Ontology-Based News Recommendation

Wouter IJntema
wouterijntema@gmail.com

Frank Goossen
frankgoossen@gmail.com

Flavius Frasincar
frasincar@ese.eur.nl

Frederik Hogenboom
fhogenboom@ese.eur.nl

Erasmus University Rotterdam
PO Box 1738, NL-3000
Rotterdam, the Netherlands

ABSTRACT

Recommending news items is traditionally done by term-based algorithms like TF-IDF. This paper concentrates on the benefits of recommending news items using a domain ontology instead of using a term-based approach. For this purpose, we propose Athena, which is an extension to the existing Hermes framework. Athena employs a user profile to store terms or concepts found in news items browsed by the user. Based on this information, the framework uses a traditional method based on TF-IDF, and several ontology-based methods to recommend new articles to the user. The paper concludes with the evaluation of the different methods, which shows that the new ontology-based method that we propose in this paper performs better (w.r.t. accuracy, precision, and recall) than the traditional method and, with the exception of one measure (recall), also better than the other considered ontology-based approaches.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering, Relevance feedback*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Representation languages*

General Terms

Design, Experimentation

Keywords

Recommender systems, User profiling, Ontology

1. INTRODUCTION

In the last decade, the Web has become increasingly important in delivering news to individuals. Many people read news articles for different purposes and the Web is the best platform to find them. However, these news items are not personalized for one's interests. In this paper we propose an approach based on rich semantics for delivering the most interesting news items to the user.

Recommending news items can be done by calculating the similarity between the current news item and the previously browsed news items. Traditionally, this similarity is calculated by an algorithm that is content-based, which practically means that every word in a news item is taken into account. However, a news item often contains key concepts that capture the semantic context of the article. Recommenders that focus on the key concepts might produce faster and more accurate recommendations than the content-based recommenders, since they don't need to consider all words, and unlike words, concepts are not ambiguous. Such an approach is called a semantic-based recommendation system. Other recommendation systems are either collaborative or hybrid, and are outside the scope of this paper.

In [7], we introduced the Hermes framework, which provides a semantic method for personalizing news items. It uses an ontology to store concepts and their relations to the news items. Our paper focuses on a new way of recommending, based on concepts found in the news items, by employing some of the functionalities offered by Hermes.

In order to recommend news items, first we model the user's browsing behavior. By recording a history of read news items, a profile of the user can be made. Based on this profile, it is possible to propose new news items that the user might find interesting. The goal of our research is to investigate the benefit of recommending news items by using domain ontology-based recommenders with respect to traditional term-based recommenders, and to determine which of the ontology-based recommenders performs best.

In this paper we propose Athena, which is an extension of the Hermes framework. Athena is able to observe user behavior and generate recommendations based on this behavior. The program uses a traditional term-based recommender and several semantic-based recommendation algorithms to compare unread news items with the user profile. The news items having the highest similarity with the user profile are recommended to the user.

The structure of this paper is as follows. First, we discuss the related work in Sect. 2. Section 3 presents the Athena framework, the Hermes framework, and the Hermes News Portal (HNP), which is the implementation of the Hermes framework. After that, Sect. 4 describes the implementation

EDBT 2010, March 22–26, 2010, Lausanne, Switzerland.

of Athena as a plug-in for the HNP. Then, Sect. 5 gives the evaluation of the implemented methods. Section 6 concludes the paper and proposes future work.

2. RELATED WORK

Recommending news items or other documents based on the user's interest has attracted the attention of many researchers. Several adaptive Web-based news services have been developed which focus on personal recommendation of news items. These systems vary in application domain, platform, development methodology, levels of adaptivity, etc. We identify four categories in recommendation systems, content-based, semantic-based, collaborative, and hybrid systems. In this paper, we limit the discussion to content-based and semantic-based recommendation methods.

