Chapter 4
Semantics-Aware Content-Based Recommender Systems

Marco de Gemmis, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro

4.1 Introduction

Content-based recommender systems (CBRSs) rely on item and user descriptions (content) to build item representations and user profiles to suggest items similar to those a target user already liked in the past. The basic process of producing content-based recommendations consists in matching up the attributes of the target user profile, in which preferences and interests are stored, with the attributes of the items. The result is a relevance score that predicts the target user's level of interest in those items. Usually, attributes for describing an item are features extracted from metadata associated to that item, or textual features extracted directly from the item description. The content extracted from metadata is often too short and not sufficient to correctly define the user interests, while the use of textual features involves a number of complications when learning a user profile due to natural language ambiguity. Polysemy, synonymy, multi-word expressions, named entity recognition and disambiguation are inherent problems of traditional keyword-based profiles, which are not able to go beyond the usage of lexical/syntactic structures to infer the user interest in topics.

The ever increasing interest in *semantic technologies* and the availability of several open knowledge sources, such as Wikipedia, DBpedia, Freebase, and BabelNet have fueled recent progress in the field of CBRSs. Novel research works have introduced semantic techniques that shift from a *keyword*-based to a *concept*-based representation of items and user profiles. These observations make very relevant the integration of proper techniques for deep content analytics borrowed

M. de Gemmis • P. Lops (✉) • C. Musto • F. Narducci • G. Semeraro
Department of Computer Science, University of Bari "Aldo Moro", Bari, Italy
e-mail: marco.degemmis@uniba.it; pasquale.lops@uniba.it; cataldo.musto@uniba.it; fedelucio.narducci@uniba.it; giovanni.semeraro@uniba.it

F. Ricci et al. (eds.), *Recommender Systems Handbook*,
DOI 10.1007/978-1-4899-7637-6_4

from Natural Language Processing (NLP) and Semantic Technologies, which is one of the most innovative lines of research in *semantic recommender systems* [61].

We roughly classify semantic techniques into *top-down* and *bottom-up* approaches. Top-down approaches rely on the integration of external knowledge, such as machine readable dictionaries, taxonomies (or IS-A hierarchies), thesauri or ontologies (with or without value restrictions and logical constraints), for annotating items and representing user profiles in order to capture the semantics of the target user information needs. The main motivation behind top-down approaches is the challenge of providing recommender systems with the linguistic knowledge and common sense knowledge, as well as the cultural background which characterize the human ability of interpreting documents expressed in natural language and reasoning on their meaning.

On the other side, bottom-up approaches exploit the so-called geometric metaphor of meaning to represent complex syntagmatic and paradigmatic relations between words in high-dimensional vector spaces. According to this metaphor, each word (and each document as well) can be represented as a point in a vector space. The peculiarity of these models is that the representation is learned by analyzing the context in which the word is used, in a way that terms (or documents) similar to each other are close in the space. For this reason bottom-up approaches are also called distributional models. One of the great virtues of these approaches is that they are able to induce the semantics of terms by analyzing their use in large corpora of textual documents using unsupervised mechanisms, as evidenced by the recent advances of machine translation techniques [52, 83].

This chapter describes a variety of semantic approaches, both top-down and bottom-up, and shows how to leverage them to build a new generation of semantic CBRSs that we call *semantics-aware content-based recommender systems*.

4.2 Overview of Content-Based Recommender Systems

This section reports an overview of the basic principles for building CBRSs, the main techniques for representing items, learning user profiles and providing recommendations. The most important limitations of CBRSs are also discussed, while the semantic techniques useful to tackle those limitations are introduced in the next sections.

The high level architecture of a content-based recommender system is depicted in Fig. 4.1. 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

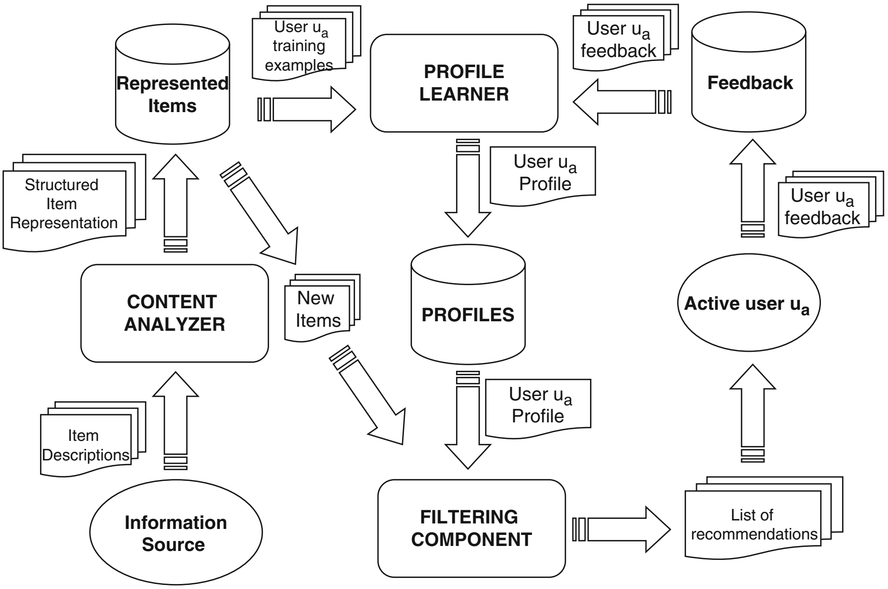


Fig. 4.1 High level architecture of a content-based recommender

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 [86], 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 [113] 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 (computed using some similarity metrics [57]), 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 [6, 118]. 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* [51] 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 [58]). 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 [111]. 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. After the user profile has been learned, the FILTERING COMPONENT predicts whether a new item 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).

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 enables the system to take into account the dynamic nature of user preferences.

4.2.1 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). 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 $D = \{d_1, d_2, \ldots, d_N\}$ denote a set of documents or corpus, and $T = \{t_1, t_2, \ldots, t_n\}$ 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 $\overrightarrow{d_j} = \langle w_{1j}, w_{2j}, \ldots, w_{nj} \rangle$, 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 [117]:

- 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:

$$\text{TF-IDF}(t_k, d_j) = \underbrace{\text{TF}(t_k, d_j)}_{\text{TF}} \cdot \underbrace{log \frac{N}{n_k}}_{\text{IDF}} \tag{4.1}$$

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.

$$\text{TF}(t_k, d_j) = \frac{f_{k,j}}{max_z f_{z,j}} \tag{4.2}$$

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 Eq. (4.1) are usually normalized by cosine normalization:

$$w_{k,j} = \frac{\text{TF-IDF}(t_k, d_j)}{\sqrt{\sum_{s=1}^{|T|} \text{TF-IDF}(t_s, d_j)^2}} \tag{4.3}$$

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:

$$sim(d_i, d_j) = \frac{\sum_k w_{ki} \cdot w_{kj}}{\sqrt{\sum_k w_{ki}^2} \cdot \sqrt{\sum_k w_{kj}^2}} \tag{4.4}$$

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.

4.2.2 Methods for Learning User Profiles

Machine learning techniques generally used in the task of inducing content-based profiles, are well-suited for text categorization [119]. In a machine learning approach to text categorization, an inductive process automatically builds a text classifier from a set of *training documents*, i.e. documents labeled with the categories they belong to.

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). Classifiers can be also adopted with a set of categories which is not binary. Besides the use of classifiers, other machine learning algorithms, such as linear regression, can be adopted to predict numerical ratings. The most used learning algorithms in content-based recommender systems are based on probabilistic methods, relevance feedback and k-nearest neighbors [6].

4.2.2.1 Probabilistic Methods

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 $P(c|d)$:

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)} \tag{4.5}$$

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 [12, 70].

There are two commonly used working models of the naïve Bayes classifier, the *multivariate Bernoulli* event model and the *multinomial* event model [77]. 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 [77]. The way the multinomial event model uses its document vector to calculate $P(c_j|d_i)$ is as follows:

$$P(c_j|d_i) = P(c_j) \prod_{t_k \in V_{d_i}} P(t_k|c_j)^{N_{(d_i,t_k)}} \tag{4.6}$$

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 [129].

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 [40]. Another advantage of the naïve Bayes approach is that it is very efficient and easy to implement compared to other learning methods.

4.2.2.2 Relevance Feedback

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 [113], are commonly adopted by content-based recommender systems. The general principle is to let 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 [119]. 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:

$$\omega_{ki} = \beta \cdot \sum_{\{d_j \in POS_i\}} \frac{w_{kj}}{|POS_i|} - \gamma \cdot \sum_{\{d_j \in NEG_i\}} \frac{w_{kj}}{|NEG_i|} \tag{4.7}$$

where w_{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_i, β and γ are control parameters that allow to set the relative importance of *all* positive and negative examples. To assign a class $\tilde{c}$ to a document d_j, the similarity between each prototype vector $\overrightarrow{c_i}$ and the document vector $\overrightarrow{d_j}$ is computed and $\tilde{c}$ 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 [108].

4.2.2.3 Nearest Neighbors

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.

4.2.3 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. Furthermore, CBRSs based on a string matching approach suffer from problems of:

  - POLYSEMY, the presence of multiple meanings for one word;
  - SYNONYMY, multiple words with the same meaning;
  - MULTI-WORD EXPRESSIONS, the difficulty to assign the correct properties to a sequence of two or more words whose properties are not predictable from the properties of the individual words;
  - ENTITY IDENTIFICATION or NAMED ENTITY RECOGNITION, the difficulty to locate and classify elements in text into pre-defined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, etc.
  - ENTITY LINKING or NAMED ENTITY DISAMBIGUATION, the difficulty of determining the identity (often called the reference) of entities mentioned in text.

- 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 *lack of 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.

4.3 Top-Down Semantic Approaches

There is an ever increasing interest in using a deep domain knowledge as part of the recommendation process, in order to deal with the main problems of CBRSs (i.e., limited content analysis, overspecialization) and generate more accurate recommendations. To this purpose, several CBRSs:

- incorporate ontological knowledge, ranging from simple linguistic ontologies, to more complex domain-specific ones [81];
- leverage unstructured or semi-structured encyclopedic knowledge sources, such as Wikipedia [120];
- try to exploit the wealth of the so-called Linked Open Data cloud [39].

The following sections provide an overview of CBRSs, with the aim of imposing a degree of order on the diversity of the knowledge sources and techniques exploited for the representation of items and user profiles. Section 4.3.1 describes the role of ontologies for defining advanced CBRSs, by highlighting the main advantages and drawbacks, while recommendation approaches leveraging encyclopedic knowledge are described in Sect. 4.3.2, with the proposal of new ontological resources which can be effectively used for improving CBRSs. Finally, more recent approaches based on the Linked Open Data cloud are discussed in Sect. 4.3.3.

4.3.1 Approaches Based on Ontological Resources

The leading role of linguistic knowledge is highlighted by the wide use of WordNet [84], which is mostly adopted for the semantic interpretation of content by using Word Sense Disambiguation (WSD) algorithms. In [36, 37], WordNet and WSD algorithms are used to integrate linguistic knowledge in the process of learning user profiles. The basic building block for WordNet is the SYNSET (SYNonym SET), which represents a specific meaning of a word. Hence, items are represented according to a synset-based vector space model, and the user profile includes those synsets that turn out to be most indicative of the user preferences. In addition to the better performance of synset-based profiles, the advantage is that synset-based representations are inherently multilingual. Indeed, concepts (word meanings) remain the same across different languages, while terms used for describing them change in each specific language. Using lexical resources such as MultiWordNet [9], which associates a unique identifier to each possible sense (meaning) of a word, regardless the original language, it is possible to define a bridge between different languages. In [71], a WSD algorithm exploiting MultiWordNet as sense repository is integrated in the design of MARS (MultilAnguage Recommender System), a cross-language recommender system whose effectiveness is comparable to a classical monolingual content-based recommender. Similarly, in [75] the authors present a personal agent for a multilingual news Web site, which adopts a synset-based document representation obtained through a Word Domain Disambiguation algorithm [74] which exploits MultiWordNet.

