

# An Investigation on the Serendipity Problem in Recommender Systems

Marco de Gemmis<sup>a</sup>, Pasquale Lops<sup>a</sup>, Giovanni Semeraro<sup>a</sup>, Cataldo Musto<sup>a</sup>

<sup>a</sup>*Department of Computer Science, University of Bari Aldo Moro  
Via E. Orabona 4, I-70125 Bari, Italy*

---

## Abstract

Recommender systems are filters which suggest items or information that might be interesting to users. These systems analyze the past behavior of a user, build her profile that stores information about her interests, and exploit that profile to find potentially interesting items. The main limitation of this approach is that it may provide accurate but likely obvious suggestions, since recommended items are similar to those the user already knows. In this paper we investigate this issue, known as *overspecialization* or *serendipity problem*, by proposing a strategy that fosters the suggestion of surprisingly interesting items the user might not have otherwise discovered.

The proposed strategy enriches a graph-based recommendation algorithm with background knowledge that allows the system to deeply understand the items it deals with. The hypothesis is that the infused knowledge could help to discover hidden correlations among items that go beyond simple feature similarity and therefore promote non-obvious suggestions. Two evaluations are performed to validate this hypothesis: an in-vitro experiment on a subset of the HETREC2011-MOVIELENS-2K dataset, and a preliminary user study. Those evaluations show that the proposed strategy actually promotes non-obvious suggestions, by narrowing the accuracy loss.

*Keywords:* Recommender Systems, Serendipity Problem, Knowledge Representation, Spreading Activation, Affective Feedback, Facial Expressions

---

---

*Email addresses:* marco.degemmis@uniba.it (Marco de Gemmis),  
pasquale.lops@uniba.it (Pasquale Lops), giovanni.semeraro@uniba.it (Giovanni Semeraro), cataldo.musto@uniba.it (Cataldo Musto)

## 1. The Filter Bubble and the Serendipity Problem

In the book “*The Filter Bubble: What the Internet Is Hiding from You*”, Eli Pariser argues that Internet is limiting our horizons [58]. He worries that personalized filters, such as Google search or Facebook delivery of news from our friends, create individual universes of information for each of us, in which we are fed only with information we are familiar with and that confirms our beliefs. These filters are opaque, that is to say, we do not know what is being hidden from us, and may be dangerous because they threaten to deprive us from *serendipitous* encounters that spark creativity, innovation, and the democratic exchange of ideas. Similar observations have been previously made by Gori and Witten [32] and extensively developed in their book “Web Dragons, Inside the Myths of Search Engine Technology” [81], where the metaphor of search engines as modern *dragons* or gatekeepers of a treasure is justified by the fact that “the immense treasure they guard is society’s repository of knowledge” and all of us accept dragons as mediators when having access to that treasure. But most of us do not know how those dragons work, and all of us (probably the search engines’ creators, either) are not able to explain the reason why a specific web page ranked first when we issued a query. This gives rise to the so called *bubble of Web visibility*, where people who want to promote visibility of a Web site fight against heuristics adopted by most popular search engines, whose details and biases are closely guarded trade secrets.

Also recommender systems, which suggest to users items or information they might be interested in [61], give their contribution to the filter bubble [42, 60]. These systems analyze a user’s past behavior, maybe find others who have a similar history, and use that information to provide suggestions. For example, if you tell the Internet Movie Database (IMDb)<sup>1</sup> that you like the movie *Star Trek into Darkness*, it will suggest movies liked by other people who liked that movie (“People who liked this also liked...” in Figure 1), most of whom are probably science-fiction fans. Furthermore, one of those recommendations is a movie of the same saga, which is likely to be already known to the user. The user will be provided with items within her existing range of interests and her tendency towards a certain behavior is reinforced by creating a self-referential loop. This drawback is known as *overspecialization* or *serendipity problem* [52], and stems from the fact that the goal of the

---

<sup>1</sup>[www.imdb.com](http://www.imdb.com)

The screenshot shows the IMDb page for the movie "Star Trek Into Darkness" (2013). At the top, there is a search bar and a navigation menu with options like "Movies, TV & Showtimes", "Celebs, Events & Photos", "News & Community", and "Watchlist". The main content area features a large movie poster on the left and a detailed description on the right. The description includes the title "Into Darkness - Star Trek", the year "(2013)", the original title "Star Trek Into Darkness", and the genre "Action, Adventure, Sci-Fi". It also displays a user rating of 7.8 and a Metascore of 72/100. Below the description, there is a section titled "People who liked this also liked..." which shows a grid of movie posters for other Star Trek titles, including "Star Trek (2009)", "Star Trek: The Motion Picture", "Star Trek: The Next Generation", "Star Trek: Voyager", "Star Trek: Enterprise", and "Star Trek: Generations".

Figure 1: IMDb suggestions for the movie *Star Trek into Darkness*

system is to find items that best match the model of user preferences in order to improve accuracy, regardless of the actual usefulness of the suggestions. The importance of taking into account factors, other than accuracy, which contribute to the perceived quality of recommendations is emphasized in recent research [15, 38, 87]. One of these factors is serendipity, that can be seen as the experience of receiving unexpected suggestions helping the user to find surprisingly interesting items she might not have otherwise discovered, or that would have been really hard to discover [36]. Serendipity has been recognized as a goal that often conflicts with accuracy [26], therefore it is important that systems were designed and evaluated by taking into account the need of properly balancing these two factors. As an extreme case, let us consider random recommendations, which improve serendipity but cause a

drastic loss in accuracy, making the system actually ineffective.

In this paper, we investigate the following two issues related to the serendipity problem:

1. does the inclusion of knowledge into the recommendation process (which aims at providing the system with deeper understanding of the items it deals with) help to find serendipitous, non-obvious and at the same time effective suggestions?
2. how to measure the perception of serendipity, i.e. how to assess the acceptance of suggestions, given that providing non-obvious recommendation can hurt the accuracy of the system?

The main contributions of the paper with respect to the above mentioned issues are:

1. the definition of a strategy based on a knowledge intensive process, called *Knowledge Infusion* (KI) [68], that automatically builds a machine-readable *background knowledge*, the memory of the recommender system, exploited by a reasoning algorithm to find meaningful hidden correlations among items. The hypothesis is that, if the recommendation process exploits the discovered associations rather than classical feature similarities or co-rating statistics, more serendipitous suggestions can be provided to the user;
2. the in-depth analysis of the results of both an in-vitro experimental evaluation on a benchmark dataset and a preliminary user study carried out in order to validate the proposed hypothesis. In particular, the user study assessed the serendipity of suggestions by means of a tool which allows to gather implicit user feedback through the analysis of their facial expressions.

The analysis of the items in the recommendation lists produced by the proposed strategy leads us to conclude that they show an acceptable balance of serendipity and accuracy.

## 2. Background: Serendipity in Recommender Systems

Several definitions of serendipity have been proposed in recommender systems literature. A commonly agreed one, proposed by Herlocker et al. [36], describes serendipitous recommendations as the ones helping the user

to find surprisingly interesting items she might not have discovered by herself. McNee et al. [52] identify serendipity as the experience of receiving an unexpected and fortuitous item recommendation, while Shani and Gunawardana [71] state that serendipity involves a positive emotional response of the user about novel items and measures how surprising these recommendations are.

According to these definitions, serendipity in recommender systems is characterized by *interestingness* of items and the *surprise* for users who get *unexpected* suggestions. Therefore, in our work we define serendipitous suggestions those which are both *attractive* and *unexpected*. While attractiveness is usually determined in terms of closeness to the user profile [49], the assessment of unexpectedness of recommendations is not immediate. Previous studies agreed on defining unexpectedness as the deviation from a benchmark model or primitive prediction method that generates expected recommendations [29, 53]. For example, in case of a movie recommender system, expected recommendations could be blockbusters seen by many people, or movies related to those already seen by the user, such as sequels, or those with same genre and director.

For the evaluation of the proposed strategy, unexpectedness will be measured with respect to benchmark models based on popularity and average rating of items (Section 5.1.2).

In order to make clearer the adopted definition of serendipity, it is useful to point out the differences with related notions of *novelty* and *diversity*.

The *novelty* of a piece of information generally refers to how different it is with respect to “what has been previously seen” by a user or a community. Novelty occurs when a recommender system suggests to the active user an unknown item that she might have autonomously discovered [36, 78]. Let us consider a recommender system that simply suggests movies directed by the user’s favorite director. If the system recommends a movie the user was not aware of, the movie will be novel, but not serendipitous. On the other hand, a movie by a young, not very popular director is more likely to be serendipitous (and also novel).

*Diversity* represents the variety present in a list of recommendations [3, 26, 87]. Methods for the diversification of suggestions are generally used to avoid homogeneous lists, in which all the items suggested are very similar to each other. This may reduce the overall quality of the recommendation list because none of the alternative suggestions will be liked, in case the user wants something different from the usual. Although diversity is very different

from serendipity, a relationship between the two notions exists, in the sense that providing the user with a diverse list can facilitate unexpectedness [2]. Continuing with our example, we can reasonably assume that users could be surprised to some extent when seeing a romantic movie within the list of science-fiction movies shown in Figure 1. However, the diversification of recommendations does not necessarily imply serendipity since diverse items could all fall into the range of user preferences.

The acquisition of information in an accidental or serendipitous manner is a recognized information seeking metaphor investigated in literature [23, 27, 77]. In particular, Toms suggests four strategies to induce serendipity in the search process [77]:

- *Blind luck* or role of chance, implemented via a random information node generator. In the context of recommender systems, that strategy might be implemented by providing random suggestions;
- *Pasteur principle*, i.e. “chance favors only the prepared mind”, meaning that sudden flashes of insight do not just happen, but they are the product of preparation. Recommender systems could implement the “prepared mind” paradigm by applying information about user preferences in different contexts. For example, if the system knows that a user is interested in science-fiction movies, it might exploit that information when the user is looking for a hotel as well, and suggest the Hilton in Las Vegas because it hosts a Star Trek flight simulator;
- *Anomalies and exceptions*, that might be implemented using distance measures able to identify items dissimilar to those the user liked in the past;
- *Reasoning by analogy*, which implies an abstraction mechanism allowing the system to discover the applicability of an existing schema to a new situation.

In this paper we propose an approach related to the Pasteur principle. It is grounded on the idea that the capability of an algorithm to produce serendipitous suggestions could be improved by the Knowledge Infusion process described in Section 3, which provides the system with a memory of world facts and linguistic competencies, and therefore contributes to build the prepared mind.

The recommendation algorithm adopted is Random Walk with Restarts [50], augmented with the infused knowledge to build an advanced item correlation matrix where more significant associations are inferred by the prepared mind, compared to the standard item similarity computation. Section 4 describes the details of the whole recommendation process.

### 3. The Knowledge Infusion Process

The Knowledge Infusion (KI) process builds a computer-understandable knowledge repository which constitutes the cultural and linguistic background of the system. The repository is automatically fed by information obtained from several knowledge sources freely available, such as Wikipedia. The main motivation for this choice, compared to the adoption of specific handcrafted ontologies, is the willingness to design a general strategy which allows to update the knowledge repository easily, as well as to plug in additional sources, without changing the overall organization and implementation of the process.

KI consists of two steps:

1. *Knowledge Extraction and Harmonization*: Linguistic knowledge is extracted from WordNet [25], while encyclopedic knowledge is obtained from Wikipedia. Due to the different organization of the sources (articles in Wikipedia, synsets in WordNet), an harmonization phase turns the extracted concepts in a homogeneous format. Linguistic knowledge is useful to recognize *general* concepts into item descriptions, while encyclopedic knowledge is useful to recognize *specific* concepts or named entities, usually not included in a dictionary. More details are provided in Section 3.1;
2. *Reasoning*: It allows to make inference on the background knowledge and item descriptions, in order to discover information potentially useful for the recommendation step. More details are provided in Section 3.2.

#### 3.1. Knowledge Extraction and Harmonization

The heterogeneity of the knowledge sources involved in the process requires:

- the identification of the *basic unit* representing a *concept* in each specific source;

- the adoption of a unique representation model for all the basic units in the sources.

As for the first issue, the idea is that the basic unit of a knowledge source corresponds to a primary concept it represents. We consider an article as basic unit for Wikipedia, since it provides details about an entity (“Alan Turing”), a world fact (“Normandy landings”), or a generic concept (“Computer”). For WordNet, the basic unit is the synset, which provides the short description of a generic concept (gloss) and lists all the synonyms expressing that concept.

As for the representation model, being each basic unit a fragment of text, we adopt the standard bag-of-words (BOW) model and tf-idf as term weighting scheme. We call the BOW representation of a basic unit a *Cognitive Unit* (CU), because it provides the machine-readable format of the concepts on which the reasoning mechanism works. The name stems from the Adaptive Control of Thought (ACT) theory by J. R. Anderson, according to which information in the long term memory of human beings is encoded as cognitive units that form an interconnected network [5]. The reasoning algorithm of the system is inspired by that theory.

