A Semantic Approach for News Recommendation

Flavius Frasincar
Wouter IJntema
Frank Goossen
Frederik Hogenboom
Erasmus University Rotterdam, the Netherlands

ABSTRACT

News items play an increasingly important role in the current business decision processes. Due to the large amount of news published every day it is difficult to find the new items of one's interest. One solution to this problem is based on employing recommender systems. Traditionally, these recommenders use term extraction methods like TF-IDF combined with the cosine similarity measure. In this chapter, we explore semantic approaches for recommending news items by employing several semantic similarity measures. We have used existing semantic similarities as well as proposed new solutions for computing semantic similarities. Both traditional and semantic recommender approaches, some new, have been implemented in Athena, an extension of the Hermes news personalization framework. Based on the performed evaluation, we conclude that semantic recommender systems in general outperform traditional recommenders systems with respect to accuracy, precision, and recall, and that the new semantic recommenders have a better F-measure than existing semantic recommenders.

INTRODUCTION

Finding the news items of interest is a critical task in many business processes. One such process is business intelligence which aims to gather, analyse, and use company-related data in order to support decision making (Luhn, 1958). While a lot of this information is represented by company internal data (e.g., product sales, costs, incomes, etc.), in the recent years, we observed a growing focus of attention for company external data whose processing is aimed to answer questions as how is the company perceived by the public? (business marketing), how are competitors reported in the media? (competitive intelligence), what are possible collaborators in other countries? (business internationalization), etc., (Saggion, Funk, Maynard, & Bontcheva, 2007) (Pang & Lee, 2008) (Castellanos, Gupta, Wang, & Dayal, 2010). News items, as rich sources of external company-related information, are increasingly exploited in business intelligence tasks.

The Web is one of the most popular platforms for distributing and consuming news items. There are several factors that contributed to this success story as for example the reduced cost for distributing and accessing news items, Web availability on a multitude of browsing platforms, world-wide information delivery and consumption, short amount of time required for news publication, etc. Unfortunately, the Web's success is also the cause of one of its most serious liabilities: the large number of daily published news items makes the process of finding the ones relevant to particular interests difficult. For business intelligence, companies are only interested in news items deemed relevant for their analytical processes, which for competitive reasons should be made available with minimal delay times.

One possible solution to deal with the news items overload problem is the use of recommender systems, which aim to propose previously unseen items, in our case news items, that are of interest to a certain user. Typically such recommenders employ a user profile and aim to recommend news items that best match this user profile. Currently, there are four types of recommender systems: content-based, collaborative filtering, semantics-based, and hybrid (Adomavicius & Tuzhilin, 2005). While the user profile is usually represented by the user's previously browsed items, the recommendation methods differ per employed recommendation method. The content-based recommenders propose items based on the lexical content of the previously viewed items, semantic recommenders use the semantic information of the earlier browsed items, collaborative filtering recommenders exploit profile similarities between different users, and hybrid recommenders are combinations of the previous recommenders.

In this chapter we focus on recommenders that use the information content in news items, be it lexical (as in content-based approaches) or semantic (as in semantics-based approaches). While content-based recommenders have previously been thoroughly investigated, it is only in the last years that researchers started to focus on semantics-based approaches for recommender systems. Also, a comprehensive study that compares the content-based recommenders with semantics-based recommenders is currently missing. Therefore one of the aims of this chapter is to produce such an investigation in the context of recommending news items. In addition, we would like to investigate multiple semantics-based approaches and compare their performances. The collaborative filtering and hybrid recommenders are considered outside the scope of this chapter.

In previous work (IJntema, Goossen, Frasincar, & Hogenboom, 2010) we have proposed a semantic recommender for news called Ranked-based Semantic Recommender (RSR). In this chapter we extend our previous work by considering not only the concepts directly related to the concepts from the user profile but also the concepts directly related to the concepts present in unread news items, which can help recommend more relevant news items than before. Our research is circumscribed to Hermes, a framework for news personalization that we have developed during the last five years (Borsje, Levering, & Frasincar, 2008; Frasincar, Borsje, & Levering, 2009; Schouten et al., 2010). For this purpose we have developed Athena, which extends Hermes with news recommendation functionality.

The chapter is organized as follows. In the first section we discuss the background on recommendation methods, including content-based recommenders and semantics-based recommender systems, with a special attention being given to news recommenders. In the next section we present a new semantic recommender for news items. We describe the evaluation we performed using the implementation of the proposed recommender as well as existing content-based and semantics-based recommenders in the following section. The last two sections discuss future work and present our conclusions.

BACKGROUND

Recommendation helps users to focus on what is interesting by selecting new content based on previously read news articles, Web pages, research papers or other kind of documents. In this chapter we focus on recommendation of news items. First we discuss the user profile that is used to collect information about the interests of the user. Secondly a detailed description of content based-recommendation and semantics-based recommendation is given. The third and fourth part of this section discuss Hermes, a framework for building personalized news services, and Athena, an extension to Hermes which provides a news recommendation system employing both content- and semantics-based recommendation methods.

User Profile

In order to recommend news items to the user, a user profile needs to be constructed. The user's interests can be determined based on the news items which have been read. How the user profile is represented depends on the recommendation approach employed. For the content-based recommendation method the user profile consists of terms with corresponding frequencies. Semantics-based recommendation methods rely on the concepts that appear in the news items. For concept equivalence, binary cosine, and Jaccard, the user profile consists of all concepts that appear in the news items that have been read. The semantic relatedness approach uses a vector with distinct concepts and assigns a weight to each concept. In a similar way the rank-based semantic recommender assigns a rank to each concept, which is also stored in a vector.

Content-Based Recommendation

Term Frequency-Inverse Document Frequency (TF-IDF) (Salton & Buckley, 1988) is a well-known term weighting method which is often used in information retrieval. It is employed to determine the importance of a word within a document relative to the frequency of the word within a collection of documents (or corpus). TF-IDF is often used in conjunction with the cosine similarity measure in order to compare the similarity between two documents.

Many content-based recommenders make use of TF-IDF and the cosine similarity measure for news personalization. Before the TF-IDF values are calculated, first the stop words are removed, followed by stemming the remaining words. The latter means, determining the root of each word, such as ‘recommending’, ‘recommender’, and ‘recommended’ all become ‘recommend’, with the advantage that the TF-IDF values are not calculated for each individual morphological form.