YourNews [1] is a personalized news system, that employs a content-based approach, which intends to increase the transparency of adapted news delivery by allowing the user to adapt the user profile. Another content-based approach is News Dude [2], which is a personal news recommending agent, that utilizes TF-IDF in combination with the Nearest Neighbor algorithm in order to recommend news items to the user. [3] states, supported by Singhal's findings [12], that the performance of TF-IDF, which is employed in YourNews and NewsDude, decreases as the length of the article, and the number of words, increases. In addition to this, by ignoring the semantics of a text, news items that are semantically related to the news items in the user profile, fail to be recommended by the system.

[8] provides a practical approach to measure the relatedness or similarity between RSS news items. Their method is based on the semantic relatedness between RSS items. As in our approach, they determine the relationships between words, using WordNet [6]. Their focus is on the linguistic neighborhood of a word, in which general relationships as synonymy, hyponymy, and meronymy between words are considered. The difference with our approach is that we make use of an ontology. Besides the general relationships between words, the ontology covers specific relationships like *is-competitor-of*, *has-product*, etc. Despite this difference, their method is applicable in our context, and therefore we will compare both approaches.

In [10] ontological user profiling is employed for recommending academic research papers. While is-a relationships are rich in semantics, we find this approach limited, as it fails to consider other types of concept relationships. The authors propose a classification algorithm, based on the k-Nearest Neighbor classifier, that assigns topics to papers. In our approach, GATE [4] is employed to classify the content of an article by using several language processing techniques. This enables the system to not only recommend full articles, but also possibly recommend a snippet of an article. Another difference lies in the construction of the user profile, as in [10], the user can adjust the profile. However, as [1] explains, adjusting the user profile might harm the quality of the recommendations, so in our approach the user is not allowed to change the profile. Recommendations are made by combining collaborative filtering techniques with limited semantic-based recommendations, that only employ is-a relations, while our system solely employs semantic-based recommendation techniques that utilize more types of relationships between concepts.

3. ATHENA

In this paper we propose Athena, which is an extension to the Hermes framework. Subsection 3.1 explains the Hermes framework and how it contributes to the recommendation of news items. Subsection 3.2 explains how the user profile is constructed. In subsection 3.3 and subsection 3.4 we discuss some existing content-based respectively, semantic-based recommendation methods. In subsection 3.5 we introduce the ranked recommendation method, our semantic-based recommendation method.

3.1 Hermes

Athena is an extension to the Hermes framework [7], a framework used to build a news personalization service. The system can be described by input, internal processing, and output. The input is composed of predefined RSS feeds of news items and concepts selected by the user. The internal processing is the classification of these news items using concepts from a knowledge base. The output is defined as the personalized news items based on the selected concepts.

3.1.1 The Ontology

The Hermes framework offers a semantic-based approach for retrieving news items related, directly or indirectly, to the concepts of interests from a domain ontology, which is called the knowledge base [7]. The ontology consists of classes, e.g., *Company* and *CEO*, and the relationship between these classes, like *isCEOOf* and its inverse *hasCEO*. A concept is defined as either a class or an instance of a class, e.g., *Company* and *Microsoft*. The knowledge base is constructed and maintained by a domain expert, with financial information obtained from Yahoo! Finance.

3.1.2 The Hermes News Portal

The Hermes News Portal (HNP) is a Java implementation of the Hermes framework [7]. It allows the user to query the news and view the knowledge base. It uses Jena for manipulating and reasoning with the OWL ontologies. For querying, it employs SPARQL and tSPARQL [7], which adds time functionalities to the queries. The classification of the news articles is done using GATE [4] and the WordNet [6] semantic lexicon.

3.2 User Profile Construction

Recommending news items starts with building a user profile. A user profile can be defined by keeping track of which articles the user has read so far. Those articles will provide us with information about the user's interests. The user profile is constructed in different ways. For concept equivalence, binary cosine, and Jaccard, the profile is a set of concepts from the articles the user has read. The semantic relatedness approach creates a vector with the distinct concepts from the user profile and assigns a weight to each concept. The ranked recommendation method also uses a vector of distinct concepts from the read articles and assigns a rank to each concept. The difference in user profile construction between the latter two approaches, is the method used to compute the corresponding weights.

