i^2 = E_i^1 \bigcup \{e_4, e_5, e_6, e_7, e_8\}$
$E_i^3 = E_i^2 \bigcup \{e_9, e_{10}, e_{11}\}$
Starting from those item graph-based representations we define the two different feature mappings which are described in what follows.

Entity-based item neighborhood mapping. In this mapping each feature refers to an entity in E and the corresponding score represents the weight associated to that entity in G_i^h. The resulting feature vector $\phi_E(G_i^h)$ is:

$$\phi_E(G_i^h) = (w_{i,e_1}, w_{i,e_2}, ...w_{i,e_m}, ..., w_{i,e_t})$$

where the weight associated to the generic entity e_m is computed as follows:

$$w_{i,e_m} = \sum_{l=1}^{h} \alpha_l \cdot c_{l,e_m}$$

with

$$\alpha_l = \frac{1}{1 + \log(l)}$$

and

$$c_{l,e_m} = |\{(e_n, p, e_m) \mid e_n \in \widehat{E}_i^{l-1} \ \wedge \ e_m \in \widehat{E}_i^{l}\} \bigcup \{(e_m, p, e_n) \mid e_m \in \widehat{E}_i^{l} \ \wedge \ e_n \in \widehat{E}_i^{l-1}\}|$$

where $\widehat{E}_i^l = E_i^l \setminus E_i^{l-1}$ is the set of entities *exactly* l hops far from i.
In particular, c_{l,e_m} corresponds to the number of triples connecting e_m to entities in the previous hop $(l-1)$, whether e_m appears either as subject or object of the triple. In other words, c_{l,e_m} can be seen as the *occurrence* of the entity e_m in the item neighborhood at distance l. The more the entity e_m is connected to neighboring entities of i, the more it

is descriptive of i. α_l can be seen as a decay factor depending on the distance l from the item i, whose aim is to incrementally penalize farther entities from the item. It allows us to take into account the *locality* of those entities in the graph neighborhood. The closer an entity e_m to the item i, the stronger its relatedness to it. We use a logarithmic decay. Indeed, the discount factor can also be parametrized defining a specific weight for each hop. In such case, an optimal combination of weights can be found.

With reference to example showed in Figure 2, the c_{l,e_m} values are computed as follows: $c_{1,e_1} = 1$, $c_{1,e_2} = 1$, $c_{1,e_3} = 1$, $c_{2,e_4} = 1$, $c_{2,e_5} = 2$, $c_{2,e_6} = 1$, $c_{2,e_7} = 2$, $c_{2,e_8} = 1$, $c_{3,e_9} = 2$, $c_{3,e_{10}} = 2$, $c_{3,e_{11}} = 1$. All the others are zero. The presented graph embedding is an adaptation of the one presented in [Ostuni et al. 2014], in this work we use a logarithmic discount factor instead of a parametric one.

Path-based item neighborhood mapping. Differently from the previous case, in this mapping we represent a feature as a sequence of nodes in G. Given two entities e_1 and e_n, we consider the sequence of nodes $e_1 \cdot e_2 \cdot \ldots \cdot e_{n-1} \cdot e_n$ met while traversing the graph to go from e_1 to e_n and we refer to such sequence as *path*. In this mapping, a feature is then represented by a path. In particular, in this mapping each feature refers to several variants of paths rooted in the item node. We first collect all the paths rooted in i which can be indicated as sequence of entities $i \cdot e_1 \cdot e_2 \cdot \ldots \cdot e_{n-1} \cdot e_n$. Then, starting from those paths we define various features considering sub-paths of the original paths. Specifically we form sub-paths composed by only those entities progressively farther from the item. Considering the path given above we build the following features: $e_1 \cdot e_2 \cdot \ldots \cdot e_{n-1} \cdot e_n$, $e_2 \cdot \ldots \cdot e_{n-1} \cdot e_n$, ..., $e_{n-1} \cdot e_n$, e_n. The rationale behind this choice is that it allows to explicitly represent substructures shared between items with no overlapping in their immediate neighborhoods but somehow connected at further distance. Items connected to the same entities have same common structures because both closer and further entities are shared. Items connected to different entities which are however linked directly or at a farther distance to same entities share less or none sub-paths depending on how much far the common entities are, if any.

More formally, let P_i be the set of paths rooted in i and P_i^* be the list of all possible sub-paths extracted from them. We use $p_m(i)$ and $p_m^*(i)$ to refer to the $m-th$ elements in P_i and P_i^*, respectively. Then, the feature mapping for item i is:

$$\phi_P(G_i^h) = (w_{i,p_1^*}, w_{i,p_2^*}, ...w_{i,p_m^*}, ..., w_{i,p_t^*})$$

where each w_{i,p_m^*} is computed as:

$$w_{i,p_m^*} = \frac{\#p_m^*(i)}{|p_m| - |p_m^*|}$$

where $|p_m|$ indicates the length of path p_m and $\#p_m^*(i)$ the occurrence of $p_m^*(i)$ in P_i^*. The denominator is a discounting factor which takes into account the difference between the original path p_m and its sub-path p_m^*. The shorter the sub-path the more the discount because it contains entities farther from the item.
With respect to item i we have:
$P_i = \{i{\cdot}e_1{\cdot}e_4{\cdot}e_9,\ i{\cdot}e_1{\cdot}e_5,\ i{\cdot}e_2{\cdot}e_6,\ i{\cdot}e_2{\cdot}e_7{\cdot}e_9,\ i{\cdot}e_2{\cdot}e_7{\cdot}e_{10},\ i{\cdot}e_3{\cdot}e_7{\cdot}e_{10},\ i{\cdot}e_3{\cdot}e_8{\cdot}e_{10},\ i{\cdot}e_3{\cdot}e_8{\cdot}e_{11}\}$
$P_i^* = [e_1 \cdot e_4 \cdot e_9,\ e_4 \cdot e_9,\ e_9,\ e_1 \cdot e_5,\ e_5,\ e_2 \cdot e_6,\ e_6,\ e_2 \cdot e_7 \cdot e_{10},\ e_7 \cdot e_{10},\ e_{10},\ e_2 \cdot e_7 \cdot e_9,\ e_7 \cdot e_9,\ e_9,\ e_3 \cdot e_7 \cdot e_{10},\ e_7 \cdot e_{10},\ e_{10},\ e_3 \cdot e_8 \cdot e_{10},\ e_8 \cdot e_{10},\ e_{10},\ e_3 \cdot e_7 \cdot e_{11},\ e_7 \cdot e_{11},\ e_{11}]$

4.2. Feature Combination

Each final feature vector x_i is obtained by concatenating a vector of collaborative features $\phi_{col}(i)$ to the item neighborhood mapping vector $\phi(G_i^h)$. Collaborative features are simply added by encoding in the feature vector those users who downloaded that item. The collaborative feature vector regarding the generic item is then:

$$\phi_{col}(i) = (w_{i,u_1}, w_{i,u_2}, ..., w_{i,u_1})$$

where $w_{i,u_1} = 1$ if user u_1 downloaded item i.