The TF-IDF value of a word can be calculated as follows. First we determine the term frequency (TF) $f_{i,j}$ for a term t_i within a news article a_j:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

where $n_{i,j}$ is the number of occurrences of term t_i in news article a_j and the denominator is the total number of terms in the document. The second step is the calculation of the inverse document frequency (IDF), which is the relative importance of a term in a set of news items. This is computed as follows:

$$idf_i = \log \frac{|A|}{|\{a : t_i \in a\}|} \tag{2}$$

where the numerator is the total number of news items and the denominator denotes the number of news items containing term t_i. The final value is computed by taking the product of the term frequency and inverse document frequency:

$$tfidf_{i,j} = tf_i \times idf_i \tag{3}$$

In order to obtain the user profile, one has to calculate the TF-IDF values for each term in the news items that user has read. The user profile consists of a relatively large number of words (e.g., 100 words, as we have used later in our experiments) with the highest TF-IDF value. Subsequently an unread news item, which can be represented by vector N can be compared to the user profile P by computing the cosine similarity between these vectors N and P:

$$similarity = \frac{P \cdot N}{\|P\| \times \|N\|} \tag{4}$$

where the numerator is the dot product of vectors and the denominator is the product of the magnitude of vectors. The news items with the highest similarity are considered to match best with the user profile.

Implementations

Many existing systems employ content-based methods in order to recommend content to the user. They differ in aspects like article representation, user profile representation, and similarity measure. We discuss several existing methods and the similarities and differences with our implementation.

YourNews (Ahn, Brusilovsky, Grady, He, & Syn, 2007) is a personalized news system. It employs TF-IDF for representing news items and the user profile, and cosine similarity measure to compute the degree of similarity between news items and the user profile. Differently than other traditional approaches it aims to increase the transparency of recommended news items by allowing the user to inspect and modify the user profile. Unfortunately this added functionality seems to harm the system, as users observe a lower system performance when making use of this functionality.

NewsDude (Billsus & Pazzani, 1999) is a personalized news recommender agent. It uses a two step approach for making recommendations, first it employs the user's short term interests to find relevant news items and if this returns an empty result it filters news items based on the user's long term interests. For short term model construction NewsDude uses TF-IDF in combination with Nearest Neighbour (NN), which is able to represent user's multiple interests and takes in consideration the changing user's short term interests (concept drift problem). Long term interests or user's general interests are modeled by means of the Naïve Bayes classifier.

Personalized Recommender System (PRES) (van Meteren & van Someren, 2000) is another example of a news personalization system that uses TF-IDF and the cosine similarity measure. A specific aspect of this system is that each time a news items is added to the profile, the weights of the terms previously stored in the profile are diminished by a certain factor. This diminishing factor aims to decrease the importance of terms originating from news items read before the current news item in order to allow for possible changes of user interests over time. The optimal diminishing factor is determined by experimentation.

TF-IDF favors long documents in the cosine similarity computations over short ones. While it is true that long documents have in general more information and possibly select many relevant documents, TF-IDF reduces the chance of relevant short documents to be selected (Singhal, Salton, Mitra, & Buckley, 1996). Also, the vector space model is prone to produce many false negatives, as it does not take into account the term semantics, failing for example to consider synonyms of the query terms for occurrence in documents.

Semantics-Based Recommendation

In traditional content-based recommendation the degree of interestingness of a news item is determined by considering all terms in a document. In semantics-based recommendation only the most important words, called concepts, are considered. Furthermore, semantics are added by providing an underlying knowledge base, which contains relations between these concepts. The availability of concepts and relations to the recommendation process makes it possible to introduce news items to the user, which are semantically related to the ones read. For instance a user interested in news about Apple might also be interested in news about Microsoft, because both are of type Company and the knowledge base contains a relation defining a competitor relation between those two companies.

To illustrate how concepts can be used in the recommendation process, we explain three simple methods. The first is based on concept equivalence, followed by binary cosine and then Jaccard. With the semantic relatedness approach and our own ranked semantic recommendation method we show how relations between concepts can be employed in the recommendation process.

Concept Equivalence

The first method we discuss is a simple technique we proposed in (IJntema et al., 2010), where only equivalent concepts are considered. The idea is to recommend only news items which contain concepts appearing in the user profile. Each concept is stored in the underlying ontology. We define the ontology by the following set of concepts (concepts have the ontology properties attached):

$$C = \{c_1, c_2, c_3, \cdots, c_n\} \tag{5}$$

A concept is present in a news item if one of its lexical representations is found in the news item. The news article can be defined by a set of p concepts:

$$A = \{c_1^a, c_2^a, c_3^a, \cdots, c_p^a\} \tag{6}$$

The user profile consists of q concepts found in the news items read by the user and is defined by:

$$U = \{c_1^u, c_2^u, c_3^u, \cdots, c_q^u\} \tag{7}$$

Due to the use of sets it is easy to compute the similarity between the news item and the user profile. In this method it is only relevant whether or not a concept from the user profile exists in the unread news item and if it does, it is considered to be interesting. The similarity between a news article and the user profile can consequently be computed by:

$$Similarity(U, A) = \begin{cases} 1 & \text{if } |U \cap A| > 0 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Binary Cosine

In the previous subsection we have shown how TF-IDF is often used in conjunction with the cosine similarity measure. In a similar fashion we can employ binary cosine to compute the similarity between two sets of concepts:

$$B(U, A) = \frac{|U \cap A|}{|U| \times |A|} \tag{9}$$

where $|U \cap A|$ is the number of concepts in the intersection of the user profile and the unread news article and $|U| \times |A|$ is the product of the number of concepts in respectively U and A. The returned value gives an indication of the interestingness of the article compared to the news items the user has read so far.

Jaccard

Analogous to the binary cosine measure, Jaccard (Jaccard, 1901) computes the similarity between two sets of concepts as follows:

$$J(U, A) = \frac{|U \cap A|}{|U \cup A|} \tag{10}$$

where $|U \cap A|$ is the number of concepts in the intersection of U and A and $|U \cup A|$ represents the number of concepts in the union of U and A. Unlike concept equivalence, binary cosine and Jaccard take into account the number of concepts found in a news item.

Semantic Relatedness

(Getahun, Tekli, Chbeir, Viviani, & Yetongnon, 2009) proposes a method to determine the similarity between two texts which takes into account the semantic neighborhood of a concept. In this approach only linguistic relations, i.e., synonymy, hyponymy, and meronymy, are considered, while in our approach many more types of relations are covered by the ontology. Calculating the similarity between the user profile and a news article based on the semantic neighborhood of concepts is applicable to our approach.

