Chapter 3
Content-based Recommender Systems: State of the Art and Trends

Pasquale Lops, Marco de Gemmis and Giovanni Semeraro

Abstract Recommender systems have the effect of guiding users in a personalized way to interesting objects in a large space of possible options. *Content-based* recommendation systems try to recommend items similar to those a given user has liked in the past. Indeed, the basic process performed by a content-based recommender consists in matching up the attributes of a user profile in which preferences and interests are stored, with the attributes of a content object (item), in order to recommend to the user new interesting items. This chapter provides an overview of content-based recommender systems, with the aim of imposing a degree of order on the diversity of the different aspects involved in their design and implementation. The first part of the chapter presents the basic concepts and terminology of content-based recommender systems, a high level architecture, and their main advantages and drawbacks. The second part of the chapter provides a review of the state of the art of systems adopted in several application domains, by thoroughly describing both classical and advanced techniques for representing items and user profiles. The most widely adopted techniques for learning user profiles are also presented. The last part of the chapter discusses trends and future research which might lead towards the next generation of systems, by describing the role of User Generated Content as a way for taking into account evolving vocabularies, and the challenge of feeding users with serendipitous recommendations, that is to say surprisingly interesting items that they might not have otherwise discovered.

Pasquale Lops
Department of Computer Science, University of Bari “Aldo Moro”, Via E. Orabona, 4, Bari (Italy)
e-mail: lops@di.uniba.it

Marco de Gemmis
Department of Computer Science, University of Bari “Aldo Moro”, Via E. Orabona, 4, Bari (Italy)
e-mail: degemmis@di.uniba.it

Giovanni Semeraro
Department of Computer Science, University of Bari “Aldo Moro”, Via E. Orabona, 4, Bari (Italy)
e-mail: semeraro@di.uniba.it

F. Ricci et al. (eds.), *Recommender Systems Handbook*,
DOI 10.1007/978-0-387-85820-3_3, © Springer Science+Business Media, LLC 2011

3.1 Introduction

The abundance of information available on the Web and in Digital Libraries, in combination with their dynamic and heterogeneous nature, has determined a rapidly increasing difficulty in finding what we want when we need it and in a manner which best meets our requirements.

As a consequence, the role of user modeling and personalized information access is becoming crucial: users need a personalized support in sifting through large amounts of available information, according to their interests and tastes.

Many information sources embody recommender systems as a way of personalizing their content for users [73]. Recommender systems have the effect of guiding users in a personalized way to interesting or useful objects in a large space of possible options [17]. Recommendation algorithms use input about a customer's interests to generate a list of recommended items. At Amazon.com, recommendation algorithms are used to personalize the online store for each customer, for example showing programming titles to a software engineer and baby toys to a new mother [50].

The problem of recommending items has been studied extensively, and two main paradigms have emerged. *Content-based* recommendation systems try to recommend items similar to those a given user has liked in the past, whereas systems designed according to the *collaborative* recommendation paradigm identify users whose preferences are similar to those of the given user and recommend items they have liked [7].

In this chapter, a comprehensive and systematic study of content-based recommender systems is carried out. The intention is twofold:

- to provide an overview of state-of-the-art systems, by highlighting the techniques which revealed the most effective, and the application domains in which they have adopted;
- to present trends and directions for future research which might lead towards the next generation of content-based recommender systems.

The chapter is organized as follows. First, we present the basic concepts and terminology related to content-based recommenders. A classical framework for providing content-based recommendations is described, in order to understand the main components of the architecture, the process for producing recommendations and the advantages and drawbacks of using this kind of recommendation technique. Section 3.3 provides a thorough review of the state of the art of content-based systems, by providing details about the classical and advanced techniques for representing items to be recommended, and the methods for learning user profiles. Section 3.4 presents trends and future research in the field of content-based recommender systems, while conclusions are drawn in Section 3.5.

3.2 Basics of Content-based Recommender Systems

Systems implementing a content-based recommendation approach analyze a set of documents and/or descriptions of items previously rated by a user, and build a model or profile of user interests based on the features of the objects rated by that user [63]. The profile is a structured representation of user interests, adopted to recommend new interesting items. The recommendation process basically consists in matching up the attributes of the user profile against the attributes of a content object. The result is a relevance judgment that represents the user's level of interest in that object. If a profile accurately reflects user preferences, it is of tremendous advantage for the effectiveness of an information access process. For instance, it could be used to filter search results by deciding whether a user is interested in a specific Web page or not and, in the negative case, preventing it from being displayed.

3.2.1 A High Level Architecture of Content-based Systems

Content-based Information Filtering (IF) systems need proper techniques for representing the items and producing the user profile, and some strategies for comparing the user profile with the item representation. The high level architecture of a content-based recommender system is depicted in Figure 3.1. The recommendation process is performed in three steps, each of which is handled by a separate component:

- CONTENT ANALYZER – When information has no structure (e.g. text), some kind of pre-processing step is needed to extract structured relevant information. The main responsibility of the component is to represent the content of items (e.g. documents, Web pages, news, product descriptions, etc.) coming from information sources in a form suitable for the next processing steps. Data items are analyzed by feature extraction techniques in order to shift item representation from the original information space to the target one (e.g. Web pages represented as keyword vectors). This representation is the input to the PROFILE LEARNER and FILTERING COMPONENT;
- PROFILE LEARNER – This module collects data representative of the user preferences and tries to generalize this data, in order to construct the user profile. Usually, the generalization strategy is realized through machine learning techniques [61], which are able to infer a model of user interests starting from items liked or disliked in the past. For instance, the PROFILE LEARNER of a Web page recommender can implement a relevance feedback method [75] in which the learning technique combines vectors of positive and negative examples into a prototype vector representing the user profile. Training examples are Web pages on which a positive or negative feedback has been provided by the user;
- FILTERING COMPONENT – This module exploits the user profile to suggest relevant items by matching the profile representation against that of items to be recommended. The result is a binary or continuous relevance judgment (com-

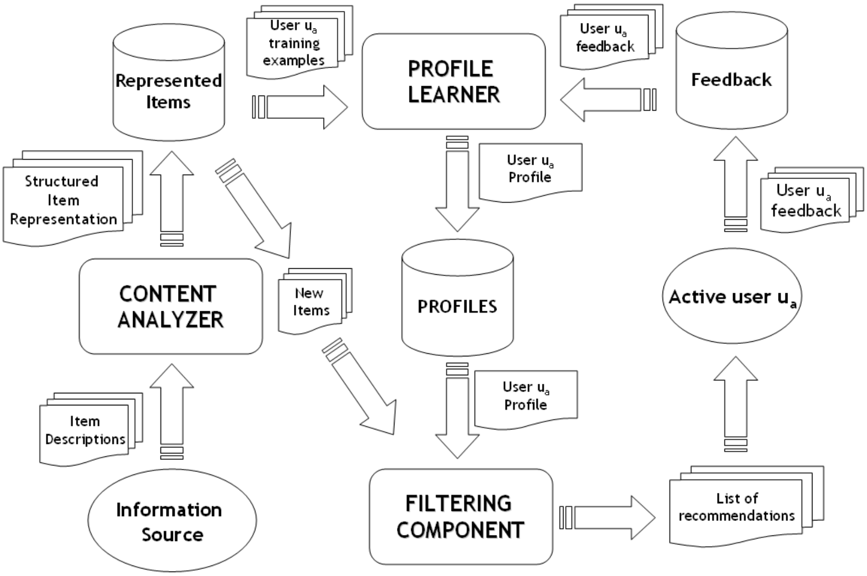

Fig. 3.1: High level architecture of a Content-based Recommender

puted using some similarity metrics [42]), the latter case resulting in a ranked list of potentially interesting items. In the above mentioned example, the matching is realized by computing the cosine similarity between the prototype vector and the item vectors.

The first step of the recommendation process is the one performed by the CONTENT ANALYZER, that usually borrows techniques from Information Retrieval systems [80, 6]. Item descriptions coming from *Information Source* are processed by the CONTENT ANALYZER, that extracts features (keywords, n-grams, concepts, ...) from unstructured text to produce a structured item representation, stored in the repository *Represented Items*.

In order to construct and update the *profile* of the *active user* u_a (user for which recommendations must be provided) her reactions to items are collected in some way and recorded in the repository *Feedback*. These reactions, called *annotations* [39] or *feedback*, together with the related item descriptions, are exploited during the process of learning a model useful to predict the actual relevance of newly presented items. Users can also explicitly define their areas of interest as an initial profile without providing any feedback.

Typically, it is possible to distinguish between two kinds of relevance feedback: positive information (inferring features liked by the user) and negative information (i.e., inferring features the user is not interested in [43]).

Two different techniques can be adopted for recording user's feedback. When a system requires the user to explicitly evaluate items, this technique is usually referred to as "explicit feedback"; the other technique, called "implicit feedback", does not require any *active* user involvement, in the sense that feedback is derived from monitoring and analyzing user's activities.

Explicit evaluations indicate how relevant or interesting an item is to the user [74]. There are three main approaches to get explicit relevance feedback:

- *like/dislike* – items are classified as "relevant" or "not relevant" by adopting a simple binary rating scale, such as in [12];
- *ratings* – a discrete numeric scale is usually adopted to judge items, such as in [86]. Alternatively, symbolic ratings are mapped to a numeric scale, such as in Syskill & Webert [70], where users have the possibility of rating a Web page as *hot*, *lukewarm*, or *cold*;
- *text comments* – Comments about a single item are collected and presented to the users as a means of facilitating the decision-making process, such as in [72]. For instance, customer's feedback at Amazon.com or eBay.com might help users in deciding whether an item has been appreciated by the community. Textual comments are helpful, but they can overload the active user because she must read and interpret each comment to decide if it is positive or negative, and to what degree. The literature proposes advanced techniques from the affective computing research area [71] to make content-based recommenders able to automatically perform this kind of analysis.

Explicit feedback has the advantage of simplicity, albeit the adoption of numeric/symbolic scales increases the cognitive load on the user, and may not be adequate for catching user's feeling about items. Implicit feedback methods are based on assigning a relevance score to specific user actions on an item, such as saving, discarding, printing, bookmarking, etc. The main advantage is that they do not require a direct user involvement, even though biasing is likely to occur, e.g. interruption of phone calls while reading.