More recent works still rely on WordNet to define semantic recommender systems. In [25], a semantic approach to news recommendation making use of WordNet is investigated. WordNet synsets are used to compute similarities between unread news articles and articles stored in user profiles by adopting the Wu and Palmer semantic similarity measure [130]. However, in order to cope with the lack of support for named entities, the authors extend the WordNet-based recommendation approach with a similarity based on page counts for named entities stemming from a Web search engine. WordNet and WSD are also adopted in [27] to compute the semantic similarity between short microblog posts in order to recommend tweets related to what a user has issued or trending topics.

In spite of the advantages provided by WordNet, there are several limitations related to its limited coverage for named entities, events, contemporary terms, and in general specific knowledge. With the advent of the Semantic Web [10], ontologies emerged as powerful means for representing domain knowledge in many areas, and for this reason several approaches have been proposed to incorporate ontological knowledge in recommender systems. Ontologies are used to describe domain-specific knowledge and they are commonly handled as hierarchies of concepts with attributes and relations, which establish a terminology to define semantic networks of interrelated concepts and instances. In general, when a domain model is represented as an ontology, items and user models consist of a subset of concepts from the domain ontology, possibly with associated values characterizing their importance. In [82], the recommendation of on-line academic research papers is performed by leveraging a research topic ontology, based on the computer science classifications, for representing both items and user profiles. The match is based on the correlation between the topics in the user profile and those associated to the papers. The same process is adopted in [22, 23] to recommend news. Item descriptions are vectors of TF-IDF scores in the space of ontology concepts, user profiles are represented in the same space, and the item-profile matching is performed as a cosine-based vector similarity, differently from the strategy in [21, 24], in which item and user spaces are clustered in order to build implicit communities of interest which enable recommendation based on the similarities among them. In [121], the similarity between an item and a user profile is based on the existence of the same concepts or related concepts, according to their position in a three-level ontology, while a more advanced recommendation method is described in [16], where a spreading activation algorithm is adopted on ontology-profiles to suggest interesting and novel items to the user. Spreading activation is used in [26] as well, where the propagation from a small number of initial concepts (those which received the user feedback) to other related domain concepts allows to provide finer recommendations and to tackle the cold start problem. The novelty of the approach relies on the definition of a set of contextualized propagation strategies, ranging from the horizontal propagation among siblings, to the anisotropic vertical one among ancestors and descendants, which permits user interests to be propagated differently upward and downward.

The use of ontologies for adding a semantic dimension to items and user profiles may be beneficial for limiting some of the problems of CBRSs and providing better recommendations. Ontology-based user profiles are less ambiguous, and the structure of the ontology may be adopted to define measures able to estimate how semantically related two concepts are. Different types of measures are provided in the literature, ranging from link-based (e.g. Wu and Palmer, Leacock and Chodorow) to node-based ones (e.g. Resnik, Jiang and Conrath, Lin). More details about those measures can be found in [19].

On the other hand, there are difficulties which hinder the use of ontologies in recommender systems. The development of rich and expressive domain-specific ontologies is a time consuming task which has to be performed by human experts, and there are also the onerous tasks of ontology population and maintenance to perform [63]. Hence, there is an increasing attention of many researchers towards the integration of world knowledge which may be extracted from online collaborative resources, in order to exploit the richness of such resources to come up with *semantics-aware recommender systems*.

4.3.2 Approaches Based on Unstructured or Semi-Structured Encyclopedic Knowledge

Studies in Artificial Intelligence (AI) have already recognized the importance of knowledge for problem solving. Back in the early years of AI research, Buchanan and Feigenbaum [18] formulated the knowledge-as-power hypothesis, which postulated that "The power of an intelligent program to perform its task well depends primarily on the quantity and quality of knowledge it has about that task".

Many knowledge sources have become available in the last years, both structured and unstructured [Open Directory Project (ODP), Yahoo!Web Directory, and Wikipedia]. The use of external knowledge sources can be useful to better understand the information items (documents, news, product descriptions) and to extract more meaningful features, in order to design advanced content-based filtering methods able to provide better recommendations. Among unstructured knowledge sources, Wikipedia emerges as the most used source of information for several tasks [8, 42, 59, 96]. The main advantages of using Wikipedia, rather than conventional document archives, as a knowledge source are:

- it is freely available on the Web;
- it is a wide-coverage resource which is under constant development by the community;
- it is available in several languages, hence can be seen as a multilingual corpus;
- it is very accurate [50].

On the other hand, Wikipedia knowledge is available in textual form written by humans for humans, and enough common-sense knowledge is needed to correctly understand the meaning of articles. For this reason, natural language understanding capabilities are required for the interpretation of Wikipedia pages and for making them *machine processable*.

The problem of extracting and using knowledge contained in Wikipedia was studied by several researchers [33, 46, 49]. Different techniques have been defined, which exploit the encyclopedic knowledge contained in Wikipedia for *selecting the most accurate semantic features* to represent the items, or for *generating new semantic features* to enrich the item representation.

The most prominent approaches which perform *feature selection* are Wikify! [33] and Tagme [46]. Wikify! allows to identify important concepts in a text representation by using keyword extraction, and then to link these concepts to the corresponding Wikipedia pages by exploiting WSD techniques. More specifically, Wikify! is a system for automatically cross-referencing documents with Wikipedia [85]. The system is trained on Wikipedia articles, and thus learns to disambiguate and detect links in the same way as Wikipedia editors [45].

Tagme [46] augments a text representation with pertinent hyperlinks to Wikipedia pages, by implementing an anchor disambiguation algorithm which exploits inter-relations between Wikipedia pages, as well as other heuristics. The main advantage of Tagme is its ability to annotate texts which are short and poorly composed, such as snippets coming from search engine result pages, tweets, news, etc.

An approach which leverages Wikipedia knowledge to *generate* new features for enriching items representation is Explicit Semantic Analysis (ESA) [49]. ESA provides a fine-grained semantic representation of text documents as a weighted vector of concepts derived from Wikipedia. Specifically, concepts correspond to Wikipedia articles, e.g. such as WOODY ALLEN, APPLE INC., or MACHINE LEARNING. Explicit Semantic Analysis resembles the well known Latent Semantic Analysis technique [35], whose representation is based on *latent* (and not comprehensible) features, rather than *explicit* (and comprehensible) concepts derived from Wikipedia (concepts explicitly defined and manipulated by humans).

In [48, 49], ESA was adopted for computing semantic relatedness of natural language texts, with better performance with respect to a keyword-based approach. In [43], ESA is adopted to enrich documents and queries to enhance traditional bag-of-words-based retrieval models, while in [8], ESA is used for enriching bag-of-words representing news or blog feeds before their clustering. ESA was also effectively used to augment the bag-of-words representation with Wikipedia-based features in the text categorization task [49].

Finally, the availability of Wikipedia knowledge in several languages and the multilingual alignment of Wikipedia articles allow to have cross-lingual and multilingual services. Potthast et al. [109] proposed a Wikipedia-based multilingual retrieval model for the analysis of cross-language similarity. They demonstrated that, given a query in a specific language, the most similar documents from a corpus in another language were properly ranked. They used Cross-Language Explicit Semantic Analysis (CL-ESA), an extension of ESA for cross-language retrieval. Recently, ESA was also used to develop the Cross-language Service Retriever tool (CroSeR), to support the cross-language linking of e-Government services to the Linked Open Data cloud [98].

4.3.2.1 Explicit Semantic Analysis

The idea behind ESA is to view an encyclopedia as a collection of concepts, each of which accompanied with a large body of text (the article content). The power of ESA is the capability of representing the Wikipedia knowledge base in a way that is directly used by machines, without the need for manually encoded common-sense knowledge. The gist of the technique is to use the high-dimensional space defined by these concepts in order to represent the meaning of natural language texts. ESA allows to leverage Wikipedia knowledge by defining relationships between terms and Wikipedia articles.

More formally, given a set of basic concepts $C = \{c_1, c_2, \ldots, c_n\}$, a term t is represented by a vector of weights $< w_1, w_2, \ldots, w_n >$, where w_i represents the strength of association between t and c_i. The set of concepts C are one to one associated to documents $D = \{d_1, d_2, \ldots, d_n\}$ (the Wikipedia articles). Hence, a sparse matrix T is built, called *ESA-matrix*, where each column corresponds to a concept (title of Wikipedia article), and each row corresponds to a term (word) that occurs in $\bigcup_{i=1\ldots n} d_i$. The entry $T[i,j]$ of the matrix represents the TF-IDF of term t_i in document d_j. Finally, length normalization is applied to each column to disregard differences in document length. This allows to define the semantics of a term t_i as a point in the n-dimensional semantic space of Wikipedia concepts. The weighed vector corresponding to a term t_i is called *semantic interpretation vector*. The semantics of a text fragment $< t_1, t_2, \ldots, t_k >$ (i.e. a sentence, a paragraph, an entire document) is obtained by computing the centroid (average vector) of the semantic interpretation vectors of the individual terms occurring in the fragment. This definition allows to partially perform WSD [49].

As an example consider the text fragment of a news title “Apple patents a Tablet Mac”. Without deep knowledge of hi-tech industry and gadgets, one finds it hard to predict the content of the news item. Using Wikipedia it is possible to identify the following related concepts: APPLE COMPUTER (with the correct identification of the concept representing the computer company rather than the fruit), MAC OS, LAPTOP, AQUA (the GUI of Mac OS X), IPOD, and APPLE NEWTON (the name of Apple’s early personal digital assistant).

4.3.2.2 CBRSs Leveraging Encyclopedic Knowledge

Even though the above mentioned indexing methods have been adopted for several tasks, they are not yet widely used in the context of learning user profiles and providing recommendations. However, CBRSs may benefit of the Wikipedia-based representation. Indeed, the feature generation process, adopted for example by ESA, can lead to richer item representations, able to improve the overlap between items and profiles. Indeed the new features allow to match items that did not share any keyword with the profile before the feature generation process. ESA is also able to introduce new related concepts for generating less obvious and more serendipitous (unexpected) recommendations.

In [91], an enhanced semantic TV-show representation for Personalized Electronic Program Guides is proposed. ESA is used to enrich the textual descriptions associated to TV shows with additional features extracted from Wikipedia, in order to improve the ranking of the most relevant items for each program genre. ESA is exploited to enrich a classic bag-of-words representation with 20, 40, or 60 new features, and it was adopted to enrich German TV-show descriptions. To this purpose, the German Wikipedia dump (released on October 13th, 2010 with a size of approximately 7.5 GB) was processed in order to obtain the corresponding German ESA-matrix. Results showed that the enhanced bag-of-words representation outperforms the classical bag-of-words one in terms of precision.

Besides the improvement of accuracy, the work carried out in [97] shows that, leveraging encyclopedic knowledge for representing user interests allows to introduce serendipitous topics and to obtain more understandable and transparent user profiles. Transparency is defined as the extent to which keywords in the user profile reflect the actual user interests. In that work user interests have been gathered from Facebook profiles by extracting both interests explicitly declared by users and those implicitly inferred from posts and other published content. The feature generation process implemented by ESA helps to introduce new serendipitous topics of interests, while the feature selection process implemented by Tagme helps to obtain more comprehensible user profiles, more representative of user interests.

These results are confirmed in the user study presented in [96], in which both ESA and Tagme are effectively used to improve the performance of a news recommender. News titles are extracted from a set of RSS feeds, and the profile of interests is built by extracting information from the Facebook and Twitter accounts of the user. The information extracted (news, posts, tweets) are represented using keywords, ESA concepts or Tagme concepts, respectively. The representation obtained by Tagme outperforms the others in terms of transparency and accuracy. This is probably due to the ability of Tagme to effectively annotate very short texts, such as news titles.

The ability of the ESA technique to cope with the cold-start problem is shown in [105], in which a CBRS in the context of non-fiction multimedia recommendation of TED lectures is presented. Using ESA as indexing method for titles and descriptions of talks allows to obtain the best performance with respect to other semantic representations, and this shows that a representation of items based on external knowledge is significantly more useful than the domain knowledge captured intrinsically by the other semantic methods.