Some Natural Language Processing operations are applied on basic units to obtain the corresponding CUs:

- *Wikipedia*: Title and full text of an article are processed by tokenization, stopword elimination, lemmatization, named entity recognition. Simple heuristics (not described for brevity) are adopted for boosting tf-idf scores of emphasized words [69], while feature selection by using tf-idf thresholding is applied to filter out less significant words.
- *WordNet*: The lemma and its synonyms, the keywords in the gloss, as well as the keywords in the example phrases, are processed by tokenization, stopword elimination, lemmatization, named entity recognition. Polysemous words originate one CU for each possible meaning.

The resulting CUs are stored in separate repositories. The main advantage of having CUs in the form of BOWs is that CU repositories can be represented by using the Vector Space Model. This provides an easy and immediate way to find relevant CUs associated with any keywords, by simply querying the CU repositories and computing relevance as cosine similarity. The whole KI process is described in Figure 2: the Knowledge Extraction



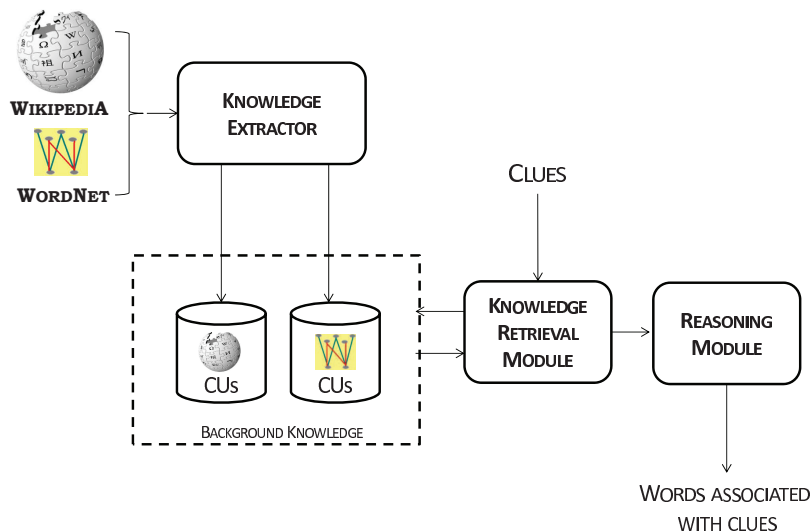


Figure 2: Architecture of the Knowledge Infusion process

and Harmonization phase is performed by the *Knowledge Extractor*; then, once the background memory is available, the reasoning step is triggered by a set of keywords, which we call *clues*, exploited to query the CU repositories in order to retrieve the most appropriate *pieces of knowledge*. Both clues and retrieved CUs are then passed to the *Reasoning Module*, which produces a new list of related keywords, which can be exploited in the recommendation process to suggest or search for related items. The advantage is that search is performed using keywords that are not necessarily included in the description of preferred items, which might allow for the selection of non-obvious and potentially serendipitous items.

### 3.2. The Reasoning Step

As introduced in previous section, the reasoning mechanism is inspired by the ACT theory [5], according to which words and their meanings are stored in the mind in a network-like structure. The algorithm implemented in the *Reasoning Module* is based on a *Spreading Activation model* [17], consisting of a network of nodes called Spreading Activation Network (SAN) on which a search process is performed. The model has been successfully adopted in Information Retrieval [19, 20].

Nodes in the SAN built for the KI process represent words or CUs, and links between them are usually weighted according to the strength of their

relationship, obtained from CU repositories.

The method that builds the SAN and activates the spreading process is described by means of the following illustrative example in a movie recommendation scenario. We start from the “seed movie” *Star Trek into Darkness* (Fig. 1), and we are looking for related items. We choose the two plot keywords (among those provided by the IMDb web site) *alien* and *battle* as clues to query the *Knowledge Retrieval* module. For the sake of simplicity, we assume that, for each clue, two CUs are retrieved from each knowledge source:

- 2 CUs from the WordNet dictionary ( $Dic_1$  and  $Dic_2$ ) corresponding to different meanings of the keyword *alien*, i.e. a stranger or an extraterrestrial being;
- 2 CUs from Wikipedia ( $Wiki_1$  and  $Wiki_2$ ) corresponding to different meanings of the keyword *alien*, i.e. the movie by Ridley Scott or the extraterrestrial life;
- 2 CUs from the WordNet dictionary ( $Dic_3$  and  $Dic_4$ ) corresponding to different meanings of the keyword *battle*, i.e. war or the battleship game;
- 2 CUs from Wikipedia ( $Wiki_3$  and  $Wiki_4$ ) corresponding to different meanings of the keyword *battle*, i.e. war or the battleship game. In this case, *battle* has the same meanings as for the dictionary, but the keywords in the CUs are different due to different content of the Wikipedia article and the WordNet synset.

The construction of the SAN is described in the following paragraphs and the result is shown in Figure 3. Initially, two source nodes labeled with the two clues are included into the SAN. Then, retrieved CUs are included in the SAN. Each CU is linked to the corresponding source node; the edge is oriented from the clue to the CU and is labeled with the cosine similarity value between the clue and the CU. At this stage of the process, edges represent associations between clues and CUs, while similarity values measure the strength of those relationships. Finally, for each CU node, word nodes labeled with terms in the BOW of the CU are included in the SAN. Links are created from the CU node towards its word nodes and labeled with tf-idf scores of words.

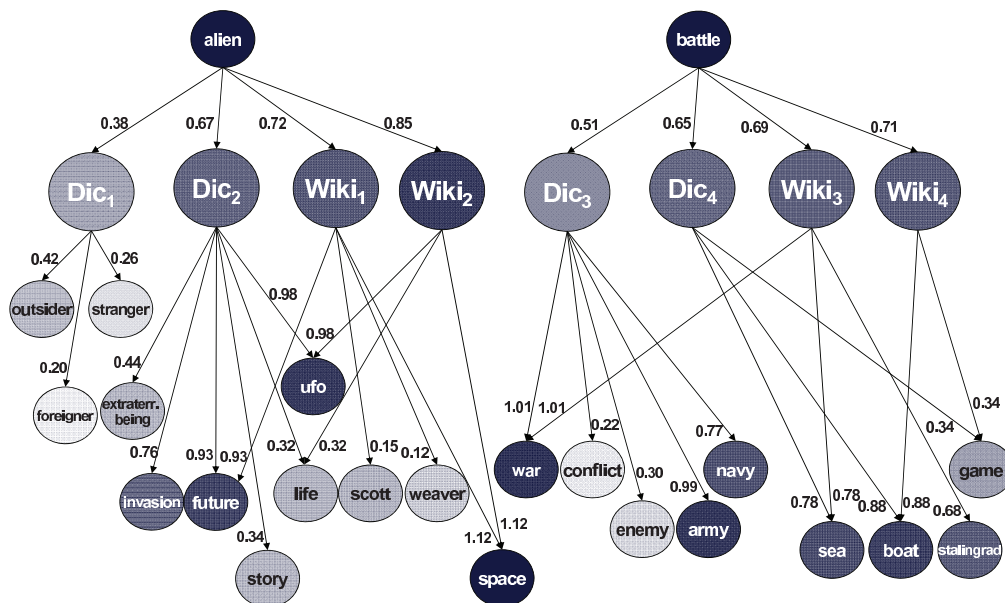


Figure 3: The illustrative SAN built starting from the clues *alien* and *battle*. Edges between a clue and a CU are weighted using the cosine similarity measure, while edges between a CU and a word node are weighted using tf-idf. Darker nodes are those with a higher level of activation.

Once the SAN is built, the reasoning process starts from the clues, which trigger the search process over the network. Each node  $n_i$  has an associated activation level  $al_i$ , which is a real number in the range  $[0.0 \dots 1.0]$ , and represents the level of stimulus of the node. At time  $tm = 1$  the SAN is initialized by setting all activation levels to 0, with exception of the clues, whose activation level is set to 1. A threshold  $F$ , a real number in the range  $[0.0 \dots 1.0]$ , determines if a node is fired, that is to say whether it can spread its activation level over the SAN. Every fired node propagates its own activation value to its neighbors as a function of both its current activation level and the weights of the edges that connect it with its neighbors, and a decay factor  $D$  that limits the propagation of the activation value through the network. The activation level of neighbors is updated accordingly. At time  $tm = 2$ , all clues are fired and the amount of activation levels spreading from them updates the activation level of CU nodes. At time  $tm = 3$ , only CU nodes whose activation levels exceed  $F$  are fired and propagate their activation values to their neighbors, i.e. word nodes. Word nodes are

ranked in descending order according to their activation values and the *top-k* nodes are selected as the most relevant words related to the clues. In the illustrative SAN depicted in Figure 3, *future, ufo, space, war, army, navy, sea, boat, stalingrad* are the most relevant keywords related to the clues *alien* and *battle*. These new keywords could be exploited to lead the graph-based recommendation algorithm to produce unexpected suggestions, as described in the following section. The spreading activation algorithm is thoroughly described in [67], where it was successfully used as the “brain” of an artificial player for a language game demanding the linguistic and cultural background knowledge typically owned by human beings.

#### 4. RWR-KI: a graph-based recommendation algorithm enhanced with KI

Graph-based techniques are becoming popular since they allow to capture transitive associations between nodes (items), thus promoting the discovery of correlations between them [21]. We adopted a Random Walk model, called Random Walk with Restarts (RWR) [50] as a recommendation technique to be enhanced by KI for discovering serendipitous items. We called the resulting algorithm Random Walk with Restarts enhanced by Knowledge Infusion (RWR-KI).

Random Walk models exploit a correlation graph between items to predict user preferences. Nodes in the correlation graph correspond to items, while edges indicate the degree of correlation between items. A *correlation matrix* is built by filling in each entry with the correlation index between item pairs. In [31] the correlation index is the number of users who co-rated the item pair, while in [85] the correlation index denotes the content similarity between movies.

Given the correlation graph and a starting point, e.g. an item preferred by the user, in the random walk model a neighbor of the starting point is randomly selected for a transition; then, a neighbor of this point is recursively selected at random for a new transition. At each step, there is some probability to return to the starting node. The sequence of randomly selected points is a random walk on the graph.

We have enhanced that model in a way that the correlation index between items can actually reflect some hidden associations discovered by the KI process, rather than using a classical similarity score based on a statistical correlation.

#### 4.1. Random walk with restarts

The algorithm simulates a random walk by moving from an item  $i$  to a similar item  $j$  in the next step of the walk. The relevance score of an item  $j$  with respect to an item  $i$  is defined as the steady-state probability  $r_{ij}$  to finally stay at item  $j$ , and the correlation matrix is interpreted as a *transition probability matrix*. Formally, given:

- a weighted graph  $G$  denoting the degree of correlation between items;
- the corresponding column normalized correlation matrix  $S$  of the graph  $G$ , in which the element  $S_{ij}$  represents the probability of  $j$  being the next state given that the current state is  $i$ ;
- a starting node  $x$ ;
- the column vector  $p^\tau$ , where  $p_i^\tau$  denotes the probability that the random walk at step  $\tau$  is at node  $i$ ;
- the starting vector  $q$ , having zeros for all elements except the starting node  $x$  set to 1;
- the probability  $\alpha$  to restart from the initial node  $x$ ,  $0 \leq \alpha \leq 1$ ;

then, Random Walk with Restarts is defined as follows:

$$p^{\tau+1} = (1 - \alpha)Sp^\tau + \alpha q \quad (1)$$

The steady-state or stationary probabilities provide the long term visit rate of each node, given a bias toward the particular starting node. This can be obtained by iterating Equation (1) until convergence, that is, until the difference between  $L_2$  norm of two successive estimates is below a certain threshold, or a maximum number of iterations is reached.

Let  $\sigma$  be the state after convergence,  $p_i^\sigma$  can be considered a measure of relatedness between the starting node  $x$  and the node  $i$ . The final result is a list of items ranked according to the stationary probability of each node after convergence. The complexity of the method is  $O(mt)$ , where  $m$  is the number of edges in the graph and  $t$  is the number of iteration steps [28].

#### 4.2. Building the correlation matrix using Knowledge Infusion

Given an item  $I$ , the idea is to exploit the keywords associated with  $I$  by KI to compute the correlation index between  $I$  and other items in the collection. We adopt a content-based model in which each item  $I$  is represented as a vector in a  $n$ -dimensional space of features [49]:

$$\vec{I} = \langle w_1, w_2, \dots, w_n \rangle \quad (2)$$

Features are keywords extracted from item descriptions, therefore the feature space is the vocabulary of the item collection, while  $w_i$  is the score of feature  $k_i$  in the item  $I$ , which measures the importance of that feature for the item.

Given a query  $q$ , the ranking function adopted for searching in the item collection is based on the BM25 probabilistic retrieval framework [62, 74]:

$$R(q, I) = \sum_{t \in q} \frac{f(t, I) \cdot (\alpha_1 + 1)}{f(t, I) + \alpha_1 \cdot (1 - b + b \frac{|I|}{avgdl})} \cdot idf(t) \quad (3)$$

where  $f(t, I)$  is frequency of the term  $t$  in the item  $I$ ,  $\alpha_1$  and  $b$  are parameters usually set to 2 and 0.75 respectively,  $avgdl$  is the average item length and  $idf(t)$  is the standard inverse document frequency of term  $t$  in the item collection.