In order to build the profile of the active user u_a, the training set TR_a for u_a must be defined. TR_a is a set of pairs $\langle I_k, r_k \rangle$, where r_k is the rating provided by u_a on the item representation I_k. Given a set of item representation labeled with ratings, the PROFILE LEARNER applies supervised learning algorithms to generate a predictive model – the *user profile* – which is usually stored in a *profile repository* for later use by the FILTERING COMPONENT. Given a new item representation, the FILTERING COMPONENT predicts whether it is likely to be of interest for the active user, by comparing features in the item representation to those in the representation of user preferences (stored in the user profile). Usually, the FILTERING COMPONENT implements some strategies to rank potentially interesting items according to the relevance with respect to the user profile. Top-ranked items are included in a *list of recommendations* L_a, that is presented to u_a. User tastes usually change in time, therefore up-to-date information must be maintained and provided to the PROFILE LEARNER in order to automatically update the user profile. Further feedback is gathered on generated recommendations by letting users state their satisfaction or dissatisfaction with items in L_a. After gathering that feedback, the learning process is performed again on the new training set, and the resulting profile is adapted to the updated user interests. The iteration of the feedback-learning cycle over time allows the system to take into account the dynamic nature of user preferences.

3.2.2 Advantages and Drawbacks of Content-based Filtering

The adoption of the content-based recommendation paradigm has several advantages when compared to the collaborative one:

- USER INDEPENDENCE - Content-based recommenders exploit solely ratings provided by the active user to build her own profile. Instead, collaborative filtering methods need ratings from other users in order to find the “nearest neighbors” of the active user, i.e., users that have similar tastes since they rated the same items similarly. Then, only the items that are most liked by the neighbors of the active user will be recommended;
- TRANSPARENCY - Explanations on how the recommender system works can be provided by explicitly listing content features or descriptions that caused an item to occur in the list of recommendations. Those features are indicators to consult in order to decide whether to trust a recommendation. Conversely, collaborative systems are black boxes since the only explanation for an item recommendation is that unknown users with similar tastes liked that item;
- NEW ITEM - Content-based recommenders are capable of recommending items not yet rated by any user. As a consequence, they do not suffer from the first-rater problem, which affects collaborative recommenders which rely solely on users’ preferences to make recommendations. Therefore, until the new item is rated by a substantial number of users, the system would not be able to recommend it.

Nonetheless, content-based systems have several shortcomings:

- LIMITED CONTENT ANALYSIS - Content-based techniques have a natural limit in the number and type of features that are associated, whether automatically or manually, with the objects they recommend. Domain knowledge is often needed, e.g., for movie recommendations the system needs to know the actors and directors, and sometimes, domain ontologies are also needed. No content-based recommendation system can provide suitable suggestions if the analyzed content does not contain enough information to discriminate items the user likes from items the user does not like. Some representations capture only certain aspects of the content, but there are many others that would influence a user’s experience. For instance, often there is not enough information in the word frequency to model the user interests in jokes or poems, while techniques for affective computing would be most appropriate. Again, for Web pages, feature extraction techniques from text completely ignore aesthetic qualities and additional multimedia information.

  To sum up, both automatic and manually assignment of features to items could not be sufficient to define distinguishing aspects of items that turn out to be necessary for the elicitation of user interests.

- OVER-SPECIALIZATION - Content-based recommenders have no inherent method for finding something unexpected. The system suggests items whose scores are high when matched against the user profile, hence the user is going to be recommended items similar to those already rated. This drawback is also called *serendipity* problem to highlight the tendency of the content-based systems to produce recommendations with a limited degree of novelty. To give an example, when a user has only rated movies directed by Stanley Kubrick, she will be recommended just that kind of movies. A "perfect" content-based technique would rarely find anything *novel*, limiting the range of applications for which it would be useful.
- NEW USER - Enough ratings have to be collected before a content-based recommender system can really understand user preferences and provide accurate recommendations. Therefore, when few ratings are available, as for a new user, the system will not be able to provide reliable recommendations.

In the following, some strategies for tackling the above mentioned problems will be presented and discussed. More specifically, novel techniques for enhancing the content representation using common-sense and domain-specific knowledge will be described (Sections 3.3.1.3-3.3.1.4). This may help to overcome the limitations of traditional content analysis methods by providing new features, such as WordNet [60, 32] or Wikipedia concepts, which help to represent the items to be recommended in a more accurate and transparent way. Moreover, the integration of user-defined lexicons, such as folksonomies, in the process of generating recommendations will be presented in Section 3.4.1, as a way for taking into account evolving vocabularies.

Possible ways to feed users with *serendipitous* recommendations, that is to say, interesting items with a high degree of novelty, will be analyzed as a solution to the over-specialization problem (Section 3.4.2).

Finally, different strategies for overcoming the new user problem will be presented. Among them, social tags provided by users in a community can be exploited as a feedback on which recommendations are produced when few or no ratings for a specific user are available to the system (Section 3.4.1.1).

3.3 State of the Art of Content-based Recommender Systems

As the name implies, content-based filtering exploits the content of data items to predict its relevance based on the user's profile. Research on content-based recommender systems takes place at the intersection of many computer science topics, especially Information Retrieval [6] and Artificial Intelligence.

From Information Retrieval (IR), research on recommendation technologies derives the vision that users searching for recommendations are engaged in an information seeking process. In IR systems the user expresses a one-off information need by giving a query (usually a list of keywords), while in IF systems the information need of the user is represented by her own profile. Items to be recommended can be very different depending on the number and types of attributes used to describe them. Each item can be described through the same small number of attributes with known set of values, but this is not appropriate for items, such as Web pages, news, emails or documents, described through unstructured text. In that case there are no attributes with well-defined values, and the use of document modeling techniques with roots in IR research is desirable.

From an Artificial Intelligence perspective, the recommendation task can be cast as a learning problem that exploits past knowledge about users. At their simplest, user profiles are in the form of user-specified keywords or rules, and reflect the long-term interests of the user. Often, it is advisable for the recommender to learn the user profile rather than impose upon the user to provide one. This generally involves the application of Machine Learning (ML) techniques, whose goal is learning to categorize new information items based on previously seen information that have been explicitly or implicitly labelled as interesting or not by the user. Given these labelled information items, ML methods are able to generate a predictive model that, given a new information item, will help to decide whether it is likely to be of interest for the target user.

Section 3.3.1 describes alternative item representation techniques, ranging from traditional text representation, to more advanced techniques integrating ontologies and/or encyclopedic knowledge. Next, recommendation algorithms suitable for the described representations will be discussed in Section 3.3.2.

3.3.1 Item Representation

Items that can be recommended to the user are represented by a set of features, also called *attributes* or *properties*. For example, in a movie recommendation application, features adopted to describe a movie are: actors, directors, genres, subject matter, . . .). When each item is described by the same set of attributes, and there is a known set of values the attributes may take, the item is represented by means of structured data. In this case, many ML algorithms can be used to learn a user profile [69].

In most content-based filtering systems, item descriptions are textual features extracted from Web pages, emails, news articles or product descriptions. Unlike structured data, there are no attributes with well-defined values. Textual features create a number of complications when learning a user profile, due to the natural language ambiguity. The problem is that traditional keyword-based profiles are unable to capture the semantics of user interests because they are primarily driven by a string matching operation. If a string, or some morphological variant, is found in both the profile and the document, a match is made and the document is considered as relevant. String matching suffers from problems of:

- POLYSEMY, the presence of multiple meanings for one word;
- SYNONYMY, multiple words with the same meaning.

The result is that, due to synonymy, relevant information can be missed if the profile does not contain the exact keywords in the documents while, due to polysemy, wrong documents could be deemed relevant.

Semantic analysis and its integration in personalization models is one of the most innovative and interesting approaches proposed in literature to solve those problems. The key idea is the adoption of knowledge bases, such as lexicons or ontologies , for annotating items and representing profiles in order to obtain a "semantic" interpretation of the user information needs. In the next section, the basic keyword-based approach for document representation will be described, followed by a review of "traditional" systems relying on that model. Then, Sections 3.3.1.3 and 3.3.1.4 will provide an overview of techniques for semantic analysis based on ontological and world knowledge, respectively.

3.3.1.1 Keyword-based Vector Space Model

Most content-based recommender systems use relatively simple retrieval models, such as keyword matching or the Vector Space Model (VSM) with basic TF-IDF weighting. VSM is a spatial representation of text documents. In that model, each document is represented by a vector in a n-dimensional space, where each dimension corresponds to a term from the overall vocabulary of a given document collection. Formally, every document is represented as a vector of term weights, where each weight indicates the degree of association between the document and the term. Let $D = \{d_1, d_2, ..., d_N\}$ denote a set of documents or corpus, and $T = \{t_1, t_2, ..., t_n\}$ be the dictionary, that is to say the set of words in the corpus. T is obtained by applying some standard natural language processing operations, such as tokenization, stopwords removal, and stemming [6]. Each document d_j is represented as a vector in a n-dimensional vector space, so $d_j = \{w_{1j}, w_{2j}, ..., d_{nj}\}$, where w_{kj} is the weight for term t_k in document d_j.

Document representation in the VSM raises two issues: weighting the terms and measuring the feature vector similarity. The most commonly used term weighting scheme, TF-IDF (Term Frequency-Inverse Document Frequency) *weighting*, is based on empirical observations regarding text [79]:

- rare terms are not less relevant than frequent terms (IDF assumption);
- multiple occurrences of a term in a document are not less relevant than single occurrences (TF assumption);
- long documents are not preferred to short documents (normalization assumption).

In other words, terms that occur frequently in one document (TF =term-frequency), but rarely in the rest of the corpus (IDF = inverse-document-frequency), are more likely to be relevant to the topic of the document. In addition, normalizing the resulting weight vectors prevent longer documents from having a better chance of retrieval. These assumptions are well exemplified by the TF-IDF function:

$$\text{TF-IDF}(t_k, d_j) = \underbrace{\text{TF}(t_k, d_j)}_{\text{TF}} \cdot \underbrace{log \frac{N}{n_k}}_{\text{IDF}} \tag{3.1}$$

where N denotes the number of documents in the corpus, and n_k denotes the number of documents in the collection in which the term t_k occurs at least once.

$$\text{TF}(t_k, d_j) = \frac{f_{k,j}}{max_z f_{z,j}} \tag{3.2}$$

where the maximum is computed over the frequencies $f_{z,j}$ of all terms t_z that occur in document d_j. In order for the weights to fall in the $[0,1]$ interval and for the documents to be represented by vectors of equal length, weights obtained by Equation (3.1) are usually normalized by cosine normalization:

$$w_{k,j} = \frac{\text{TF-IDF}(t_k, d_j)}{\sqrt{\sum_{s=1}^{|T|} \text{TF-IDF}(t_s, d_j)^2}} \tag{3.3}$$

which enforces the normalization assumption.

As stated earlier, a similarity measure is required to determine the closeness between two documents. Many similarity measures have been derived to describe the proximity of two vectors; among those measures, cosine similarity is the most widely used:

$$sim(d_i, d_j) = \frac{\sum_k w_{ki} \cdot w_{kj}}{\sqrt{\sum_k w_{ki}{}^2} \cdot \sqrt{\sum_k w_{kj}{}^2}} \tag{3.4}$$

In content-based recommender systems relying on VSM, both user profiles and items are represented as weighted term vectors. Predictions of a user's interest in a particular item can be derived by computing the cosine similarity.