4.3.2.3 BabelNet: An Encyclopedic Dictionary

Resources like Wikipedia lack full coverage for the lexicographic senses of lemmas, which is instead provided by a computational lexicon, such as WordNet . In this section we briefly describe a new resource, called BabelNet [100], which integrates the largest multilingual Web encyclopedia, i.e., Wikipedia, and the most popular computational lexicon, i.e., WordNet, to obtain a very large multilingual semantic network. BabelNet integrates the linguistic knowledge contained in WordNet and the encyclopedic knowledge contained in Wikipedia for providing an *encyclopedic dictionary*. It encodes knowledge as a labeled directed graph. *Nodes* are *concepts* extracted from WordNet and Wikipedia, i.e. word senses (synsets) available in WordNet, and encyclopedic entries (Wikipages) extracted from Wikipedia, while *edges* connecting the nodes are labeled with *semantic relations* coming from WordNet, as well as semantically unspecified relations from hyperlinked text coming from Wikipedia. Each node also contains a set of lexicalizations of the concept for different languages, e.g., APPLE for English, MANZANA for Spanish, MELA for Italian, POMME for French, ... These multilingually lexicalized concepts are called *Babel synsets*. The current version (2.0) of BabelNet covers 50 languages, and contains more than nine millions Babel synsets and 262 millions of lexico-semantic relations.

Figure 4.2 presents an excerpt of two results obtained by issuing the query "apple" to BabelNet.[1] The system returns 11 different senses of "apple", such as fruit, the British rock band, the multinational corporation, etc. Clicking on the sense allows to link to the corresponding WordNet synset or Wikipedia page in that specific language. The system also reports the set of glosses extracted from the different resources and the categories extracted from the corresponding Wikipedia pages. For each sense, its semantically related concepts may also be explored. For example, some of the concepts related to apple in the sense of the multinational corporation—*Apple Inc*—are *computer architecture, Power Mac G4, Apple ProDOS, etc*. More information about BabelNet can be found in [100].

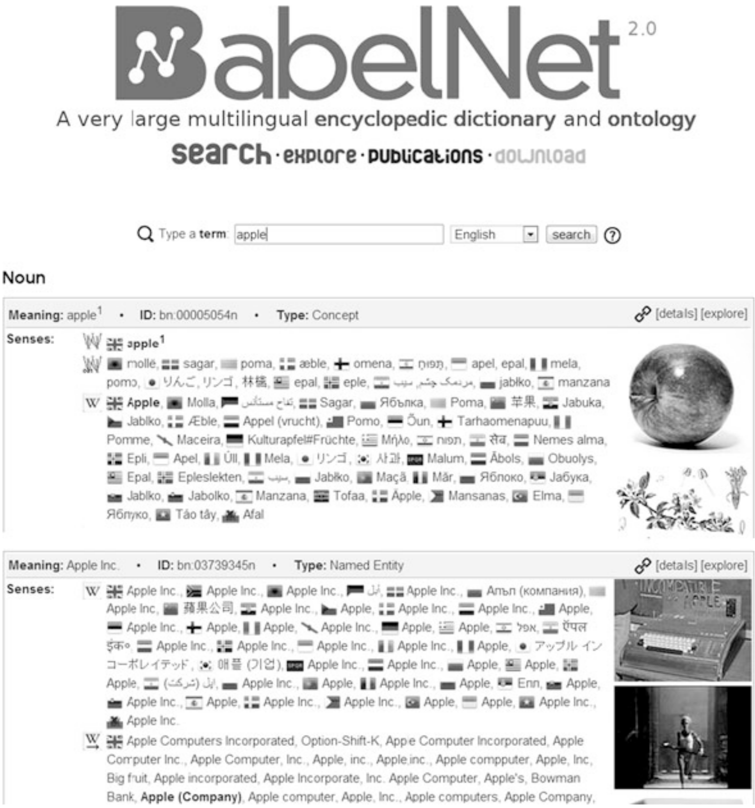
BabelNet sense inventory can be effectively used for a variety of tasks, ranging from multilingual semantic relatedness [101], to (multilingual) WSD [99, 102]. The use of BabelNet can also fuel the progress on the research on CBRSs, which could rely on knowledge-richer approaches to represent items and user profiles.

4.3.3 Approaches Based on Linked Open Data

Novel and more accessible forms of information coming from different open knowledge sources represent a new and rapidly growing piece of the big data puzzle. These new sources of open data represent an expanding trove of largely unexploited value, which paves the way to a new generation of recommender systems. Using open or pooled data from many sources, often combined and linked with proprietary big data, can help develop insights difficult to uncover with internal data alone [28]. The Linked Data community has advocated the following set of best principles for collaboratively publishing and interlinking structured data over the Web[2]:

[1] http://babelnet.org/.

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


BabelNet 2.0
A very large multilingual encyclopedic dictionary and ontology
search · explore · publications · download
Type a term
search
?
[details] [explore]
[details] [explore]

Type a term: apple English search

Noun

Meaning: apple[1] • **ID:** bn:00005054n • **Type:** Concept

Senses: **apple**[1]

mollë, sagar, poma, æble, omena, תפוח, apel, epal, mela, pomo, りんご, リンゴ, 林檎, epal, eple, سیب ,مرغک چشم ,سیب, jabłko, manzana

Apple, Molla, تفاح مستأنس, Sagar, Ябълка, Poma, 苹果, Jabuka, Jablko, Æble, Appel (vrucht), Pomo, Õun, Tarhaomenapuu, Pomme, Maceira, Kulturapfel#Früchte, Μήλο, תפוח, सेब, Nemes alma, Epli, Apel, Úll, Mela, リンゴ, 사과, Malum, Ābols, Obuolys, Epal, Epleslekten, سیب, Jabłko, Maçã, Măr, Яблоко, Јабука, Jablko, Jabolko, Manzana, Tofaa, Äpple, Mansanas, Elma, Яблуко, Táo tây, Afal

Meaning: Apple Inc. • **ID:** bn:03739345n • **Type:** Named Entity

Senses: Apple Inc., Apple Inc., Apple Inc., أبل, Apple Inc., Апъл (компания), Apple Inc, 蘋果公司, Apple Inc., Apple, Apple Inc., Apple Inc., Apple, Apple Inc., Apple, Apple, Apple Inc., Apple, Apple, אפל, ऍपल इंक०, Apple Inc., Apple Inc., Apple Inc., Apple Inc., Apple, アップル インコーポレイテッド, 애플 (기업), Apple Inc., Apple Inc., Apple, Apple, Apple, أبل (شركت), Apple Inc., Apple, Apple Inc., Apple, Епл, Apple, Apple Inc., Apple, Apple Inc., Apple Inc., Apple, Apple, Apple Inc., Apple Inc.

Apple Computers Incorporated, Option-Shift-K, Apple Computer Incorporated, Apple Computer Inc., Apple Computer, Inc., Apple, inc., Apple.inc., Apple compputer, Apple, Inc, Big fruit, Apple incorporated, Apple Incorporate, Inc. Apple Computer, Apple's, Bowman Bank, **Apple (Company)**, Apple computer, Apple, Inc., Apple computers, Apple Company,

Fig. 4.2 The result obtained by issuing the query "apple" to BabelNet

- the use of URIs (Uniform Resource Identifier) as names for things (arbitrary real-world entities);
- the use of HTTP URIs so those names can be looked up by people (dereferencing);
- the delivery of useful information upon lookup of those URIs using standards such as RDF and SPARQL;
- the inclusion of links to other URIs to discover more things.

This allows the dissemination of structured data on the Web in an interoperable manner using the Semantic Web standards [14].

Over the last years, more and more semantic data are published following the Linked Data principles, by connecting information referring to geographical locations, people, companies, book, scientific publications, films, music, TV and radio programs, genes, proteins, drugs, online communities, statistical data, and reviews in a single global data space, the *Web of Data* [13]. These datasets interlinked with each other form a global graph, called Linked Open Data cloud.

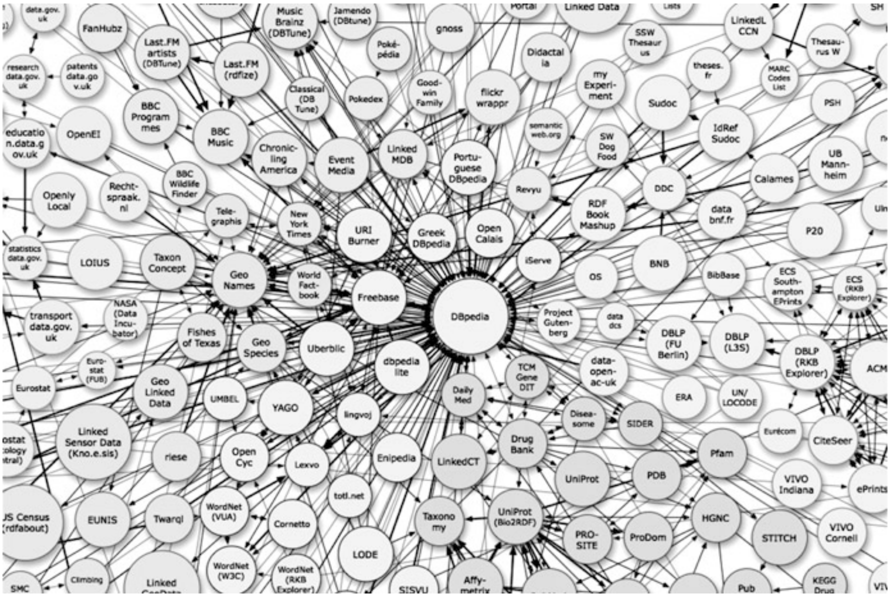


Fig. 4.3 Fragment of the Linked Open Data cloud (as of September 2011)

At the time of writing more than 2100 datasets are available with almost 62 billions of RDF triples.[3] Figure 4.3 shows a fragment of the Linked Open Data cloud, whose nucleus is represented by DBpedia.

The standard mechanism for specifying the existence and meaning of connections between items described in this data is provided by the Resource Description Framework (RDF), which allows to link things by explicitly stating the nature of the connection (typed links). For example, a hyperlink of the type *friend_of* may be set between two people. RDF statements are in the form of *subject-predicate-object* expressions, called *triples*. The *subject* denotes the resource, and the *predicate* denotes an aspect of the resource and expresses a *relationship* between the *subject* and the *object*. Relations are also called *properties*. SPARQL[4] is a SQL-like language for RDF graphs to retrieve and manipulate data stored in RDF format.

In the context of recommender systems, this is useful to interlink diverse information about users, items, and their relations, and to implement reasoning mechanisms that can support and improve the recommendation process [34]. The challenge is to investigate whether and how this large amount of wide-coverage and linked semantic knowledge can significantly improve complex filtering tasks.

[3] http://stats.lod2.eu/.

[4] http://www.w3.org/TR/rdf-sparql-query/ accessed on September 12, 2014.

4.3.3.1 CBRSs Leveraging Linked Open Data

The use of Linked Open Data for recommender systems is very recent. On one hand, the richness and the ontological nature of this data allows to enrich item descriptions and user profiles for different domains. Hence, the use of Linked Open Data helps to *fill in the gaps* in the background data, and to cope with the new user, new item and sparsity problems. On the other hand, the use of such a huge amount of interlinked data poses new challenges for well established recommendation algorithms.

One of the first attempts to leverage Linked Open Data to build recommender systems is *dbrec* [106], a music recommender system using DBpedia to provide recommendations for bands and solo artists. The system is based on the *Linked Data Semantic Distance (LDSD)* algorithm [107], which allows to provide recommendations by computing the semantic distance for all artists referenced in DBpedia. LDSD is a link-based measure; it does not take into account the semantics of the relations, the links hierarchy or other DBpedia properties. It allows explanations when computing the recommendations as a positive side effect of using Linked Open Data. Linked Open Data are also used to mitigate the data acquisition problem of both collaborative and content-based recommender systems. In [56], the architecture of a collaborative recommender system is extended by leveraging user-item connections coming from DBTune [110]; the resulting RDF graph of user-item relations is transformed into a user-item matrix exploited by the recommendation algorithm. In [95], DBpedia is used to enrich the playlists extracted from a Facebook profile with new related artists. Each artist in the original playlist is mapped to a DBpedia node, and other similar artists are selected by taking into account shared properties, such as the genre and the musical category of the artist.