Although more sophisticated and advanced methods can be used for feature combination [Beliakov et al. 2015], our experimental evaluation (see Section 5) shows the effectiveness of our choice.

5. EXPERIMENTAL EVALUATION

For the evaluation of our approach we adopted the `All Unrated Items` methodology presented in [Steck 2013]. It consists in creating a *top-N* recommendation list for each user by predicting a score for every item not rated by that particular user, whether the item appears in the user test set or not. Then, performance metrics are computed comparing recommendation lists with test data. The evaluation has been carried out using the holdout method consisting in splitting the data in two disjoint sets: the one for training and the other for testing. We used 80% of user downloads for building the training set T and remaining 20% as test data for measuring recommendation accuracy. We repeated the procedure three times by randomly drawing new training/test sets in each round and averaged the results.

For measuring recommendation accuracy we adopted the following standard performance metrics: Precision and Recall. Precision@N (P@N) is computed as the fraction of *top-N* recommended items appearing in the test set, while Recall@N (R@N) is computed as the ratio of *top-N* recommended items appearing in the test set to the number of items in the test set. Note that in such implicit feedback setting all items in the test set are relevant. In addition to the standard precision and recall metrics we also measure the Mean Reciprocal Rank (MRR) which measure the quality of the highest ranked recommendations. For each user recommendation list the Reciprocal Rank (RR) measures how early in the list is positioned the first relevant recommendation.

As pointed out by [McNee et al. 2006], the most accurate recommendations according to the standard metrics are sometimes not the recommendations that are most useful to users. In order to assess the utility of a recommender system, it is extremely important to evaluate also its capacity to suggest items that users would not readily discover for themselves, that is its ability to generate novel and unexpected results. The *Entropy-Based Novelty* (*EBN*) [Bellogín et al. 2010] expresses the ability of a recommender system to suggest less popular items, i.e. items not known by a wide number of users. In particular, for each user's recommendation list L_u, the novelty is computed as:

$$EBN_u@N = -\sum_{i \in L_u} p_i \cdot \log_2 p_i$$

where:

$$p_i = \frac{|\{s_{ui} = 1 | u \in U\}|}{|U|}$$

Particularly, p_i is the ratio of users who downloaded item *i*. The lower $EBN_u@N$, the better the novelty.

Another important quality of the system is aggregate diversity. In our work we adopt the *diversity-in-top-N* metric presented in [Adomavicius and Kwon 2012] that measures the distinct items recommended across all users. In particular we compute its normalized version with respect to the size of the item catalog. For brevity we refer to

Table I. Datasets Overview

dataset	items	avg. tags	avg. keywords	resources	synsets	categories
Freesound	21,552	6.44	11.36	16,407	20,034	54,419
Last.fm	8,640	42.09	77.33	46,109	27,708	96,942

Number of tags and keywords identified by Babelfy averaged by item, plus total number of distinct DBpedia resources, WordNet synsets and Wikipedia categories.

it as $ADiv@N$ and we compute it as follows:

$$ADiv@N = \frac{|\bigcup_u L_u|}{|I|}$$

This metric is an indicator of the level of personalization provided by a recommender system. Low values of aggregated diversity indicate that all users are being recommended almost the same few items. This corresponds to a low level of personalization of the system. Instead, high values mean that users receive very different recommendations which can be indirectly seen as a high level of personalization of the system.

All the reported metrics, besides aggregated diversity, are computed for each single user and eventually averaged.

5.1. Datasets Description

Freesound Dataset. We evaluated our approach on historical data about sound downloads collected from February 2005 to October 2013. The initial dump consisted in 3,275,092 users, 183,246 sounds and 48,636,182 downloads. However, for the purpose of our experimentation, we selected a subset of sounds that fulfilled some criteria. We selected those sound with at least two tags classified in the Freesound Ontology [Font and Oramas 2014]. After that we filtered out all sounds with less than 10 downloads to reduce the sparsity of the implicit feedback matrix and have a fairer comparison with pure collaborative filtering methods. After some further data cleansing, the final dataset consisted in 20,000 users, 21,552 items and 2,117,698 downloads[16]. The sparsity of the implicit feedback matrix was 99.51%. Statistics on the enriched knowledge graph of the final dataset are shown in Table I.

Last.fm Dataset. To recreate most of the conditions of the Freesound dataset in a typical music recommendation scenario, a new dataset is created combining user's implicit feedback, tags and textual descriptions of songs. This dataset combines a corpus of user's listening habits and song-related tags coming from Last.fm[17] [Vigliensoni and Fujinaga 2014], with a corpus of textual descriptions about songs obtained from Songfacts.com[18] [Sordo et al. 2015]. The former is an implicit feedback dataset consisting of user-song listening data, indicating the frequency a user listened to a song. For every user in the corpus we chose the users' average listening count as a threshold to identify the relevant songs for each user. From Last.fm, we only selected for our dataset user-song relations with a number of listens above each user's threshold. Moreover, only those songs that were relevant to at least 10 users, and users with at least 50 relevant songs were added to the dataset. The final dataset consisted in 5,199 users, 8,640 songs and 751,531 relations between users and songs. The sparsity of the implicit feedback matrix was 98.33%. This collaborative information was complemented with the list of top tags of every song provided by the Last.fm API, and a textual description of each

[16] A dump of the datasets is available at http://mtg.upf.edu/download/datasets/knowledge-graph-rec

[17] http://last.fm

[18] http://songfacts.com

song coming from Songfacts.com. Information about the enriched knowledge graph is shown in Table I.