The procedure (Algorithm 1) for building the correlation matrix follows three main steps:

1. selection of the most representative features (keywords) for item  $I_j$  (step 5);
2. running KI by providing those keywords as clues in order to get new keywords related to  $I_j$  (step 6);
3. retrieval of items correlated to  $I_j$  by using new keywords provided by KI as input for the ranking function (steps 7-10). The scores computed by the ranking function are used to fill in the correlation matrix.

Figure 4 depicts a fragment of the row of the correlation matrix for the movie *Star Trek into Darkness*.

Starting from the most representative keywords for that movie (*alien, battle, starship, captain, mission*), KI produces new keywords which are exploited to compute the correlation index with the other movies in the collection. New keywords may be roughly subdivided in two main topics: science-fiction (*space, future, ufo*) and conflicts/fights (*war, army, navy, boat, sea*)

---

**Algorithm 1** Algorithm for building the correlation matrix

---

```
1:  $C \leftarrow \{I_1, \dots, I_N\}$   $\triangleright I_1, \dots, I_N$  items in the collection
2:  $S \leftarrow NULL$   $\triangleright$  Initialization of the correlation matrix
3: procedure BUILDCORRELATIONMATRIX( $S, C$ )  $\triangleright$  Fills in
   the correlation matrix  $S$  for items in the collection  $C$ . Each element  $S_{ji}$ 
   is the correlation index between item  $I_j$  and item  $I_i$ 
4:   for all  $I_j \in C$  do
5:      $Features_j \leftarrow \{k_1, \dots, k_n\}$   $\triangleright \{k_1, \dots, k_n\}$  set of features for  $I_j$  given
   as clues to KI
6:      $NewFeatures_j \leftarrow KI(Features_j, m)$   $\triangleright$  List of  $m$  related
   keywords associated with clues by KI
7:      $q \leftarrow NewFeatures_j$   $\triangleright$  Query for retrieving items correlated with
   new keywords provided by KI
8:     for all  $I_i \in C \wedge I_i \neq I_j$  do  $\triangleright$  Fill in row  $I_j$  of correlation matrix
9:        $S_{ji} \leftarrow R(q, I_i)$   $\triangleright$  Correlation index between  $I_j$  and  $I_i$ 
   computed by the ranking function
10:    end for
11:  end for
12: end procedure
```

---


	THE TRUMAN SHOW	THE HUNT FOR RED OCTOBER	ALIENS	MASTER AND COMMANDER	THE X-FILES	ENEMY AT THE GATES
 <p><b>STAR TREK INTO DARKNESS</b></p> <p>IMDb keywords alien, battle, starship, captain, mission</p> <p>KI keywords space, future, ufo, war, army, navy, boat, sea, stalingrad ,...</p>	<p>Correlation index 0.43</p> <p>Keywords matched boat, future, storm at sea</p>	<p>Correlation index 0.67</p> <p>Keywords matched navy, US navy, soviet navy, sea, cold war</p>	<p>Correlation index 0.55</p> <p>Keywords matched outer space, space colony, space travel, future</p>	<p>Correlation index 0.72</p> <p>Keywords matched sea, sea battle, navy, royal navy, ship, war</p>	<p>Correlation index 0.14</p> <p>Keywords matched ufo</p>	<p>Correlation index 0.51</p> <p>Keywords matched world war II, stalingrad, german army, boat</p>

Figure 4: An illustrative example showing a fragment of the row of the correlation matrix for the movie *Star Trek into Darkness*. Each cell reports the correlation index between that movie and those on the column, and the set of plot keywords which match the query represented by the keywords produced by KI.

and *stalingrad*). While science-fiction keywords are quite understandable as clearly related to the movie, conflicts/fights keywords are probably obtained due to less obvious correlation with the input keywords *captain* and *battle*.

Our hypothesis is that this kind of correlations can lead the recommendation algorithm towards serendipitous suggestions.

## 5. Experimental evaluation

The main goal of the experimental evaluation is to validate the hypothesis that top- $N$  recommendations produced by the Random Walk with Restarts algorithm enhanced with the KI process are serendipitous. Measuring the degree of serendipity of a recommendation list is a complex task since it involves multiple dimensions upon which items are evaluated [4]. Furthermore, it is not only an issue of metrics, but it also depends on the difficulty of detecting and providing an objective assessment of the emotional response - the *pleasant surprise* - which serendipitous suggestions should convey [52].



In order to clearly define the evaluation task, we consider serendipitous suggestions those *relevant*, i.e. close to the user profile, and *unexpected* at the same time. While computing relevance is a well established issue, the problem of assessing unexpectedness could be approached in different ways. To this purpose, we designed two experiments:

1. an in-vitro experiment on a benchmark dataset, in which unexpectedness is measured as the deviation from a *standard prediction criterion* which is more likely to produce expected recommendations, as suggested by Murakami et al. [53]. For example, if the standard prediction criterion is based on a non-personalized recommender algorithm based on popularity, the most popular items will be the most expected recommendations, while the items in the long tail will be the most unexpected ones. The investigation is described in Section 5.1;
2. a study with real users aiming at assessing the actual perception of serendipity of recommendations and their *acceptance* in terms of both relevance and unexpectedness. The analysis is performed by using Noldus FaceReader<sup>TM</sup>, a tool which allows to gather implicit feedback about users' reactions to recommendations through the analysis of their facial expressions. The study is described in Section 5.2.

### 5.1. In-vitro experiment

The main aim of this experiment is to study the trade-off between relevance and unexpectedness of recommendation lists computed by RWR-KI, in order to understand whether the suggestions satisfy the personal interests of users on the one hand, and encourage the exploration of new areas of potential interests on the other hand. The results are compared to those reported by other state-of-art algorithms described in Section 5.1.4 in order to evaluate to which extent they provide serendipitous suggestions. Furthermore, we analyze the distribution of relevant, unexpected and serendipitous items within lists of different sizes. The aim of the analysis is to assess whether size is a significant factor when the goal is to provide the user with a balanced recommendation set.

#### 5.1.1. Dataset

The evaluation is performed on a subset of the HETREC2011-MOVIELENS-2K dataset, made available at the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems, HetRec 2011 [13], and

freely downloadable at [grouplens.org/datasets/hetrec-2011](http://grouplens.org/datasets/hetrec-2011). The original dataset contains 855,598 rating assignments on a 10-point Likert scale from 0.5 to 5.0 (step 0.5), provided by 2,113 users on 10,197 movies (sparsity 96.03%). Due to the adoption of a content-based approach, we were forced to crawl the plot keywords and the summary of each movie from the IMDb web site. We removed those movies for which content was not available, and obtained a subset of HETREC2011-MOVIELENS-2K containing 2,642 movies, rated by 2,113 users, who provided 593,903 ratings (sparsity: 89.4%).

As regards the content associated with items, we analyzed the vocabulary of plot keywords and discovered that 98% of terms occurred in less than 60 items. The other terms, due to their lower discriminatory power, were removed. The resulting vocabulary contained 36,075 terms.

The same analysis on the vocabulary of the summaries led us to conclude that it was mostly made up of common terms, less distinctive than plot keywords, which are not very useful to discover hidden correlation among items. Therefore, only plot keywords were used in the experiments. The average number of keywords per item was 13.65, which represents the average number of clues given as input to KI for building the SAN corresponding to a movie (step 5 of Algorithm 1).

### 5.1.2. Metrics

Relevance in the context of recommendation is a user-specific notion which can be equated to the interest of users for items [78] and can be modeled as a binary concept: either an item is liked by a user or not. According to this idea, we define an item  $i$  as relevant to user  $u$  if the rating given by  $u$  on  $i$  is greater than the average value of all ratings provided by  $u$ . Given a recommendation list  $L$  of size  $N$ , the following metric defines the relevance of  $L$  as the ratio between the size of the subset of  $L$  that contains relevant items and the size of  $L$ :

$$Relevance@N = \frac{\sum_{i \in L} R(i)}{N} \quad (4)$$

where

$$R(i) = \begin{cases} 1 & \text{if } i \text{ is relevant;} \\ 0 & \text{otherwise.} \end{cases}$$

On the other hand, unexpectedness can be defined independently of the user, based on some standard prediction criteria [53]. We adopt two criteria: *popularity* and *item average rating*.

*Popularity* of the item  $i$  is defined as the ratio between the number of users who rated  $i$  and the total number of users in the dataset. According to this criterion, the item  $i$  is *unexpected* if its popularity score is below the average popularity computed across all the items in the dataset. This means that the average value of popularity allows to split items in the dataset in two parts: the *short head*, containing the most popular (and expected) items, and the *long tail*, containing the less popular (and unexpected) items. This criterion does not take into account whether ratings assigned to the items are positive or negative. This means that an item may be popular even though it is disliked by most of the users.

The other criterion takes into account the ratings assigned to each item. The *item average rating* of the item  $i$  is the average rating provided by the users in the dataset on item  $i$  (the value is normalized using the maximum of those values in the dataset)<sup>2</sup>. According to this criterion, the *short head* includes the items whose average rating is above the average rating computed across all the items in the dataset, while the *long tail* contains items below the average, i.e. those less liked by the users, and more likely unexpected.

By adopting popularity, 69% of items in the HETREC2011-MOVIELENS-2K dataset are unexpected, while by using average rating the percentage of unexpected items decreases to 44%. This means that recommending unexpected items may be more difficult when the criterion based on ratings is adopted.

The *Unexpectedness@N* metric defines the unexpectedness of  $L$  as the ratio between the size of the subset of  $L$  that contains just unexpected items and the size of  $L$  (Eq. 5):

$$Unexpectedness@N = \frac{\sum_{i \in L} U(i)}{N} \quad (5)$$

where:

$$U(i) = \begin{cases} 1 & \text{if } i \text{ is unexpected;} \\ 0 & \text{otherwise.} \end{cases}$$

*Serendipity@N* defines the serendipity of  $L$  as the ratio between the size of the subset of  $L$  that contains serendipitous items, i.e. those relevant and unexpected at the same time, and the size of  $L$  (Eq. 6):

---

<sup>2</sup>each item in the dataset has at least 50 ratings

$$Serendipity@N = \frac{\sum_{i \in L} S(i)}{N} \quad (6)$$

where:

$$S(i) = \begin{cases} 1 & \text{if } i \text{ is serendipitous;} \\ 0 & \text{otherwise.} \end{cases}$$

### 5.1.3. Evaluation Protocol for RWR-KI

Experiments were carried out using a *per user* evaluation, scheduled as follows:

1. Correlation matrix  $S$  is built using Algorithm 1;
2. Ratings of the active user  $u_a$  (for which recommendations must be provided) are split into a training set  $Tr$  and a test set  $Ts$ ;
3.  $Tr$  is used to set the starting vector  $q$  of the RWR algorithm described in Section 4.1. As proposed in [14], the RWR algorithm can be generalized by setting more than one single starting node. Thus, we set the value of nodes corresponding to all *relevant* items for  $u_a$  to 1, i.e. those whose ratings are greater than the average rating value of  $u_a$ . Next, we normalise  $q$  so that  $\|q\| = 1$ ;
4. Random Walk on  $S$  is performed, which returns the stationary probability vector corresponding to  $u_a$  of all the items in the dataset. The probability  $\alpha$  to return to the initial node is set to 0.8, as suggested in [46], in order to reduce random walks in the neighbouring elements of  $u_a$ ;
5. From this vector, all items in  $Ts$  (i.e. those for which the ground truth is known) are selected and ranked in descending order, with the top-ranked items having the highest probability scores that correspond to the most preferred ones;
6. Performance measures are computed on top- $N$  items and averaged for all users.

The dataset partitioning technique was 5-fold cross validation [45]. The dataset is divided into 5 disjoint partitions, and at each step 4 partitions were used to set  $q$ , whereas the remaining partition was used as the test set. These steps were repeated until each one of the 5 disjoint partitions was used as  $Ts$ . Results were averaged over the 5 runs.

#### 5.1.4. Compared Algorithms

We compared RWR-KI to the following algorithms:

- **RWR based on a correlation matrix built using similarity between plot keywords (RWR-KWD)**: this algorithm exploits only the content associated with items (endogenous knowledge) to build the correlation matrix. The comparison with RWR-KWD gives us the possibility to evaluate whether the exogenous knowledge introduced by KI allows the discovery of non-obvious correlations between items, which cannot be caught by exploiting endogenous knowledge exclusively. Top- $N$  recommendations are computed as for RWR-KI;
- **Item to Item Collaborative Filtering (I2ICF)**: even though collaborative filtering algorithms do not explicitly support the notion of unexpectedness, they constitute a fairly reasonable baseline because they perform reasonably well in terms of other performance measures besides classical accuracy measures [2, 11]. Top- $N$  recommendations are computed by performing rating predictions on  $T_s$ , using the algorithm described in [65]. Adjusted Cosine measure is adopted to assess similarity between items, while the neighborhood size has been set to 20, according to experiments in [35, 65];
- **Random**: this simple baseline randomly suggests  $N$  items from  $T_s$ . The rationale for including this algorithm in the evaluation is to compare our approach to a strategy inspired by the *blind luck* principle (Section 2).