An approach which exploits Linked Open Data for computing cross-domain recommendations is described in [44, 64]. The source and target domains involved in the recommendation scenario are mapped to DBpedia by identifying the classes that belong to the domains of interest, and the relations existing between instances of such classes. Then, a semantic network is defined by querying DBpedia in order to link a specific instance in the source domain with the related instances in the target domain. The recommendation mechanism relies on a graph-based ranking algorithm on the semantic network. The authors focused on a scenario in which recommendations for music artists and tracks are adapted to places of interests, by obtaining very positive results. Similarly to *dbrec*, the approach is able to provide explanations based on the discovered semantic paths between a place of interest and the music artists in the associated semantic network.

A simpler approach to define a CBRS exploiting exclusively Linked Open Data to represent both items and user profiles is proposed in [38]. The ontological information, encoded via specific properties extracted from DBpedia and LinkedMDB [54], is adopted to perform a semantic expansion of the item descriptions, in order to catch implicit relations and hidden information, which are not detectable just looking at the nodes directly linked to the item. The evaluation of different combinations of properties revealed that more properties lead to more accurate recommendations, since this seems to mitigate the limited content analysis issue of CBRSs.

Similarly to the previous work, a CBRS fed exclusively by Linked Open Data is presented in [39]. Data coming from DBpedia [15], LinkedMDB [54] and Freebase [17] are exploited to recommend movies using an adaptation of the Vector Space Model. The RDF graph connecting movies according to some properties is represented as a three-dimensional matrix where each slice refers to an ontology property (e.g. starring, director, genre, ...) and represents its adjacency matrix. A a cell in the matrix is not null if there is a property that relates a subject (on the rows) to an object (on the columns). The weighing scheme is based on TF-IDF and the cosine similarity allows to measure the correlation between two movies. The recommendation step is performed by computing the similarity between the user profile (movies liked and disliked by the user) and movies unknown to the user. The similarity values for each property are combined in a linear fashion, and the best configuration of weights for each property is learned via a genetic algorithm. As in [38], using more ontological information leads to the best performance, and also helps to explain the recommendations by listing, for each property, the values which are common between the movies in the user profile and those suggested.

The same approach devised in [39] is effectively adopted to develop Cinemappy [104] and a recommender system for events [67]. The former is a context-aware CBRS for movies and movie theaters suggestions fed by data coming from localized DBpedia graphs, whose results are enhanced by exploiting contextual information about the user. The latter recommends events, even though some improvements were necessary to deal with the complexity of the domain, such as the social aspect, i.e. the collaborative participation about which friend will attend an event.

All the previous approaches rely on Linked Open Data to catch implicit relations which allow to increase the number of common features between items, or to implement more sophisticated reasoning mechanisms over the graphs. Ultimately, well known reasoning mechanisms for learning content-based user profiles can be adopted on the richer representations provided by leveraging Linked Open Data [89]. An interesting work which goes one step further is presented in [103]; it leverages DBpedia to extract semantic *path-based* features to eventually compute recommendations using a *learning to rank* algorithm. Starting from the common graph-based representation of the content and collaborative data models, all the paths connecting the user to an item are considered in order to have a relevance score for that item. The more paths between a user and an item, the more that item is relevant to that user.

4.3.3.2 (Other) Entity Linking Algorithms

In [1, 2], a semantically-enriched user model based on the analysis of Twitter posts is proposed. Entity linking algorithms are used to enrich and extend user models by identifying the most relevant entities mentioned in the tweets. Similarly, entity linking algorithms are adopted in [94] to enhance item representation in a context-aware content-based recommendation framework. The experimental evaluation showed that entity-based algorithms are able to improve the predictive accuracy of the recommendation framework, in both context-aware and non-contextual recommendation settings.

This section introduces some other known entity linking systems, which can be effectively used to implement semantic CBRSs.

Babelfy[5] [88] is a novel integrated approach to entity linking and word sense disambiguation. Given a lexicalized semantic network, e.g. BabelNet, the approach is based on three steps: (1) the automatic creation of semantic signatures, i.e. related concepts and named entities for each vertex of the semantic network, (2) extraction of all the linkable fragments from a given text, listing all the possible meanings according to the semantic network, and (3) linking based on a high-coherence densest subgraph algorithm.

DBpedia Spotlight [80] has been designed to connect unstructured text to the Linked Open Data cloud by using DBpedia as hub. The output is a set of Wikipedia articles related to a text retrieved by following the URI of the DBpedia instances. The annotation process works in four-stages. First, the text is analyzed in order to select the phrases that may indicate a mention to a DBpedia resource. In this step, spots that are only composed of verbs, adjectives, adverbs and prepositions are disregarded. Subsequently, a set of candidate DBpedia resources is built by mapping the spotted phrase to resources that are candidate disambiguations for that phrase. The disambiguation process uses the context around the spotted phrase to decide for the best choice amongst the candidates.

Other tools allow for the semantic annotation of natural language text, but the techniques used to perform the analysis are not described with sufficient details.

Alchemy[6] offers a NLP processing service able to analyze web pages, documents, and tweets for identifying entities, keywords, concepts, etc. If available, a link to the Linked Open Data cloud is also provided (DBpedia, Yago, Crunchbase, etc.). It also performs sentiment analysis on the input text by assigning a sentiment polarity to the entities identified into the text.

Open Calais[7] exploits NLP and machine learning to find entities within documents. The main difference with respect to other entity recognizers is that Open Calais returns facts and events hidden within the text. Open Calais consists of three main components: (1) a named entity recognizer that identifies people, companies, organizations; (2) a fact recognizer that links the text with position tags, alliance, person-political; (3) an event recognizer whose role is to identify sport, management, change events, labor actions, etc. Open Calais supports English, French and Spanish, and its assets are currently linked to DBpedia, Wikipedia, Freebase, GeoNames.

[5] http://babelfy.org.

[6] http://www.alchemyapi.com/.

[7] http://www.opencalais.com/.

NERD (Named Entity Recognition and Disambiguation)[8] [112] is a framework to unify different named entity extractors, such as Alchemy, DBpedia Spotlight, Open Calais, etc., using the NERD ontology, providing a rich set of axioms aligning the taxonomies of those tools. In the NERD ontology a manual mapping between taxonomies coming from different schemas is established, and a concept is included in the NERD ontology as soon as there are at least three extractors that use it.

4.4 Bottom-Up Semantic Approaches

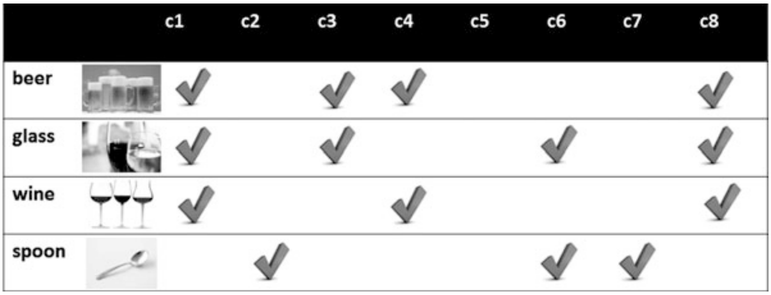
This section focuses on approaches able to produce implicit semantic representation of both items and user profiles that could be defined *lightweight* in contrast to the approaches presented in Sect. 4.3. These techniques are mainly based on the distributional hypothesis, according to which the meaning of words depends on the contexts in which they occur. The most distinguishing aspect of these approaches lies in the fact that the semantic representation is directly learned according to the way terms are used in large corpora of data. Thus, they do not need any *human intervention*, differently from the development of an external resource for semantic content representation or the maintenance of an ontology. Bottom-up semantic approaches just need as much data as possible to learn and represent the meaning of the terms.

The following sections provide the background about Discriminative Models (Sect. 4.4.1), and the basics for the definition of a novel content-based recommendation framework that exploits the strengths of VSM, by tackling its drawbacks at the same time. A novel dimensionality reduction technique, which avoids the need for factorization, is discussed in Sect. 4.4.1.1, and a more sophisticated negation operator to model negative preferences is presented in Sect. 4.4.1.2. A survey of CBRSs built on the ground of the previous methods is finally provided in Sect. 4.4.1.3.

4.4.1 Approaches Based on Discriminative Models

Discriminative Models (DMs) rely on a simple insight: as humans infer the meaning of a word by understanding the contexts in which that word is typically used, discriminative algorithms extract information about the meaning of a word by analyzing its usage in large corpora of textual documents. This means that it is possible to infer the meaning of a term (e.g., *leash*) by analyzing the other terms it co-occurs with (*dog, animal,* etc.) [114]. In the same way, the correlation between different terms (e.g., *leash* and *muzzle*) can be inferred by analyzing the

[8] http://www.wikimeta.com/portfolio_nerd.html.



	c1	c2	c3	c4	c5	c6	c7	c8
beer	✓		✓	✓				✓
glass	✓		✓			✓		✓
wine	✓			✓				✓
spoon		✓				✓	✓	

Fig. 4.4 A term-context matrix. The analysis of the usage patterns of the terms allows to state that *beer* and *wine* or *beer* and *glass* are similar, since they are often used together

similarity between the contexts in which they are used. These approaches rely on the *distributional hypothesis* [53], according to which *"Words that occur in the same contexts tend to have similar meanings"*. This means that words are semantically similar to the extent that they share contexts.

DMs represent information about terms usage in a *term-context* matrix (Fig. 4.4), instead of a term-document matrix adopted in the classic VSM. The advantage is that the *context* is a very flexible concept which can be adapted to the specific granularity level of the representation required by the application: for example, given a word, its context could be either a single word it co-occurs with, or a sliding window of terms that surrounds it, or a sentence, or yet the whole document. In [125], it is presented an interesting survey about the three broad classes of VSM to represent semantics, related to the different types of matrix adopted: (1) term-document matrix—usually used to measure similarity of documents, (2) word-context matrix—usually used to measure similarity of terms, and (3) pair-pattern matrix—usually used to measure similarity of relations (the textual patterns in which the pair X,Y co-occurs, e.g. *X cuts Y* or *X works with Y*).

The classical VSM is the simplest DM proposed in literature, in which co-occurrences are computed by considering the whole document as context. This approach uses *syntagmatic* relations between words to assess their semantic similarity. Indeed, words with a similar meaning will tend to occur in the same document, because they are appropriate to define the particular topic of that document. Instead, the approach based on the co-occurrences computed in a context different from the document uses *paradigmatic* relations, because in a small context window we do not expect that similar words (e.g., synonyms) can co-occur, but we could expect that their surrounding words will be more or less the same.

DMs are referred to as *geometrical models* as well, since each term represented by a row of the term-context matrix can be modeled as a vector. In order to compute relatedness between terms, it is possible to exploit distributional measures that rely on the distributional hypothesis, such as spatial measures (e.g., cosine similarity, Manhattan and Euclidean distances), mutual information-based measures (e.g., Lin), or relative entropy-based measures (e.g., Kullback-Leibler divergence) [87].

On one hand, this representation has the advantage of building a language model, typically referred to as `WordSpace` [72], able to learn similarities and connections in a totally unsupervised way, but on the other hand the dimensionality of vectors when adopting finer-grained representations of contexts is a clear issue (*curse of dimensionality)*. For example, the adoption of sentences as granularity level for contexts causes an explosion of the number of dimensions of the vector space: by assuming 10–20 sentences per document on average, the dimension of the vector space would be 10–20 times the one using a classical term-document matrix. For this reason, feature selection or *dimensionality reduction* techniques must be adopted.

4.4.1.1 Dimensionality Reduction Techniques

Dimensionality reduction techniques help to transform a high-dimensional space into a lower-dimensionality one.

Latent Semantic Indexing (LSI) [35] is a technique for building a semantic vector space representation based on the application of Singular Value Decomposition (SVD) [68] on the term-document matrix. The approach, largely investigated for representing the meaning of terms through statistical computations applied to a large corpus of text, works in two steps: first, the corpus is represented into a matrix in which each row is a word and each column is a text passage (*document*). Next, SVD is applied in order to decompose the original matrix into two matrices of reduced dimensionality (obtained by selecting the largest eigenvalues) that represent the original rows (*terms*) and the original columns in terms of *latent* orthogonal factors.

As pointed out in [11], the reduced orthogonal dimensions resulting from SVD are less noisy than the original data and capture the latent associations between terms and documents.