3.3 Content-Based Recommendation

A well-known term weighting method is TF-IDF (term frequency-inverse document frequency) [11]. A classic approach in comparing documents is the use of TF-IDF to-

gether with the cosine similarity measure. TF-IDF is a statistical method used to determine the relative importance of a word within a document in a collection (or corpus) of documents.

As discussed in [1], before calculating the TF-IDF values, the stop words are being filtered from the document. After stop word removal, the remaining words are stemmed using a stemmer. This process reduces words like 'process', 'processor', 'processing', and 'processed' back to their root word 'process'.

The TF-IDF measure can be determined by first calculating the term frequency (TF), which indicates the importance of a term t_i within a document d_j. By computing the inverse document frequency (IDF), the general importance of the term in a set of documents can be captured.

The objective is to compare any new document against the user profile. Therefore a vector is calculated for the user profile. This vector contains the TF-IDF value for 100 words with the highest TF-IDF value from the documents that have been read by the user. Subsequently in the same manner a vector, based on the total set of documents, is created for the new document that is being compared to the user profile. By calculating the cosine measure of the news item and the user profile, the similarity can be determined. The articles with the highest similarity value are considered to be the most similar to the user profile and are recommended to the user.

3.4 Semantic-Based Recommendation

In traditional forms of text comparison, all words in the text are considered. In addition to this, there is no relation between different words. For instance, it is not possible to determine the relation between *Google* and *Microsoft*. But a user who is interested in news regarding his stocks in *Google*, might also be interested in news about *Microsoft*, because it is a competitor of *Google*. Using an ontology that covers those relations might therefore be useful in recommending new articles. To illustrate how we accomplished this, we will first discuss a few simple methods and then conclude with a complex method.

3.4.1 Concept Equivalence

We start with a very simple technique which only considers the equivalent concepts. The ontology contains a set of n concepts:

$$C = \{c_1, c_2, c_3, \cdots, c_n\} \ . \tag{1}$$

The user profile consists of p concepts identified by Hermes in the news previously read by the user. A concept is present in a news item if one of the concept lexical representations is found in the news item and the meaning of this lexical representation in the context of the news item corresponds to the meaning of the concept as defined in the domain ontology. The user profile can be represented as the following set:

$$U = \left\{c_1^u, c_2^u, c_3^u, \cdots, c_p^u\right\}, \text{where } c_i^u \in C \ . \tag{2}$$

A news article can also be formulated as a set of q concepts that appear in the article:

$$A = \left\{c_1^a, c_2^a, c_3^a, \cdots, c_q^a\right\}, \text{where } c_j^a \in C \ . \tag{3}$$

Using sets of concepts, makes it impossible to compute the similarity using the regular cosine measure. This measure requires a vector of values, like TF-IDF values. The interestingness of a new news item is determined by computing the intersection between the previous two sets:

$$\text{Similarity}(U, A) = \begin{cases} 1 & \text{if } |U \cap A| > 0 \\ 0 & \text{otherwise} \end{cases} \ . \tag{4}$$

If this results in 1, the article is considered interesting, otherwise it is considered not interesting.

3.4.2 Binary Cosine

To compute the similarity between two texts, we can also use the binary cosine similarity coefficient:

$$B(U, A) = \frac{|U \cap A|}{|U| \times |A|} \ , \tag{5}$$

where $|U \cap A|$ represents the number of concepts in the intersection of U and A, and $|U|$ and $|A|$ are respectively the number of concepts in U and A.

3.4.3 Jaccard

The Jaccard similarity coefficient can be computed in a similar manner:

$$J(U, A) = \frac{|U \cap A|}{|U \cup A|} \ , \tag{6}$$

where $|U \cap A|$ is the number of concepts in the intersection of U and A, and $|U \cup A|$ is the number of concepts in the union of U and A. Jaccard computes the number of elements in the intersection of the concepts found in the user profile and the news item, relatively to the number of concepts in the union of these two sets.