3.3.1.2 Review of Keyword-based Systems

Several keyword-based recommender systems have been developed in a relatively short time, and it is possible to find them in various fields of applications, such as news, music, e-commerce, movies, etc. Each domain presents different problems, that require different solutions.

In the area of *Web recommenders*, famous systems in literature are *Letizia* [49], *Personal WebWatcher* [62, 63], *Syskill & Webert* [70, 68], *ifWeb* [4], *Amalthea* [66], and *WebMate* [23]. *Letizia* is implemented as a web-browser extension that tracks the user's browsing behavior and builds a personalized model consisting of keywords related to the user's interests. It relies on implicit feedback to infer the user's preferences. For example, bookmarking a page is interpreted as strong evidence for the user's interests in that page. In a similar way, *Personal WebWatcher* learns individual interests of users from the Web pages they visit, and from documents lying one link away from the visited pages. It processes visited documents as positive examples of the user's interests, and non-visited documents as negative examples. *Amalthaea* uses specific filtering agents to assist users in finding interesting information as well. User can specify filtering agents by providing pages (represented as weighted vectors) closely related to their interests.

The same approach is adopted by *Syskill & Webert*, that represents documents with the 128 most informative words (the "informativeness" of words in documents is determined in several different ways). Advanced representation techniques are adopted by *ifWeb*, that represents profiles in the form of a weighted semantic network. It supports explicit feedback and takes into account not only interests, but also explicit *dis*interests. Another interesting aspect is that it incorporates a mechanism for temporal decay, i.e. it ages the interests as expressed by the user. A different approach for representing user interests is adopted by *WebMate*, that keeps track of user interests in different domains by learning a user profile that consists of the keyword vectors that represents positive training examples. A profile of *n* keyword vectors can correctly represent up to *n* independent user interests.

In the field of *news filtering*, noteworthy recommender systems are *NewT* [87], *PSUN* [90], *INFOrmer* [91], *NewsDude* [12], *Daily Learner* [13], and *YourNews* [2]. *NewT* (News Tailor) allows users to provide positive and negative feedback on articles, part of articles, authors or sources. Several filtering agents are trained for different types of information, e.g. one for political news, one for sports, etc. In the same way *YourNews*, a more recent system for personalized news access, maintains a separate interest profile for 8 different topics (National, World, Business, etc.). The user interest profile for each topic is represented as a weighted prototype term vector extracted from the user's news view history. *N* articles from users' past views are collected, and the 100 top-weighted terms are extracted to generate the final prototype vectors. The system maintains short-term profiles by considering only the 20 most recently viewed news item, whereas long-term profiles consider all past views. The system can use profiles to suggest *recent* and *recommended* news.

Learning short-term and long-term profiles is quite typical of news filtering systems. *NewsDude* learns a short-term user model based on TF-IDF (cosine similarity), and a long-term model based on a naïve Bayesian classifier by relying on an initial training set of interesting news articles provided by the user. The news source is Yahoo! News. In the same way *Daily Learner*, a learning agent for wireless information access, adopts an approach for learning two separate user-models. The former, based on a Nearest Neighbor text classification algorithm, maintains the short-term interests of users, while the latter, based on a naïve Bayesian classifier, represents the long-term interests of users and relies on data collected over a longer period of time.

Among systems using a more complex representation for articles or profiles, *PSUN* and *INFOrmer* are worth to note. *PSUN* adopts an alternative representation for articles. Profiles are provided initially by presenting the system with some articles the user finds interesting. Recurring words in these articles are recorded by means of *n-grams* stored in a network of mutually attracting or repelling words, whose degree of attraction is determined by the number of co-occurrences. Each user has multiple profiles that compete via a genetic algorithm, requiring explicit feedback. *INFOrmer* uses a semantic network for representing both user profiles and articles. A spreading activation technique [25] is used to compare articles and profiles, and a relevance feedback mechanism may be used to adapt the behavior of the system to user's changing interests. The pure spreading activation model consists of a network data structure consisting of nodes interconnected by links, that may be labeled and/or weighted. The processing starts by labeling a set of *source nodes* with activation weights and proceeds by iteratively propagating that activation to other nodes linked to the source nodes, until a termination condition ends the search process over the network.

A variety of content-based recommender systems exist in other application domains. *LIBRA* [65] implements a naïve Bayes text categorization method for book recommendation that exploits the product descriptions obtained from the Web pages of the Amazon on-line digital store. *Re:Agent* [16] is an intelligent email agent that can learn actions such as filtering, downloading to palmtops, forwarding email to voicemail, etc. using automatic feature extraction. Re:Agent users are required only to place example messages in folders corresponding to the desired actions. Re:Agent learns the concepts and decision policies from these folders. *Citeseer* [15] assists the user in the process of performing a scientific literature search, by using word information and analyzing common citations in the papers. *INTIMATE* [53] recommends movies by using text categorization techniques to learn from movie synopses obtained from the Internet Movie Database[1]. In order to get recommendations, the user is asked to rate a minimum number of movies into six categories: terrible, bad, below average, above average, good and excellent. In the same way, *Movies2GO* [67] learns user preferences from the synopsis of movies rated by the user. The innovative aspect of the system is to integrate voting schemes [93], designed to allow multiple individuals with conflicting preferences arrive at an acceptable compromise, and adapt them to manage conflicting preferences in a single user.

In the music domain, the commonly used technique for providing recommendations is collaborative filtering (see Last.fm[2] and MyStrands[3] systems). The most noticeable system using (manual) content-based descriptions to recommend music is Pandora[4]. The main problem of the system is scalability, because the music annotation process is entirely done manually. Conversely, *FOAFing the music* [21, 22] is able to recommend, discover and explore music content, based on user profiling via *Friend of a Friend* (FOAF)[5] descriptions, context-based information extracted from music related RSS feeds, and content-based descriptions automatically extracted from the audio itself.

In order to complete the survey of content-based recommender systems adopting the simple keyword-based vector space representation, we should also men-

[1] http://www.imdb.com

[2] http://www.last.fm

[3] http://www.mystrands.com

[4] http://www.pandora.com

[5] http://www.foaf-project.org

tion some hybrid recommender systems that combine collaborative and content-based methods, such as *Fab* [7], *WebWatcher* [45], *P-Tango* [24], *ProfBuilder* [99], *PTV* [89], *Content-boosted Collaborative Filtering* [56], *CinemaScreen* [78] and the one proposed in [94].

The most important lesson learned from the analysis of the main systems developed in the last 15 years is that keyword-based representation for both items and profiles can give accurate performance, provided that a sufficient number of evidence of user interests is available. Most content-based systems are conceived as text classifiers built from training sets including documents which are either positive or negative examples of user interests. Therefore, accurate recommendations are achieved when training sets with a large number of examples are available, which guarantee reliable "syntactic" evidence of user interests. The problem with that approach is the "lack of intelligence". When more advanced characteristics are required, keyword-based approaches show their limitations. If the user, for instance likes "French impressionism", keyword-based approaches will only find documents in which the words "French" and "impressionism" occur. Documents regarding Claude Monet or Renoir exhibitions will not appear in the set of recommendations, even though they are likely to be very relevant for that user. More advanced representation strategies are needed in order to equip content-based recommender systems with "semantic intelligence", which allows going beyond the syntactic evidence of user interests provided by keywords.

In the next sections, we will examine possible ways to infuse knowledge in the indexing phase by means of ontologies and encyclopedic knowledge sources.

3.3.1.3 Semantic Analysis by using Ontologies

Semantic analysis allows learning more accurate profiles that contain references to concepts defined in external knowledge bases. The main motivation for this approach is the challenge of providing a recommender system with the cultural and linguistic background knowledge which characterizes the ability of interpreting natural language documents and reasoning on their content.

In this section, a review of the main strategies adopted to introduce some semantics in the recommendation process is presented. The description of these strategies is carried out by taking into account several criteria:

- the type of knowledge source involved (e.g. lexicon, ontology, etc.);
- the techniques adopted for the annotation or representation of the items;
- the type of content included in the user profile;
- the item-profile matching strategy.

SiteIF [52] is a personal agent for a multilingual news Web site. To the best of our knowledge, it was the first system to adopt a sense-based document representation in order to build a model of the user interests. The external knowledge source involved in the representation process is MultiWordNet, a multilingual lexical database where English and Italian senses are aligned. Each news is automatically associated with a list of MultiWordNet synsets by using Word Domain Disambiguation [51]. The user profile is built as a semantic network whose nodes represent synsets found in the documents read by the user. During the matching phase, the system receives as input the synset representation of a document and the current user model, and it produces as output an estimation of the document relevance by using the Semantic Network Value Technique [92].

ITR (ITem Recommender) is a system capable of providing recommendations for items in several domains (e.g., movies, music, books), provided that descriptions of items are available as text documents (e.g. plot summaries, reviews, short abstracts) [27, 83]. Similarly to SiteIF, ITR integrates linguistic knowledge in the process of learning user profiles, but Word Sense Disambiguation rather than Word Domain Disambiguation is adopted to obtain a sense-based document representation. The linguistic knowledge comes exclusively from the WordNet lexical ontology. Items are represented according to a synset-based vector space model, called bag-of-synsets (BOS), that is an extension of the classical bag-of-words (BOW) one [8, 84]. In the BOS model, a synset vector, rather than a word vector, corresponds to a document. The user profile is built as a Naïve Bayes binary text classifier able to categorize an item as interesting or not interesting. It includes those synsets that turn out to be most indicative of the user preferences, according to the value of the conditional probabilities estimated in the training phase. The item-profile matching consists in computing the probability for the item of being in the class "interesting", by using the probabilities of synsets in the user profile.

SEWeP (Semantic Enhancement for Web Personalization) [31] is a Web personalization system that makes use of both the usage logs and the semantics of a Web site's content in order to personalize it. A domain-specific taxonomy of categories has been used to semantically annotate Web pages, in order to have a uniform and consistent vocabulary. While the taxonomy is built manually, the annotation process is performed automatically. SEWeP, like SiteIF and ITR, makes use of the lexical knowledge stored in WordNet to "interpret" the content of an item and to support the annotation/representation process. Web pages are initially represented by keywords extracted from their content, then keywords are mapped to the concepts of the taxonomy. Given a keyword, a WordNet-based word similarity measure is applied to find the "closest" category word to that keyword. SEWeP does not build a personal profile of the user, rather it discovers navigational patterns. The categories which have been "semantically associated" to a pattern are used by the SEWeP recommendation engine to expand the recommendation set with pages characterized by the thematic categories that seem to be of interest for the user.