The semantic neighborhood of a concept c_i is defined as all concepts directly related to c_i including c_i and can be denoted as:

$$N(c_i) = \{c_1^i, c_2^i, c_3^i, \cdots, c_n^i\} \tag{11}$$

A news item a_k, which consists of m concepts can be described as the following set:

$$A_k = \{c_1^k, c_2^k, c_3^k, \cdots, c_m^k\} \tag{12}$$

In order to compare two new items n_i and $n_{j,}$ a vector in n-dimensional space can be created according to the vector space model:

$$V_l = \left(\left\langle c_1^l, w_1^l \right\rangle, \cdots, \left\langle c_p^l, w_p^l \right\rangle\right) \tag{13}$$

where $l \in \{i, j\}$ and w_i represents the weight associated to the concept c_i and $p = |A_i \cup A_j|$, which is the number of distinct concepts in A_i and A_j. (the set of concepts in n_i and the set of concepts in n_j, respectively). The weights are calculated as follows:

$$w_i = \begin{cases} 1 & \text{if } freq(c_i \text{ in } A_j) > 0 \\ \max_j (ES(c_i, c_j)) & \text{otherwise} \end{cases} \tag{14}$$

If the concept c_i occurs once or more in A_j the weight assigned is equal to 1, otherwise it is calculated according to the maximum enclosure similarity, which takes into account the semantic neighborhood of a concept:

$$ES(c_i, c_j) = \frac{|N(c_i) \cap N(c_j)|}{|N(c_i)|} \tag{15}$$

Once the weights are computed, the similarity between the news items a_i and a_j is determined by using the following equation:

$$\mathrm{SemRel}(a_i, a_j) = \cos(V_i, V_j) = \frac{V_i \cdot V_j}{\|V_i\| \times \|V_j\|} \in [0,1] \quad (16)$$

where the numerator represents the dot product of the vectors V_i and V_j and the denominator is the product of the magnitude of each vector.

Compared to the previous discussed approaches, this method has the advantage of taking into account the semantics of a text by also considering the related concepts to the concepts appearing in the text. The user profile is defined by the set of concepts appearing in the read news items.

Implementations

Some existing systems use semantics-based methods in order to recommend content to the user. They differ in aspects like article representation, user profile representation and similarity measure. We briefly describe existing methods and the commonalities and differences with our implementation.

PersoNews (Banos, Katakis, Bassiliades, Tsoumakas, & Vlahavas, 2006) is a personalized news reader that is based on semantic filtering and machine learning. First the reader filters news items that contain lexical representations associated to the selected concepts of interest from a taxonomy. Then, it applies the Naïve Bayes classification algorithm in order to determine if an article is interesting. In this approach the user is expected to manually update the concept lexical representations, which is a laborious process especially when the taxonomy size is large.

Quickstep (Middleton, Shadbolt, & De Roure, 2004) is recommender system for academic research papers. Academic papers are classified by means of the boosted IBk classifier (Aha, Kibler, & Albert, 1991) which makes use of k-Nearest Neighbour (k-NN) algorithm applied on a vector space representation of papers. Paper topics are stored in an ontology and are also used for modeling user interests. This ontology is also exploited to solve the user or item cold-start problems, by offering an initial set of user interests based on the topics of authors' previous papers. The recommendation is based on correlations between user's topic of interest and the paper's topics. Quickstep considers only type relationships failing to exploit other ontology relationships (e.g., part-of, domain specific relationships, etc.) that are also rich in semantics.

PVA, a self-adaptive Personal View Agent (Chen & Chen, 2002) introduces a recommender system that tries to recommend Web pages based on personal views. The world view is a predefined category hierarchy. A personal view is a subset of the world view. After Web pages are collected by a proxy, they are classified using the Automatic Classifier for Internet Resource Discovery (ACIRD) (Lin et al., 1998) in order to determine the category they belong to. The constructed user profile is compared with unviewed pages by using the cosine similarity measure. Different from our approach, no Semantic Web techniques are employed. In addition, only hierarchical structures are used, while our recommendation system employs many more types of relationships, such as 'CompetitorOf', 'ProductBy', 'CEOOf', etc.

Unlike previous semantic approaches that are based on machine learning, we use linguistic techniques for text classification. Differently than machine learning approaches linguistic techniques offer easy explanations of why a certain news item is classified to a certain topic or why a news item is recommended to a certain user. Also, linguistic techniques are easily extensible to incorporate information semantics capturing the meaning behind textual descriptions. In this way we aim to improve the classifications and recommendations with respect to both precision and recall compared to traditional recommenders based on TF-IDF. We exploit General Architecture for Text Engineering (GATE) (Cunningham, Maynard, Bontcheva, & Tablan, 2002) in order to classify news and extract concept lexical representations in an automated manner from WordNet (Fellbaum, 1998). Also, we use more ontology relationships than just type relationships better capturing the news semantics.

Hermes

Hermes (Borsje et al., 2008; Frasincar et al., 2009; Schouten et al., 2010) is a framework that can be used for building a personalized news service. Based on a set of concepts, selected by the user, Hermes is able to determine which news items are relevant. In order to do so Hermes uses state-of-the-art classification techniques for categorizing news items from various sources. At the heart of this process lies the knowledge base, the formal representation of our domain.

The knowledge base is a domain ontology consisting of classes, relationships and instances of classes. For instance in the financial ontology used as an example in this chapter 'Company' and 'Product' are classes and between them there exists a relation like 'hasProduct' and its inverse 'isProducedBy'. We define a concept as being a class or an instance of a class, such as 'Google' is an instance of 'Company' and 'iPhone' is an instance of 'Product'. The knowledge base has initially been extracted from Yahoo! Finance and is maintained by a domain expert. At the current moment it contains more than 300 concepts and their descriptions.

The Hermes News Portal (HNP) is a Java implementation of the Hermes framework. It allows the user to formulate queries and execute them in order to retrieve relevant news items. When news items are gathered from various RSS feeds, they are classified using a GATE (Cunningham et al., 2002) text processing pipeline and the WordNet (Fellbaum, 1998) semantic lexicon. GATE has the advantage of having a modular structure. This allows developers, aside from using out-of-the-box components, to build their own components and adapt the system to their needs. The components that are part of the pipeline used by Hermes are in the usage order: Document Reset, ANNIE English Tokeniser, ANNIE Gazetteer, ANNIE Sentence Splitter, ANNIE Part-Of-Speech Tagger, ANNIE Named Entity (NE) Transducer, and the ANNIE Ortho Matcher. Furthermore, the implementation employs various Semantic Web techniques. The ontology is represented in OWL (Bechhofer et al., 2004), which is queried by using SPARQL (Prud'hommeaux & Seaborne, 2008) and tSPARQL (Borsje et al., 2008), an extension to SPARQL with time-based functionalities.

Athena

Athena is our extension to the Hermes framework for news item recommendation. In Athena we use two types of recommenders, content-based recommenders and semantics-based recommenders. The first type of recommenders determine the similarity between the user profile and a news item based on the frequency of words, while the second type of recommenders determine the similarity between the user profile and a news item based on the meaning of the text by employing concepts and relations.