5.2. Experiment settings

As mentioned in Section 4, each user model is learnt using the Linear Support Vector Regression method. In particular we adopted the efficient *LIBLINEAR*[19] library and chose the *L2-regularized Support Vector Regression* [Ho and Lin 2012]. The tuning of the model hyper-parameters of the learning algorithm was performed through cross-validation on validation data obtained by selecting the 15% of feedback for each user from the training data. We set the parameters C and e by using a grid-search varying C from 0.1 to 1000 with step 10 and $e = \{0.1, 0.01\}$ (tolerance of termination criterion). Before the training we performed some pre-processing on the feature vectors. We removed those features appearing in less then 5 sounds and scaled all features to the range $[0, \ldots, 1]$ using min-max normalization. Finally each feature vector was normalized to unit length using the L2 norm.

Regarding the run time performances of the entire recommender for the Freesound experiment, the highest computation time (corresponding to the path-based feature mapping with 3-hops) lasted about 28 minutes, from feature extraction to recommendation generation, on a dedicated server machine with 4 Xeon quad-core 2.93GHz processors and 32GB RAM. Since each user model is learnt independently, the learning process is highly parallelizable. Moreover, being a model-based recommender, each user model learning can be performed offline periodically once a certain number of new feedbacks are accumulated for that specific user. The implementation of the recommendation algorithm presented in this work is available on GitHub [20].

In the following we describe the experiments we carried out to evaluate our approach. In particular we are interested in evaluating the impact of semantic enrichment of the original data on the recommendation quality and the differences among the two feature mapping methods we implemented. Furthermore, we compare our approach with state of the art algorithms for implicit feedback scenarios.

5.3. Sound Recommendation Experiment

Evaluation of the semantic item description enhancement. To evaluate the impact of the various features and information sources we built several variants of item feature vectors by varying: the information sources considered, the size of the item neighborhood graphs (number of hops) and the feature mapping method. In addition, we built a content-based approach purely based on 352 low-level audio features[21] extracted from the sound signal by using Essentia [Bogdanov et al. 2013]. In this approach, predictions are computed by aggregating the Euclidean distances between the sounds downloaded by the user and the target sound to recommend. All the results are reported in Table II.

Looking at the accuracy results we see that there are no marked differences among all the feature vector variants. Noteworthy is that without considering the collaborative information (`noCollab`) the accuracy drops significantly. In addition, when considering only collaborative features accuracy performances are comparable with respect to hybrid feature combination variants. The best hybrid semantic version `Ent(fso+wn+db/keyw+tags/h=3)` is slightly better than pure collaborative (+0.8% in terms of P@10). Regarding the comparison of the two mapping methods, the Entity-based item neighborhood mapping has generally slightly higher accuracy than the

[19] http://www.csie.ntu.edu.tw/~cjlin/liblinear/

[20] https://github.com/sisinflab/lodreclib

[21] https://www.freesound.org/docs/api/analysis_example.html#all-descriptors

Table II. Freesound Results

Approach	Enrichment	h-hops	MRR	P@10	R@10	EBN@10	ADiv@10
Ent	fso	h=3	0.303	0.113	0.065	2.791	0.257
Ent	fso+wn+db/tags	h=3	0.303	0.115	0.066	2.617	0.332
Ent	fso+wn+db/tags	h=4	0.302	0.114	0.065	2.507	0.368
Ent	fso+wn+db/keyw+tags	h=3	**0.306**	**0.118**	**0.067**	2.426	0.361
Ent	fso+wn+db/keyw+tags	h=4	**0.306**	0.117	0.066	2.303	0.391
Path	fso	h=3	0.301	0.113	0.065	2.750	0.287
Path	fso+wn+db/tags	h=3	0.301	0.114	0.064	2.279	0.461
Path	fso+wn+db/tags	h=4	0.292	0.106	0.059	1.863	**0.556***
Path	fso+wn+db/keyw+tags	h=3	0.304	0.116	0.065	2.019	0.461
Path	fso+wn+db/keyw+tags	h=4	0.296	0.111	0.061	**1.618***	0.532
Collab			0.293	0.110	0.062	2.890	0.181
Ent - noCollab	fso+wn+db/keyw+tags	h=3	0.154	0.058	0.034	0.384	0.591
Path - noCollab	fso+wn+db/keyw+tags	h=3	0.151	0.049	0.028	**0.369**	**0.670**
VSM	keyw+tags	h=1	0.301	0.116	0.066	2.621	0.305
VSM - noCollab	keyw+tags	h=1	0.151	0.055	0.032	0.389	**0.670**
Audio Sim			0.022	0.004	0.002	0.382	0.044

*Accuracy, Novelty and Aggregate Diversity results for different versions of the Freesound dataset. Best values in each column are in bold. The * symbol indicates best values for hybrid and collaborative configurations. `Ent` and `Path` refers to graph embedding options; `fso`, `wn` and `db` to the initial Freesound Ontology, WordNet and DBpedia respectively; `tags` to item tags, and `keyw` to text description keywords; `h` indicates the length of the h-hop neighborhood graph; `Collab` means that only collaborative features are considered; `noCollab` that no collaborative features are considered; `VSM` refers to Vector Space Model embedding; `Audio Sim` to the audio-based approach.*

Path-based one. We can also note that considering too far entities does not improve accuracy. In fact, in both the two feature mapping when four hops are considered the results drop slightly with respect to three hops. Finally, we see that the semantic expansion of tags and terms do not improve consistently accuracy with respect to the usage of pure keywords and tags combined with collaborative information. The semantic configuration with highest accuracy (`Ent(fso+wn+db/keyw+tags/h=3)`) is only 0.2% better in terms of P@10 with respect to `VSM keyw+tags`. We can also observe that the pure audio based approach (`Audio Sim`) has by far lower performances than all the others. All the differences between the hybrid graph embeddings and the other baselines are statistically significant ($p < 0.01$) according to the paired t-test.