#### 5.1.5. Discussion of results

Figure 5 shows the relevance-unexpectedness tradeoff for recommendation lists of different sizes. For readability, the graph reports the metrics defined in Section 5.1.2 as percentages. For each algorithm we report the results of four runs, each corresponding to a different size  $N = 5, 10, 15, 20$  of the recommendation list.

All algorithms are biased toward relevance, but RWR-KI and RANDOM seem better balanced than RWR-KWD and I2ICF, regardless of the criteria adopted for the definition of unexpectedness. In particular, RWR-KI significantly dominates all the algorithms ( $p < 0.001$  using the Wilcoxon test) as regards unexpectedness. The fact that RWR-KI outperforms RWR-KWD is certainly due to knowledge infusion since both the approaches use the

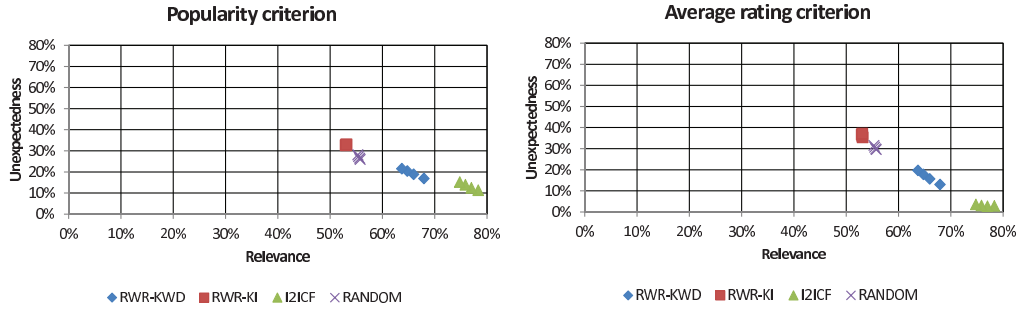


Figure 5: Percentage of relevant and unexpected items in recommendation lists of size  $N = 5, 10, 15, 20$ . The list size is not explicitly reported in the graph since the variance of results of four runs for each algorithm is relatively small.

same recommendation algorithm, thus confirming our intuition that exogenous knowledge might help to produce non-obvious suggestions. The worst performance of I2ICF in terms of unexpectedness confirms its bias towards mostly liked items [26].

These results suggest that in a multi-objective decision-making problem, RWR-KI would be the best approach, if unexpectedness is given higher weight than relevance, while I2ICF and RWR-KWD are more appropriate for recommendation scenarios where accuracy is more important than unexpectedness. In general, all the observed trends do not depend on the size of the recommendation lists, in the sense that the results of each algorithm for  $N = 5, 10, 15, 20$  are very similar.

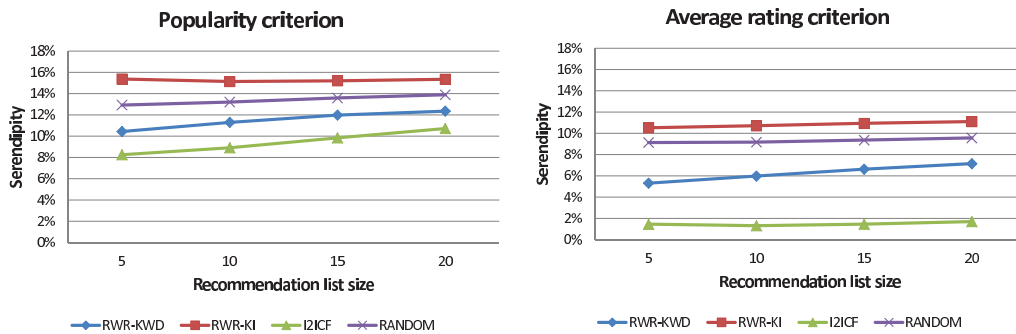


Figure 6: Percentage of serendipitous items in recommendation lists of different sizes.

Results of the evaluation of serendipity, presented in Figure 6, show that RWR-KI achieves the best performance compared to the other approaches

( $p < 0.001$ ), regardless of the criterion for defining unexpectedness. In general, algorithms with higher unexpectedness (RWR-KI and RANDOM) reach higher serendipity than those more biased towards relevance. It seems that there is more chance that the algorithms designed for finding unexpected items suggest relevant items as well, than algorithms designed for finding relevant items suggest unexpected items as well. As for the previous analysis, the size of the recommendation list does not affect the performance of the algorithms: the percentage of serendipitous items in lists of different sizes is quite uniform for all the tested algorithms.

To sum up, the main outcomes of the experiment are:

- RWR-KI dominates all the tested algorithms in the task of finding serendipitous recommendations, showing better balancing of relevance and unexpectedness;
- the size of the recommendation list is not a significant factor for providing users with a higher ratio of serendipitous items, meaning that even short lists (size 5 or 10) contain serendipitous items.

Although this comparative evaluation allowed us to assess the ability of suggesting serendipitous items, a deeper analysis of the recommendation lists is required to decide whether they can be provided to users as final recommendation sets. For example, a list containing serendipitous items, as well as a high percentage of not relevant ones, is not suitable as a recommendation set. Therefore, in the next section we present a study of the distribution of relevant, unexpected and serendipitous items within the recommendation lists.

#### 5.1.6. Anatomy of recommendation lists

The analysis focuses on recommendations lists of size 5 or 10, which are mostly used as final recommendation sets shown to the user. We define a list as *serendipitous* if at least 20% of its items are serendipitous. Figure 7 presents the number of *serendipitous lists* having size=5 provided by each tested algorithm, together with the corresponding distribution of serendipitous items.

Regardless of the criteria for unexpectedness, RWR-KI produces the highest number of serendipitous lists. By looking at the popularity criterion, 48% of users (1025 out of 2113) received at least one serendipitous recommendation, while this percentage decreases to 36%, when average rating is adopted.

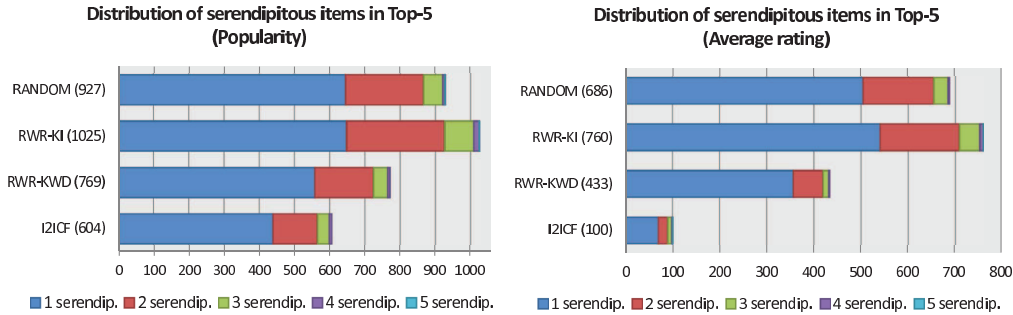


Figure 7: Distribution of serendipitous items inside serendipitous lists of 5 items, for both the unexpectedness criteria. For each tested algorithm, the number of serendipitous lists is reported in parenthesis.

It is worth to note the good performance of RANDOM, which provided 44% of users with serendipitous lists (32% when using average rating). Once again results confirm that I2ICF is biased towards most popular or most liked items: only 28% of the lists are serendipitous (5% with average rating). As for the distribution of serendipitous items in the lists, most of the top-5 recommendations produced by all the algorithms contains only 1 serendipitous item, but RWR-KI is the algorithm that produces the highest number of lists having more than 1 serendipitous suggestion.

Figures 8 and 9 show the composition of serendipitous lists in terms of percentage of relevant or unexpected items.

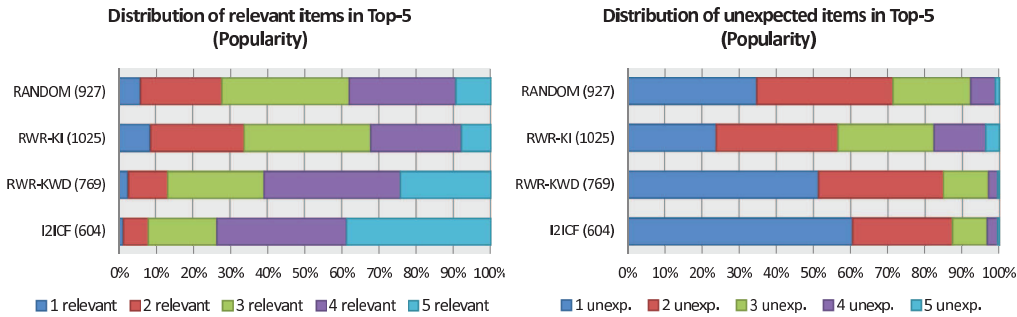


Figure 8: Distribution of relevant/unexpected items inside serendipitous lists of 5 items. Unexpectedness criterion is popularity.

Results for both unexpectedness criteria confirm the primacy of RWR-KI and RANDOM. Over 40% of the lists produced by RWR-KI have at least



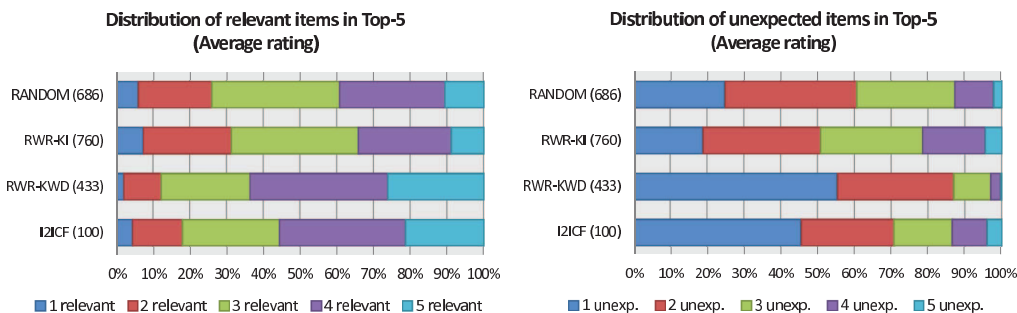


Figure 9: Distribution of relevant/unexpected items inside serendipitous lists of 5 items. Unexpectedness criterion is average rating.

3 unexpected items, while over 60% contain at least 3 relevant items. The RANDOM approach has similar performance. The same analysis performed on top-10 recommendations revealed similar trends (results are not reported for brevity).

Finally, we present some results about *not serendipitous* lists. Indeed, those recommendation lists may be worthless if they do not contain any items the users may like, and there is the risk that the advantage of surprising some users is obtained at the price of disappointing most of them.

Figure 10 shows the box plot of the number of relevant and unexpected items in lists produced by each algorithm, using popularity as unexpectedness criterion. Upper and lower ends of boxes represent the 3<sup>rd</sup> and 1<sup>st</sup> quartile, respectively. Whiskers extend to the most extreme data point which is no more than 1.5 times the interquartile range. Median is depicted with a solid line. Empty circles are outliers.

The main outcome is that for all the algorithms the median of the number of relevant items is greater than or equal to 2, which is acceptable for list of size 5. The number of unexpected items is low for all the algorithms, with a slightly better performance by RWR-KI. Similar results (not reported for brevity) were observed for the other unexpectedness criterion.

Even if the results clearly show that RWR-KI overcomes the other algorithms, a surprisingly good performance is observed for the random strategy. This raises the following issues: *is our strategy actually different from making random suggestions? Does the difference in performance justify the difference in complexity of the strategies?* As a consequence, we conducted a study with real users, in which these two strategies are compared, aiming at assessing

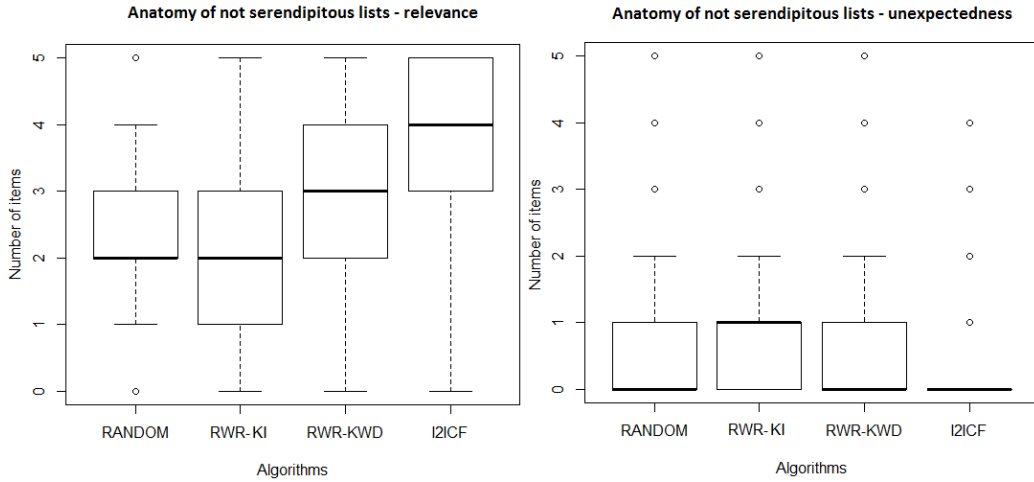


Figure 10: Anatomy of not serendipitous lists. Unexpectedness criterion is popularity.

the actual *perception* of serendipity of recommendations and their *acceptance* by users.