Quickstep [58] is a system for the recommendation of on-line academic research papers. The system adopts a research paper topic ontology based on the computer science classifications made by the DMOZ open directory project[6] (27 classes used). Semantic annotation of papers consists in associating them with class names within the research paper topic ontology, by using a k-Nearest Neighbor classifier. Interest profiles are computed by correlating previously browsed research papers with

[6] http://www.dmoz.org/

their classification. User profiles thus hold a set of topics and interest values in these topics. The item-profile matching is realized by computing a correlation between the top three interesting topics in the user profile and papers classified as belonging to those topics. *Foxtrot* [58] extends the Quickstep system by implementing a paper search interface, a profile visualization interface and an email notification, in addition to the Web page recommendation interface. Profile visualization is made possible because profiles are represented in ontological terms understandable to the users.

Informed Recommender [1] uses consumer product reviews to make recommendations. The system converts consumers' opinions into a structured form by using a translation ontology, which is exploited as a form of knowledge representation and sharing. The ontology provides a controlled vocabulary and relationships among words to describe: the consumer's skill level and experience with the product under review. To this purpose, the ontology contains two main parts: *opinion quality* and *product quality*, which formalize the two aforementioned aspects. A text-mining process automatically maps sentences in the reviews into the ontology information structure. The system does not build a profile of the user, rather it computes a set of recommendations on the basis of a user's request, e.g. the user asks about the quality of specific features of a product. Informed Recommender is able to answer to query and also recommends the best product according to the features the user is concerned with. Two aspects make this work noteworthy: one is that ontological knowledge can model different points of view according to which items can be annotated, the other is the use of review comments in the form of free text.

News@hand [18] is a system that adopts an ontology-based representation of item features and user preferences to recommend news. The annotation process associates the news with concepts belonging to the domain ontologies. A total of 17 ontologies have been used: they are adaptations of the IPTC ontology[7], which contains concepts of multiple domains such as education, culture, politics, religion, science, sports, etc. It is not clear whether the annotation process is performed manually or by means of automated techniques such as text categorization. Item descriptions are vectors of TF-IDF scores in the space of concepts defined in the ontologies. User profiles are represented in the same space, except that a score measures the intensity of the user interest for a specific concept. Item-profile matching is performed as a cosine-based vector similarity.

A recommender system for *Interactive Digital Television* is proposed in [14], where the authors apply reasoning techniques borrowed from the Semantic Web in order to compare user preferences with items (TV programs) in a more flexible way, compared to the conventional syntactic metrics. The TV programs available during the recommendation process are annotated by metadata that describe accurately their main attributes. Both the knowledge about the TV domain and the user profiles are represented using an OWL ontology. Ontology-profiles provide a formal representation of the users' preferences, being able to *reason* about them and *discover* extra knowledge about their interests. The recommendation phase exploits

[7] IPTC ontology, http://nets.ii.uam.es/neptuno/iptc/

the knowledge stored in the user profile to discover hidden semantic associations between the user's preferences and the available products. The inferred knowledge is processed and a spreading activation technique is adopted to suggest products to the user. The noteworthy aspect of this work is that ontology-profiles improve flat lists-based approaches which are not well structured to foster the discovery of new knowledge.

The JUMP System [10, 9] is capable of intelligent delivery of contextualized and personalized information to knowledge workers acting in their day-to-day working environment on non-routinary tasks. The information needs of the JUMP user is represented in the form of a complex query, such as a *task support request*, rather than a user profile. An example of complex query is "I have to prepare a technical report for the VIKEF project". The semantic analysis of both documents and user information needs is based on a domain ontology in which concepts are manually annotated using WordNet synsets. The mapping between documents and domain/lexical concepts is performed automatically by means of Word Sense Disambiguation and Named Entity Recognition procedures, which exploit the lexical annotations in the domain ontology. The matching between concepts in the user request and documents is based on the relations in the domain ontology. For the processing of the example query, all instances of the concepts "technical report" and "project", and relations among these instances are extracted from the ontology.

The leading role of linguistic knowledge is highlighted by the wide use of WordNet, which is mostly adopted for the semantic interpretation of content by using word sense disambiguation. On the other hand, the studies described above showed that the great potential provided by WordNet is not sufficient alone for the full comprehension of the user interests and for their contextualization in the application domain. Domain specific knowledge is also needed. Ontologies play the fundamental role of formalizing the application domain, being exploited for the semantic descriptions of the items and for the representation of the concepts (i.e. classes and their instances) and relationships (i.e. hierarchical links and properties) identified in the domain. In conclusion, all studies which incorporated either linguistic or domain-specific knowledge or both in content-based filtering methods provided better and more accurate results compared to traditional content-based methods. This encourages researchers to design novel filtering methods which formalize and contextualize user interests by exploiting external knowledge sources such as thesauri or ontologies.

3.3.1.4 Semantic Analysis by using Encyclopedic Knowledge Sources

Common-sense and domain-specific knowledge may be useful to improve the effectiveness of natural language processing techniques by generating more informative features than the mere bag of words. The process of learning user profiles could benefit from the *infusion* of exogenous knowledge (externally supplied), with respect to the classical use of endogenous knowledge (extracted from the documents themselves). Many sources of world knowledge have become available in recent years.

Examples of general purpose knowledge bases include the Open Directory Project (ODP), Yahoo! Web Directory, and Wikipedia.

In the following we provide a brief overview of novel methods for generating new advanced features using world knowledge, even though those methods are not yet used in the context of learning user profiles.

Explicit Semantic Analysis (ESA) [34, 35] is a technique able to provide a fine-grained semantic representation of natural language texts in a high-dimensional space of natural (and also comprehensible) concepts derived from Wikipedia . Concepts are defined by Wikipedia articles, e.g., ITALY, COMPUTER SCIENCE, or RECOMMENDER SYSTEMS. The approach is inspired by the desire to augment text representation with massive amounts of world knowledge. In the case of Wikipedia as knowledge source, there are several advantages, such as its constant development by the community, the availability in several languages, and its high accuracy [37]. Empirical evaluations showed that ESA leads to substantial improvements in computing word and text relatedness, and in the text categorization task across a diverse collection of datasets. It has also been shown that ESA enhanced traditional BOW-based retrieval models [30].

Another interesting approach to add semantics to text is proposed by the Wikify! system [59, 26], which has the ability to identify important concepts in a text (keyword extraction), and then link these concepts to the corresponding Wikipedia pages (word sense disambiguation). The annotations produced by the Wikify! system can be used to automatically enrich documents with references to semantically related information. A Turing-like test to compare the quality of the system annotations to manual annotations produced by Wikipedia contributors has been designed. Human beings are asked to distinguish between manual and automatic annotations. Results suggest that the computer and human-generated Wikipedia annotations are hardly distinguishable, which indicates the high quality of the Wikify! system's annotations.

To the best of our knowledge, there are no (content-based) recommender systems able to exploit the above mentioned advanced semantic text representations for learning profiles containing references to world facts. The positive results obtained exploiting the advanced text representations in several tasks, such as semantic relatedness, text categorization and retrieval, suggest that similar positive results could be also obtained in the recommendation task. It seems a promising research area, not yet explored.

In [47], Wikipedia is used to estimate similarity between movies, in order to provide more accurate predictions for the Netflix Prize competition. More specifically, the content and the hyperlink structure of Wikipedia articles are exploited to identify similarities between movies. A similarity matrix containing the degree of similarity of each movie-movie pair is produced, and the prediction of user ratings from this matrix is computed by using a k-Nearest Neighbors and a Pseudo-SVD algorithm. Each of these methods combines the similarity estimates from Wikipedia with ratings from the training set to predict ratings in the test set. Unfortunately, these techniques did not show any significant improvement of the overall accuracy.

In [88], a quite complex, but not yet complete, approach for filtering RSS feeds and e-mails is presented. More specifically, the authors present an approach exploiting Wikipedia to automatically generate the user profile from the user's document collection. The approach mainly consists of four steps, namely the Wikipedia indexing, the profile generation, the problem-oriented index database creation, and the information filtering. The profile generation step exploits the collection of documents provided by the user, which implicitly represents a set of topics interesting for the user. A set of terms is extracted from each document, then similar Wikipedia articles are found by using the ESA algorithm. The system then extracts the list of Wikipedia categories from the articles and clusters these categories in order to get a subset of categories corresponding to one topic in the user profile. The user can also check her own profile and add or remove categories in order to refine topics. For each topic in the user profile, a problem-oriented Wikipedia corpus is created and indexed, and represents the base for filtering information.

In [85], a different approach to exploit Wikipedia in the content analysis step is presented. More specifically, the idea is to provide a *knowledge infusion* process into content-based recommender systems, in order to provide them with the *cultural* background knowledge that hopefully allows a more accurate content analysis than classical approaches based on words. The encyclopedic knowledge is useful to recognize specific domain-dependent concepts or named entities, especially in those contexts for which the adoption of domain ontologies is not feasible. Wikipedia entries have been modeled using Semantic Vectors, based on the `WordSpace` model [77], a vector space whose points are used to represent semantic concepts, such as words and documents. Relationships between words are then exploited by a spreading activation algorithm to produce new features that can be exploited in several ways during the recommendation process.

3.3.2 Methods for Learning User Profiles

Machine learning techniques, generally used in the task of inducing content-based profiles, are well-suited for text categorization [82]. In a machine learning approach to text categorization, an inductive process automatically builds a text classifier by learning from a set of *training documents* (documents labeled with the categories they belong to) the features of the categories.

The problem of learning user profiles can be cast as a binary text categorization task: each document has to be classified as interesting or not with respect to the user preferences. Therefore, the set of categories is $C = \{c_+, c_-\}$, where c_+ is the positive class (user-likes) and c_- the negative one (user-dislikes).

In the next sections we review the most used learning algorithms in content-based recommender systems. They are able to learn a function that models each user's interests. These methods typically require users to label documents by assigning a relevance score, and automatically infer profiles exploited in the filtering process to rank documents according to the user preferences.

3.3.2.1 Probabilistic Methods and Naïve Bayes

Naïve Bayes is a probabilistic approach to inductive learning, and belongs to the general class of Bayesian classifiers. These approaches generate a probabilistic model based on previously observed data. The model estimates the *a posteriori* probability, $P(c|d)$, of document d belonging to class c. This estimation is based on the a priori probability, $P(c)$, the probability of observing a document in class c, $P(d|c)$, the probability of observing the document d given c, and $P(d)$, the probability of observing the instance d. Using these probabilities, the Bayes theorem is applied to calculate $P(c|d)$:

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)} \tag{3.5}$$

To classify the document d, the class with the highest probability is chosen:

$$c = argmax_{c_j} \frac{P(c_j)P(d|c_j)}{P(d)}$$

$P(d)$ is generally removed as it is equal for all c_j. As we do not know the value for $P(d|c)$ and $P(c)$, we estimate them by observing the training data. However, estimating $P(d|c)$ in this way is problematic, as it is very unlikely to see the same document more than once: the observed data is generally not enough to be able to generate good probabilities. The naïve Bayes classifier overcomes this problem by simplifying the model through the independence assumption: all the words or tokens in the observed document d are conditionally independent of each other given the class. Individual probabilities for the words in a document are estimated one by one rather than the complete document as a whole. The conditional independence assumption is clearly violated in real-world data, however, despite these violations, empirically the naïve Bayes classifier does a good job in classifying text documents [48, 11].