The use of LSI in the area of CBRSs has been already investigated in several research work [47, 78], and it has been demonstrated that it is able to outperform other techniques, regardless the application domain. In [122], a feature profile of a user is built using both collaborative and content features, and LSI is exploited to detect the dominant features of a user. Recommendations are provided according to this dimensionally-reduced feature profile, with a better performance with respect to both collaborative and content-based as well as hybrid algorithms. Recently, LSI has been effectively adopted as the content-based component of a hybrid algorithm for recommending TV-shows [7, 32], as well as in the task of recommending source code examples according to user requirements [79]. However, Terzi et al. [124] showed that LSI can underperform compared to other approaches when the set of available data is small and the textual content is too short. This outcome confirms the insight that DMs, regardless the dimensionality technique they adopt, are effective when a lot of data about terms usage is available.

Regardless of its effectiveness, LSI suffers from scalability issues inherited from the use of SVD for dimensionality reduction. Consequently, research has been oriented towards the investigation of more scalable and incremental techniques, such as those based on Random Projections (RP) [126], which has its theoretical basis

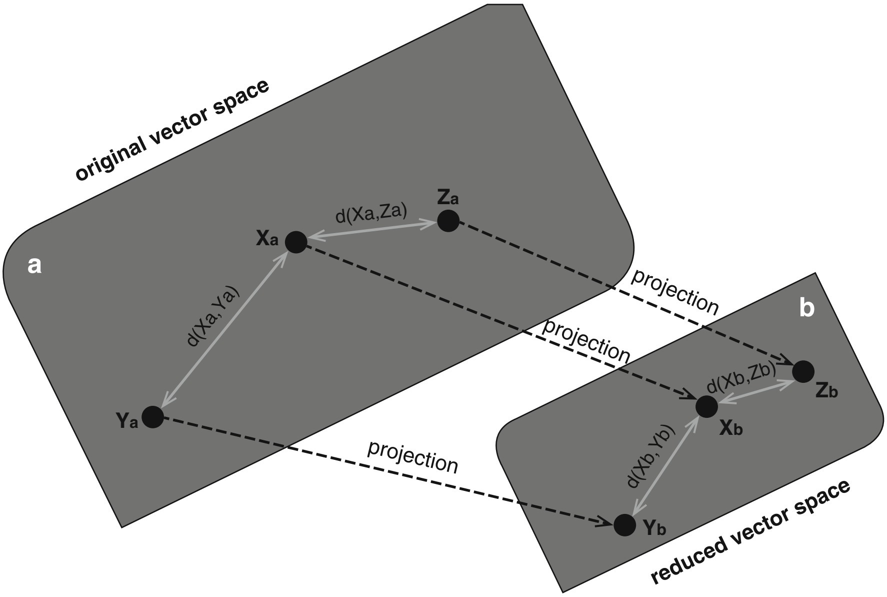


Fig. 4.5 A visual explanation of the Johnson-Lindenstrauss lemma. *Z* is the nearest point to *X* in the reduced vector space, as in the original space, even though the *numerical* value of their pairwise similarity is different

in the Hecht-Nielsen's studies about near-orthogonality [55]. These approaches, originally proposed for clustering text documents [69], do not need factorization, and are based on the insight that a high-dimensional vector space can be *randomly* projected into a space of lower dimensionality without compromising distance metrics. By following this approach, a high-dimensional matrix M of size $n \times m$ is transformed into a reduced k-dimensional matrix M^* as follows:

$$M_{n,m} \times R_{m,k} = M^*_{n,k} \tag{4.8}$$

where the row vectors of R are built in a pseudo-random way (more details follow). According to the Johnson and Lindenstrauss' lemma [62], when the random matrix R is built by following specific constraints, distances between points in the reduced vector space are nearly preserved, i.e. remains proportional with respect to those in the original space (see Fig. 4.5), thus it is still possible to perform similarity computations between points in the reduced vector space with a minimum loss of significance, balanced by the gain in efficiency.

This important outcome has been experimentally confirmed in several works [66, 73]. Despite its advantages, the use of RP is still not widespread compared to SVD. In [29, 123], RP is applied to collaborative filtering, while in [105], RP is used to build an item to item similarity matrix leveraging the reduced vector space representation.

RP was used as dimensionality reduction technique for a discriminative model called Random Indexing (RI) [115, 116]. This strategy, based on Kanerva's work on sparse distributed representations [65], is an incremental technique for creating small-scale `WordSpaces` that merges the advantages of discriminative models with the efficiency of dimensionality reduction based on RP. Similarly to LSI, RI represents terms and documents as points in a *semantic* vector space that is built according to the distributional hypothesis. However, differently from it, RI uses RP instead of SVD as technique for dimensionality reduction. Thus, the heavyweight decomposition performed by SVD is here replaced by an incremental (but effective) algorithm as RP, which performs the same process with less computational cost. Thanks to RI it is possible to represent terms (and documents) through a $n \times k$ term-context matrix, which is more compact than the original $n \times m$ term-document matrix, since k is typically set lower than m. One of the strongest points of RI is its flexibility, since the dimension k is a simple parameter thus it can be adapted to the available computational resources, as well as to the requirements of the specific application domain. Basically, the larger the vector space, the higher the precision in representing word similarities, and the higher the computational resources required to represent and update the model.

The k-dimensional representation is obtained by using the following incremental strategy:

1. A k-dimensional randomly generated context vector is assigned to each context. This vector is sparse, high-dimensional and ternary, which means that its elements have values in $\{-1, 0, 1\}$. Values are distributed in a random way, but the number of non-zero elements has to be much smaller. Specifically, a very common choice is to use a Gaussian distribution for the elements of the context vectors. However, much simpler distributions (zero mean distributions with unit variance) can also be used [3];
2. The vector space representation of a *term* is obtained by summing the context vectors of all the contexts which contain the term;
3. The vector space representation of a *document* is obtained by summing the vector space representation of all the terms (created in step 2) which occur in the document.

Step 2 allows to build a `WordSpace`, while step 3 allows to build a `DocSpace`. Both the spaces have the same dimension. In a `WordSpace` it is possible to compute similarities between different terms, while in a `DocSpace` this is possible for documents. The approach is totally incremental: when a new document comes into play, the algorithm randomly generates a new context vector for it (step 1) and updates the `WordSpace`. The technique is scalable because the calculation of the vector space representation of this new document does not need to generate again the whole vector space, but it is simply obtained by summing the context vectors of the terms that occurs in it.

4.4.1.2 Modeling Negation

The above mentioned novel representation inherits a classical issue of VSM, since the information coming from negative evidence (i.e., items user dislikes) is not taken into account. This is an important aspect for recommender systems, since user profiles are built by modeling positive, as well as negative user preferences. Several works rely on an adaptation of the Rocchio algorithm [113] to incrementally refine the user profiles by exploiting positive and negative feedback provided by users. The problems with the Rocchio algorithm is related to the extensive tuning of parameters needed for being effective and to the lack of solid theoretical foundations of the method. Negative relevance feedback is also discussed in [41], in which the idea of representing negation by subtracting an unwanted vector from a query emerged, even if nothing about *how much to subtract* is stated. This is a problem which we try to clarify using the following example, inspired by Widdows [127].

Let us suppose to have a `WordSpace` built on a corpus of documents related to music (in order to leave disambiguation problems out of this discussion). Consider the term vectors of the two words *rock* and *pop*. The query (or profile) (*rock NOT pop*) should allow to represent *rock* only by the aspects of its meaning which are different from, and preferably unrelated to, those of *pop*. If we subtract the whole vector *pop* from *rock*, we might remove features of *rock* which we wanted to keep. Instead, we should subtract exactly the right amount to make the unwanted vector *pop* irrelevant to the desired result. This removal operation is called *vector negation*, which is related to the concept of *orthogonality*, and it is proposed in [127], according to the principles of Quantum Logic. Meanings are unrelated to one another if they have no features in common at all, precisely when their vectors are orthogonal. Hence, we need to make our final query vector (*rock NOT pop*) orthogonal to *pop*. Geometrically, this corresponds to the *orthogonal projection* of the vector *rock* onto the vector *pop*, that is the vector λpop ($\lambda \in \Re$):

$$\lambda = \frac{rock \cdot pop}{pop \cdot pop} \tag{4.9}$$

From this definition, (*rock NOT pop*) is represented as the vector ($rock - \lambda\ pop$), which is orthogonal to the vector *pop*. For simplicity, we do not discuss here about ambiguity problems (e.g. rock could refer also to geology). More details can be found in [127].

4.4.1.3 CBRSs Leveraging Discriminative Models

One of the first attempt to define a CBRS using discriminative models is presented in [120], in which the process of learning user profiles benefits from the infusion of exogenous knowledge coming from Wikipedia. The knowledge contained in Wikipedia is processed using the Semantic Vectors package [128], in order to build a `WordSpace` model in which related words are close to each other in that space.

A more complete approach using discriminative models based on RI and the above mentioned negation operator is described in [92], which presents a novel content-based recommendation framework called enhanced Vector Space Model (eVSM). In eVSM, RI is used to build a user profile in an incremental way, i.e. by summing all the *document vectors* representing documents liked by that user. More complex models were defined by introducing the negation operator to represent in the user profile both positive and negative preferences. To this purpose, instead of a single vector representing a user profile, two vectors were defined, one for positive preferences ($\mathbf{p}_{+u}$) and one for negative ones ($\mathbf{p}_{-u}$).

The same approach was also used to build language-independent user profiles [90], by assuming that in every language each term often co-occurs with the same other terms (expressed in different languages, of course). Hence, representing a content-based user profile in terms of the co-occurrences of its terms, user preferences become inherently independent from the language and this is sufficient to provide the user with cross-language recommendations. Thus, profiles learnt on English movies were used to recommend Italian movies, and vice versa. Results were accurate and comparable to a classical monolingual recommendation setting. This highlights the power of the approach, which is able to tackle a complex multilingual recommendation task without using any complex operations, such as translation or semantic indexing based on WSD [71].

Recently, the eVSM framework has been further evolved to manage contextual information. In [93], contextual eVSM extends eVSM with a context-aware post-filtering algorithm [5]. More specifically, a semantic representation of the context is built and used to influence non-contextual recommendations. The intuition behind the context representation is that there exists a set of terms that are likely more descriptive than others to model items relevant in a certain context. For example, it is likely that restaurant descriptions containing terms such as *candlelight* or *sea view* are more relevant if the user is looking for a restaurant suitable for a romantic dinner. Experiments demonstrated that contextual eVSM is able to outperform non-contextual baselines in most experimental settings, as well as the state of the art algorithm for context-aware collaborative recommendation proposed in [4].

DMs were also adapted to face the sparsity problem of context-aware recommender systems, which need large datasets of contextually tagged ratings, i.e. ratings for items provided in the different contextual situations. In [30], it is described an approach based on the intuition that, when making recommendations in a particular situation, it can be considered as relevant not only the ratings provided by the users in that situation but also to *reuse* ratings provided in similar situations. The similarity among contextual conditions is estimated by identifying the "meaning" of a condition by means of its implicit semantics, that is captured by the usage of the concept. Experiments demonstrated good performance of the proposed approach, which was further improved in [31].

4.5 Summary and Comparison of Approaches

In the previous sections we analyzed top-down and bottom-up semantic approaches for facing well-known problems of CBRSs (i.e., limited content analysis, overspecialization).

In Table 4.1, pros a cons of each approach are summarized with respect to several criteria: transparency of the models, coverage of topics, complexity of NLP techniques required, ease of applying reasoning mechanisms for discovering relationships between items and profiles, support for multilinguality.

In order to capture the semantics of the user information needs, recommender systems based on top-down approaches can exploit different types of exogenous knowledge that allow advanced concept-based content representation: ontological resources, encyclopedic knowledge, and the Linked Open Data cloud. Conversely, recommender systems based on bottom-up semantic approaches rely on methods able to induce the semantics of terms by analyzing their use in large corpora of documents, i.e. they rely on the so-called distributional hypothesis: *words that occur in the same contexts tend to have similar meanings*.