Novelty and aggregate diversity results instead show more interesting insights. We observe that the semantic expansion, with both feature mappings, results in an improving of both novelty and aggregated diversity. In fact, the semantic enriched variant (`fso+wn+db+keyw+tags/h=4`) has much better novelty and diversity than considering only the original tagging ontology (`fso`). Furthermore, with respect to the variants without semantic expansion, that is the variants based only on keywords and tags, the usage of semantic expansion improves considerably novelty and diversity. Hence, thanks to this exploitation of the knowledge graph we are able to recommend good items which are also not so popular. We also see that the Path-based embedding has better performances than the Entity-based one. Such approaches allow to explore better the long tail distribution of items and to increase the personalization of the system. The variants without collaborative information are the ones with better novelty and diversity. The reason behind this behavior is that pure content-based approaches are not influenced by popularity biases. However, when using only content data the system recommends unpopular but very inaccurate items. Good novelty without accuracy does not imply good recommendation quality. Finally, the usage of only collaborative infor-

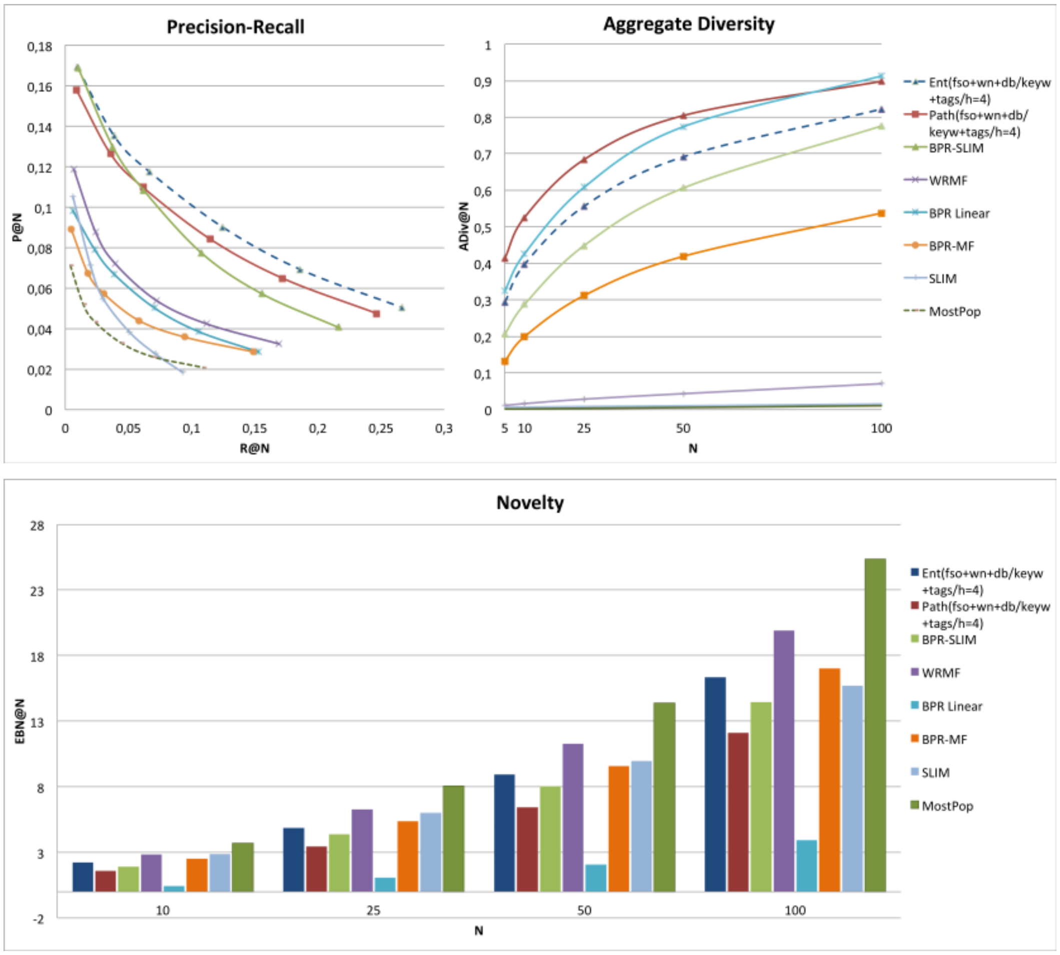

Fig. 3. Precision-Recall, Novelty and Aggregate Diversity plots in Freesound dataset

mation has much lower catalog coverage (aggregate diversity) than feature vectors containing also semantic features. For example `Path(fso+wn+db+keyw+tags/h=4)` has comparable performances in terms of accuracy with respect to `Collab` but considerably better catalog coverage and novelty (lower EBN).

To conclude, we can state that the semantic expansion, especially when combined with the Path-based mapping, improves recommendation quality in terms of novelty and aggregated diversity. The intuition behind these results is that the semantic expansion allows the system to find items semantically related to the ones in the user profile. Conversely, when using only keyword or tag-based representations the system is able to retrieve only those few items with an exact keyword/tag match with those liked by the user. Thus, the system is unable to widely explore the item space to find those items which are semantically related to the ones liked by the user.

Comparison with other methods. We compared our approach with several state of the art recommendation algorithms. `MostPop` is a popularity-based baseline which provides the same recommendation to all users based on the global popularity of items. `BPR-MF` [Rendle et al. 2009] is a matrix factorization-based method optimized with

Bayesian Personalized Ranking optimization criterion. `WRMF` is a weighted matrix factorization method [Hu et al. 2008]. `SLIM` [Ning and Karypis 2012] uses a Sparse Linear method for learning a sparse aggregation coefficient matrix. `BPR-SLIM` is similar to `SLIM` but it uses the BPR optimization criterion. `BPR Linear` is a hybrid matrix factorization method able to work with sparse datasets [Gantner et al. 2010]. We used keywords and tags as item attribute data. The computation of the recommendations for all these comparative algorithms has been done with the publicly available software library *MyMediaLite*[22].

Figure 3 shows precision-recall, novelty and aggregated diversity plots. In those plots we report the competitive algorithms used for comparison and the `Ent(fso+wn+db/keyw+tags/h=4)` and `Path(fso+wn+db+keyw+tags/h=4)` configurations which we chose as representative for our approach due to its performances in terms of novelty and aggregate diversity.