Athena is written in Java as a plug-in for the existing Hermes News Portal (HNP), which is the implementation of the Hermes framework. The HNP provides Athena with classified news items, which are used in the semantics-based recommendation methods. The content-based method relies on news items from which the stop words are removed and the remaining words are stemmed. For the latter we have used an implementation of the Krovetz Stemmer (Guzman-Lara, 2007). Due to the object-oriented design of Athena it is easily extended with new recommendation methods.

The recommendations tab in Figure 1 corresponds to the Athena plug-in in HNP. The user is shown three subtabs. The first is a list with news items sorted by date in ascending or descending order. For each news item the user is provided with the title, the content, and the date. If the user double clicks a news item it is registered in the user profile and opened in the Web browser.

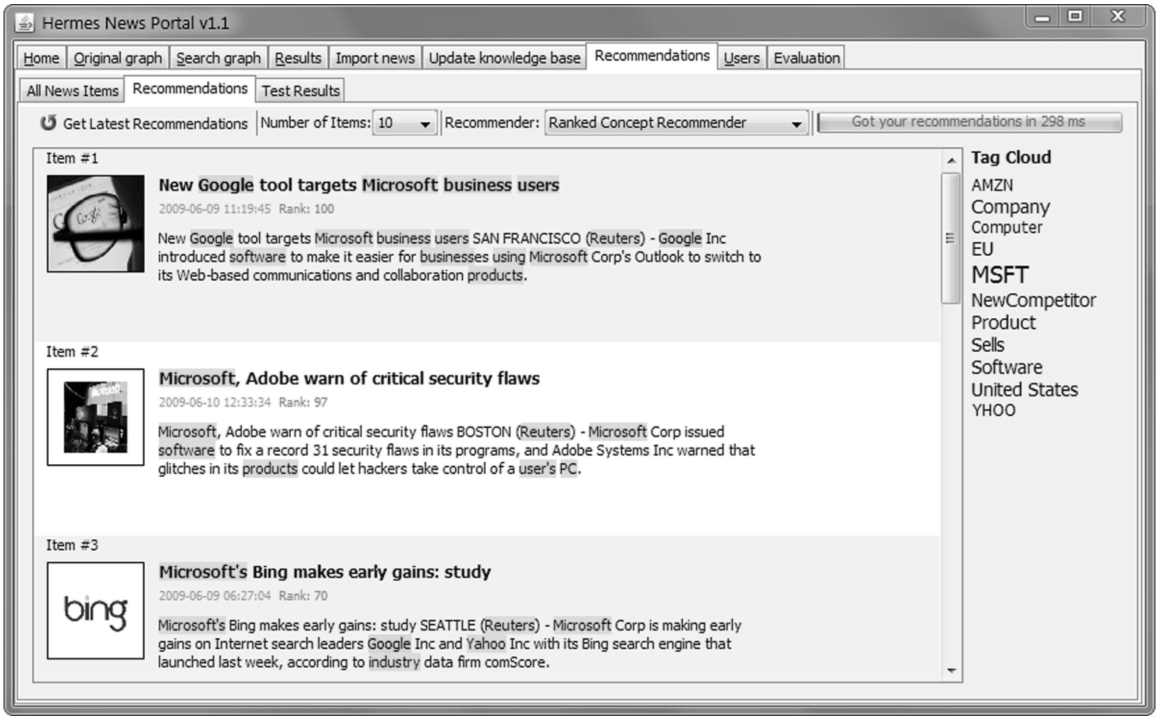

Figure 1. Athena Recommendations

The second tab contains the recommendation functionality. The user is able to select which recommendation method he prefers and how many news items should be shown. Clicking the refresh button starts the recommendation process by determining which news items are of most interest to the user. Generally this process does not take longer than a few seconds, depending on the number of articles and concepts. To give the user insight into his user profile we display a concept list on the right panel. It is similar to the well-known tag cloud. The font size of the concept is dependent on the number of articles read containing this concept.

Once the recommended news items are determined, they are shown in a list, displaying the title, date, and content. An additional feature is to highlight concepts from the user profile (yellow) and concepts related to them (green), which makes it easy to scan news items for concepts of interest. In the third tab we built a test environment used to evaluate the recommendation methods.

RANKED SEMANTIC RECOMMENDATION 2

In Ranked Semantic Recommendation (RSR), we rely on an intuitive approach described in (Bra, Aerts, Houben, & Wu, 2000) applied to adaptive hypermedia. Even though it is used in a different research field, the idea suits our purpose. When applied to our problem, the method assumes that the user reading news articles about a concept means that he might also be interested in directly related concepts. A user profile can be constructed by increasing the rank, which determines the importance of a concept in the user profile, with each article the user has read about a concept and concepts related to them. In addition to increasing the ranks of concepts which appear in the article and concepts related to them, we also assume that if a concept is neither referred to in the text nor is it related to a concept from the text, the rank should be decreased reflecting the user's current disinterest in the concept.

The method proposed in this chapter is an extension to Ranked Semantic Recommendation 1 (RSR1) (IJntema et al., 2010). The difference between RSR1 and RSR2 is that the latter also takes into account the concepts related to the concepts from the unread news items rather than only the concepts that appear in the article. For instance, if the concept ‘Microsoft’ appears in a news item, RSR2 will also take into account related concepts like ‘Google’, ‘Apple’, ‘Windows’ or ‘Steve Ballmer’. We discuss this in more detail later in this section.

We first start with explaining how the extended user profile is constructed followed by how to determine the rank for each concept and end with computing the similarity between an article and the extended user profile. We define the set of related concepts to c_i as follows:

$$r(c_i) = \{c_1^i, c_2^i, \cdots, c_k^i\} \tag{17}$$

The user profile U is the set of concepts that appear in the news items read by the user. Subsequently we define R as the set with concepts related to the concepts from the user profile U:

$$R = \bigcup_{u_i \in U} r(u_i) \tag{18}$$

Finally the extended user profile U_R is the union of both the user profile U and the related concepts of the concepts from the user profile R:

$$U_R = U \cup R \tag{19}$$

The next step is computing the rank for each concept in the extended user profile. In order to do this we organize the concepts in a matrix, where the rows represent concepts appearing in read news items (user profile) and the columns represent concepts from the extended user profile:

	$\mathbf{e_1}$	$\mathbf{e_2}$	…	$\mathbf{e_q}$
$\mathbf{u_1}$	r_{11}	r_{12}	…	r_{1q}
$\mathbf{u_2}$	r_{21}	r_{22}	…	r_{2q}
⋮	⋮	⋮	⋮	⋮
$\mathbf{u_m}$	r_{m1}	r_{m2}	…	r_{mq}