An important difference between the two approaches is related to transparency: the explicit concept-based representation of both items and profiles allows the definition of less ambiguous user profiles and is particularly useful for estimating the semantic similarity between user preferences and item features. Furthermore, the advanced content representation has an impact on the accuracy of recommendations, allows to mitigate the limited content analysis problem, and also helps to provide well-structured explanations of recommendations in terms of matched concepts.

Bottom-up approaches do not allow an explicit representation of concepts, but the meaning of a word is inferred by analyzing its co-occurrence with context features (other words, larger textual units, or documents). Hence, the semantics is implicitly encoded as high-dimensional vectors learned from large corpora of documents. This is the main limitation of these approaches, that do not allow an intelligible explanation of the recommendations.

Another problem is that high-dimensional vectors require novel dimensionality reduction techniques, in order to improve scalability. In [92], we described Random Indexing, a dimensionality reduction technique which avoids the need for factorization, and we showed how effective user profiles can be built in an incremental way by distinguishing between positive and negative user preferences. The novel content-based recommendation framework based on Random Indexing was able to outperform state-of-the-art techniques, and to easily implement language-independent context-aware recommender systems [90]. This is a relevant advantage of bottom-up approaches, which do not require to perform complex NLP tasks such as translation or WSD in order to provide cross-language recommendations.

Furthermore, an important distinction between the two approaches must be considered with respect to the capability of discovering novel relationships among items and profiles, beyond the simple similarity. Indeed, both the approaches give the possibility of inferring new information (e.g. new words or concepts not

Table 4.1 Overview of semantic approaches for CBRSs

Approach	Description	References	Pros	Cons
Top-down	Use of external knowledge to improve the representation of items and user profiles	Ontological resources [16, 21–27, 36, 37, 71, 75, 82, 121]	– High transparency – Standard NLP techniques required, WSD at most – Easily allow reasoning, due to hierarchical organization of concepts	– Limited coverage of named entities, events, etc. – No multilinguality support, unless specific resources are adopted
		Encyclopedic knowledge [91, 96, 97, 105]	– High transparency – Wide coverage of topics – Multilinguality typically supported – Allows reasoning	– More NLP effort needed than using specific ontologies due to unstructured information
		Linked Open Data cloud and entity linking [1, 2, 38, 39, 44, 56, 64, 67, 94, 95, 103, 104, 106]	– Very high transparency – Wide coverage of topics – Allows deeper reasoning especially for cross-domain recommendation	– Strong NLP effort required for linking data to the Linked Open Data cloud
Bottom-up	Use of implicit semantic representation of items and user profiles. The meaning of a term is inferred by analyzing its usage	Latent semantic indexing [7, 32, 47, 78, 79, 122]	– Allows reasoning by capturing latent associations between terms and documents – Multilinguality typically supported	– Low transparency – A lot of data about terms usage required for building a wide language model – Standard NLP effort, but poor scalability due to the need for dimensionality reduction
		Random indexing [29–31, 90, 92, 93, 105, 120, 123]	– Standard NLP effort and good scalability because factorization is notrequired – Allows reasoning – Support for multilinguality based on co-occurrence of terms	– Low transparency – A lot of data about terms usage required for building a wide language model

explicitly included in the item descriptions), which could be exploited to discover those associations, but the reasoning process could be performed in a different way, depending on the type of knowledge it is based on. For example, ontologies represent the domain knowledge in a more formal way, due to their structured representation, and easily allow reasoning, even at an abstract level, by navigating the concept hierarchy. Obviously, reasoning is influenced by the usually limited coverage of topics, because of the cost of the human-based tasks of building, maintaining and populating ontologies. Hence, the research is moving towards the exploitation of freely available knowledge sources, such as Wikipedia. Encyclopedic knowledge covers a wider range of topics compared to ontologies, is generally multilingual, but requires more NLP effort to analyze unstructured information in order to select or even *generate* semantic features to effectively represent items and user profiles. This capability of generating new semantic features, besides those that can be found in item descriptions, can be exploited to discover unexpected and non-trivial relationships between items and between items and user profiles. However, the NLP effort for performing this task is higher compared to using ontologies, due to the absence of an explicit organization of concepts. Similarly, the lack of a structured representation of concepts in bottom-up approaches does not make the implementation of reasoning capabilities as easy as for ontology-based approaches. Anyway, the fact that discriminative models are able to catch *latent* associations between terms can help to find not-trivial correlations among items [120]. On the other side, the graph-based organization of the Linked Open Data cloud facilitates the adoption of even sophisticated reasoning mechanisms, as the one described in [103], that allow deeper reasoning connecting data in different domains and promote cross-domain recommendations. A significant effort here is due to need of linking data to the Linked Open Data cloud.

4.6 Conclusions and Future Challenges

This chapter was structured around two different approaches for introducing semantics in CBRSs: top-down and bottom-up. Both approaches have advantages and drawbacks, and pose new challenges in the development of CBRSs. Many other recommendation scenarios may benefit from semantic-based approaches. In the context of sentiment analysis, concept-based approaches proved to be superior to purely syntactical techniques [20], hence recommender systems which rely on the analysis of opinions written in natural language for extracting user preferences and affective states, might effectively adopt all the techniques presented in this chapter to provide better suggestions.

In conclusion, research on content-based recommender systems produced a variety of solid methods, some of which having their roots in NLP foundations, but still poses some interesting challenges:

- Definition of recommendation methods able to reason on the graph structure of the Linked Open Data cloud to discover latent connections among items and user profiles, as suggested in [39]. Those emerging relations could be exploited for cross-domain recommendations or diversification of suggestions. As an example, the Linked Open Data-enabled Recommender Systems Challenge of the 11th European Semantic Web Conference has shown how Linked Open Data and semantic technologies can boost the creation of a new breed of knowledge-enabled and content-based recommender systems. In particular, one of the tasks of the challenge was devoted to the design of Linked Open Data-enabled recommender systems whose effectiveness was evaluated by considering a combination of both accuracy of the recommendation list and the diversity of items belonging to it. Diversity is a very popular topic in content-based recommender systems, which usually suffer from overspecialization;
- Definition of content-based methods for mining microblogging data and deep analysis of text reviews. In particular, aspect-based opinion mining and sentiment analysis techniques can support the design of recommendation methods that take into account the evaluation of aspects of items expressed in text reviews. As an example, "Aspect Based Sentiment Analysis" was one of the tasks of SemEval 2014, and was devoted to evaluate methods for automated detection of both aspects and the sentiment expressed towards each aspect in text reviews of laptops and restaurants. These methods could be exploited for implicit rating of aspects and can support the development of multi-criteria recommendation techniques;
- Definition of personality-based recommendation methods based on automated recognition of personality. Content-based methods can be exploited to detect personality markers in language through the extraction of linguistic features associated with personality traits [76]. Automated modeling of personality from text can ease the development of systems that incorporate personality aspects into recommendation methods to enhance both recommendation quality and user experience [60]. The design of personality-based and emotion-aware personalized services is an emerging research topic, as shown also by the recent EMPIRE workshops hold in conjunction with the Conference on User Modelling, Adaptation and Personalization.

We hope that this chapter may stimulate the research community to adopt and effectively integrate the discussed techniques in several recommendation scenarios in order to foster future innovations in the area of CBRSs.

Acknowledgements The authors would like to express their deep gratitude to Professor Michael J. Pazzani and Dr. Daniel Billsus for their seminal work on machine learning for user modeling and content-based recommendation systems, which inspired many ideas developed in this chapter.

References

1. Abel, F., Gao, Q., Houben, G.J., Tao, K.: Analyzing User Modeling on Twitter for Personalized News Recommendations. In: J.A. Konstan, R. Conejo, J. Marzo, N. Oliver (eds.) Proc. of the 19th International Conference on User Modeling, Adaption and Personalization, *Lecture Notes in Computer Science*, vol. 6787, pp. 1–12. Springer (2011)
2. Abel, F., Gao, Q., Houben, G.J., Tao, K.: Semantic Enrichment of Twitter Posts for User Profile Construction on the Social Web. In: G. Antoniou, M. Grobelnik, E.P.B. Simperl, B. Parsia, D. Plexousakis, P.D. Leenheer, J.Z. Pan (eds.) Proc. of the 8th Extended Semantic Web Conference, ESWC 2011, Part II, *Lecture Notes in Computer Science*, vol. 6644, pp. 375–389. Springer (2011)
3. Achlioptas, D.: Database-friendly random projections. In: P. Buneman (ed.) PODS. ACM (2001)
4. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. ACM Trans. Inf. Syst. **23**(1), 103–145 (2005). DOI 10.1145/1055709.1055714. URL http://doi.acm.org/10.1145/1055709.1055714
5. Adomavicius, G., Tuzhilin, A.: Context-Aware Recommender Systems. In: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (eds.) Recommender Systems Handbook, pp. 217–253. Springer (2011)
6. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley (1999)
7. Bambini, R., Cremonesi, P., Turrin, R.: A Recommender System for an IPTV Service Provider: a Real Large-Scale Production Environment. In: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (eds.) Recommender Systems Handbook, pp. 299–331. Springer (2011)
8. Banerjee, S., Ramanathan, K., Gupta, A.: Clustering Short Texts Using Wikipedia. In: Proc. of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '07, pp. 787–788. ACM, New York, NY, USA (2007). DOI 10.1145/1277741.1277909. URL http://doi.acm.org/10.1145/1277741.1277909
9. Bentivogli, L., Pianta, E., Girardi, C.: MultiWordNet: Developing an Aligned Multilingual Database. In: First International Conference on Global WordNet, Mysore, India (2002)
10. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American **284**(5), 28–37 (2001)
11. Berry, M.W.: Large-scale Sparse Singular Value Computations. International Journal of Supercomputer Applications **6**(1), 13–49 (1992)
12. Billsus, D., Pazzani, M.: Learning Probabilistic User Models. In: Proc. of the Workshop on Machine Learning for User Modeling. Chia Laguna, IT (1997). URL citeseer.nj.nec.com/billsus96learning.html
13. Bizer, C.: The Emerging Web of Linked Data. IEEE Intelligent Systems **24**(5), 87–92 (2009)
14. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story So Far. Int. J. Semantic Web Inf. Syst. **5**(3), 1–22 (2009)
15. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - A crystallization point for the Web of Data. Web Semant. **7**(3), 154–165 (2009). DOI 10.1016/j.websem.2009.07.002. URL http://dx.doi.org/10.1016/j.websem.2009.07.002
16. Blanco-Fernandez, Y., Pazos-Arias, J.J., Gil-Solla, 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)
17. Bollacker, K.D., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a Collaboratively Created Graph Database for Structuring Human Knowledge. In: J.T.L. Wang (ed.) Proc. of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, pp. 1247–1250. ACM (2008)
18. Buchanan, B.G., Feigenbaum, E.: Forward. In: R. Davis, D. Lenat (eds.) Knowledge-Based Systems in Artificial Intelligence. McGraw-Hill (1982)