## 5.2. User study

The aim of the study is twofold:

- to assess the *acceptance* of recommendations produced by RWR-KI and RANDOM, the algorithms that excelled on the in-vitro evaluation. This is achieved by gathering explicit feedback from users through a questionnaire. Results are presented in Section 5.2.3;
- to measure the *perception* of serendipity of recommendations. This is achieved by gathering implicit feedback from users through a tool able to detect their emotions when exposed to recommendations. Results are presented in Section 5.2.4.

### 5.2.1. Users and dataset

The experimental units were 40 master students in engineering, architecture, economy, computer science and humanities; 26 male (65%) and 14 female (35%), with an age distribution ranging from 20 to 35. None of them had been previously exposed to the system used in our study.

We collected from [IMDb.com](http://IMDb.com) some details (poster, keywords, cast, director, etc.) of 2,135 movies released between 2006 and 2011. The size of the

vocabulary of plot keywords was 32,583 and the average number of keywords per item was 12.33, which is comparable to that of the in-vitro experiment.

### 5.2.2. Procedure

We ran a *between subjects* controlled experiment, in which half of the users was randomly assigned to test RWR-KI, and the other half was assigned to test the RANDOM approach (control group). The experimental units were blinded since they did not know which algorithm is used to generate their recommendations. The recommendation algorithm was the only *independent variable* in the experiment, while the quality metrics used to assess the acceptance of recommendations and the perception of serendipity were the *dependent variables*.

Users interacted with a web application which showed details of movies randomly selected from the dataset and collected ratings on a 5-point Likert scale (1=strongly dislike, 5=strongly like). Once the user provided 20 ratings, if she was assigned to the RWR-KI group, the ratings were used to set the starting vector of the random walk algorithm, as described in Section 5.1.3, otherwise the ratings were simply discarded. The rating step was performed for both the groups in order to avoid any possible bias. Five recommendations are given to each user in the two groups, showing the poster and the title of the movies.

Recommended items were displayed one at a time, and users were asked to reply to two questions to assess their *acceptance* in terms of *relevance* and *unexpectedness*. Relevance was evaluated by asking the standard question “Do you like this movie?”, while for unexpectedness the question was: “Have you ever heard about this movie?”. If the user never heard about that movie, the system allowed her to have access to other movie details, such as cast, director, actors and plot, and the answer of the user to the first question was interpreted as the degree of potential interest in that movie. If a user liked a recommended item, and she never heard about that movie, it is likely a pleasant surprise for her, and hence it fits with our definition of serendipitous recommendation. Users were not asked directly if they found some recommendations surprising, since it might be difficult for them to explicitly assess the unexpectedness or the surprise. On the other side, we decided to analyze the signals coming from their facial expressions in order to get an implicit signal of surprise.

Whenever an item was shown to the user, the system started recording a video of the face of the user, which was stopped when the answers to both

the questions were provided. Hence, for each user 5 videos were collected which have been analyzed by means of the Noldus FaceReader™ system to assess her *emotional response* to that suggestion. Obviously, users did not know in advance that their facial expressions would be analyzed. They were just informed that a high definition web camera would have recorded their interaction with the system. At the end of the experiment, we disclosed the goal of the evaluation, and asked users the permission to analyze the videos.

### 5.2.3. Analysis of the questionnaires

The perceived quality of the two algorithms is assessed by computing the metrics of relevance, unexpectedness and serendipity defined in Section 5.1.2. According to the ResQue model proposed in [16], these metrics belong to the category *Perceived System Qualities*, subcategory *Quality of Recommended Items*. *Relevance*, also called *perceived accuracy*, measures the extent to which users feel the recommendations match their interests and preferences. Unexpectedness and serendipity refer to *novelty* or *discovery* dimension of the ResQue model, and represent the extent to which users receive new, interesting and surprising suggestions.

Results are reported in Table 1. The main outcome is that RWR-KI outperforms RANDOM in terms of serendipity, with a more marked difference compared to the in-vitro evaluation. The value of serendipity is noteworthy since almost half of the recommendations are deemed serendipitous by users. Furthermore, RWR-KI shows a better relevance-unexpectedness trade-off than RANDOM, which is more unbalanced towards unexpectedness.

Table 1: Metrics computed on the answers provided in the questionnaire. A Mann-Whitney U test confirmed that the results are statistically significant ( $p < 0.05$ ).

<b>Metric</b>	<b>RWR-KI</b>	<b>RANDOM</b>
Relevance	0.69	0.46
Unexpectedness	0.72	0.85
Serendipity	0.46	0.35

Figure 11 presents the distribution of serendipitous items within *serendipitous lists*.

Almost all users (19 out of 20) in the two groups received at least one serendipitous suggestion, but the composition of the lists provided by the two algorithms is different. Most of the RWR-KI lists contains 2 or 3 serendipitous items, while most of those randomly produced has only 1 or 2 serendipitous items.

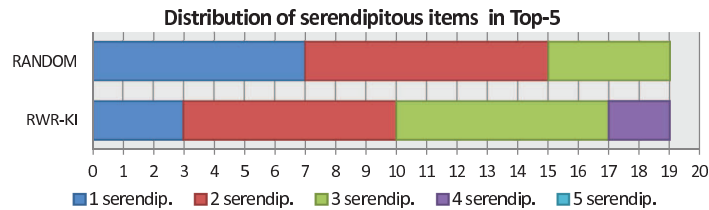


Figure 11: Distribution of serendipitous items inside serendipitous lists.

itous items. Moreover, by analyzing only relevance, we observed that 79% of RWR-KI lists contains at least 3 relevant items, while this percentage decreases to 42% for RANDOM (the complete analysis of relevance is not reported for brevity).

The main conclusion of the questionnaire analysis is that recommendations produced by RWR-KI seem to be well accepted by users, who perceived the difference with respect to random suggestions.

#### 5.2.4. Analysis of the user emotions

The FaceReader™ recognizes the six categories of emotions proposed by Ekman [22], i.e. happiness, anger, sadness, fear, disgust and surprise, besides a neutral state. The classification accuracy is about 90% on the Radboud Faces Database [47].

Given a video of  $t$  seconds, the output is the distribution of a person’s emotions during time  $t$ , as shown in Figure 12.

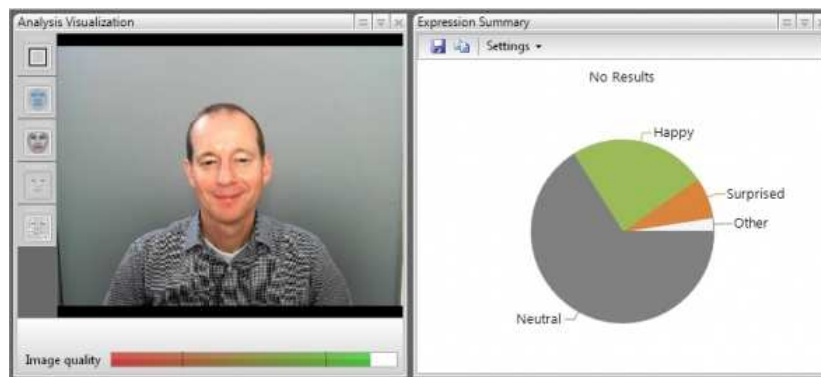


Figure 12: Analysis of emotions by FaceReader™.

Our hypothesis is that facial expressions of users might convey a mixture

of emotions that helps to measure the perception of serendipity of recommendations. We associated serendipity with *surprise* and *happiness*, the only two emotions, among those suggested by Ekman, which are reasonably related to the pleasant surprise serendipity should excite. In the ResQue model this quality is called *attractiveness*, and refers to recommendations capable of evoking a positive emotion of interest or desire.

We filtered out 41 (out of 200) videos in which users provided feedback on a recommendation in less than 5 seconds, therefore actually evaluating the suggestion in a shallow way. For each one of the remaining 159 videos, FaceReader™ computed the set of detected emotions together with the corresponding duration. The distribution of emotions associated with serendipitous recommendations provided by RWR-KI and RANDOM, reported in Figure 13, is computed as follows: for each emotion  $e_i$  detected during the visualization of serendipitous recommendation  $r_j$ , we recorded its duration  $d_{ij}$ . Then, the total duration of  $e_i$  is obtained as  $T_{e_i} = \sum_j d_{ij}$ . The percentages reported in Figure 13 are computed as the ratio between  $T_{e_i}$  and the total duration of videos showing serendipitous recommendations.

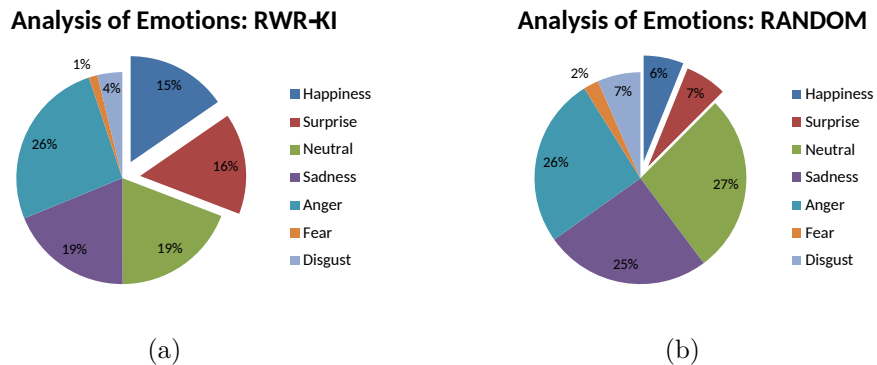


Figure 13: Analysis of emotions associated with serendipitous recommendations.

We note that users testing RWR-KI revealed more surprise and happiness than users receiving random suggestions (16% vs. 7% for surprise, 15% vs. 6% for happiness), and this confirms the results of the questionnaires: RWR-KI provided more serendipitous suggestions than RANDOM.

The distribution of emotions over non-serendipitous suggestions, computed as for serendipitous ones, is reported in Figure 14. We observe that there is a general decrease of surprise and happiness compared to serendipitous ones for both the algorithms.

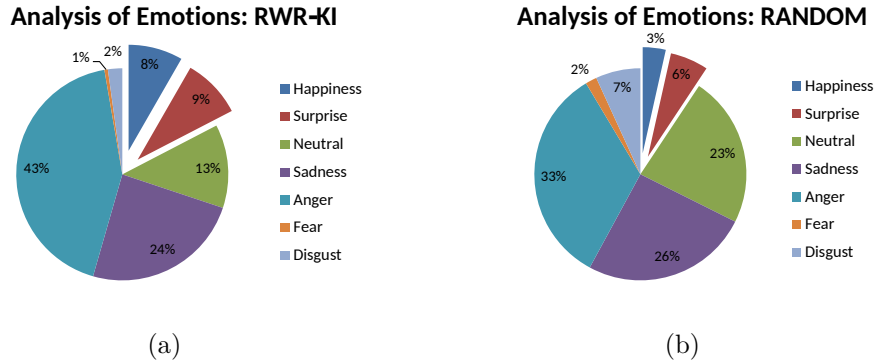


Figure 14: Analysis of emotions associated with non-serendipitous recommendations.

In general, we can observe that there is a marked difference of positive emotions between the two algorithms, as well as between serendipitous and non-serendipitous suggestions, regardless of the algorithm.

We were quite puzzled by the high percentage of negative emotions (sadness and anger), which are the dominant ones besides the neutral state. The analysis of videos revealed that the high presence of negative emotions might be due to the fact that users were very concentrated on the task to accomplish and assumed a troubled expression. However, it is also known that *personality* is a factor which might affect the way people express emotions [64]. The high presence of negative emotions might be due to the involvement of users with personality traits characterized by the tendency to experience unpleasant emotions easily. Hence, we performed a deeper investigation by asking the users to answer the Big Five personality traits questionnaire [18] and to indicate to what extent they agree with each statement on a 3-point Likert scale (low, medium, high), so that we can assess their personality characteristics, i.e., *conscientiousness*, *agreeableness*, *extroversion*, *neuroticism* and *openness to experience*. After that, we studied possible relationships between personality traits and emotions experienced by users. To this purpose we used two dichotomous categorical variables for emotions, one for positive emotions (i.e. happiness and surprise), and one for negative ones (i.e. sadness and anger). We did not take into account neither the neutral state, nor fear and disgust since their occurrence was negligible (see Figures 13 and 14). We also used one categorical variable for each personality trait, and we computed the joint distribution of emotions and personality traits using a contingency table containing the number of users with a specific value of

that personality trait (e.g. neuroticism=high or extraversion=low) who experienced positive or negative emotions. The analysis does not report any significant relation between the two variables and this led us to conclude that the high percentage of negative emotions does not depend on the presence of users who tend to experience unpleasant emotions easily, such as neurotic ones.