Table 1. Rank Matrix

The ranks r_{11} through r_{mq} are determined as follows:

$$r_{ij} = w_i \times \begin{cases} +0.90 & \text{if } e_j = u_i \\ +0.75 & \text{if } e_j \neq u_i \text{ and } e_j \in r(u_i) \\ -0.05 & \text{otherwise} \end{cases} \tag{20}$$

where the weight w_i is equal to the number of read articles containing the concept u_i and is contained in the following vector:

$$W = (w_1, w_2, \cdots, w_m) \tag{21}$$

The values 0.9, 0.75, and -0.05 are determined empirically by evaluating the recommender performance (i.e., F-measure) on the training set (explained later in the chapter) by trying all combinations between -1 and +1 with a step of 0.05 and are different from RSR1 (as the methods are different). By taking the sum of each column in the matrix, we can compute the final rank for each concept in the extended user profile:

$$Rank(e_j) = \sum_{i=1}^{m} r_{ij} \tag{22}$$

Before comparing the extended user profile with an unread news article, we have to ensure that the ranks are between 0 and 1. Therefore we normalize the ranks as follows:

$$v_i^{new} = \frac{v_i^{old} - \min\limits_{v_u^{old} \in V_U^{old}} (v_u^{old})}{\max\limits_{v_u^{old} \in V_U^{old}} (v_u^{old}) - \min\limits_{v_u^{old} \in V_U^{old}} (v_u^{old})}, v_i^{new} \in V_U \tag{23}$$

where V_U is a vector containing the ranks as determined by Eq. 22.

The last step of the approach is comparing an unread news item with the just created extended user profile in order to determine which articles are of interest to the user. We define the news article as:

$$A = \{a_1, a_2, \cdots a_t\} \tag{24}$$

The new semantics-based recommender proposed in this chapter, RSR2, extends RSR1 (IJntema et al., 2010) by also including the concepts related to the concepts appearing in the news item. This results in a set E:

$$E = \bigcup_{a_i \in A} r(a_i) \tag{25}$$

The extended article is then defined as:

$$A_E = A \cup E \tag{26}$$

Similar to the user profile a vector with ranks is defined for the extended news article:

$$V_{AE} = (s_1, s_2, \cdots, s_t) \tag{27}$$

where

$$s_i = \begin{cases} Rank(e_i) & \text{if } e_i \in A_E \\ 0 & \text{if } e_i \notin A_E \end{cases} \tag{28}$$

The rank corresponding to the concepts from the extended user profile in the vector V_{AE} is equal to the value in the matrix we constructed in the first step if it appears in the unread news item or is related to one of the concepts that appear in the news item, otherwise it is equal to zero. Subsequently we compare vector V_U and vector V_{AE} in order determine the similarity between the user profile and the unread article. We compute to what degree the article fits the user profile by dividing the sum of the ranks of the concepts in A_E by the sum of the ranks in U_R:

$$Similarity(V_{AE}, V_U) = \frac{\sum_{v_a \in V_{AE}} v_a}{\sum_{v_u \in V_U} v_u} \tag{29}$$

The article with the highest similarity is considered to be most interesting to the user.

We conclude this section with an example of RSR2. The example is simplified in order to illustrate the method at hand by limiting the number of concepts and it therefore does not reflect the contents of our knowledge base, which contains many more concepts and relationships. The user profile is defined as follows:

$$U = \{\text{Yahoo!}, \text{Obama}, \text{China}\}$$

The weights W for the corresponding concepts are:

$$W = (4,3,2)$$

which means that the user in this example has read four articles which contained the concept 'Yahoo!', three articles with 'Obama' and two articles with 'China'. The sets of related concepts for each concept in the profile are as follows:

$$r(\text{Yahoo!}) = \{\text{Google}, \text{Apple}\}$$
$$r(\text{Obama}) = \{\text{USA}\}$$
$$r(\text{China}) = \{\text{USA}\}$$

The set R with related concepts to the concept in U is defined as:

$$R = r(\text{Yahoo!}) \cup r(\text{Obama}) \cup r(\text{China}) = \{\text{Google}, \text{Apple}, \text{USA}\}$$

By combining R with U the extended user profile is constructed:

$$U_R = U \cup R = \{\text{Yahoo!}, \text{Obama}, \text{China}, \text{Google}, \text{Apple}, \text{USA}\}$$

The table below shows how the rank for each concept can be computed by applying Equations 20 and 22. The columns show the extended user profile and the rows show the user profile.

	Yahoo!	**Obama**	**China**	**Google**	**Apple**	**USA**
Yahoo!	3.6	-0.2	-0.2	3.0	3.0	-0.2
Obama	-0.15	2.7	-0.15	-0.15	-0.15	2.25
China	-0.1	-0.1	1.8	-0.10	-0.10	1.5
Rank	3.35	2.4	1.45	2.75	2.75	3.55

Table 2. Example Rank Matrix

After normalizing these ranks within a range of [0,1], the following vector is constructed:

$$V_U = (0.905, 0.452, 0, 0.619, 0.619, 1.000)$$

The next step is to determine the interestingness of an unread article based on this user profile. The article that is being examined for this example contains three concepts and can be represented as:

$$A = \{\text{Google}, \text{Apple}, \text{Toyota}\}$$

Let us assume for this example that the concepts related to these concepts are:

$$r(\text{Google}) = \{\text{Yahoo!}, \text{Apple}\}$$
$$r(\text{Apple}) = \{\text{Yahoo!}, \text{Google}\}$$
$$r(\text{Toyota}) = \{\text{Prius}\}$$

The union of *A* with these related concepts (extended article) becomes:

$$A_E = \{\text{Google}, \text{Apple}, \text{Toyota}, \text{Yahoo!}, \text{Prius}\}$$

The corresponding vector is determined by looking up the value for each concept in V_U:

$$V_A = (0.619, 0.619, 0, 0.905, 0)$$

The similarity between article *A* and the user profile is now computed by dividing the sums of both vectors V_U and V_A:

$$Similarity = \frac{0.619 + 0.619 + 0 + 0.905 + 0}{0.905 + 0.452 + 0 + 0.619 + 0.619 + 1.000} = 0.596$$

EVALUATION

The goal of this chapter is to evaluate and compare traditional recommenders and semantic recommenders for news items. For this purpose we have developed a test environment based on the implementation from the previous section. The test environment is based on a supervised learning method.

Setup

The performed tests are based on a corpus of 300 news items that have been assembled by the designer of the test. We have used 5 users with different but well-defined interests in our experiments. An example of a user interest is “Google and all its competitors”. Each user has manually rated the news items as relevant or non-relevant for his interest.