19. Budanitsky, A., Hirst, G.: Evaluating WordNet-based Measures of Lexical Semantic Relatedness. Computational Linguistics **32**(1), 13–47 (2006)
20. Cambria, E., Schuller, B., Liu, B., Wang, H., Havasi, C.: Knowledge-Based Approaches to Concept-Level Sentiment Analysis. IEEE Intelligent Systems **28**(2), 12–14 (2013)
21. Cantador, I., Bellogín, A., Castells, P.: A Multilayer Ontology-based Hybrid Recommendation Model. AI Communications **21**(2), 203–210 (2008)
22. 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)
23. Cantador, I., Bellogín, A., Castells, P.: Ontology-based Personalised and Context-aware Recommendations of News Items. In: Proc. of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01, pp. 562–565. IEEE Computer Society (2008)
24. Cantador, I., Szomszor, M., Alani, H., Fernández, M., Castells, P.: Enriching Ontological User Profiles with Tagging History for Multi-domain Recommendations. In: Proc. of the 1st International Workshop on Collective Semantics: Collective Intelligence & the Semantic Web (2008)
25. Capelle, M., Hogenboom, F., Hogenboom, A., Frasincar, F.: Semantic News Recommendation Using Wordnet and Bing Similarities. In: S.Y. Shin, J.C. Maldonado (eds.) Proc. of the 28th Annual ACM Symposium on Applied Computing, SAC '13, pp. 296–302. ACM (2013)
26. Cena, F., Likavec, S., Osborne, F.: Anisotropic Propagation of User Interests in Ontology-based User Models. Inf. Sci. **250**, 40–60 (2013)
27. Chen, X., Li, L., Xu, G., Yang, Z., Kitsuregawa, M.: Recommending Related Microblogs: A Comparison Between Topic and WordNet based Approaches. In: J. Hoffmann, B. Selman (eds.) Proc. of the Twenty-Sixth AAAI Conference on Artificial Intelligence. AAAI Press (2012)
28. Chui, M., Manyika, J., Kuiken, S.V.: What executives should know about open data. McKinsey Quarterly, January 2014 (2014)
29. Ciesielczyk, M., Szwabe, A., Prus-Zajaczkowski, B.: Interactive Collaborative Filtering with RI-based Approximation of SVD. In: Proc. of the 3rd International Conference on Computational Intelligence and Industrial Application (PACIIA), pp. 243–246. IEEE Press (2010)
30. Codina, V., Ricci, F., Ceccaroni, L.: Exploiting the Semantic Similarity of Contextual Situations for Pre-filtering Recommendation. In: S. Carberry, S. Weibelzahl, A. Micarelli, G. Semeraro (eds.) Proc. of the 21st International Conference on User Modeling, Adaptation, and Personalization, UMAP 2013, *Lecture Notes in Computer Science*, vol. 7899, pp. 165–177. Springer (2013)
31. Codina, V., Ricci, F., Ceccaroni, L.: Local Context Modeling with Semantic Pre-filtering. In: Q. Yang, I. King, Q. Li, P. Pu, G. Karypis (eds.) Seventh ACM Conference on Recommender Systems, RecSys '13, pp. 363–366. ACM (2013)
32. Cremonesi, P., Turrin, R., Airoldi, F.: Hybrid Algorithms for Recommending New Items. In: Proc. of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems, pp. 33–40. ACM (2011)
33. Csomai, A., Mihalcea, R.: Linking Documents to Encyclopedic Knowledge. IEEE Intelligent Systems **23**(5), 34–41 (2008). DOI 10.1109/MIS.2008.86. URL http://dx.doi.org/10.1109/MIS.2008.86
34. de Gemmis, M., Di Noia, T., Lops, P., T.Lukasiewicz, Semeraro, G. (eds.): Proc. of the International Workshop on Semantic Technologies meet Recommender Systems & Big Data, Boston, USA, November 11, 2012, *CEUR Workshop Proceedings*, vol. 919. CEUR-WS.org (2012)
35. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science **41**(6), 391–407 (1990)

36. 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.
37. Degemmis, M., Lops, P., Semeraro, G., Basile, P.: Integrating Tags in a Semantic Content-based Recommender. In: P. Pu, D.G. Bridge, B. Mobasher, F. Ricci (eds.) Proc. of the 2008 ACM Conference on Recommender Systems, RecSys 2008, pp. 163–170. ACM (2008)
38. Di Noia, T., Mirizzi, R., Ostuni, V.C., Romito, D.: Exploiting the Web of Data in Model-based Recommender Systems. In: P. Cunningham, N.J. Hurley, I. Guy, S.S. Anand (eds.) Proc. of the Sixth ACM Conference on Recommender Systems, RecSys '12, pp. 253–256. ACM (2012)
39. Di Noia, T., Mirizzi, R., Ostuni, V.C., Romito, D., Zanker, M.: Linked Open Data to Support Content-based Recommender Systems. In: V. Presutti, H.S. Pinto (eds.) I-SEMANTICS 2012 - 8th International Conference on Semantic Systems, pp. 1–8. ACM (2012)
40. 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)
41. Dunlop, M.D.: The Effect of Accessing Nonmatching Documents on Relevance Feedback. ACM Trans. Inf. Syst. **15**, 137–153 (1997)
42. Egozi, O., Gabrilovich, E., Markovitch, S.: Concept-based Feature Generation and Selection for Information Retrieval. In: Proc. of the 23rd National Conference on Artificial Intelligence - Volume 2, AAAI'08, pp. 1132–1137. AAAI Press (2008). URL http://dl.acm.org/citation.cfm?id=1620163.1620248
43. Egozi, O., Markovitch, S., Gabrilovich, E.: Concept-Based Information Retrieval using Explicit Semantic Analysis. ACM Transactions on Information Systems **29**(2), 8:1–8:34 (2011).
44. Fernández-Tobías, I., Kaminskas, M., Cantador, I., Ricci, F.: A Generic Semantic-based Framework for Cross-domain Recommendation. In: I. Cantador, P. Brusilovsky, T. Kuflik (eds.) HetRec '11 Proc. of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems, pp. 25–32. ACM New York (2011)
45. Fernando, S., Hall, M., Agirre, E., Soroa, A., Clough, P., Stevenson, M.: Comparing Taxonomies for Organising Collections of Documents. In: M. Kay, C. Boitet (eds.) Proc. of the 24th International Conference on Computational Linguistics, COLING 2012, pp. 879–894. Indian Institute of Technology Bombay (2012). URL http://www.aclweb.org/anthology/C12-1054
46. Ferragina, P., Scaiella, U.: Fast and Accurate Annotation of Short Texts with Wikipedia Pages. IEEE Software **29**(1), 70–75 (2012)
47. Foltz, P.W., Dumais, S.T.: Personalized Information Delivery: an Analysis of Information Filtering Methods. Communications of the ACM **35**(12), 51–60 (1992)
48. Gabrilovich, E., Markovitch, S.: Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In: M.M. Veloso (ed.) Proc. of the 20th International Joint Conference on Artificial Intelligence, pp. 1606–1611 (2007)
49. Gabrilovich, E., Markovitch, S.: Wikipedia-based Semantic Interpretation for Natural Language Processing. Journal of Artificial Intelligence Research (JAIR) **34**, 443–498 (2009)
50. Giles, J.: Internet Encyclopaedias Go Head to Head. Nature **438**(7070), 900–901 (2005). URL http://dx.doi.org/10.1038/438900a
51. 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
52. Halevy, A.Y., Norvig, P., Pereira, F.: The Unreasonable Effectiveness of Data. IEEE Intelligent Systems **24**(2), 8–12 (2009)
53. Harris, Z.S.: Mathematical Structures of Language. Interscience, New York, (1968)
54. Hassanzadeh, O., Consens, M.P.: Linked Movie Data Base. In: C.Bizer, T. Heath, T. Berners-Lee, K. Idehen (eds.) Proc. of the WWW2009 Workshop on Linked Data on the Web, LDOW 2009, *CEUR Workshop Proceedings*, vol. 538. CEUR-WS.org (2009)

55. Hecht-Nielsen, R.: Context Vectors: General Purpose Approximate Meaning Representations Self-organized from Raw Data. Computational Intelligence: Imitating Life, IEEE Press pp. 43–56 (1994)
56. Heitmann, B., Hayes, C.: Using Linked Data to Build Open, Collaborative Recommender Systems. In: AAAI Spring Symposium: Linked Data Meets Artificial Intelligence, pp. 76–81. AAAI (2010)
57. 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)
58. 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)
59. Hu, J., Fang, L., Cao, Y., Zeng, H., Li, H., Yang, Q., Chen, Z.: Enhancing Text Clustering by Leveraging Wikipedia Semantics. In: S. Myaeng, D.W. Oard, F. Sebastiani, T. Chua, M. Leong (eds.) Proc. of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '08, pp. 179–186. ACM (2008)
60. Hu, R., Pu, P.: A study on user perception of personality-based recommender systems. In: User Modeling, Adaptation, and Personalization, 18th International Conference, UMAP 2010, Big Island, HI, USA, June 20–24, 2010. Proceedings, pp. 291–302 (2010).
61. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: Recommender systems: An introduction. Cambridge University Press (2010)
62. Johnson, W., Lindenstrauss, J.: Extensions of Lipschitz Maps into a Hilbert Space. Contemporary Mathematics (1984)
63. Jones, D., Bench-Capon, T., Visser, P.: Methodologies for Ontology Development (1998)
64. Kaminskas, M., Fernández-Tobías, I., Ricci, F., Cantador, I.: Knowledge-based Music Retrieval for Places of Interest. In: C.C.S. Liem, M. Müller, S.K. Tjoa, G. Tzanetakis (eds.) Proc. of the 2nd International ACM workshop on Music information retrieval with user-centered and multimodal strategies, MIRUM '12, pp. 19–24 (2012)
65. Kanerva, P.: Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors. Cognitive Computation **1**(2), 139–159 (2009)
66. Kaski, S.: Dimensionality Reduction by Random Mapping: Fast Similarity Computation for Clustering. In: Proc. of the International Joint Conference on Neural Networks, vol. 1, pp. 413–418. IEEE (1998)
67. Khrouf, H., Troncy, R.: Hybrid Event Recommendation using Linked Data and User Diversity. In: Q. Yang, I. King, Q. Li, P. Pu, G. Karypis (eds.) Seventh ACM Conference on Recommender Systems, RecSys '13, pp. 185–192. ACM (2013)
68. Klema, V., Laub, A.: The Singular Value Decomposition: its Computation and Some Applications. IEEE Transactions on Automatic Control **25**(2), 164–176 (1980)
69. Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Honkela, J., Paatero, V., Saarela, A.: Self Organization of a Massive Document Collection. IEEE Transactions on Neural Networks **11**(3), 574–585 (2000)
70. Lewis, D.D., Ringuette, M.: A Comparison of Two Learning Algorithms for Text Categorization. In: Proc. of the Annual Symposium on Document Analysis and Information Retrieval, pp. 81–93. Las Vegas, US (1994)
71. Lops, P., Musto, C., Narducci, F., de Gemmis, M., Basile, P., Semeraro, G.: MARS: a MultilAnguage Recommender System. In: P. Brusilovsky, I. Cantador, Y. Koren, T. Kuflik, M. Weimer (eds.) HetRec '10 Proc. of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems, pp. 24–31. ACM New York (2010)
72. Lowe, W.: Towards a Theory of Semantic Space. In: Proc. of the Twenty-Third Annual Conference of the Cognitive Science Society, pp. 576–581. Lawrence Erlbaum Associates (2001)
73. Magen, A.: Dimensionality Reductions that Preserve Volumes and Distance to Affine Spaces, and their Algorithmic Applications. In: Randomization and approximation techniques in computer science, pp. 239–253. Springer (2002)