With reference to the accuracy results we notice that our two approaches largely outperforms the others. The only method which is close to the approaches we propose is `BPR-SLIM` which slightly outperforms `Path(fso+wn+db+keyw+tags/h=4)` for low values of recommendation list length ($N = 5, 10$). All differences between our approach and the other methods are statistically significant ($p < 0.01$) according to the paired t-test. With respect to the Novelty plot, our approach has much better novelty than all the other collaborative filtering algorithms but `BPR Linear` which however have much lower accuracy. Our approach outperforms most of the collaborative filtering algorithms in terms of aggregated diversity. It is able to achieve a coverage of almost 80% and 90% for $N = 50$ and $N = 100$, respectively. The approach closer to ours is `BPR Linear` that for $N = 100$ reaches same performances. Also, `BPR-SLIM` and `BPR-MF` have acceptable diversity results. Instead, all the others have very low diversity results meaning that they focus mostly on a few specific items and recommend them to all users indiscriminately.

Summing up, the experimental results show that our approach is able to give more accurate and at the same time less popular recommendations, than collaborative filtering methods. It is able to better find good recommendations in the long tail. Effective recommendation systems should promote novel and relevant items taken primarily from the tail of the distribution. In addition, our approach shows much higher aggregated diversity which can be seen as a higher personalization of the system.

5.4. Music Recommendation Experiment

The recommendation algorithms we propose have been further validated on the Last.fm dataset. We performed the same experiments on this dataset to assess the applicability of the approach to other musical contexts.

Evaluation of the semantic item description enhancement. As we may notice from the results shown in Table III, Entity-based embedding, `Collab`, and `VSM tags` approaches have very similar performance in terms of precision and recall. The first two Entity-based embedding variants have slightly higher MRR than `VSM tags`, meaning that they better locate relevant items in the top positions. Analogously to the previous sounds recommendation task, the approaches exploiting semantic expansion outperform the others in terms of novelty and aggregated diversity. The same tendency of the previous experiment is observed with the Entity-based and Path-based item neighborhood mappings. The Path-based approaches have lower precision, but much better novelty and aggregated diversity. Moreover, it is very interesting to observe that for both embedding options if we expand the graph by means of farther entities (h=4) pre-

[22] http://www.mymedialite.net/.

Table III. Last.fm Results

Approach	Enrichment	h-hops	MRR	P@10	R@10	EBN@10	ADiv@10
Ent	wn+db/tags	h=2	**0.612**	0.321	**0.122**	2.414	0.357
Ent	wn+db/tags	h=3	**0.612**	0.319	0.121	2.383	0.374
Ent	wn+db/tags	h=4	0.599	0.314	0.119	2.356	0.389
Ent	wn+db/keyw+tags	h=3	0.604	0.315	0.114	2.448	0.316
Ent	wn+db/keyw+tags	h=4	0.601	0.312	0.113	2.424	0.331
Path	wn+db/tags	h=3	0.570	0.287	0.108	2.112	0.479
Path	wn+db/tags	h=4	0.537	0.260	0.097	**1.911***	**0.544***
Path	wn+db/keyw+tags	h=3	0.570	0.289	0.104	2.173	0.411
Path	wn+db/keyw+tags	h=4	0.537	0.259	0.093	1.942	0.484
Collab			0.597	0.313	0.113	2.664	0.240
Ent - noCollab	wn+db/tags	h=3	0.292	0.114	0.043	0.983	0.703
Path - noCollab	wn+db/tags	h=3	0.285	0.113	0.043	**0.981**	**0.736**
VSM	tags	h=1	0.610	**0.322**	**0.122**	2.454	0.346
VSM	keyw	h=1	0.599	0.309	0.112	2.642	0.249

*Accuracy, Novelty and Aggregate Diversity results for different versions of the Last.fm dataset. Best values in each column are in bold. The * symbol indicates best values for hybrid and collaborative configurations.*

cision decreases whilst novelty and diversity improve. It is noteworthy thet differently from the results of the Freesound experiment, here we obtain higher accuracy with the approach that uses only tags and not keywords. Our interpretation of this trend is that, as shown in Table I, the number of tags in the Freesound dataset is somehow scarce, and the addition of keywords taken from the textual descriptions improves the annotation of the items. On the other side, in the Last.fm dataset, the set of tags is already very rich, then the addition of keywords introduces noise within the items description thus deteriorating the accuracy of recommendations. Also in this experiment we can observe that when no collaborative feature is used, accuracy is significantly worse even if novelty and diversity seem to be better. We may confirm from results in both experiments that collaborative features are a very strong signal for the accuracy of the recommendations. Nonetheless, the inclusion of semantic features allows the system to further improve accuracy and provide novel and diverse recommendations, thus better leveraging the long tail. All the differences between the hybrid graph embeddings and the other baselines are statistically significant ($p < 0.01$) according to the paired t-test.

Comparison with other methods. We compared our approach with the same set of state of the art algorithms presented in the sound recommendation experiment. Based on the observations made in the previous paragraph, we used for this experiment only tags as item attribute data for `BPR Linear`. Figure 4 shows precision-recall, novelty and aggregated diversity plots of the comparison with the other methods. We compare the competitive algorithms with the `Ent(wn+db/tags/h=3)` and `Path(wn+db/tags/h=3)` configurations which in this scenario results to be the most representative for our approach. Results are pretty similar to the ones observed in the sound recommendation experiment. Our two approaches largely outperforms the others in terms of accuracy. `BPR-SLIM` and `SLIM` have performance similar to our Entity-based mapping approach for low values of recommendation list length (N = 5, 10), and slightly higher that the Path-based one. All differences between our approaches and the other methods are statistically significant ($p < 0.01$) according to the paired t-test. Our approaches have much better novelty results than all other collaborative filtering algorithms but `BPR Linear`, which again has much lower accuracy. In terms of aggregated diversity, our approach outperforms most of the collaborative filtering algorithms. `BPR Linear` achieves