For each user we split the news items corpus in two different sets: 60% of the news items are the training set and 40% of the news items are the test set. We have split the news items in such a way that two sets are filled with relatively equal number of interesting news items (which are user dependent). The training set is used to learn the (extended) user profile, each news item marked by the user as relevant will contribute to the construction of the (extended) user profile. The test set is used to evaluate how well each recommender performs. The recommenders compute the similarity between the news items (from the test set) and the previously computed (extended) user profile (based on the training set). If the computed similarity value is higher than a predefined cut-off value the news item is recommended, otherwise the news item is ignored.

The evaluation of the different recommenders is performed by measuring accuracy, precision, recall (also known as sensitivity), specificity, and F-measure (the harmonic mean of precision and recall). In order to compute these measures we have used a confusion matrix for each user which stores the true positives, true negatives, false positives, and false negatives. Using these measurements we evaluate each recommender and compare them to each other.

Results

Table 3 shows the results of the evaluations of the considered recommenders. The reported results are user-based averages of the considered measures. RSR1 scores better than TF-IDF in terms of accuracy, precision, recall, and equally good for specificity. RSR2, which extends RSR1 so that the indirect concepts that appear in previously unseen news items are also considered in the recommendations, as expected, improves the recall of RSR1, although by decreasing the precision. Nevertheless, the F-measure of RSR2 is higher than RSR1, which means that the increase in recall in higher than the loss in precision. Also, by analyzing the F-measures it is clear that the Jaccard, RSR1, and RSR2 perform better than TF-IDF, showing that the (advanced) semantic recommenders outperform traditional recommenders.

When comparing the winners for each of the investigated performance measures we notice that the best recommenders for accuracy are RSR1 and RSR2, for precision is RSR1, for recall is concept equivalence, for specificity are TF-IDF, Jaccard, and RSR1, and for F-measure is RSR2. RSR algorithms score well on accuracy as they make relatively small amount of errors for both recommended news as well as discarded news items. For precision, RSR1 algorithm scores the best for precision as most recommended news items are relevant. The good results for recall obtained by the concept equivalence are due to the optimistic nature of the algorithm: any news item which involves previously viewed concepts (in news) is recommended. TF-IDF, Jaccard, and RSR1 score well on specificity as these algorithms do not recommend most of the non-relevant news items. The best performing recommender with respect to the F-Measure is RSR2, the newly introduced algorithm in this chapter. Based on these evaluations we suggest the use of RSR2 for news items recommendations in a business intelligence setup feeding on news items as it provides the best trade-off between precision and recall.

The conclusions are drawn for this particular experimental setup. Despite the experiment limitations (in the number of users and news items), it clearly shows the good performance of RSR1 and RSR2 with respect to the other considered recommenders.

	Accuracy	**Precision**	**Recall**	**Specificity**	**F-Measure**
TF-IDF	90%	90%	45%	99%	60%
Concept Equivalence	44%	22%	98%	32%	36%
Binary Cosine	47%	23%	95%	36%	37%
Jaccard	93%	92%	58%	99%	71%
Semantic Relatedness	57%	26%	92%	47%	41%
RSR 1	94%	93%	62%	99%	74%
RSR 2	94%	80%	86%	97%	83%

Table 3. Evaluation results

FUTURE RESEARCH DIRECTIONS

In the future we plan to extend the evaluation with a larger user base and a statistical test. Furthermore, the results presented in this chapter are dependent on the quality of the used knowledge base. A manual maintenance of the knowledge base is a time-intensive and expensive process. Also, as the information in a business domain is continuously changing, it is imperative to provide an automatic solution for the knowledge base maintenance problem. In (Schouten et al., 2010) we have devised lexico-semantic rules that allow for extracting information from news items and used them for updating the knowledge base, closing thus the information processing loop. At the current moment we are extending this language, formally defining its grammar, and applying it for extracting financial events from news.

Another possible research direction relates to considering the importance of terms in news items. For example concepts appearing in the title of news items are possibly more important than the ones that appear in the body of new items. Also, we would like to experiment with advanced traditional weighting schemes that outperform TF-IDF as logarithmic TF functions (Buckley, Allan, & Salton, 1995), in the context of semantic recommendations. Another research direction closely related to the term weighting method is the considered similarity function. We would like to evaluate alternatives for cosine similarity as Lnu.ltu which seem to remove some of the cosine similarity bias favoring long documents over short documents (Singhal, Buckley, & Mitra, 1996).

Regarding RSR, our semantics-based recommender, we would like to consider not only concepts directly related to concepts from the user profile or unread news items but also concepts that are indirectly related (via one or more concepts) to the concepts from the user profile or unread news items. Also, for building such concept neighborhoods the nature of semantic relationships needs to be taken in account (as for example a gloss relationship provides for less semantic similarity than a hyponym or hypernym relationship). By using machine learning techniques the semantic weights between the knowledge base concepts can be learned providing for a more accurate representation of the semantic neighborhood of a concept. These improved specifications can be exploited to recommend more relevant news items, further improving recall.

As additional further work we would like to consider other types of news recommendation services as collaborative filtering or hybrid approaches. In this way we will be able to compare semantic recommenders to these recommenders too. As hybrid approaches combining content-based recommendations and collaborative filtering have already been widely studied, we would like to investigate hybrid recommenders that combine semantic recommenders with collaborative filtering for addressing the user or item cold start problems. Also, we would like to investigate the performance of this type of recommenders with respect to existing hybrid recommenders.

CONCLUSION

In this chapter we have presented several semantic recommenders that can be employed for recommending news items in a business intelligence process. News items, due to their timeliness and rich information content can provide for a competitive advantage in business decision activities. Currently there are hundreds of news items posted daily on the Web, which, combined with a historical analysis of news information, pose considerable challenges to decision makers facing time pressure while making important decisions that may impact companies for many years to come. News recommender systems are useful tools in such processes offering almost instantaneously access to the news items of interest.

A selection of traditional and semantic recommenders for news has been implemented in Athena, an extension of Hermes. Hermes is a framework for building news personalization services that makes use of advanced natural language processing techniques and Semantic Web technologies. Athena benefits from the Hermes classification of news items with respect to domain concepts as well as the available concept semantic relationships it exploits in similarity measures. As traditional methods we have implemented TF-IDF with cosine similarity, and as semantic recommenders we have implemented concept equivalence, binary cosine, Jaccard, semantic relatedness, and ranked semantic recommenders.

In a previous paper we have proposed the Ranked Semantic Recommender which have been proven to perform better not only than the traditional recommenders but also than many of the existing semantic recommenders (IJntema et al., 2010). In this chapter we go one step further, by improving our previous algorithm with respect to recall at the expense of a slight decline in precision. This has been achieved by taking into account not only the concepts directly found in unseen news items but also the concepts directly related to these ones when computing the similarity to the (extended) user profile. The new recommender has a better F-measure than any of the considered traditional and semantic recommenders.