74. 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)
75. Magnini, B., Strapparava, C.: Improving User Modelling with Content-based Techniques. In: M. Bauer, P.J. Gmytrasiewicz, J. Vassileva (eds.) Proc. of the 8th International Conference of User Modeling, *Lecture Notes in Computer Science*, vol. 2109, pp. 74–83. Springer (2001)
76. Mairesse, F., Walker, M.A., Mehl, M.R., Moore, R.K.: Using linguistic cues for the automatic recognition of personality in conversation and text. J. Artif. Intell. Res. (JAIR) **30**, 457–500 (2007). DOI 10.1613/jair.2349. URL http://dx.doi.org/10.1613/jair.2349
77. McCallum, A., Nigam, K.: A Comparison of Event Models for Naïve Bayes Text Classification. In: Proc. of the AAAI/ICML-98 Workshop on Learning for Text Categorization, pp. 41–48. AAAI Press (1998)
78. McCarey, F., Cinnéide, M., Kushmerick, N.: Recommending Library Methods: An Evaluation of the Vector Space Model (VSM) and Latent Semantic Indexing (LSI). In: M. Morisio (ed.) Proc. of the 9th International Conference on Software Reuse, ICSR 2006, *Lecture Notes in Computer Science*, vol. 4039, pp. 217–230. Springer (2006)
79. McMillan, C., Poshyvanyk, D., Grechanik, M.: Recommending Source Code Examples via API Call Usages and Documentation. In: Proc. of the 2nd International Workshop on Recommendation Systems for Software Engineering, pp. 21–25. ACM (2010)
80. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: DBpedia Spotlight: Shedding Light on the Web of Documents. In: C. Ghidini, A.N. Ngomo, S.N. Lindstaedt, T. Pellegrini (eds.) Proceedings the 7th International Conference on Semantic Systems, I-SEMANTICS 2011, pp. 1–8. ACM (2011)
81. Middleton, S.E., De Roure, D., Shadbolt, N.R.: Ontology-based Recommender Systems. In: S. Staab, R. Studer (eds.) Handbook on ontologies, pp. 477–498. Springer (2004)
82. 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)
83. Mikolov, T., Le, Q.V., Sutskever, I.: Exploiting Similarities among Languages for Machine Translation. CoRR **abs/1309.4168** (2013)
84. Miller, G.: WordNet: An On-Line Lexical Database. International Journal of Lexicography **3**(4) (1990). (Special Issue)
85. Milne, D., Witten, I.H.: Learning to Link with Wikipedia. In: J.G. Shanahan, S. Amer-Yahia, I. Manolescu, Y. Zhang, D.A. Evans, A. Kolcz, K. Choi, A. Chowdhury (eds.) Proc. of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, pp. 509–518. ACM (2008)
86. Mitchell, T.: Machine Learning. McGraw-Hill, New York (1997)
87. Mohammad, S., Hirst, G.: Distributional Measures of Semantic Distance: A Survey. CoRR **abs/1203.1858** (2012)
88. Moro, A., Raganato, A., Navigli, R.: Entity Linking meets Word Sense Disambiguation: a Unified Approach. Transactions of the Association for Computational Linguistics **2**, 231–244 (2014)
89. Musto, C., Basile, P., Lops, P., de Gemmis, M., Semeraro, G.: Linked Open Data-enabled Strategies for Top-N Recommendations. In: T. Bogers, M. Koolen, I. Cantador (eds.) Proc. of the 1st Workshop on New Trends in Content-based Recommender Systems, ACM RecSys 2014, *CEUR Workshop Proceedings*, vol. 1245, pp. 49–56. CEUR-WS.org (2014)
90. Musto, C., Narducci, F., Basile, P., Lops, P., de Gemmis, M., Semeraro, G.: Cross-Language Information Filtering: Word Sense Disambiguation vs. Distributional Models. In: R. Pirrone, F. Sorbello (eds.) AI*IA, *Lecture Notes in Computer Science*, vol. 6934, pp. 250–261. Springer (2011)
91. Musto, C., Narducci, F., Lops, P., Semeraro, G., de Gemmis, M., Barbieri, M., Korst, J.H.M., Pronk, V., Clout, R.: Enhanced Semantic TV-Show Representation for Personalized Electronic Program Guides. In: Judith.Masthoff, B. Mobasher, M.C. Desmarais, R. Nkambou (eds.) Proc. of the 20th International Conference on User Modeling, Adaptation, and Personalization, UMAP 2012, *Lecture Notes in Computer Science*, vol. 7379, pp. 188–199. Springer (2012)

92. Musto, C., Semeraro, G., Lops, P., de Gemmis, M.: Random Indexing and Negative User Preferences for Enhancing Content-Based Recommender Systems. In: C. Huemer, T. Setzer (eds.) Proc. of the 12th International Conference on Electronic Commerce and Web Technologies, EC-Web 2011, *Lecture Notes in Business Information Processing*, vol. 85, pp. 270–281. Springer (2011)
93. Musto, C., Semeraro, G., Lops, P., de Gemmis, M.: Contextual eVSM: a Content-Based Context-Aware Recommendation Framework Based on Distributional Semantics. In: C. Huemer, P. Lops (eds.) Proc. of the 14th International Conference on E-Commerce and Web Technologies, EC-Web 2013, *Lecture Notes in Business Information Processing*, vol. 152, pp. 125–136. Springer (2013)
94. Musto, C., Semeraro, G., Lops, P., de Gemmis, M.: Combining Distributional Semantics and Entity Linking for Context-Aware Content-Based Recommendation. In: Proc. of the 22nd International Conference on User Modeling, Adaptation, and Personalization, UMAP 2014, *Lecture Notes in Computer Science*, vol. 8538, pp. 381–392. Springer (2014)
95. Musto, C., Semeraro, G., Lops, P., de Gemmis, M., Narducci, F.: Leveraging Social Media Sources to Generate Personalized Music Playlists. In: C. Huemer, P. Lops (eds.) Proc. of the 13th International Conference on E-Commerce and Web Technologies, EC-Web 2012, *Lecture Notes in Business Information Processing*, vol. 123, pp. 112–123. Springer (2012)
96. Narducci, F., Musto, C., Semeraro, G., Lops, P., de Gemmis, M.: Exploiting Big Data for Enhanced Representations in Content-Based Recommender Systems. In: C. Huemer, P. Lops (eds.) Proc. of the 14th International Conference on E-Commerce and Web Technologies, EC-Web 2013, *Lecture Notes in Business Information Processing*, vol. 152, pp. 182–193. Springer (2013)
97. Narducci, F., Musto, C., Semeraro, G., Lops, P., de Gemmis, M.: Leveraging Encyclopedic Knowledge for Transparent and Serendipitous User Profiles. In: S. Carberry, S. Weibelzahl, A. Micarelli, G. Semeraro (eds.) Proc. of the 21st International Conference on User Modeling, Adaptation, and Personalization, UMAP 2013, *Lecture Notes in Computer Science*, vol. 7899, pp. 350–352. Springer (2013)
98. Narducci, F., Palmonari, M., Semeraro, G.: Cross-Language Semantic Retrieval and Linking of e-Gov Services. In: H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. Parreira, L. Aroyo, N. Noy, C. Welty, K. Janowicz (eds.) The Semantic Web - ISWC 2013, *Lecture Notes in Computer Science*, vol. 8219, pp. 130–145. Springer Berlin Heidelberg (2013).
99. Navigli, R., Jurgens, D., Vannella, D.: SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. In: Proc. of the 7th International Workshop on Semantic Evaluation (SemEval 2013), in conjunction with the 2nd Joint Conference on Lexical and Computational Semantics (*SEM 2013), pp. 222–231. Atlanta, USA (2013)
100. Navigli, R., Ponzetto, S.P.: BabelNet: The automatic Construction, Evaluation and Application of a Wide-coverage Multilingual Semantic Network. Artif. Intell. **193**, 217–250 (2012)
101. Navigli, R., Ponzetto, S.P.: BabelRelate! A Joint Multilingual Approach to Computing Semantic Relatedness. In: J. Hoffmann, B. Selman (eds.) Proc. of the Twenty-Sixth AAAI Conference on Artificial Intelligence, AAAI-12. AAAI Press (2012)
102. Navigli, R., Ponzetto, S.P.: Joining Forces Pays Off: Multilingual Joint Word Sense Disambiguation. In: Proc. of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 1399–1410. Jeju, Korea (2012)
103. Ostuni, V.C., Di, T., Sciascio, E.D., Mirizzi, R.: Top-N Recommendations from Implicit Feedback Leveraging Linked Open Data. In: Q. Yang, I. King, Q. Li, P. Pu, G. Karypis (eds.) Seventh ACM Conference on Recommender Systems, RecSys ’13, pp. 85–92. ACM (2013)
104. Ostuni, V.C., Di Noia, T., Mirizzi, R., Romito, D., Sciascio, E.D.: Cinemappy: a Context-aware Mobile App for Movie Recommendations boosted by DBpedia. In: M. de Gemmis, T. Di Noia, P. Lops, T. Lukasiewicz, G. Semeraro (eds.) Proc. of the International Workshop on Semantic Technologies meet Recommender Systems & Big Data, ACM RecSys, *CEUR Workshop Proceedings*, vol. 919, pp. 37–48. CEUR-WS.org (2012)

105. Pappas, N., Popescu-Belis, A.: Combining Content with User Preferences for Non-Fiction Multimedia Recommendation: A Study on TED Lectures. Multimedia Tools and Applications (2014)
106. Passant, A.: dbrec - Music Recommendations Using DBpedia. In: P.F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J.Z. Pan, I. Horrocks, B. Glimm (eds.) The Semantic Web - ISWC 2010 - 9th International Semantic Web Conference, Revised Selected Papers, Part II, *Lecture Notes in Computer Science*, vol. 6497, pp. 209–224. Springer (2010)
107. Passant, A.: Measuring Semantic Distance on Linking Data and Using it for Resources Recommendations. In: AAAI Spring Symposium: Linked Data Meets Artificial Intelligence, pp. 93–98. AAAI (2010)
108. 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
109. Potthast, M., Stein, B., Anderka, M.: A Wikipedia-based Multilingual Retrieval Model. In: C. Macdonald, I. Ounis, V. Plachouras, I. Ruthven, R.W. White (eds.) Proc. of the 30th European conference on Advances in information retrieval, ECIR 2008, *Lecture Notes in Computer Science*, vol. 4956, pp. 522–530. Springer (2008)
110. Raimond, Y., Sandler, M.B.: A Web of Musical Information. In: J.P. Bello, E. Chew, D. Turnbull (eds.) International Conference on Music Information Retrieval, pp. 263–268 (2008)
111. Rich, E.: User Modeling via Stereotypes. Cognitive Science **3**, 329–354 (1979)
112. Rizzo, G., Troncy, R.: NERD: a Framework for Unifying Named Entity Recognition and Disambiguation Extraction Tools. In: W. Daelemans, M. Lapata, L. Màrquez (eds.) Proc. of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 73–76. Association for Computational Linguistics (2012)
113. 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)
114. Rubenstein, H., Goodenough, J.B.: Contextual Correlates of Synonymy. Commun. ACM **8**(10), 627–633 (1965).
115. Sahlgren, M.: An Introduction to Random Indexing. In: Proc. of the Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE (2005)
116. 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 University (2006)
117. Salton, G.: Automatic Text Processing. Addison-Wesley (1989)
118. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)
119. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys **34**(1) (2002)
120. Semeraro, G., Lops, P., Basile, P., Gemmis, M.d.: Knowledge Infusion into Content-based Recommender Systems. In: L.D. Bergman, A. Tuzhilin, R.D. Burke, A. Felfernig, L. Schmidt-Thieme (eds.) Proc. of the 2009 ACM Conference on Recommender Systems, RecSys 2009, New York, NY, USA, October 23–25, 2009, pp. 301–304. ACM (2009)
121. Shoval, P., Maidel, V., Shapira, B.: An Ontology-Content-based Filtering Method. International Journal of Information Theories and Applications **15**, 303–314 (2008)
122. Symeonidis, P.: Content-based Dimensionality Reduction for Recommender Systems. In: Data Analysis, Machine Learning and Applications, pp. 619–626. Springer (2008)
123. Szwabe, A., Ciesielczyk, M., Janasiewicz, T.: Semantically Enhanced Collaborative Filtering Based on RSVD. In: P. Jedrzejowicz, N.T. Nguyen, K. Hoang (eds.) Proc. of Computational Collective Intelligence. Technologies and Applications - Third International Conference, ICCCI 2011, Part II, *Lecture Notes in Computer Science*, vol. 6923, pp. 10–19. Springer (2011)

124. Terzi, M., Ferrario, M., Whittle, J.: Free Text In User Reviews: Their Role In Recommender Systems. In: Proc. of the Workshop on Recommender Systems and the Social Web, 3rd ACM Conf. on Recommender Systems, pp. 45–48 (2011)
125. Turney, P.D., Pantel, P.: From Frequency to Meaning: Vector Space Models of Semantics. J. Artif. Intell. Res. (JAIR) **37**, 141–188 (2010)
126. Vempala, S.S.: The Random Projection Method, vol. 65. American Mathematical Society (2004)
127. Widdows, D.: Orthogonal Negation in Vector Spaces for Modelling Word-Meanings and Document Retrieval. In: E.W. Hinrichs, D. Roth (eds.) Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, pp. 136–143 (2003)
128. Widdows, D., Cohen, T.: The Semantic Vectors Package: New Algorithms and Public Tools for Distributional Semantics. In: Proc. of the 4th IEEE International Conference on Semantic Computing, ICSC 2010, pp. 9–15. IEEE (2010)
129. 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)
130. Wu, Z., Palmer, M.S.: Verb Semantics and Lexical Selection. In: J. Pustejovsky (ed.) 32nd Annual Meeting of the Association for Computational Linguistics, pp. 133–138. Morgan Kaufmann Publishers / ACL (1994)