In order to better evaluate the results obtained by the analysis of the user emotions and to what extent they are actually able to identify serendipitous recommendations, we compared them with the results obtained by administering the questionnaires. The analysis was performed on the 159 recommendations for which FaceReader<sup>TM</sup> was able to detect a reliable set of emotions (Section 5.2.4). In order to deem a recommendation as *serendipitous* by taking into account the emotions conveyed by facial expressions we performed the following steps: 1) we grouped emotions into three groups, i.e. positive - happiness and surprise, negative - sadness and anger, and very negative - fear and disgust; 2) we discarded the time the user expressed a neutral emotion and we computed the average duration of the three groups of emotions; 3) we deem a recommendation as serendipitous if the total duration of positive emotions is greater or equal than the average duration computed at the previous step. We arranged the results of the questionnaires and those obtained by FaceReader<sup>TM</sup> in a  $2 \times 2$  contingency table (Table 2). 69 out of 159 recommendations were deemed as serendipitous by the explicit feedback provided by the users, while the remaining 90 were classified as non-serendipitous. The analysis of the user emotions correctly classifies 30 out of 69 serendipitous recommendations, and 71 out of 90 non-serendipitous ones.

Table 2: Contingency table. **Q**=Questionnaires, **E**=Emotions.

	<b>Serend. (E)</b>	<b>Non-serend. (E)</b>	<b>Row total</b>
<b>Serend. (Q)</b>	30	39	69
<b>Non-serend. (Q)</b>	19	71	90
<b>Column total</b>	49	110	159

We used the Cohen’s kappa coefficient ( $K$ ) to measure the pairwise agreement between the classification of serendipitous and non-serendipitous items obtained by the questionnaires and that obtained by the analysis of the user emotions.  $K$  is a more robust measure than simple percent agreement calculation, since it takes into account the agreement occurring by chance. Data in Table 2 show a moderate strength of agreement ( $K = 0.232$ ).



Despite the limitation of the study due to the low number of participants, the preliminary results show on one side the ability of the implicit feedback acquired by the facial expressions to discriminate between serendipitous and non-serendipitous suggestions provided by the two different algorithms, and on the other side a tendency of the user emotions to agree with the results obtained by the explicit feedback provided by the questionnaires, thus revealing that emotions could help to assess the actual perception of serendipity.

## 6. Related work

We support the idea of *programming for serendipity*, proposed by Campos and de Figueiredo, who suggested to introduce serendipity in information seeking systems in an *operational* way [12]. Although some efforts for enhancing search engines and recommender systems with operationally-induced serendipity have been made, no computational model for serendipity stood out. In the following sections, we analyze the literature of recommender systems on this topic and then we review some large-scale commercial systems including strategies for serendipitous discoveries. Furthermore, since we investigated the problem whether facial expressions could help to assess the actual perception of serendipity, we also discuss related literature on using implicit affective feedback in recommender and search systems.

### 6.1. Programming for serendipity

Determining the filter bubble and finding unexpected recommendations out of the bubble is one of the most common strategies of programming for serendipity. One of the first attempts in that direction was the development of MAX [12], a software agent that mimics the browsing behavior of users navigating the Web just for the sake of wandering. MAX exploits interests contained in user profiles, one for each domain of interest (the bubble), and adopts retrieval techniques and heuristic search to find useful and not known information on the Web for stimulating serendipitous insights. The wandering process starts with a Google search of randomly chosen words from the profile in order to select pages that have more cross-domain integration, which allow to spark new interests. A similar approach is described in [41], in which a method for locating unexpected items from clusters similar to a user cluster (the bubble) has been explored. The AURALIST framework for music recommendation is also based on the same strategy [87]. A declustering algorithm aims to determine musical bubbles (clusters of artists the user

listens to) in order to recommend artists outside established cluster groups. The *Outside-The-Box* system [1] suggests items not falling into regions of interests (the bubble) the user is familiar with. Unfamiliarity can arise either when a user does not like items in that region and chooses not to rate them, or when the user has not been exposed enough to the region. In the latter case that region is likely to contain serendipitous items.

Another strategy generally adopted to obtain unexpected results is that of mixing together different features. This resembles the intuitive action of mixing colors, ingredients, and sounds, that may yield unexpected results. Similarly to the approach proposed in MAX, Oku and Hattori propose a fusion-based recommender system which suggests items that have the mixed features of two user-input items [55].

Some collaborative approaches have been developed as well. In [43, 44], the authors propose a strategy for suggesting surprisingly interesting items to a user by identifying purchase history logs of users who have similar preferences and a high degree of purchase precedence (i.e., purchasing the same items earlier) relative to that user. These users are called “innovators” since they become aware of items well before their release, and purchase them soon after their release. The method assigns higher weights to innovators, and can rank these novel items first in the recommendation list. This should help to find items that match the latest user preferences, but also items she might not have otherwise discovered.

Some work adopt graph-based methods, similarly to our approach, but rely only on *endogenous* knowledge to discover serendipitous items. The TANGENT recommendation algorithm [56] selects nodes in a graph connecting users with movies they like, giving high scores to nodes that are well connected to the older choices of the user, and at the same time well connected to unrelated choices, in order to broaden the user horizons. This strategy allows the recommendation of items close enough to a user’s current interests, but also towards a new area that the user has not discovered yet. This is similar to our approach, in which the Random Walk with Restarts algorithm exploits user preferences as starting nodes and the correlation matrix built by KI as a transition probability matrix. Graph-based techniques have been used for recommending serendipitous mobile apps in [9]. The approach generates serendipitous recommendations based on apps installed on a target user’s phone and using an app-app similarity graph. The main intuition behind the method is that, if there exists a path connecting two apps on a user’s phone, apps along this path which are not already downloaded by the

user, are good candidates for serendipitous recommendations. The shortest-path connecting two apps is selected in order to reduce the overall cost to traverse from a given source node to a given destination node. The cost is represented by the similarity between apps and the edges with low similarity are taken into account to reach the destination. As in our approach, the similarity between items does not depend on rating patterns, but rather on item descriptions, albeit no external knowledge sources have been used. Similarly to the previous approach, [75] describes a strategy for guided exploration of music preferences that allows serendipitous encounters. The process starts by allowing users to select target genres towards which they want to initiate an exploration (intent of the user). User preferences graphs built adopting similarities based on preferred genres are adopted to detect the shortest path towards a selected target genre. Using the selected path a predefined number of artists are selected per path node in order to form a sequence of suggestions that are finally presented to the user. The main problem of this kind of approaches is that the lack of a *general* background knowledge limits the reasoning process to the specific domain of the recommender and, as a consequence, also narrows the search space of unexpected items.

Other approaches try to introduce serendipity in the recommendation process by still using graph-based techniques, but relying on *exogenous* knowledge, as in our approach. In [51], the author proposes to exploit the rich link structure of the Linked Open Data cloud in order to explore deep and novel connections between concepts, with the aim of identifying *interesting patterns* (i.e. content patterns) in graphs connecting information about user profiles and program metadata that would lead to serendipitous recommendations. Several ways for finding content patterns are proposed, even though the work is yet in a very preliminary stage. The maximization of serendipity is also investigated in [79], where the authors propose an approach based on the definition of a huge set of unexpected and surprising relationships between items, modeled on the ground of the properties encoded in DBpedia. Specifically, in this work the authors introduce the concept of *renso* relationships between two concepts, i.e. concepts connected through a n-hop path in the Linked Open Data cloud. The algorithm generates location-based music recommendations based on the identification of other songs connected with the current position of the users by browsing the novel graph of serendipitous *renso* relationships. The distinctive feature of these approaches is that the “reasoning for serendipity”, explicitly driven by *predefined* relationships in DBpedia, is limited in some way. We prefer to adopt a more “open”

approach, which introduces both WordNet and Wikipedia concepts in a reasoning process, which is “free” of specific relationships, following implicit ones discovered among items through the exogenous knowledge.

Graph-based techniques have been also used in the context of search in order to find serendipitous results, especially combined with User Generated Content (UGC). In [10], the authors build an entity network by extracting data from Wikipedia and Yahoo! Answers, and propose an algorithm based on random walk with restart to retrieve entity recommendations from the network. The network is enriched with metadata about *sentiment*, writing quality, and topical category. In particular, the authors investigate whether entities which convey more emotion provide better results, and they found that it is not enough to select only emotionally-evocative items in order to catch the user’s interest in terms of unexpected results. Other authors suggest an emotional-oriented search strategy, that could allow to discover unexpected results that “make the user happy” [34]. In their position paper, Hauff and Houben suggest to use sentiment analysis techniques to identify emotional topics within Wikipedia articles and to select those evoking emotional feelings. Sentiment analysis is supported by semantic analysis of outgoing links to measure how related an article is to the other articles it contains links to. This can help to find seemingly unrelated articles that can be suggested as unexpected results. Other types of UGC that can promote the discovery of unexpected information are folksonomies. For instance, in [30], the author shows that serendipity in Flickr can be improved through the exploitation of Wikipedia URLs as translation sources, while in [84] it is proposed a serendipity-oriented recommendation method that exploits tags as metadata attached to items, similarly to the idea proposed in [70]. These works showed that the exploitation of UGC could be beneficial to serendipitous discovery. Especially Wikipedia has the potential to support “informed” search which can drive the user toward unexpected results. Compared to other type of UGC, such as folksonomies, Wikipedia has the advantage of showing a high level of accuracy, due to the presence of editors, that makes it a more trustworthy source of information. This is another reason why we ground our KI process on Wikipedia, but it could be interesting to include also other UGC sources, especially those that can provide emotional information, such as user comments or reviews.

Other approaches leveraging graph-based knowledge representation exist, even though they are not specifically focused on the serendipity improvement. However, they propose interesting strategies to exploit exogenous knowledge

and come up with new features which could help to promote serendipity. In [59], DBpedia properties are exploited to compute the semantic similarity between artists in a music recommender system, while in [54], DBpedia is used to enrich playlists extracted from a Facebook profile with new related artists by taking into account shared properties (e.g. genre or musical category of the artist). A graph-based representation is also adopted in [57], where both a collaborative and a content-based data model exploiting DBpedia are adopted to extract semantic path-based features connecting users to items, where the more paths between a user and an item, the more the relevance of that item to that user. A graph-based approach aiming at suggesting items with the best trade-off between accuracy, similarity, diversity is presented in [72]. The model is based on the definition of “a cost flow” over the graph, which is in turn defined according to several criteria, such as the similarity between items, the fact that a specific item is in the long tail, and a measure of how wide is the range of interests of a specific user. Given such a cost flow, each user is recommended the items with the lower cost.

Finally, we want to point out the main differences between our KI approach and the one previously adopted in [39], which grounded the search for potentially serendipitous items on the similarity between the item descriptions and the user profile. The recommendation algorithm categorized an item as liked or disliked according to the similarity with the user profile. The idea was to suggest those items on which the categorization was more uncertain, since they were likely not known to the user and might result to be the most serendipitous ones. Therefore, the approach relied exclusively on endogenous knowledge, i.e. item descriptions, and did not exploit any additional knowledge source.

## 6.2. Commercial Systems implementing Serendipity Strategies

The concept of serendipity is very interesting also for the most important Web companies such as eBay, Amazon, Google and Facebook.

eBay is testing serendipitous shopping in the context of the *Discover* project, an alternate view of eBay’s inventory [82]. With Discover, eBay is proposing products the user will be interested in and that are at the same time a complete surprise to her, by avoiding very popular items. In order to identify those items, the algorithm implemented in Discover takes into account several factors, such as how much people interact with a listing (clicking or returning to it, forwarding it to friends), or the textual analysis (a longer description of the product indicates more passion about the item). The

hope is to recreate the unexpected discoveries that are familiar to shoppers in brick-and-mortar stores, in which “serendipity is built into the layout of the store”.

Similarly, Amazon tries to replicate on the online library the serendipity experiences that sometimes characterize the offline activity of visiting a library, by opening up the content for more exposure, which may lead to unexpected discoveries. For example, Amazon uses the *Statistically Improbable Phrases (SIPs)* - the most distinctive phrases in the text of books - which allow an exploration starting from a content item, that can lead to the discovery of additional unexpected content.

Another interesting service that, similarly to our system, exploits exogenous knowledge is that developed by Clever Sense<sup>3</sup> [48], recently acquired by Google. The heart of the platform is the Serendipity Engine which learns interests and preferences of users based on their interactions with various sources, including Facebook and Twitter, in order to provide more surprising recommendations.

This is the goal of Facebook as well, that acquired Glancee<sup>4</sup>, in order to make easier to meet interesting people around you, empowering serendipity and pioneering social discovery.

### 6.3. *Implicit Affective Feedback in Search and Recommender Systems*

The advances in computer vision techniques and algorithms for emotion detection have enabled the usage of facial expressions as a direct source of information about the affective state of the user [24, 86]. This kind of implicit affective feedback has been exploited in several domains, such as consumer behavior research [80], to detect the emotional response of the user to an observed or consumed item. In recommender systems literature, emotional feedback is mainly associated with multimedia content [73, 76] and plays different roles related to the acquisition of user preferences:

1. As a source of affective metadata for item modeling and building a preference model;
2. As an implicit relevance feedback for assessing user satisfaction.