Even though the evaluation was performed with a limited number of users, the results obtained with our semantic recommenders show a better performance of this class of recommenders with respect to traditional content-based recommenders. This is due to the fact that differently than terms, concepts have well-defined semantics and this provides for a more precise definition of user interests, which boosts the precision of the recommendations. Also, concepts have semantic relationships to other concepts that allow considering the “invisible” news information that have the potential to improve the recall rates. In our approach we have used concepts related to the ones of interest or the ones present in unread news items.

REFERENCES

Adomavicius, G., & Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering, 17*(6), 734-749.

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-Based Learning Algorithms *Machine Learning, 6*(1), 37-66.

Ahn, J., Brusilovsky, P., Grady, J., He, D., & Syn, S. Y. (2007). Open User Profiles for Adaptive News Systems: Help or Harm? In *16th International Conference on World Wide Web (WWW 2007)* (pp. 11-20): ACM.

Banos, E., Katakis, I., Bassiliades, N., Tsoumakas, G., & Vlahavas, I. P. (2006). PersoNews: A Personalized News Reader Enhanced by Machine Learning and Semantic Filtering. In *5th International Conference on Ontologies, DataBases and Applications of Semantics (ODBASE 2006)* (Vol. 4275, pp. 975-982): Springer.

Bechhofer, S., Harmelen, F. v., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., et al. (2004). *OWL Web Ontology Language Reference* W3C Recommendation 10 February 2004.

Billsus, D., & Pazzani, M. J. (1999). A Personal News Agent that Talks, Learns and Explains. In *Third International Conference on Autonomous Agents (Agents 1999)* (pp. 268-275): ACM.

Borsje, J., Levering, L., & Frasincar, F. (2008). Hermes: a Semantic Web-Based News Decision Support System. In *23rd Annual ACM Symposium on Applied Computing (SAC 2008)* (pp. 2415-2420): ACM.

Bra, P. D., Aerts, A. T. M., Houben, G. J., & Wu, H. (2000). Making General-Purpose Adaptive Hypermedia Work. In *World Conference on the WWW and Internet (WebNet 2000)* (pp. 117-123): AACE.

Buckley, C., Allan, J., & Salton, G. (1995). Automatic Routing and Retrieval Using Smart: TREC-2. *Information Porcessing and Management, 31*(3), 315-326.

Castellanos, M., Gupta, C., Wang, S., & Dayal, U. (2010). Leveraging Web Streams for Contractual Situational Awareness in Operational BI. In *EDBT/ICDT International Workshop on Business Intelligence and the Web (BEWEB 2010)*: ACM.

Chen, C. C., & Chen, M. C. (2002). PVA: A Self-Adaptive Personal View Agent. *Journal of Intelligent Information Systems, 18*(2/3), 173–194.

Cunningham, H., Maynard, D., Bontcheva, K., & Tablan, V. (2002). GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *40th Anniversary Meeting of the Association for Computational Linguistics (ACL 2002)* (pp. 168-175): ACL.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*: MIT.

Frasincar, F., Borsje, J., & Levering, L. (2009). A Semantic Web-Based Approach for Building Personalized News Services. *International Journal of E-Business Research, 5*(3), 35-53.

Getahun, F., Tekli, J., Chbeir, R., Viviani, M., & Yetongnon, K. (2009). Relating RSS News/Items. In *9th International Conference on Web Engineering (ICWE 2009)* (Vol. 5648, pp. 442-452): Springer.

Guzman-Lara, S. (2007). KStem Java Implementation. from http://ciir.cs.umass.edu/cgi-bin/downloads/downloads.cgi

IJntema, W., Goossen, F., Frasincar, F., & Hogenboom, F. (2010). Ontology-Based News Recommendation. In *EDBT/ICDT International Workshop on Business Intelligence and the Web (BEWEB 2010)*: ACM.

Jaccard, P. (1901). Étude Comparative de la Distribution Florale dans une Portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, *37*, 547-579.

Lin, S.-H., Shih, C.-S., Chen, M. C., Ho, J.-M., Ko, M.-T., & Huang, Y.-M. (1998). Extracting Classification Knowledge of Internet Documents with Mining Term Associations: A Semantic Approach. In *21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1998)* (pp. 241-249): ACM.

Luhn, H. P. (1958). A Business Intelligence System. *IBM Journal of Research and Development 2*(4), 314-319.

Middleton, S. E., Shadbolt, N. R., & De Roure, D. C. (2004). Ontological User Profiling in Recommender Systems. *ACM Transactions on Information Systems*, *22*(1), 54-88.

Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, *2*(1-2), 1-135.

Prud'hommeaux, E., & Seaborne, A. (2008). *SPARQL Query Language for RDF*: W3C Recommendation 15 January 2008.

Saggion, H., Funk, A., Maynard, D., & Bontcheva, K. (2007). Ontology-Based Information Extraction for Business Intelligence. In *6th International Semantic Web Conference (ISWC 2007)* (pp. 843-856): Springer.

Salton, G., & Buckley, C. (1988). Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, *24*(5), 513-523.

Schouten, K., Ruijgrok, P., Borsje, J., Frasincar, F., Levering, L., & Hogenboom, F. (2010). A Semantic Web-Based Approach for Personalizing News. In *25th Symposium on Applied Computing (SAC 2010)* (pp. 854-861): ACM.

Singhal, A., Buckley, C., & Mitra, M. (1996). Pivoted Document Length Normalization. In *19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1996)* (pp. 21-29): ACM.

Singhal, A., Salton, G., Mitra, M., & Buckley, C. (1996). Document Length Normalization. *Information Processing & Management*, *32*(5), 619-633.

van Meteren, R., & van Someren, M. (2000). Using Content-Based Filtering for Recommendation. In *ECML/MLnet Workshop on Machine Learning in the New Information Age 2010 (MLNIA 2010)* (pp. 47-56).

ADDITIONAL READING

Billsus, D., & Pazzani, M. J. (1999). A Hybrid User Model for News Story Classification. In *Seventh International Conference on User Modeling (UM 1999)* (pp. 99-108): Springer.

Bogers, T., & Bosch, A. v. d. (2007). Comparing and Evaluating Information Retrieval Algorithms for News Recommendation. In *2007 ACM Conference on Recommender Systems (RecSys 2007)* (pp. 141-144): ACM.

Cantador, I., Bellogín, A., & Castells, P. (2002). News@hand: A Semantic Web Approach to Recommending News. In *5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2002)* (Vol. 5149, pp. 279-283): Springer.