There are two commonly used working models of the naïve Bayes classifier, the *multivariate Bernoulli* event model and the *multinomial* event model [54]. Both models treat a document as a vector of values over the corpus vocabulary, V, where each entry in the vector represents whether a word occurred in the document, hence both models lose information about word order. The multivariate Bernoulli event model encodes each word as a binary attribute, i.e., whether a word appeared or not, while the multinomial event model counts how many times the word appeared in the document. Empirically, the multinomial naïve Bayes formulation was shown to outperform the multivariate Bernoulli model. This effect is particularly noticeable for large vocabularies [54]. The way the multinomial event model uses its document vector to calculate $P(c_j|d_i)$ is as follows:

$$P(c_j|d_i) = P(c_j) \prod_{w \in V_{d_i}} P(t_k|c_j)^{N_{(d_i,t_k)}} \tag{3.6}$$

where $N_{(d_i,t_k)}$ is defined as the number of times word or token t_k appeared in document d_i. Notice that, rather than getting the product of all the words in the corpus vocabulary V, only the subset of the vocabulary, V_{d_i}, containing the words that appear in the document d_i, is used.

A key step in implementing naïve Bayes is estimating the word probabilities $P(t_k|c_j)$. To make the probability estimates more robust with respect to infrequently encountered words, a smoothing method is used to modify the probabilities that would have been obtained by simple event counting. One important effect of smoothing is that it avoids assigning probability values equal to zero to words not occurring in the training data for a particular class. A rather simple smoothing method relies on the common Laplace estimates (i.e., adding one to all the word counts for a class). A more interesting method is Witten-Bell [100]. Although naïve Bayes performances are not as good as some other statistical learning methods such as nearest-neighbor classifiers or support vector machines, it has been shown that it can perform surprisingly well in the classification tasks where the computed probability is not important [29]. Another advantage of the naïve Bayes approach is that it is very efficient and easy to implement compared to other learning methods.

Although the classifiers based on the multinomial model significantly outperform those based on the multivariate one at large vocabulary sizes, their performance is unsatisfactory when: 1) documents in the training set have different lengths, thus resulting in a rough parameter estimation; 2) handling rare categories (few training documents available). These conditions frequently occur in the user profiling task, where no assumptions can be made on the length of training documents, and where obtaining an appropriate set of negative examples (i.e., examples of the class c_-) is problematic. Indeed, since users do not perceive having immediate benefits from giving negative feedback to the system [81], the training set for the class c_+ (user-likes) may be often larger than the one for the class c_- (user-dislikes). In [46], the authors propose a multivariate Poisson model for naïve Bayes text classification that allows more reasonable parameter estimation under the above mentioned conditions. We have adapted this approach to the case of user profiling task [36].

The naïve Bayes classifier has been used in several content-based recommendation systems, such as *Syskill & Webert* [70, 68], *NewsDude* [12], *Daily Learner* [13], *LIBRA* [65] and *ITR* [27, 83].

3.3.2.2 Relevance Feedback and Rocchio's Algorithm

Relevance feedback is a technique adopted in Information Retrieval that helps users to incrementally refine queries based on previous search results. It consists of the users feeding back into the system decisions on the relevance of retrieved documents with respect to their information needs.

Relevance feedback and its adaptation to text categorization, the well-known Rocchio's formula [75], are commonly adopted by content-based recommender systems. The general principle is to allow users to rate documents suggested by the recommender system with respect to their information need. This form of feedback can subsequently be used to incrementally refine the user profile or to train the learning algorithm that infers the user profile as a classifier.

Some linear classifiers consist of an explicit profile (or prototypical document) of the category [82]. The Rocchio's method is used for inducing linear, profile-style classifiers. This algorithm represents documents as vectors, so that documents with similar content have similar vectors. Each component of such a vector corresponds to a term in the document, typically a word. The weight of each component is computed using the TF-IDF term weighting scheme. Learning is achieved by combining document vectors (of positive and negative examples) into a prototype vector for each class in the set of classes C. To classify a new document d, the similarity between the prototype vectors and the corresponding document vector representing d are calculated for each class (for example by using the cosine similarity measure), then d is assigned to the class whose document vector has the highest similarity value.

More formally, Rocchio's method computes a classifier $\overrightarrow{c_i} = \langle \omega_{1i}, \ldots, \omega_{|T|i} \rangle$ for the category c_i (T is the *vocabulary*, that is the set of distinct terms in the training set) by means of the formula:

$$\omega_{ki} = \beta \cdot \sum_{\{d_j \in POS_i\}} \frac{\omega_{kj}}{|POS_i|} - \gamma \cdot \sum_{\{d_j \in NEG_i\}} \frac{\omega_{kj}}{|NEG_i|} \tag{3.7}$$

where ω_{kj} is the TF-IDF weight of the term t_k in document d_j, POS_i and NEG_i are the set of positive and negative examples in the training set for the specific class c_j, β and γ are control parameters that allow to set the relative importance of *all* positive and negative examples. To assign a class $\widetilde{c}$ to a document d_j, the similarity between each prototype vector $\overrightarrow{c_i}$ and the document vector $\overrightarrow{d_j}$ is computed and $\widetilde{c}$ will be the c_i with the highest value of similarity. The Rocchio-based classification approach does not have any theoretic underpinning and there are guarantees on performance or convergence [69].

Relevance feedback has been used in several content-based recommendation systems, such as *YourNews* [2], *Fab* [7] and *NewT* [87].

3.3.2.3 Other Methods

Other learning algorithms have been used in content-based recommendation systems. A very brief description of the most important algorithms follows. A thorough review is presented in [64, 69, 82].

Decision trees are trees in which internal nodes are labeled by terms, branches departing from them are labeled by tests on the weight that the term has in the test document, and leaves are labeled by categories. Decision trees are learned by recursively partitioning training data, that is text documents, into subgroups, until those subgroups contain only instances of a single class. The test for partitioning data is run on the weights that the terms labeling the internal nodes have in the document. The choice of the term on which to operate the partition is generally made according to an information gain or entropy criterion [101]. Decision trees are used in the *Syskill & Webert* [70, 68] recommender system.

Decision rule classifiers are similar to decision trees, because they operates in a similar way to the recursive data partitioning approach decribed above. An advantage of rule learners is that they tend to generate more compact classifiers than decision trees learners. Rule learning methods usually attempt to select from all the possible covering rules (i.e. rules that correctly classify all the training examples) the "best" one according to some minimality criterion.

Nearest neighbor algorithms, also called lazy learners, simply store training data in memory, and classify a new unseen item by comparing it to all stored items by using a similarity function. The "nearest neighbor" or the "k-nearest neighbors" items are determined, and the class label for the unclassified item is derived from the class labels of the nearest neighbors. A similarity function is needed, for example the cosine similarity measure is adopted when items are represented using the VSM. Nearest neighbor algorithms are quite effective, albeit the most important drawback is their inefficiency at classification time, since they do not have a true training phase and thus defer all the computation to classification time. *Daily Learner* [13] and *Quickstep* [58] use the nearest neighbor algorithm to create a model of the user's short term interest and for associating semantic annotations of papers with class names within the ontology, respectively.

3.4 Trends and Future Research

3.4.1 The Role of User Generated Content in the Recommendation Process

Web 2.0 is a term describing the trend in the use of World Wide Web technology that aims at promoting information sharing and collaboration among users. According to Tim O'Reilly[8], the term "Web 2.0" means putting the user in the center, designing software that critically depends on its users since the content, as in Flickr, Wikipedia, Del.icio.us, or YouTube, is contributed by thousands or millions of users. That is why Web 2.0 is also called the "participative Web". O'Reilly[9] also defined Web 2.0 as "the design of systems that get better the more people use them".

One of the forms of User Generated Content (UGC) that has drawn more attention from the research community is *folksonomy*, a taxonomy generated by users who collaboratively annotate and categorize resources of interests with freely chosen keywords called *tags*.

Despite the considerable amount of researches done in the context of recommender systems, the specific problem of integrating tags into standard recommender

[8] http://radar.oreilly.com/archives/2006/12/web-20-compact.html, Accessed on March 18, 2009

[9] http://www.npr.org/templates/story/story.php?storyId=98499899, Accessed on March 18, 2009

system algorithms, especially content-based ones, is less explored than the problem of recommending tags (i.e. assisting users for annotation purposes) [98, 95].

Folksonomies provide new opportunities and challenges in the field of recommender systems (see Chapter 19). It should be investigated whether they might be a valuable source of information about user interests and whether they could be included in user profiles. Indeed, several difficulties of tagging systems have been identified, such as polysemy and synonymy of tags, or the different expertise and purposes of tagging participants that may result in tags at various levels of abstraction to describe a resource, or the chaotic proliferation of tags [40].

3.4.1.1 Social Tagging Recommender Systems

Several methods have been proposed for taking into account user tagging activity within content-based recommender systems.

In [28], the user profile is represented in the form of a tag vector, with each element indicating the number of times a tag has been assigned to a document by that user. A more sophisticated approach is proposed in [57], which takes into account tag co-occurrence. The matching of profiles to information sources is achieved by using simple string matching. As the authors themselves foresee, the matching could be enhanced by adopting WORDNET.

In the work by Szomszor et al. [96], the authors describe a movie recommendation system built purely on the keywords assigned to movies via collaborative tagging. Recommendations for the active user are produced by algorithms based on the similarity between the keywords of a movie and those of the tag-clouds of movies she rated. As the authors themselves state, their recommendation algorithms can be improved by combining tag-based profiling techniques with more traditional content-based recommender strategies.

In [33], different strategies are proposed to build tag-based user profiles and to exploit them for producing music recommendations. Tag-based user profiles are defined as collections of tags, which have been chosen by a user to annotate tracks, together with corresponding scores representing the user interest in each of these tags, inferred from tag usage and frequencies of listened tracks.

While in the above described approaches only a single set of popular tags represents user interests, in [102] it is observed that this may not be the most suitable representation of a user profile, since it is not able to reflect the multiple interests of users. Therefore, the authors propose a network analysis technique (based on clustering), performed on the personal tags of a user to identify her different interests.