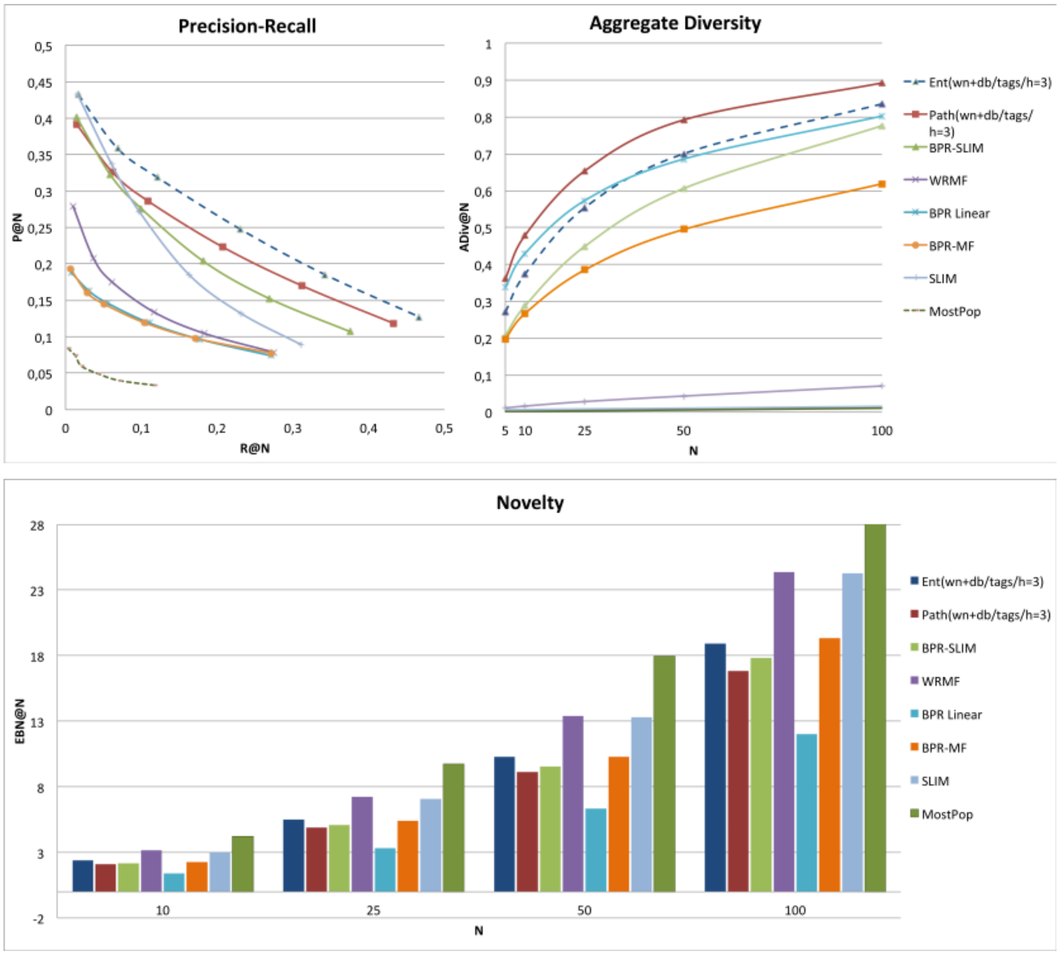

Fig. 4. Precision-Recall, Novelty and Aggregate Diversity plots in Last.fm dataset

similar diversity, but much lower accuracy. Summing up, our approach is able to recommend less popular items with higher accuracy than other collaborative filtering algorithms also in this recommendation scenario. Therefore, our approach is able to improve the level of personalization of the recommended items, and better explore the long tail also for songs recommendation.

6. RELATED WORK

Ontology-based and semantics-aware recommendation systems have been proposed in many works in the past. In [Middleton et al. 2009] an ontological recommender system is presented that makes use of semantic user profiles to compute collaborative recommendations with the effect of mitigating cold-start and improving overall recommendation accuracy. In [Mobasher et al. 2004] the authors present a *semantically enhanced collaborative filtering* approach, where structured semantic knowledge about items is used in conjunction with user-item ratings to create a combined similarity measure for item comparisons. In [Ziegler et al. 2004] taxonomic information is used to represents the user's interest in categories of products. Consequently, user similarity is determined by common interests in categories and not by common interests in items. In [Anand et al. 2007] the authors present an approach that infers user preferences from rating data using an item ontology. The system collaboratively generates recommendations using the ontology and infers preferences during similarity computation. Another hybrid ontological recommendation system is proposed in [Cantador et al. 2008] where user preferences and item features are described by semantic concepts to obtain users' clusters corresponding to implicit *Communities of Interest*. In all of these works, the experiments prove an accuracy improvement over traditional memory-based collaborative approaches especially in presence of sparse datasets. In the last few years with the availability of `Linked Open Data` (`LOD`) datasets, a new class of recommender systems has emerged which can be named as `LOD`-based recommender systems. One of the first approaches that exploits `Linked Open Data` for building recommender systems is [Heitmann and Hayes 2010]. In [Fernández-Tobías et al. 2011] the authors present a knowledge-based framework leveraging `DBpedia` for computing cross-domain recommendations. In [Di Noia et al. 2012a; Di Noia et al. 2012b] a model-based approach and a memory-based one to compute content-based recommendations are presented leveraging `LOD` datasets. Another `LOD` content-based method is presented in [Ostuni et al. 2014] which defines a neighborhood-based graph kernel for matching graph-based item representations. Two hybrid approaches have been presented lately. In [Ostuni et al. 2013] the authors show how to compute top-N recommendations from implicit feedback using linked data sources and in [Khrouf and Troncy 2013] the authors propose an event recommendation system based on linked data and user diversity. In [Rowe 2014] the authors propose a semantic-aware extension of the SVD++ model, named SemanticSVD++, which incorporates semantic categories of items into the model. The model is able also to consider the evolution over time of user's preferences. Finally, another interesting direction about the usage of `LOD` for content-based RSs is explored in [Musto et al. 2014] where the authors present Contextual eVSM, a content-based context-aware recommendation framework that adopts a semantic representation based on distributional models and entity linking techniques. In particular entity linking is used to detect entities in free text and map them to `LOD`.