As for the first issue, the idea is to acquire affective features that are included in the item profile and might be exploited for user modeling. In [76], a

---

<sup>3</sup>[www.thecleversense.com](http://www.thecleversense.com)

<sup>4</sup>[www.glancee.com](http://www.glancee.com)

feature vector is acquired, that represents the valence, arousal and dominance dimensions (identified by Russell [63]) of the emotive response of a user to an item; then the user model is inferred by machine learning algorithms trained on the item profiles and the explicit ratings given to the consumed items. The detected emotion can be used in two ways: item categorization (the item  $i$  is funny because it induces happiness in most of the users) and user modeling (the user  $u$  likes items that induce sadness). In [40], a probabilistic emotion recognition algorithm based on facial expressions was employed to detect emotions of users watching video clips. The level of expressed emotions associated with items were used as features to detect personal highlights in the videos. The main issue that these and other similar studies addressed [83] is the identification of a valid set of affective features that allows the definition of an effective user model for the canonical (relevant/non-relevant) item categorization. The main challenge from both a user modeling and decision making perspective is how to represent the whole affective state of the user in terms of emotions, mood, and personality.

As for the second issue, the main motivation for assessing user’s relevance by means of emotions detection techniques is that, since satisfaction is an internal mental state, techniques that can disclose feelings without any bias are expected to be a reliable source of implicit feedback. In fact, the emotional response is hardly alterable by the user. Furthermore, face detection is unobtrusive because usually the user is monitored by a camera, and then recorded videos are analyzed by a facial expression recognition system. Pioneer studies on this topic are those made by Arapakis et al. [6, 7, 8]. They introduced a method to assess the topical relevance of videos in accordance to a given query using facial expressions showing users satisfaction or dissatisfaction. Based on facial expressions recognition techniques, basic emotions were detected and compared with the ground truth. They investigated also the feasibility of using reactions derived from both facial expressions and physiological signals as implicit indicators of topical relevance. We adopt a similar approach for serendipity detection, but we consider also signals that might convey surprise, besides user satisfaction. There are some attempts to design user studies with real users for assessing serendipity, especially for music recommendation or retrieval [66, 87], but they adopt the traditional approach of filling in a survey with specific options related to serendipity (e.g.: “Something I would never have listened to otherwise”). To the best of our knowledge, our approach is the first attempt to associate serendipity with implicit feedback detected from facial expressions.

## 7. Conclusion and future work

In this paper we have proposed a strategy for dealing with the overspecialization problem of recommender systems. We have defined an algorithm, named RWR-KI, which exploits a Knowledge Infusion process, for enhancing a graph-based recommendation algorithm with the aim of suggesting serendipitous, i.e. accurate and unexpected, items. Offline experiments on a benchmark dataset demonstrate that the proposed algorithm produces more serendipitous suggestions than other collaborative or content-based recommendation algorithms, showing better balancing of relevance and unexpectedness. Furthermore, about half of the recommendation lists of 5 items computed by RWR-KI contain at least one serendipitous suggestion, while the rest of the lists contain 2 relevant items on average. This allows us to conclude that those lists are suitable as final recommendation sets.

A preliminary user study was performed to assess both the acceptance and the actual perception of serendipity of recommendations, through the administration of questionnaires and the analysis of users' emotions, respectively. The main result is that recommendations produced by RWR-KI are well accepted by users, since 69% of suggested items were deemed relevant, and 46% were judged as serendipitous. As regards the analysis of emotions, the results showed a moderate agreement between the positive feedback acquired through the questionnaires and the presence of positive emotions, such as happiness and surprise, thus revealing that they could help to assess the actual perception of serendipity.

Future work regards both the evaluation and the extension of KI with other knowledge sources.

As for the evaluation, we are planning to extend the user study by involving a larger sample of real users (not only students), in order to collect a higher number of observations and to increase the significance of the correlation between the implicit emotional feedback and the explicit feedback provided by questionnaires. Another point that could be further investigated is how the individual knowledge sources alone contribute to the system effectiveness. This aspect will be deepened through separate experimental sessions in which the background memory includes only one CU repository at a time. Therefore, each run of the experiment will evaluate KI based only on one knowledge source. Furthermore, we would like to evaluate knowledge infusion combined with other recommendation algorithms. For instance, the item-item collaborative filtering algorithm [65] could be enhanced by KI by



replacing similarity scores between items with “relatedness” scores computed by KI (and stored in the correlation matrix).

As for possible KI improvements, we would like to extend the knowledge repository, in order to include in the reasoning process also *specific* (domain-oriented) knowledge, besides the general knowledge provided by Wikipedia and WordNet. We foresee three possible directions:

1. *structured* exogenous knowledge - as discussed in Section 6.1, we do not consider any *predefined* connection among concepts. In fact, links in the SAN connect clues to CUs, and then CUs are connected to their most representative keywords based only on similarity scores. The reasoning process could be improved by creating links among CUs, representing relationships among concepts defined within knowledge sources. For instance, the *hypernymy* relation among synsets allows to expand the SAN by including also other CUs, besides those most similar to clues, according to some generalization/specialization strategy. To this purpose, other structured sources could be included such as Freebase<sup>5</sup>, BabelNet<sup>6</sup> and DBpedia;
2. *endogenous* knowledge from item descriptions - concepts recognized within textual description of items could be also included in the reasoning process but, while for Wikipedia or WordNet it is quite straightforward to define CUs, i.e. articles or synsets, this task is not so immediate for item descriptions, because there are different possibilities, corresponding to distinct features of items. Whatever strategy is adopted, it is still necessary to define a way to connect in the SAN this new kind of *endogenous* CUs with *exogenous* CUs representing general concepts. For instance, let us suppose that one CU is created from the textual description of an item, e.g. the plot summary of a movie. One possibility for connecting the item CU to WordNet CUs is to adopt Word Sense Disambiguation (WSD) on the text, in order associate synsets to words, as shown in [33]. In this way, an item could be connected to all the CUs corresponding to synsets recognized by WSD;
3. *endogenous* knowledge from users (e.g. user reviews) - another kind on endogenous knowledge that could be considered is User-Generated Content such as comments, discussions or reviews. Currently, we are

---

<sup>5</sup>[www.freebase.com](http://www.freebase.com)

<sup>6</sup>[babelnet.org](http://babelnet.org)

working on NLP methods for aspect-based sentiment analysis [37]. We have developed a method (not yet published) for recognizing specific aspects discussed by users in the reviews about an item, such as food, service, location for a restaurant, and computing the sentiment they expressed on those aspects. A profile of the item is built in terms of aspects and corresponding summarized opinions (e.g. food: positive, service: positive, location: negative). The idea is that the item profile could be a CU, therefore two items could be connected in the SAN because they have similar profiles (i.e. users have similar opinions of their attributes), according to some similarity measure, even if they serve different types of food.

## References

- [1] Z. Abbassi, S. Amer-Yahia, L.V.S. Lakshmanan, S. Vassilvitskii, C. Yu, Getting Recommender Systems to Think Outside the Box, in: Proceedings of the ACM Conference on Recommender Systems, RecSys 2009, New York, USA, ACM, 2009, pp. 285–288.
- [2] P. Adamopoulos, A. Tuzhilin, On Unexpectedness in Recommender Systems: Or How to Expect the Unexpected, in: Proceedings of the ACM RecSys 2011 Workshop on Novelty and Diversity in Recommender Systems (DiveRS), volume 816 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2011, pp. 11–18.
- [3] G. Adomavicius, Y. Kwon, Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques, *IEEE Transactions on Knowledge and Data Engineering* 24 (2012) 896–911.
- [4] G. Adomavicius, N. Manouselis, Y. Kwon, Multi-Criteria Recommender Systems, in: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Eds.), *Recommender Systems Handbook*, Springer, 2011, pp. 769–803.
- [5] J.R. Anderson, A Spreading Activation Theory of Memory, *Journal of Verbal Learning and Verbal Behavior* 22 (1983) 261–295.
- [6] I. Arapakis, K. Athanasakos, J.M. Jose, A comparison of general vs personalised affective models for the prediction of topical relevance, in:

Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, ACM, 2010, pp. 371–378.

- [7] I. Arapakis, I. Konstas, J.M. Jose, Using facial expressions and peripheral physiological signals as implicit indicators of topical relevance, in: Proceedings of the 17th International Conference on Multimedia 2009, Vancouver, Canada, ACM, 2009, pp. 461–470.
- [8] I. Arapakis, Y. Moshfeghi, H. Joho, R. Ren, D. Hannah, J.M. Jose, Integrating facial expressions into user profiling for the improvement of a multimodal recommender system, in: Proceedings of the 2009 IEEE International Conference on Multimedia and Expo, ICME 2009, New York City, NY, USA, IEEE, 2009, pp. 1440–1443.
- [9] U. Bhandari, K. Sugiyama, A. Datta, R. Jindal, Serendipitous recommendation for mobile apps using item-item similarity graph, in: Information Retrieval Technology - 9th Asia Information Retrieval Societies Conference, AIRS 2013, volume 8281 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 440–451.
- [10] I. Bordino, Y. Mejova, M. Lalmas, Penguins in sweaters, or serendipitous entity search on user-generated content, in: Proceedings of the 22Nd ACM International Conference on Conference on Information and Knowledge Management, CIKM '13, ACM, New York, NY, USA, 2013, pp. 109–118.
- [11] R.D. Burke, Hybrid Recommender Systems: Survey and Experiments, *User Modeling and User-Adapted Interaction* 12 (2002) 331–370.
- [12] J. Campos, A.D. de Figueiredo, Searching the Unsearchable: Inducing Serendipitous Insights, in: Proceedings of the Workshop Program at the Fourth International Conference on Case-Based Reasoning, ICCBR, 2001, pp. 159–164.
- [13] I. Cantador, P. Brusilovsky, T. Kuflik, 2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec), in: Proceedings of the ACM conference on Recommender systems, RecSys 2011, ACM, New York, NY, USA, 2011, pp. 387–388.

- [14] I. Cantador, I. Konstas, J.M. Jose, Categorising Social Tags to Improve Folksonomy-based Recommendations, *Journal of Web Semantics* 9 (2011) 1–15.
- [15] P. Castells, J. Wang, R. Lara, D. Zhang, Workshop on Novelty and Diversity in Recommender Systems - DiveRS, in: *Proceedings of the 5th ACM Conference on Recommender Systems, RecSys 2011*, ACM, 2011, pp. 393–394.
- [16] L. Chen, P. Pu, A User-Centric Evaluation Framework of Recommender Systems, in: *Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI)*, volume 612 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2010, pp. 14–21.
- [17] A.M. Collins, E.F. Loftus, A Spreading Activation Theory of Semantic Processing, *Psychological Review* 82 (1975) 407–428.
- [18] P.T. Costa, R.R. McCrae, Revised NEO Personality Inventory (NEO PI-R) and NEO Five-Factor Inventory (NEO FFI): Professional Manual, Psychological Assessment Resources, 1992.
- [19] F. Crestani, Application of Spreading Activation Techniques in Information Retrieval, *Artificial Intelligence* 11 (1997) 453–482.
- [20] F. Crestani, P.L. Lee, Searching the Web by Constrained Spreading Activation, *Information Processing and Management* 36 (2000) 585–605.
- [21] C. Desrosiers, G. Karypis, A Comprehensive Survey of Neighborhood-based Recommendation Methods, in: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Eds.), *Recommender Systems Handbook*, Springer, 2011, pp. 107–144.
- [22] P. Ekman, Basic Emotions, in: T. Dalgleish, M.J. Power (Eds.), *Handbook of Cognition and Emotion*, John Wiley & Sons, 1999, pp. 45–60.
- [23] S. Erdelez, Investigation of Information Encountering in the Controlled Research Environment, *Information Processing and Management* 40 (2004) 1013–1025.

- [24] B. Fasel, J. Luetttin, Automatic facial expression analysis: a survey, *Pattern Recognition* 36 (2003) 259–275.
- [25] C. Fellbaum, *WordNet: An Electronic Lexical Database*, MIT Press, 1998.
- [26] D. Fleder, K. Hosanagar, Blockbuster Culture’s Next Rise or Fall: the Impact of Recommender Systems on Sales Diversity, *Management Science* 55 (2009) 697–712.
- [27] A. Foster, N. Ford, Serendipity and Information Seeking: an Empirical Study, *Journal of Documentation* 59 (2003) 321–340.
- [28] Y. Fujiwara, M. Nakatsuji, M. Onizuka, M. Kitsuregawa, Fast and Exact Top-k Search for Random Walk with Restart, *PVLDB* 5 (2012) 442–453.
- [29] M. Ge, C. Delgado-Battenfeld, D. Jannach, Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity, in: *Proceedings of the ACM Conference on Recommender Systems*, ACM, 2010, pp. 257–260.
- [30] F. Gobbo, Serendipitous Browsing: Stumbling through Wikipedia, in: *Proceedings of the Workshop BOF - Between Ontologies and Folksonomies: Tools and Architectures for Managing and Retrieving Emerging Knowledge in Communities*, volume 312 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2007, pp. 44–47.
- [31] M. Gori, A. Pucci, ItemRank: A Random-Walk Based Scoring Algorithm for Recommender Engines, in: *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence*, Hyderabad, India, January 6-12, 2007, Morgan Kaufmann, 2007, pp. 2766–2771.
- [32] M. Gori, I.H. Witten, The Bubble of Web Visibility, *Communications of the ACM* 48 (2005) 115–117.
- [33] G.Semeraro, M. Degemmis, P. Lops, P. Basile, Combining Learning and Word Sense Disambiguation for Intelligent User Profiling, in: *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence*, Hyderabad, India, Morgan Kaufmann, 2007, pp. 2856–2861.