Cantador, I., Bellogín, A., & Castells, P. (2008). Ontology-based Personalised and Context-aware Recommendations of News Items. In *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI 2008)* (Vol. 1, pp. 562-565): IEEE Computer Society.

Chedrawy, Z., & Abidi, S. S. R. (2009). A Web Recommender System for Recommending, Predicting and Personalizing Music Playlists. In *10th International Conference on Web Information Systems Engineering (WISE 2009)* (Vol. 5802, pp. 335-342): Springer.

Chen, Y.-S., & Shahabi, C. (2001). Automatically Improving the Accuracy of User Profiles with Genetic Algorithm. In *IASTED International Conference on Artificial Intelligence and Soft Computing (ASC 2001)* (pp. 21-24): ACTA Press.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., & Sartin, M. (1999). Combining Content-based and Collaborative Filters in an Online Newspaper In *ACM SIGIR Workshop on Recommender Systems (SIGIR-Rec 1999)*.

Conlan, O., O'Keeffe, I., & Tallon, S. (2006). Combining Adaptive Hypermedia Techniques and Ontology Reasoning to produce Dynamic Personalized News Services. In *4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2006)* (Vol. 4018, pp. 81-90): Springer.

Das, A. S., Datar, M., Garg, A., & Rajaram, S. (2007). Google News Personalization: Scalable Online Collaborative Filtering. In *16th International Conference on World Wide Web (WWW 2007)* (pp. 271-280): ACM.

Gauch, S., Chaffee, J., & Pretschner, A. (2003). Ontology-Based Personalized Search and Browsing. *Web Intelligence and Agent Systems, 1*(3/4), 219-234.

Gauch, S., Speretta, M., Chandramouli, A., & Micarelli, A. (2007). User Profiles for Personalized Information Access. In *The Adaptive Web* (Vol. 4321, pp. 54-89): Springer.

Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition, 5*(2), 199-220.

Guarino, N. (1998). Formal Ontology and Information Systems. In *First International Conference Formal Ontology in Information Systems (FOIS 1998)* (pp. 3-15): IOS Press.

Hopfgartner, F., & Jose, J. M. (2010). Semantic User Profiling Techniques for Personalised Multimedia Recommendation. *Multimedia Systems, 16*(4/5), 255-274.

Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). *Recommender Systems: An Introduction*: Cambridge University Press.

Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D., & Allan, J. (2000). Language Models for Financial News Recommendation. In *Conference on Information and Knowledge Management (CIKM 2000)* (pp. 389-396): ACM.

Liu, J., Dolan, P., & Pedersen, E. R. (2010). Personalized News Recommendation Based on Click Behavior. In *14th International Conference on Intelligent User Interfaces (IUI 2010)* (pp. 31-40): ACM.

Middleton, S. E., De Roure, D. C., & Shadbolt, N. R. (2001). Capturing Knowledge of User Preferences: Ontologies in Recommender Systems. In *First International Conference on Knowledge Capture (K-CAP 2001)* (pp. 100-107): ACM.

Middleton, S. E., Alani, H., & De Roure, D. C. (2002). Exploiting Synergy Between Ontologies and Recommender Systems. In *WWW Semantic Web Workshop 2002 (SWW 2002)* (Vol. 55): CEUR-WS.org.

Middleton, S. E., Shadbolt, N. R., & De Roure, D. C. (2003). Capturing Interest Through Inference and Visualization: Ontological User Profiling in Recommender Systems. In *2nd International Conference on Knowledge Capture (K-CAP 2003)* (pp. 62-69): ACM.

Resnick, P., & Varian, H. R. (1997). Recommender Systems. *Communications of the ACM, 40*(3), 56-58.

Sieg, A., Mobasher, B., & Burke, R. (2007). Web Search Personalization with Ontological User Profiles. In *Sixteenth ACM Conference on Conference on Information and Knowledge Management (CIKM 2007)* (pp. 525-534): ACM.

Tan, A.-H., & Teo, C. (1998). Learning User Profiles for Personalized Information Dissemination. In *International Joint Conference on Neural Networks (IJCNN 1998)* (pp. 183-188): IEEE Computer Society.

Trajkova, J., & Gauch, S. (2004). Improving Ontology-Based User Profiles. In *7th International Conference on Adaptivity, Personalization and Fusion of Heterogeneous Information (RIAO 2004)* (pp. 380-390): CID.

Wang, J., Li, Q., Chen, Y. P., & Lin, Z. (2010). Recommendation in Internet Forums and Blogs. In *48th Annual Meeting of the Association for Computational Linguistics* (pp. 257-265): ACL.

Wang, J., Li, Q., Chen, Y. P., Liu, J., Zhang, C., & Lin, Z. (2010). News Recommendation in Forum-Based Social Media. In *Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2010)* (pp. 1449-1454): AAAI.

Ziegler, C.-N. (2005). Semantic Web Recommender Systems. In *Joint EDBT/ICDT PhD Workshop 2004* (Vol. 3268, pp. 78-89).

Ziegler, C.-N., Lausen, G., & Konstan, J. A. (2008). On Exploiting Classification Taxonomies in Recommender Systems. *AI Communications, 21*(2/3), 97-125.

KEY TERMS AND DEFINITIONS

Concept: A concept is a class or an instance of a class from a domain ontology. For example, a class is ‘Company’ and an instance of this class is ‘Google’.

Content-Based Recommendation: A content-based recommendation is based solely on the terms that appear in the text content of an item. Terms have often associated importance values based on weighting schemes, e.g., the frequencies of a term in an item. A popular content-based recommendation technique is TF-IDF combined with the cosine similarity measure.

Domain Ontology: A domain ontology is a formal explicit specification of a shared domain conceptualization. It is composed of concepts and concept relationships that characterize a certain domain.

Ranked Semantic Recommendation: A ranked semantic recommender is a news recommendation method where concepts and relations from an ontology are employed for determining the user’s interests and recommending items previously unseen. It makes use of domain ontology concepts and their relationships that goes beyond the user profile/item content.

Related Concept: A related concept is a concept that is connected by any relationship to the current concept. For instance ‘Apple’ and ‘iPhone’ are connected through the ‘hasProduct’ relation. The related concepts are used to better specify the context of the user interests and item information.

Semantics-Based Recommendation: A semantics-based recommendation is a recommendation technique that uses concepts and relations between them to determine the meaning of the text, rather than by only analyzing terms occurrences as is the case for content-based recommendations.

Semantic Web: The Semantic Web is a collection of methods and technologies that enable machines to understand the ‘meaning’ of Web content. It aims to make available Web information for processing by intelligent Web agents.

User Profile: A user profile is a formal description of the user’s current interests often based on previously read items. Based on the user profile, recommender systems suggest to the user items that match his domain of interest.