An overview about techiques for music recommendation and similarity based on music contextual data is given in [Knees and Schedl 2013]. In [Kaminskas and Ricci 2012] the authors provide a description of various tools and techniques that can be used for addressing the research challenges posed by context-aware music retrieval and recommendation. A survey about techniques for the generation of music playlists is given in [Bonnin and Jannach 2014]. In particular, the authors provide a review of the literature on automated playlist generation and a categorization of the existing approaches. A context-aware music recommender system which infers contextual information based on the most recent sequence of songs liked by the user is presented in [Hariri et al. 2012]. More recently, a playlist generation algorithm with the goal of maximizing coherence and personalization of the playlist has been presented in [Jannach et al. 2015]. Finally, in [Aghdam et al. 2015] a technique for adapting recommendations to contextual changes based on hierarchical hidden Markov models is presented.

7. CONCLUSION

We have presented a hybrid approach to recommend musical items, i.e. sounds and songs, by exploiting the information encoded within a knowledge graph. We conducted various experiments on two different datasets, the one of sounds coming from Freesound.org, the other one of songs gathered from Last.fm and Songfacts.com. They may be considered as representative of the two classes of users we find the music domain: producers looking for sounds to create new music and consumers looking for new songs to listen to.

Information coming from item descriptions and tags have been enriched with data coming from two external knowledge graphs: DBpedia and WordNet. Entity Linking tools have been adopted to extract relevant entities from textual sources associated to musical items, namely tags and text descriptions, thus creating a new graph encoding the knowledge associated to users, items and their mutual interactions. We then developed a recommendation engine that combines different features, that is semantic content-based ones extracted from the resulting knowledge graph and collaborative information from implicit user feedback. An evaluation with two explicit feature mappings, *entity-based item neighborhood* and *path-based item neighborhood*, has been conducted on both datasets in order to asses the performance of the system in terms of accuracy, diversity and novelty.

Experimental results in sounds and songs recommendation show that the proposed approach is able to improve the quality of the recommended list with respect to state of the art collaborative filtering algorithms and with respect to other content-based baselines. Our results also show that the data related to the music knowledge domain encoded in freely available datasets such as DBpedia or WordNet have reached a quality level that makes possible its usage in the creation of recommendation engines whose target are either music producers or music consumers. The semantic enrichment of the initial knowledge graph performed by means of entity linking techniques is a good choice to boost the performances of the system in terms of novelty and aggregate diversity. A knowledge-based approach can improve the degree of personalization in the recommendations of musical items from various points of view such as prediction accuracy, catalog coverage and promote long tail recommendations. We have presented a methodology that achieves these objectives by combining semantic knowledge with collaborative information.

Summing up, knowledge graphs can be a useful tool when properly leveraged within recommender systems for musical items. Indeed, the graph-based nature of the information they contain, on the one hand, makes possible a linkage to other graphs thus resulting in an easy plugging of new content-based data. On the other hand, by exploring the graph new connections and commonalities between items and users can be discovered and exploited while computing the recommendation list.

ACKNOWLEDGMENTS

The authors would like to thank Gabriel Vigliensoni for providing the user's listening habits corpus.

REFERENCES

Gediminas Adomavicius and YoungOk Kwon. 2012. Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques. *IEEE Trans. Knowl. Data Eng.* 24, 5 (2012), 896–911.

Mehdi Hosseinzadeh Aghdam, Negar Hariri, Bamshad Mobasher, and Robin D. Burke. 2015. Adapting Recommendations to Contextual Changes Using Hierarchical Hidden Markov Models. In *Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16-20, 2015*. 241–244.

Sarabjot Singh Anand, Patricia Kearney, and Mary Shapcott. 2007. Generating semantically enriched user profiles for Web personalization. *ACM Trans. Internet Technol.* 7, 4, Article 22 (Oct. 2007).

Gleb Beliakov, Tomasa Calvo, and Simon James. 2015. *Recommender Systems Handbook*. Springer US, Boston, MA, Chapter Aggregation Functions for Recommender Systems, 777–808. http://dx.doi.org/10.1007/978-1-4899-7637-6_23

Alejandro Bellogín, Iván Cantador, and Pablo Castells. 2010. A Study of Heterogeneity in Recommendations for a Social Music Service. In *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec '10)*. ACM, 1–8.

Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The Semantic Web: Scientific American. *Scientific American* (May 2001).

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. DBpedia - A crystallization point for the Web of Data. *Web Semant.* 7 (September 2009), 154–165. Issue 3.

Dmitry Bogdanov, Nicolas Wack, and others. 2013. ESSENTIA: an Open-Source Library for Sound and Music Analysis. In *ACM International Conference on Multimedia (MM'13)*. 855–858.

Geoffray Bonnin and Dietmar Jannach. 2014. Automated Generation of Music Playlists: Survey and Experiments. *ACM Comput. Surv.* 47, 2, Article 26 (Nov. 2014), 35 pages.

Robin Burke. 2002. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction* 12, 4 (Nov. 2002), 331–370.

Iván Cantador, Alejandro Bellogín, and Pablo Castells. 2008. A multilayer ontology-based hybrid recommendation model. *AI Commun. Special Issue on Rec. Sys.* 21, 2-3 (April 2008), 203–210.

Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, and Davide Romito. 2012a. Exploiting the Web of Data in Model-based Recommender Systems. In *Proceedings of the Sixth ACM Conference on Recommender Systems (RecSys '12)*. ACM, New York, NY, USA, 253–256.

Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito, and Markus Zanker. 2012b. Linked open data to support content-based recommender systems. In *Proceedings of the 8th International Conference on Semantic Systems (I-SEMANTICS '12)*. ACM, New York, NY, USA, 1–8.

Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. 2011. A generic semantic-based framework for cross-domain recommendation. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec '11)*. ACM, New York, NY, USA, 25–32.

Frederic Font and Sergio Oramas. 2014. Extending Tagging Ontologies with Domain Specific Knowledge. *Interantional Semantic Web Conference (ISWC 2014)* (2014), 1–4.