- [34] C. Hauff, G.J. Houben, Serendipitous Browsing: Stumbling through Wikipedia, in: Proceedings of the Searching4Fun Workshop, European Conference on Information Retrieval (ECIR), volume 836 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2012, pp. 21–24.
- [35] J.L. Herlocker, J.A. Konstan, J. Riedl, An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms, *Information Retrieval* 5 (2002) 287–310.
- [36] L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating Collaborative Filtering Recommender Systems, *ACM Transactions on Information Systems* 22 (2004) 5–53.
- [37] M. Hu, B. Liu, Mining opinion features in customer reviews, in: Proceedings of the 19th National Conference on Artificial Intelligence, AAAI’04, AAAI Press, 2004, pp. 755–760.
- [38] N. Hurley, M. Zhang, Novelty and Diversity in Top-N Recommendation - Analysis and Evaluation, *ACM Transactions on Internet Technologies* 10 (2011) 14.
- [39] L. Iaquinta, M. de Gemmis, P. Lops, G. Semeraro, M. Filannino, P. Molino, Introducing Serendipity in a Content-Based Recommender System, in: 8th International Conference on Hybrid Intelligent Systems, IEEE Computer Society, 2008, pp. 168–173.
- [40] H. Joho, J. Staiano, N. Sebe, J.M. Jose, Looking at the viewer: analysing facial activity to detect personal highlights of multimedia contents, *Multimedia Tools Appl.* 51 (2011) 505–523.
- [41] J. Kamahara, T. Asakawa, S. Shimojo, H. Miyahara, A Community-Based Recommendation System to Reveal Unexpected Interests, in: 11th International Conference on Multi Media Modeling, IEEE Computer Society, 2005, pp. 433–438.
- [42] T. Kamishima, S. Akaho, H. Asoh, J. Sakuma, Enhancement of the neutrality in recommendation, in: Proceedings of the ACM RecSys 2012 Workshop on Human Decision Making in Recommender Systems (Decisions), volume 893 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2012, pp. 8–14.

- [43] N. Kawamae, Serendipitous Recommendations Via Innovators, in: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2010, pp. 218–225.
- [44] N. Kawamae, H. Sakano, T. Yamada, Personalized Recommendation based on the Personal Innovator Degree, in: Proceedings of the ACM Conference on Recommender Systems, ACM, 2009, pp. 329–332.
- [45] R. Kohavi, C.H. Li, Oblivious Decision Trees, Graphs, and Top-Down Pruning, in: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, August 20-25 1995, 2 Volumes, Morgan Kaufmann, 1995, pp. 1071–1079.
- [46] I. Konstas, V. Stathopoulos, J.M. Jose, On Social Networks and Collaborative Recommendation, in: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, ACM, 2009, pp. 195–202.
- [47] O. Langner, R. Dotsch, G. Bijlstra, D.H.J. Wigboldus, S.T. Hawk, A. van Knippenberg, Presentation and Validation of the Radboud Faces Database, *Cognition and Emotion* 24 (2010).
- [48] G. Lawton, In the News. Simplifying Mobile Recommendation Technology with AI, *IEEE Intelligent Systems* 26 (2011) 8–9.
- [49] P. Lops, M. de Gemmis, G. Semeraro, Content-based Recommender Systems: State of the Art and Trends, in: F. Ricci, L. Rokach, B. Shapira, P. Kantor (Eds.), *Recommender Systems Handbook*, Springer, 2011, pp. 73–105.
- [50] L. Lovasz, Random Walks on Graphs: a Survey, *Combinatorics* 2 (1996) 1–46.
- [51] V. Maccatrozzo, Burst the Filter Bubble: Using Semantic Web to Enable Serendipity, in: Proceedings of the 11th International Conference on The Semantic Web - Volume Part II, ISWC’12, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 391–398.
- [52] S.M. McNee, J. Riedl, J.A. Konstan, Being Accurate is not Enough: How Accuracy Metrics have Hurt Recommender Systems, in: *Extended*

Abstracts Proceedings of the 2006 Conference on Human Factors in Computing Systems, ACM, 2006, pp. 1097–1101.

- [53] T. Murakami, K. Mori, R. Orihara, Metrics for Evaluating the Serendipity of Recommendation Lists, in: *New Frontiers in Artificial Intelligence*, volume 4914 of *Lecture Notes in Computer Science*, Springer, 2008, pp. 40–46.
- [54] C. Musto, G. Semeraro, P. Lops, M. de Gemmis, F. Narducci, Leveraging Social Media Sources to Generate Personalized Music Playlists, in: *Proc. of the 13th International Conference on E-Commerce and Web Technologies, EC-Web 2012*, volume 123 of *Lecture Notes in Business Information Processing*, Springer, 2012, pp. 112–123.
- [55] K. Oku, F. Hattori, Fusion-based Recommender System for Improving Serendipity, in: *Proceedings of the ACM RecSys 2011 Workshop on Novelty and Diversity in Recommender Systems (DiveRS)*, volume 816 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2011, pp. 19–26.
- [56] K. Onuma, H. Tong, C. Faloutsos, TANGENT: A Novel, 'Surprise me', Recommendation Algorithm, in: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2009, pp. 657–666.
- [57] V.C. Ostuni, T.D. Noia, E.D. Sciascio, R. Mirizzi, Top-N Recommendations from Implicit Feedback Leveraging Linked Open Data, in: *Seventh ACM Conference on Recommender Systems, RecSys '13*, ACM, 2013, pp. 85–92.
- [58] E. Parisier, *The Filter Bubble: What the Internet is Hiding from You*, The Penguin Press, HC, 2011.
- [59] A. Passant, dbrec - Music Recommendations Using DBpedia, in: *The Semantic Web - ISWC 2010 - 9th International Semantic Web Conference, Revised Selected Papers, Part II*, volume 6497 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 209–224.
- [60] P. Resnick, J. A. Konstan, A. Jameson, Panel on the Filter Bubble, in: *The ACM Recommender Systems Conference Blog*, [acmrecsys.wordpress.com/2011/10/25/panel-on-the-filter-bubble](http://acmrecsys.wordpress.com/2011/10/25/panel-on-the-filter-bubble), 2011.



- [61] F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Eds.), *Recommender Systems Handbook*, Springer, 2011.
- [62] S.E. Robertson, H. Zaragoza, The Probabilistic Relevance Framework: BM25 and Beyond, *Foundations and Trends in Information Retrieval* 3 (2009) 333–389.
- [63] J. Russell, Evidence for a three-factor theory of emotions, *Journal of Research in Personality* 11 (1977) 273–294.
- [64] C.L. Rusting, R.J. Larsen, Personality and Cognitive Processing of Affective Information, *Personality and Social Psychology Bulletin* (1998) 200–213.
- [65] B.M. Sarwar, G. Karypis, J.A. Konstan, J. Riedl, Item-based Collaborative Filtering Recommendation Algorithms, in: *Proceedings of the Tenth International World Wide Web Conference, WWW 10*, Hong Kong, China, May 1-5, 2001, ACM, 2001, pp. 285–295.
- [66] M. Schedl, D. Hauger, D. Schnitzer, A model for serendipitous music retrieval, in: *Proceedings of the 2nd Workshop on Context-awareness in Retrieval and Recommendation, CaRR '12*, ACM, New York, NY, USA, 2012, pp. 10–13.
- [67] G. Semeraro, M. de Gemmis, P. Lops, P. Basile, An Artificial Player for a Language Game, *IEEE Intelligent Systems* 27 (2012) 36–43.
- [68] G. Semeraro, P. Lops, P. Basile, M. de Gemmis, Knowledge Infusion into Content-based Recommender Systems, in: *Proceedings of the ACM Conference on Recommender Systems, RecSys 2009*, ACM, 2009, pp. 301–304.
- [69] G. Semeraro, P. Lops, P. Basile, M. de Gemmis, On the Tip of my Thought: Playing the Guillotine Game, in: *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, 2009, pp. 1543–1548.
- [70] G. Semeraro, P. Lops, M. de Gemmis, C. Musto, F. Narducci, A folksonomy-based recommender system for personalized access to digital artworks, *JOCCH* 5 (2012) 11.

- [71] G. Shani, A. Gunawardana, Evaluating Recommendation Systems, in: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Eds.), *Recommender Systems Handbook*, Springer, 2011, pp. 257–297.
- [72] L. Shi, Trading-off among accuracy, similarity, diversity, and long-tail: a graph-based recommendation approach, in: *Seventh ACM Conference on Recommender Systems, RecSys '13*, Hong Kong, China, October 12-16, 2013, ACM, 2013, pp. 57–64.
- [73] M. Soleymani, M. Pantic, T. Pun, Multimodal emotion recognition in response to videos, *T. Affective Computing* 3 (2012) 211–223.
- [74] K. Sparck-Jones, S. Walker, S.E. Robertson, A Probabilistic Model of Information Retrieval: Development and Comparative Experiments - Part 1 and Part 2, *Information Processing and Management* 36 (2000) 779–840.
- [75] M. Taramigkou, E. Bothos, K. Christidis, D. Apostolou, G. Mentzas, Escape the bubble: guided exploration of music preferences for serendipity and novelty, in: *Seventh ACM Conference on Recommender Systems, RecSys '13*, Hong Kong, China, October 12-16, 2013, ACM, 2013, pp. 335–338.
- [76] M. Tkalcic, A. Odic, A. Kosir, J.F. Tasic, Affective labeling in a content-based recommender system for images, *IEEE Transactions on Multimedia* 15 (2013) 391–400.
- [77] E. Toms, Serendipitous Information Retrieval, in: *Proceedings of DELOS Workshop: Information Seeking, Searching and Querying in Digital Libraries*, ERCIM Workshop Proceedings - No. 01/W001, 2000.
- [78] S. Vargas, P. Castells, Rank and Relevance in Novelty and Diversity Metrics for Recommender Systems, in: *Proceedings of the ACM Conference on Recommender Systems, RecSys 2011*, ACM, 2011, pp. 109–116.
- [79] M. Wang, T. Kawamura, Y. Sei, H. Nakagawa, Y. Tahara, A. Ohsuga, Context-aware music recommendation with serendipity using semantic relations, in: *Semantic Technology - Third Joint International Conference, JIST 2013, Revised Selected Papers*, volume 8388 of *Lecture Notes in Computer Science*, Springer, 2014, pp. 17–32.

- [80] R.A. de Wijk, V. Kooijman, R.H. Verhoeven, N.T. Holthuysen, C. de Graaf, Autonomic nervous system responses on and facial expressions to the sight, smell, and taste of liked and disliked foods, *Food Quality and Preference* 26 (2012) 196 – 203.
- [81] I.H. Witten, M. Gori, T. Numerico, *Web Dragons: Inside the Myths of Search Engine Technology*, Morgan Kaufmann, 2006.
- [82] E. Woyke, Serendipitous Shopping, *Forbes Magazine* (2011).
- [83] S. Xu, H. Jiang, F.C.M. Lau, Observing facial expressions and gaze positions for personalized webpage recommendation, in: *Proceedings of the 12th International Conference on Electronic Commerce: Roadmap for the Future of Electronic Business, ICEC '10*, ACM, New York, NY, USA, 2010, pp. 78–87.
- [84] H. Yamaba, M. Tanoue, K. Takatsuka, N. Okazaki, S. Tomita, On a serendipity-oriented recommender system based on folksonomy, *Artif. Life Robot.* 18 (2013) 89–94.
- [85] H. Yildirim, M.S. Krishnamoorthy, A Random Walk Method for Alleviating the Sparsity Problem in Collaborative Filtering, in: *Proceedings of the ACM Conference on Recommender Systems, RecSys 2008*, ACM, 2008, pp. 131–138.
- [86] Z. Zeng, M. Pantic, G.I. Roisman, T.S. Huang, A survey of affect recognition methods: Audio, visual, and spontaneous expressions, *IEEE Trans. Pattern Analysis and Machine Intelligence* 31 (2009) 39–58.
- [87] Y.C. Zhang, D.Ó. Séaghdha, D. Quercia, T. Jambor, Auralist: Introducing Serendipity into Music Recommendation, in: *Proceedings of the Fifth International Conference on Web Search and Data Mining*, ACM, 2012, pp. 13–22.



## \*Highlights (for review)

We design a Knowledge Infusion (KI) process for providing systems with background knowledge.  
We design a KI-based recommendation algorithm for providing serendipitous recommendations.  
An in-vitro evaluation shows the effectiveness of the proposed approach.  
We collected implicit emotional feedback on serendipitous recommendations.  
Results show that serendipity is moderately correlated with surprise and happiness.

Marco de Gemmis, Pasquale Lops, Giovanni Semeraro, Cataldo Musto,

An investigation on the serendipity problem in recommender systems,

Information Processing & Management,

Volume 51, Issue 5,

2015,

Pages 695-717,

ISSN 0306-4573

PUBLISHER VERSION: <https://doi.org/10.1016/j.ipm.2015.06.008>