About tag interpretation, Cantador et al. [19] proposed a methodology to select "meaningful" tags from an initial set of raw tags by exploiting WORDNET, Wikipedia and Google. If a tag has an exact match in WORDNET, it is accepted, otherwise possible misspellings and compound nouns are discovered by using the Google "did you mean" mechanism (for example the tag *sanfrancisco* or *san farncisco* is corrected to *san francisco*). Finally, tags are correlated to their appropriate Wikipedia entries.

In the work by de Gemmis et al. [36], a more sophisticated approach, implementing a *hybrid* strategy for learning a user profile from both (static) content and tags associated with items rated by that user, is described. The authors include in the user profile not only her *personal* tags, but also the tags adopted by other users who rated the same items (*social* tags). This aspect is particularly important when users who contribute to the folksonomy have different expertise in the domain. The inclusion of social tags in the personal profile of a user allows also to extend the pure content-based recommendation paradigm toward a hybrid content-collaborative paradigm [17]. Furthermore, a solution to the challenging task of identifying user interests from tags is proposed. Since the main problem lies in the fact that tags are freely chosen by users and their actual meaning is usually not very clear, the authors suggested to semantically interpret tags by means of a Word Sense Disambiguation algorithm based on WORDNET. A similar hybrid approach, combining content-based profiles and interests revealed through tagging activities is also described in [38].

Some ideas on how to analyze tags by means of WORDNET in order to capture their intended meanings are also reported in [20], but suggested ideas are not supported by empirical evaluations. Another approach in which tags are semantically interpreted by means of WORDNET is the one proposed in [104]. The authors demonstrated the usefulness of tags in collaborative filtering, by designing an algorithm for neighbor selection that exploits a WORDNET-based semantic distance between tags assigned by different users.

We believe that it could be useful to investigate more on the challenging task of identifying the meaning of tags by relying on different knowledge sources such as WordNet or Wikipedia. Moreover, new strategies for integrating tags in the process of learning content-based profiles should be devised, by taking into account the different nature of personal, social and expert tags. Indeed, personal tags are mostly subjective and inconsistent, expert tags, on the other hand, are an attempt to be objective and consistent. Social tags leads to some form of coherence [5].

Another interesting research direction might be represented by the analysis of tags as a powerful kind of feedback to infer user profiles. Tags that express user opinions and emotions, such as boring, interesting, good, bad, etc., could represent a user's degree of satisfaction with an item. Techniques from the affective computing research area are needed.

3.4.2 Beyond Over-specializion: Serendipity

As introduced in Section 3.2.2, content-based systems suffer from over-specialization, since they recommend only items similar to those already rated by users. One possible solution to address this problem is the introduction of some randomness. For example, the use of genetic algorithms has been proposed in the context of information filtering [87]. In addition, the problem with over-specialization is not only that the content-based systems cannot recommend items that are different from anything the user has seen before. In certain cases, items should not be recommended if they are too similar to something the user has already seen, such as a different news article describing the same event. Therefore, some content-based recommender systems, such as Daily-Learner [13], filter out items if they are too similar to something the user has seen before. The use of redundancy measures has been proposed by Zhang et al. [103] to evaluate whether a document that is deemed to be relevant contains some novel information as well. In summary, the *diversity* of recommendations is often a desirable feature in recommender systems.

Serendipity in a recommender can be seen as the experience of receiving an unexpected and fortuitous item recommendation, therefore it is a way to diversify recommendations. While people rely on exploration and luck to find new items that they did not know they wanted (e.g. a person may not know she likes watching talk shows until she accidentally turns to David Letterman), due to over-specialization, content-based systems have no inherent method for generating serendipitous recommendations, according to Gup's theory [41].

It is useful to make a clear distinction between *novelty* and *serendipity*. As explained by Herlocker [42], novelty occurs when the system suggests to the user an unknown item that she might have autonomously discovered. A serendipitous recommendation helps the user to find a surprisingly interesting item that she might not have otherwise discovered (or it would have been really hard to discover). To provide a clear example of the difference between novelty and serendipity, consider a recommendation system that simply recommends movies that were directed by the user's favorite director. If the system recommends a movie that the user was not aware of, the movie will be novel, but probably not serendipitous. On the other hand, a recommender that suggests a movie by a new director is more likely to provide serendipitous recommendations. Recommendations that are serendipitous are by definition also novel.

We look at the serendipity problem as the challenge of *programming* for serendipity, that is to find a manner to introduce serendipity in the recommendation process in an *operational* way. From this perspective, the problem has not been deeply studied, and there are really few theoretical and experimental studies.

Like Toms explains [97], there are three kinds of information searching:

1. seeking information about a well-defined object;
2. seeking information about an object that cannot be fully described, but that will be recognized at first sight;
3. acquiring information in an accidental, incidental, or serendipitous manner.

It is easy to realize that serendipitous happenings are quite useless for the first two ways of acquisition, but are extremely important for the third kind. As our discussion concerns the implementation of a serendipity-inducing strategy for a content-based recommender, the appropriate metaphor in a real-world situation could be one of a person going for shopping or visiting a museum who, while walking around seeking nothing in particular, would find something completely new that she has never expected to find, that is definitely interesting for her. Among different approaches which have been proposed for "operationally induced serendipity", Toms suggests four strategies, from simplistic to more complex ones [97]:

- Role of chance or *blind luck*, implemented via a random information node generator;
- Pasteur principle ("chance favors the prepared mind"), implemented via a user profile;
- Anomalies and exceptions, partially implemented via poor similarity measures;
- Reasoning by analogy, whose implementation is currently unknown.

In [44], it is described a proposal that implements the "Anomalies and exceptions" approach, in order to provide serendipitous recommendations alongside classical ones, thus providing the user with new entry points to the items in the system. The basic assumption is that the lower is the probability that user knows an item, the higher is the probability that a specific item could result in a serendipitous recommendation. The probability that a user knows something semantically near to what the system is confident she knows is higher than the probability of something semantically far. In other words, it is more likely to get a serendipitous recommendation by providing the user with something less similar to her profile. Following this principle, the basic idea underlying the system proposed in [44] is to ground the search for potentially *serendipitous items* on the similarity between the item descriptions and the user profile. The system is implemented as a naïve Bayes classifier, able to categorize an item as interesting (class c_+) or not (class c_-), depending on the a-posteriori probabilities computed by the classifier. In order to integrate Toms' "poor similarity" within the recommender, the item-profile matching produces a list of items ranked according to the a-posteriori probability for the class c_+. That list contains on the top the most similar items to the user profile, i.e. the items whose classification score for the class c_+ is high. On the other hand, the items for which the a-posteriori probability for the class c_- is higher, are ranked down in the list. The items on which the system is more uncertain are the ones for which the difference between the two classification scores for c_+ and c_- tends to zero. Therefore it is reasonable to assume that those items are not known by the user, since the system was not able to clearly classify them as relevant or not. The items for which the lowest difference between the two classification scores for c_+ and c_- is observed are the most uncertainly categorized, thus it might result to be the most serendipitous.

Regarding serendipity evaluation, there is a level of emotional response associated with serendipity that is difficult to capture, therefore an effective serendipity measurement should move beyond the conventional accuracy metrics and their associated experimental methodologies. New user-centric directions for evaluating new emerging aspects in recommender systems, such as serendipity of recommendations, are required [55]. Developing these measures constitutes an interesting and important research topic (see Chapter 8).

In conclusion, the adoption of strategies for realizing operational serendipity is an effective way to extend the capabilities of content-based recommender systems in order to mitigate the over-specialization problem, by providing the user with surprising suggestions.

3.5 Conclusions

In this chapter we surveyed the field of content-based recommender systems, by providing an overview of the most important aspects characterizing that kind of systems. Although there is a bunch of recommender systems in different domains, they share in common a means for representing items to be recommended and user profiles. The paper discusses the main issues related to the representation of items, starting from simple techniques for representing structured data, to more complex techniques coming from the Information Retrieval research area for unstructured data. We analyzed the main content recommender systems developed in the last 15 years, by highlighting the reasons for which a more complex "semantic analysis" of content is needed in order to go beyond the syntactic evidence of user interests provided by keywords. A review of the main strategies (and systems) adopted to introduce some semantics in the recommendation process is carried out, by providing evidence of the leading role of linguistic knowledge, even if a more specific knowledge is mandatory for a deeper understanding and contextualization of the user interests in different application domains. The latest issues in advanced text representation using sources of world knowledge, such as Wikipedia, have been highlighted, albeit they have not yet used in the context of learning user profiles. In order to complete the survey, a variety of learning algorithms have been described as well.

The last part of the chapter is devoted to the discussion of the main trends and research for the next generation of content-based recommender systems. More specifically, the chapter presents some aspects of the Web 2.0 (r)evolution, that changed the game for personalization, since the role of people evolved from passive consumers of information to that of active contributors. Possible strategies to integrate user-defined lexicons, such as folksonomies, as a way for taking into account evolving vocabularies are debated, by presenting some recent works and possible ideas for further investigations.

Finally, a very specific aspect of content recommender systems is presented. Due to the nature of this kind of systems, they can only recommend items that score highly against a user's profile, thus the user is limited to being recommended items similar to those already rated. This shortcoming, called over-specialization, prevent these systems to be effectively used in real world scenarios. Possible ways to feed users with surprising and unexpected (serendipitous) recommendations are analyzed.

To conclude this survey, we want to underline the importance of research in language processing for advanced item representation in order to get more reliable recommendations. Just as an example, it is worth to cite the news published by the U.S. Patent and Trademark Office regarding series of intriguing patent applications from Google Inc. One of this patents, namely the Open Profile, for instance, would consider a user profile like "I really enjoy hiking, especially long hikes when you can camp out for a few days. Indoor activities don't interest me at all, and I really don't like boring outdoor activities like gardening". Using smart language-processing algorithms to detect the user's sentiments ("enjoy" or "don't like" near "hiking" or "gardening") and other linguistic cues, the system would then potentially serve up active outdoor sports-related ads to this user but avoid ads about more hobbyist-oriented activities [3].

We hope that the issues presented in this chapter will contribute to stimulate the research community about the next generation of content-based recommendation technologies.