Frederic Font, Gerard Roma, Perfecto Herrera, and Xavier Serra. 2012. Characterization of the Freesound online community. *2012 3rd International Workshop on Cognitive Information Processing (CIP)* (2012), 1–6.

Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. 2010. Learning Attribute-to-Feature Mappings for Cold-Start Recommendations. In *Proceedings of the 2010 IEEE International Conference on Data Mining (ICDM '10)*. IEEE Computer Society, Washington, DC, USA, 176–185.

Andres Garcia-Silva, Oscar Corcho, Harith Alani, and Asuncion Gomez-Perez. 2012. Review of the state of the art: discovering and associating semantics to tags in folksonomies. *The Knowledge Engineering Review* 27 (3 2012), 57–85. Issue 01.

Negar Hariri, Bamshad Mobasher, and Robin D. Burke. 2012. Context-aware music recommendation based on latenttopic sequential patterns. In *Sixth ACM Conference on Recommender Systems, RecSys '12, Dublin, Ireland, September 9-13, 2012*. 131–138.

Benjamin Heitmann and Conor Hayes. 2010. Using Linked Data to Build Open, Collaborative Recommender Systems. In *AAAI Spring Symposium: Linked Data Meets Artificial Intelligence*.

Chia-Hua Ho and Chih-Jen Lin. 2012. Large-scale linear support vector regression. *Journal of Machine Learning Research* 13 (2012), 3323–3348.

Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM '08)*. 263–272.

Dietmar Jannach, Lukas Lerche, and Iman Kamehkhosh. 2015. Beyond "Hitting the Hits": Generating Coherent Music Playlist Continuations with the Right Tracks. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15)*. ACM, New York, NY, USA, 187–194.

Marius Kaminskas and Francesco Ricci. 2012. Contextual music information retrieval and recommendation: State of the art and challenges. *Computer Science Review* 6, 2-3 (2012), 89–119.

Houda Khrouf and Raphaël Troncy. 2013. Hybrid Event Recommendation Using Linked Data and User Diversity. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*. ACM, New York, NY, USA, 185–192.

Peter Knees and Markus Schedl. 2013. A Survey of Music Similarity and Recommendation from Music Context Data. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)* 10, 1 (2013).

Sean M. McNee, John Riedl, and Joseph A. Konstan. 2006. Being Accurate is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems (CHI EA '06)*. ACM, New York, NY, USA, 1097–1101.

Stuart E Middleton, David De Roure, and Nigel R Shadbolt. 2009. Ontology-Based Recommender Systems. *Handbook on Ontologies* 32, 6 (2009), 779–796.

George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM* 38, 11 (Nov. 1995), 39–41.

Bamshad Mobasher, Xin Jin, and Yanzan Zhou. 2004. Semantically Enhanced Collaborative Filtering on the Web. In *Web Mining: From Web to Semantic Web*, Bettina Berendt, Andreas Hotho, Dunja Mladenic, Maarten Someren, Myra Spiliopoulou, and Gerd Stumme (Eds.). Lecture Notes in Computer Science, Vol. 3209. Springer Berlin Heidelberg, 57–76.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation : a Unified Approach. *Transactions of the Association for Computational Linguistics (TACL)* (2014).

Cataldo Musto, Giovanni Semeraro, Pasquale Lops, and Marco de Gemmis. 2014. Combining Distributional Semantics and Entity Linking for Context-Aware Content-Based Recommendation. In *User Modeling, Adaptation, and Personalization - 22nd International Conference, UMAP 2014, Aalborg, Denmark, July 7-11, 2014. Proceedings*. 381–392.

Xia Ning and George Karypis. 2012. Sparse linear methods with side information for top-n recommendations. In *Proceedings of the sixth ACM conference on Recommender systems (RecSys '12)*. 155–162.

Sergio Oramas, Luis Espinosa-anke, Mohamed Sordo, Horacio Saggion, and Xavier Serra. 2016. ELMD : An Automatically Generated Entity Linking Gold Standard Dataset in the Music Domain. In *In Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*.

Vito Claudio Ostuni, Tommaso Di Noia, Eugenio Di Sciascio, and Roberto Mirizzi. 2013. Top-N Recommendations from Implicit Feedback Leveraging Linked Open Data. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*. ACM, New York, NY, USA, 85–92.

Vito Claudio Ostuni, Tommaso Di Noia, Roberto Mirizzi, and Eugenio Di Sciascio. 2014. A Linked Data Recommender System using a Neighborhood-based Graph Kernel. In *The 15th International Conference on Electronic Commerce and Web Technologies (Lecture Notes in Business Information Processing)*. Springer-Verlag.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI '09)*. AUAI Press, Arlington, Virginia, United States, 452–461.

Matthew Rowe. 2014. SemanticSVD++: incorporating semantic taste evolution for predicting ratings. In *2014 IEEE/WIC/ACM International Conferences on Web Intelligence, WI 2014*.

Giovanni Semeraro, Pasquale Lops, Pierpaolo Basile, and Marco de Gemmis. 2009. Knowledge infusion into content-based recommender systems. In *Proceedings of the third ACM conference on Recommender systems (RecSys '09)*. ACM, New York, NY, USA, 301–304.

John Shawe-Taylor and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA.

Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions. *Knowledge and Data Engineering, IEEE Transactions on* 27, 2 (Feb 2015), 443–460.

Mohamed Sordo, Sergio Oramas, and Luis Espinosa-Anke. 2015. Extracting Relations from Unstructured Text Sources for Music Recommendation. In *Proceedings of the 20th International Conference on Applications of Natural Language to Information Systems, NLDB 2015*. Springer International Publishing, Cham, 369–382.

Harald Steck. 2013. Evaluation of recommendations: rating-prediction and ranking. In *RecSys*. 213–220.

Gabriel Vigliensoni and Ichiro Fujinaga. 2014. Identifying Time Zones in a Large Dataset of Music Listening Logs. In *Proceedings of the First International Workshop on Social Media Retrieval and Analysis (SoMeRA '14)*. ACM, New York, NY, USA, 27–32.

Cai-Nicolas Ziegler, Georg Lausen, and Lars Schmidt-Thieme. 2004. Taxonomy-driven computation of product recommendations. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management (CIKM '04)*. ACM, New York, NY, USA, 406–415.