References

1. Aciar, S., Zhang, D., Simoff, S., Debenham, J.: Informed Recommender: Basing Recommendations on Consumer Product Reviews. IEEE Intelligent Systems **22**(3), 39–47 (2007)
2. Ahn, J., Brusilovsky, P., Grady, J., He, D., Syn, S.Y.: Open User Profiles for Adaptive News Systems: Help or Harm? In: C.L. Williamson, M.E. Zurko, P.F. Patel-Schneider, P.J. Shenoy (eds.) Proceedings of the 16th International Conference on World Wide Web, pp. 11–20. ACM (2007)
3. Anderson, M.: Google Searches for Ad Dollars in Social Networks. IEEE Spectrum **45**(12), 16 (2008)
4. Asnicar, F., Tasso, C.: ifWeb: a Prototype of User Model-based Intelligent Agent for Documentation Filtering and Navigation in the Word Wide Web. In: C. Tasso, A. Jameson, C.L. Paris (eds.) Proceedings of the First International Workshop on Adaptive Systems and User Modeling on the World Wide Web, Sixth International Conference on User Modeling, pp. 3–12. Chia Laguna, Sardinia, Italy (1997)
5. Aurnhammer, M., Hanappe, P., Steels, L.: Integrating Collaborative Tagging and Emergent Semantics for Image Retrieval. In: Proceedings of the WWW 2006 Collaborative Web Tagging Workshop (2006)
6. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley (1999)
7. Balabanovic, M., Shoham, Y.: Fab: Content-based, Collaborative Recommendation. Communications of the ACM **40**(3), 66–72 (1997)
8. Basile, P., Degemmis, M., Gentile, A., Lops, P., Semeraro, G.: UNIBA: JIGSAW algorithm for Word Sense Disambiguation. In: Proceedings of the 4th ACL 2007 International Workshop on Semantic Evaluations (SemEval-2007), Prague, Czech Republic, pp. 398–401. Association for Computational Linguistics (2007)
9. Basile, P., de Gemmis, M., Gentile, A., Iaquinta, L., Lops, P., Semeraro, G.: An Electronic Performance Support System Based on a Hybrid Content-Collaborative Recommender System. Neural Network World: International Journal on Neural and Mass-Parallel Computing and Information Systems **17**(6), 529–541 (2007)
10. Basile, P., de Gemmis, M., Gentile, A., Iaquinta, L., Lops, P.: The JUMP project: Domain Ontologies and Linguistic Knowledge @ Work. In: Proceedings of the 4th Italian Semantic Web Applications and Perspectives - SWAP 2007, CEUR Workshop Proceedings. CEUR-WS.org (2007)
11. Billsus, D., Pazzani, M.: Learning Probabilistic User Models. In: Proceedings of the Workshop on Machine Learning for User Modeling. Chia Laguna, IT (1997). URL `citeseer.nj.nec.com/billsus96learning.html`
12. Billsus, D., Pazzani, M.J.: A Hybrid User Model for News Story Classification. In: Proceedings of the Seventh International Conference on User Modeling.Banff, Canada (1999)
13. Billsus, D., Pazzani, M.J.: User Modeling for Adaptive News Access. User Modeling and User-Adapted Interaction **10**(2-3), 147–180 (2000)
14. Blanco-Fernandez, Y., Pazos-Arias J. J., G.S.A., Ramos-Cabrer, M., Lopez-Nores, M.: Providing Entertainment by Content-based Filtering and Semantic Reasoning in Intelligent Recommender Systems. IEEE Transactions on Consumer Electronics **54**(2), 727–735 (2008)

15. Bollacker, K.D., Giles, C.L.: CiteSeer: An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications. In: K. Sycara, M. Wooldridge (eds.) Proceedings of the Second International Conference on Autonomous Agents, pp. 116–123. ACM Press (1998)
16. Boone, G.: Concept Features in Re:Agent, an Intelligent Email Agent. In: K. Sycara, M. Wooldridge (eds.) Proceedings of the Second International Conference on Autonomous Agents, pp. 141–148. ACM Press (1998)
17. Burke, R.: Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction **12**(4), 331–370 (2002)
18. Cantador, I., Bellogín, A., Castells, P.: News@hand: A Semantic Web Approach to Recommending News. In: W. Nejdl, J. Kay, P. Pu, E. Herder (eds.) Adaptive Hypermedia and Adaptive Web-Based Systems, *Lecture Notes in Computer Science*, vol. 5149, pp. 279–283. Springer (2008)
19. Cantador, I., Szomszor, M., Alani, H., Fernandez, M., Castells, P.: Ontological User Profiles with Tagging History for Multi-Domain Recommendations. In: Proceedings of the Collective Semantics: Collective Intelligence and the Semantic Web, CISWeb2008, Tenerife, Spain (2008)
20. Carmagnola, F., Cena, F., Cortassa, O., Gena, C., Torre, I.: Towards a Tag-Based User Model: How Can User Model Benefit from Tags? In: User Modeling 2007, *Lecture Notes in Computer Science*, vol. 4511, pp. 445–449. Springer (2007)
21. Celma, O., Ramírez, M., Herrera, P.: Foafing the Music: A Music Recommendation System based on RSS Feeds and User Preferences. In: 6th International Conference on Music Information Retrieval (ISMIR), pp. 464–467. London, UK (2005)
22. Celma, O., Serra, X.: FOAFing the Music: Bridging the Semantic Gap in Music Recommendation. Web Semantics **6**(4), 250–256 (2008)
23. Chen, L., Sycara, K.: WebMate: A Personal Agent for Browsing and Searching. In: K.P. Sycara, M. Wooldridge (eds.) Proceedings of the 2nd International Conference on Autonomous Agents, pp. 9–13. ACM Press, New York (1998)
24. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M.: Combining Content-Based and Collaborative Filters in an Online Newspaper. In: Proceedings of ACM SIGIR Workshop on Recommender Systems (1999). URL `citeseer.ist.psu.edu/claypool99combining.html`
25. Collins, A.M., Loftus, E.F.: A Spreading Activation Theory of Semantic Processing. Psychological Review **82**(6), 407–428 (1975)
26. Csomai, A., Mihalcea, R.: Linking Documents to Encyclopedic Knowledge. IEEE Intelligent Systems **23**(5), 34–41 (2008)
27. Degemmis, M., Lops, P., Semeraro, G.: A Content-collaborative Recommender that Exploits WordNet-based User Profiles for Neighborhood Formation. User Modeling and User-Adapted Interaction: The Journal of Personalization Research (UMUAI) **17**(3), 217–255 (2007). Springer Science + Business Media B.V.
28. Diederich, J., Iofciu, T.: Finding Communities of Practice from User Profiles Based On Folksonomies. In: Innovative Approaches for Learning and Knowledge Sharing, EC-TEL Workshop Proc., pp. 288–297 (2006)
29. Domingos, P., Pazzani, M.J.: On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. Machine Learning **29**(2-3), 103–130 (1997)
30. Egozi, O., Gabrilovich, E., Markovitch, S.: Concept-Based Feature Generation and Selection for Information Retrieval. In: D. Fox, C.P. Gomes (eds.) Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, pp. 1132–1137. AAAI Press (2008). ISBN 978-1-57735-368-3
31. Eirinaki, M., Vazirgiannis, M., Varlamis, I.: SEWeP: Using Site Semantics and a Taxonomy to enhance the Web Personalization Process. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 99–108. ACM (2003)
32. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press (1998)

33. Firan, C.S., Nejdl, W., Paiu, R.: The Benefit of Using Tag-Based Profiles. In: Proceedings of the Latin American Web Conference, pp. 32–41. IEEE Computer Society, Washington, DC, USA (2007). DOI http://dx.doi.org/10.1109/LA-WEB.2007.24. ISBN 0-7695-3008-7
34. Gabrilovich, E., Markovitch, S.: Overcoming the Brittleness Bottleneck using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge. In: Proceedings of the Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, pp. 1301–1306. AAAI Press (2006)
35. Gabrilovich, E., Markovitch, S.: Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In: M.M. Veloso (ed.) Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp. 1606–1611 (2007)
36. Gemmis, M.d., Lops, P., Semeraro, G., Basile, P.: Integrating Tags in a Semantic Content-based Recommender. In: Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys 2008, Lausanne, Switzerland, October 23-25, 2008, pp. 163–170 (2008)
37. Giles, J.: Internet Encyclopaedias Go Head to Head. Nature **438**, 900–901 (2005)
38. Godoy, D., Amandi, A.: Hybrid Content and Tag-based Profiles for Recommendation in Collaborative Tagging Systems. In: Proceedings of the 6th Latin American Web Congress (LA-WEB 2008), pp. 58–65. IEEE Computer Society (2008). ISBN 978-0-7695-3397-1
39. Goldberg, D., Nichols, D., Oki, B., Terry, D.: Using Collaborative Filtering to Weave an Information Tapestry. Communications of the ACM **35**(12), 61–70 (1992). URL `http://www.xerox.com/PARC/dlbx/tapestry-papers/TN44.ps`. Special Issue on Information Filtering
40. Golder, S., Huberman, B.A.: The Structure of Collaborative Tagging Systems. Journal of Information Science **32**(2), 198–208 (2006)
41. Gup, T.: Technology and the End of Serendipity. The Chronicle of Higher Education (44), 52 (1997)
42. Herlocker, L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems **22**(1), 5–53 (2004)
43. Holte, R.C., Yan, J.N.Y.: Inferring What a User Is Not Interested in. In: G.I. McCalla (ed.) Advances in Artificial Intelligence, *Lecture Notes in Computer Science*, vol. 1081, pp. 159–171 (1996). ISBN 3-540-61291-2
44. Iaquinta, L., de Gemmis, M., Lops, P., Semeraro, G., Filannino, M., Molino, P.: Introducing Serendipity in a Content-based Recommender System. In: F. Xhafa, F. Herrera, A. Abraham, M. Köppen, J.M. Benitez (eds.) Proceedings of the Eighth International Conference on Hybrid Intelligent Systems HIS-2008, pp. 168–173. IEEE Computer Society Press, Los Alamitos, California (2008)
45. Joachims, T., Freitag, D., Mitchell, T.M.: Web Watcher: A Tour Guide for the World Wide Web. In: 15th International Joint Conference on Artificial Intelligence, pp. 770–777 (1997). URL `citeseer.ist.psu.edu/article/joachims97webwatcher.html`
46. Kim, S.B., Han, K.S., Rim, H.C., Myaeng, S.H.: Some Effective Techniques for Naïve Bayes Text Classification. IEEE Trans. Knowl. Data Eng. **18**(11), 1457–1466 (2006)
47. Lees-Miller, J., Anderson, F., Hoehn, B., Greiner, R.: Does Wikipedia Information Help Netflix Predictions? In: Seventh International Conference on Machine Learning and Applications (ICMLA), pp. 337–343. IEEE Computer Society (2008). ISBN 978-0-7695-3495-4
48. Lewis, D.D., Ringuette, M.: A Comparison of Two Learning Algorithms for Text Categorization. In: Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, pp. 81–93. Las Vegas, US (1994)
49. Lieberman, H.: Letizia: an Agent that Assists Web Browsing. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 924–929. Morgan Kaufmann (1995)
50. Linden, G., Smith, B., York, J.: Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing **7**(1), 76–80 (2003)
51. Magnini, B., Strapparava, C.: Experiments in Word Domain Disambiguation for Parallel Texts. In: Proc. of SIGLEX Workshop on Word Senses and Multi-linguality, Hong-Kong, October 2000. ACL (2000)

52. Magnini, B., Strapparava, C.: Improving User Modelling with Content-based Techniques. In: Proceedings of the 8th International Conference of User Modeling, pp. 74–83. Springer (2001)
53. Mak, H., Koprinska, I., Poon, J.: INTIMATE: A Web-Based Movie Recommender Using Text Categorization. In: Proceedings of the IEEE/WIC International Conference on Web Intelligence, pp. 602–605. IEEE Computer Society (2003). ISBN 0-7695-1932-6
54. McCallum, A., Nigam, K.: A Comparison of Event Models for Naïve Bayes Text Classification. In: Proceedings of the AAAI/ICML-98 Workshop on Learning for Text Categorization, pp. 41–48. AAAI Press (1998)
55. McNee, S.M., Riedl, J., Konstan, J.A.: Accurate is not Always Good: How Accuracy Metrics have hurt Recommender Systems. In: Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (2006)
56. Melville, P., Mooney, R.J., Nagarajan, R.: Content-Boosted Collaborative Filtering for Improved Recommendations. In: Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI-02), pp. 187–192. AAAI Press, Menlo Parc, CA, USA (2002)
57. Michlmayr, E., Cayzer, S.: Learning User Profiles from Tagging Data and Leveraging them for Personal(ized) Information Access. In: Proc. of the Workshop on Tagging and Metadata for Social Information Organization, Int. WWW Conf. (2007)
58. Middleton, S.E., Shadbolt, N.R., De Roure, D.C.: Ontological User Profiling in Recommender Systems. ACM Transactions on Information Systems **22**(1), 54–88 (2004)
59. Mihalcea, R., Csomai, A.: Wikify!: Linking Documents to Encyclopedic Knowledge. In: Proceedings of the sixteenth ACM conference on Conference on Information and Knowledge Management, pp. 233–242. ACM, New York, NY, USA (2007). DOI http://doi.acm.org/10.1145/1321440.1321475. ISBN 978-1-59593-803-9
60. Miller, G.: WordNet: An On-Line Lexical Database. International Journal of Lexicography **3**(4) (1990). (Special Issue)
61. Mitchell, T.: Machine Learning. McGraw-Hill, New York (1997)
62. Mladenic, D.: Machine learning used by Personal WebWatcher. In: Proceedings of ACAI-99 Workshop on Machine Learning and Intelligent Agents (1999)
63. Mladenic, D.: Text-learning and Related Intelligent Agents: A Survey. IEEE Intelligent Systems **14**(4), 44–54 (1999)
64. Montaner, M., Lopez, B., Rosa, J.L.D.L.: A Taxonomy of Recommender Agents on the Internet. Artificial Intelligence Review **19**(4), 285–330 (2003)
65. Mooney, R.J., Roy, L.: Content-Based Book Recommending Using Learning for Text Categorization. In: Proceedings of the 5th ACM Conference on Digital Libraries, pp. 195–204. ACM Press, New York, US, San Antonio, US (2000)
66. Moukas, A.: Amalthaea Information Discovery and Filtering Using a Multiagent Evolving Ecosystem. Applied Artificial Intelligence **11**(5), 437–457 (1997)
67. Mukherjee, R., Jonsdottir, G., Sen, S., Sarathi, P.: MOVIES2GO: an Online Voting based Movie Recommender System. In: Proceedings of the Fifth International Conference on Autonomous Agents, pp. 114–115. ACM Press (2001)
68. Pazzani, M., Billsus, D.: Learning and Revising User Profiles: The Identification of Interesting Web Sites. Machine Learning **27**(3), 313–331 (1997)
69. Pazzani, M.J., Billsus, D.: Content-Based Recommendation Systems. In: P. Brusilovsky, A. Kobsa, W. Nejdl (eds.) The Adaptive Web, *Lecture Notes in Computer Science*, vol. 4321, pp. 325–341 (2007). ISBN 978-3-540-72078-2
70. Pazzani, M.J., Muramatsu, J., Billsus, D.: Syskill and Webert: Identifying Interesting Web Sites. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference, pp. 54–61. AAAI Press / MIT Press, Menlo Park (1996)
71. Picard, R.W.: Affective Computing. MIT Press (2000)
72. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In: Proceedings of ACM 1994 Conference

on Computer Supported Cooperative Work, pp. 175–186. ACM, Chapel Hill, North Carolina (1994). URL `citeseer.ist.psu.edu/resnick94grouplens.html`

73. Resnick, P., Varian, H.: Recommender Systems. Communications of the ACM **40**(3), 56–58 (1997)
74. Rich, E.: User Modeling via Stereotypes. Cognitive Science **3**, 329–354 (1979)
75. Rocchio, J.: Relevance Feedback Information Retrieval. In: G. Salton (ed.) The SMART retrieval system - experiments in automated document processing, pp. 313–323. Prentice-Hall, Englewood Cliffs, NJ (1971)
76. Rokach, L., Maimon, O., Data Mining with Decision Trees: Theory and Applications, World Scientific Publishing (2008).
77. Sahlgren, M.: The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-dimensional Vector Spaces. Ph.D. thesis, Stockholm: Stockholm University, Faculty of Humanities, Department of Linguistics (2006)
78. Salter, J., Antonoupoulos, N.: CinemaScreen Recommender Agent: Combining collaborative and content-based filtering. IEEE Intelligent Systems **21**(1), 35–41 (2006)
79. Salton, G.: Automatic Text Processing. Addison-Wesley (1989)
80. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)
81. Schwab, I., Kobsa, A., Koychev, I.: Learning User Interests through Positive Examples using Content Analysis and Collaborative Filtering (2001). URL `citeseer.ist.psu.edu/schwab01learning.html`
82. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys **34**(1) (2002)
83. Semeraro, G., Basile, P., de Gemmis, M., Lops, P.: User Profiles for Personalizing Digital Libraries. In: Y.L. Theng, S. Foo, D.G.H. Lian, J.C. Na (eds.) Handbook of Research on Digital Libraries: Design, Development and Impact, pp. 149–158. IGI Global (2009). ISBN 978-159904879-6
84. Semeraro, G., Degemmis, M., Lops, P., Basile, P.: Combining Learning and Word Sense Disambiguation for Intelligent User Profiling. In: M.M. Veloso (ed.) Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp. 2856–2861 (2007). ISBN 978-I-57735-298-3
85. Semeraro, G., Lops, P., Basile, P., Gemmis, M.d.: Knowledge Infusion into Content-based Recommender Systems. In: Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009, New York, USA, October 22-25, 2009 (2009). To appear
86. Shardanand, U., Maes, P.: Social Information Filtering: Algorithms for Automating "Word of Mouth". In: Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems, vol. 1, pp. 210–217 (1995). URL `citeseer.ist.psu.edu/shardanand95social.html`
87. Sheth, B., Maes, P.: Evolving Agents for Personalized Information Filtering. In: Proceedings of the Ninth Conference on Artificial Intelligence for Applications, pp. 345–352. IEEE Computer Society Press (1993)
88. Smirnov, A.V., Krizhanovsky, A.: Information Filtering based on Wiki Index Database. CoRR **abs/0804.2354** (2008)
89. Smith, B., Cotter, P.: A Personalized TV Listings Service for the Digital TV Age. Knowledge-Based Systems **13**, 53–59 (2000)
90. Sorensen, H., McElligott, M.: PSUN: A Profiling System for Usenet News. In: Proceedings of CIKM '95 Intelligent Information Agents Workshop (1995)
91. Sorensen, H., O'Riordan, A., O'Riordan, C.: Profiling with the INFOrmer Text Filtering Agent. Journal of Universal Computer Science **3**(8), 988–1006 (1997)
92. Stefani, A., Strapparava, C.: Personalizing Access to Web Sites: The SiteIF Project. In: Proc. of second Workshop on Adaptive Hypertext and Hypermedia, Pittsburgh, June 1998 (1998)
93. Straffin, P.D.J.: Topics in the Theory of Voting. The UMAP expository monograph series. Birkhauser (1980)

94. Symeonidis, P.: Content-based Dimensionality Reduction for Recommender Systems. In: C. Preisach, H. Burkhardt, L. Schmidt-Thieme, R. Decker (eds.) Data Analysis, Machine Learning and Applications, Studies in Classification, Data Analysis, and Knowledge Organization, pp. 619–626. Springer Berlin Heidelberg (2008). ISBN 978-3-540-78239-1
95. Symeonidis, P., Nanopoulos, A., Manolopoulos, Y.: Tag Recommendations based on Tensor Dimensionality Reduction. In: Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys 2008, Lausanne, Switzerland, October 23-25, 2008, pp. 43–50 (2008)
96. Szomszor, M., Cattuto, C., Alani, H., O'Hara, K., Baldassarri, A., Loreto, V., Servedio, V.D.P.: Folksonomies, the Semantic Web, and Movie Recommendation. In: Proceedings of the Workshop on Bridging the Gap between Semantic Web and Web 2.0 at the 4th ESWC (2007)
97. Toms, E.: Serendipitous Information Retrieval. In: Proceedings of DELOS Workshop: Information Seeking, Searching and Querying in Digital Libraries (2000)
98. Tso-Sutter, K.H.L., Marinho, L.B., Schmidt-Thieme, L.: Tag-aware Recommender Systems by Fusion of Collaborative Filtering Algorithms. In: SAC '08: Proceedings of the 2008 ACM symposium on Applied computing, pp. 1995–1999. ACM (2008). ISBN 978-1-59593-753-7
99. Wasfi, A.M.: Collecting User Access Patterns for Building User Profiles and Collaborative Filtering. In: Proceedings of the International Conference on Intelligent User Interfaces, pp. 57–64 (1999)
100. Witten, I.H., Bell, T.: The Zero-frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression. IEEE Transactions on Information Theory **37**(4) (1991)
101. Yang, Y., Pedersen, J.O.: A Comparative Study on Feature Selection in Text Categorization. In: D.H. Fisher (ed.) Proceedings of ICML-97, 14th International Conference on Machine Learning, pp. 412–420. Morgan Kaufmann Publishers, San Francisco, US, Nashville, US (1997). URL `citeseer.ist.psu.edu/yang97comparative.html`
102. Yeung, C.M.A., Gibbins, N., Shadbolt, N.: A Study of User Profile Generation from Folksonomies. In: Proc. of the Workshop on Social Web and Knowledge Management, WWW Conf. (2008)
103. Zhang, Y., Callan, J., Minka, T.: Novelty and Redundancy Detection in Adaptive Filtering. In: Proceedings of the 25th International ACM SIGIR Conference, pp. 81–88 (2002)
104. Zhao, S., Du, N., Nauerz, A., Zhang, X., Yuan, Q., Fu, R.: Improved Recommendation based on Collaborative Tagging Behaviors. In: Proceedings of International Conference on Intelligent User Interfaces, IUI, pp. 413–416. ACM (2008). ISBN 978-1-59593-987-6