3.4.4 Semantic Relatedness

In [8] the focus is on the semantic relationship between words. The semantic neighborhood of a concept $c_i \in C$ is defined as the set of concepts related to it via the synonymy ($\equiv$), hyponymy ($\prec$), and meronymy ($<<$) relations. Our ontology covers more relations than only the linguistic relations. Therefore the semantic neighborhood of concept c_i includes each concept that is directly related to the concept c_i (including c_i):

$$N(c_i) = \left\{c_1^i, c_2^i, \cdots, c_n^i\right\} \ . \tag{7}$$

A text t_k can be described by a set of concepts:

$$CS_k = \left\{c_1^k, c_2^k, \cdots, c_m^k\right\} \ . \tag{8}$$

When comparing two texts, t_i and t_j, a vector in n-dimensional space can be created, according to the vector space model:

$$V_l = (\left\langle c_1^l, w_1^l \right\rangle, \cdots, \left\langle c_p^l, w_p^l \right\rangle) \ , \tag{9}$$

where $l \in \{i, j\}$ and w_i represents the weight associated to the concept c_i and $p = |CS_i \cup CS_j|$ is the number of distinct concepts in CS_i and CS_j. If the concept c_i is referenced in CS_j then $w_i = 1$, otherwise it is computed based on the

maximum enclosure similarity it has with another concept c_j in its corresponding vector V_j. This takes into account the global semantic neighborhood of each concept as follows:

$$w_i = \begin{cases} 1 & \text{if freq}(c_i \text{ in } CS_j) > 0 \\ \max_j(\text{ES}(c_i, c_j)) & \text{otherwise} \end{cases} \quad (10)$$

where

$$\text{ES}(c_i, c_j) = \frac{|N(c_i) \cap N(c_j)|}{|N(c_i)|} \ . \quad (11)$$

Finally the similarity between t_i and t_j is computed using the cosine measure:

$$\text{SemRel}(t_i, t_j) = \cos(V_i, V_j) = \frac{V_i \cdot V_j}{||V_i|| \cdot ||V_j||} \in [0, 1] \ , \quad (12)$$

where the nominator is the dot product of both vectors and the denominator is the multiplication of the magnitude of each vector.

The advantage of this approach above concept equivalence, binary cosine, and Jaccard, is that it also takes into account the related concepts of a concept that occurs in a text.

3.5 Ranked Semantic Recommendation

[5] describes an intuitive approach in working with adaptive hypermedia. For instance when you read something about concept c_1 which is related to concept c_2 and concept c_3 you increase not only your knowledge in concept c_1 but also in the other two concepts.

Even though it is used in a different research field (adaptive hypermedia), the main idea can be applied also here. Each concept is assigned a value, this value we call the rank. For example, when a user reads about *Google*, he might also be interested in its competitors, like *Yahoo!*, but also in news about its CEO, *Eric Schmidt*. Both are considered to be in direct relation to the concept *Google*. Therefore we increase the rank for *Google*, *Yahoo!*, and *Eric Schmidt*. Unrelated concepts, i.e., concepts that are not directly connected to the current concept, also need to be addressed. This means, if a user profile consists of concepts c_1 and c_2, and the next article the user reads, contains concept c_3, which is directly related to c_1, but not related to c_2, we increase the rank of c_1, and decrease the rank of c_2. By decreasing the rank for such a concept we make the user profile adaptive to the user's main interest.

The set of related keywords to concept c_i is defined as:

$$r(c_i) = \left\{ c_1^i, c_2^i, \cdots, c_k^i \right\} \ . \quad (13)$$

R is described as the union of all related concepts to the concepts in the user profile:

$$R = \bigcup_{u_i \in U} r(u_i) \ . \quad (14)$$

And finally U_R is defined as the set of all concepts and corresponding related concepts, this is called the extended user profile:

$$U_R = U \cup R \ . \quad (15)$$

The extended user profile is used in order to be able to increase the interest of the user in certain concepts that are not in the user profile, but are related to the concepts in the user profile.

To calculate the final ranks for each concept, we organize the concepts in a matrix. This is done because we have to assign a rank to each concept in the extended user profile for each concept the user has read about. Reading about concept c_1 increases its value with 1.0. If concept c_2 is directly related to concept c_1, then its value is increased with 0.5. If there is a concept, concept c_3, in the extended profile which is neither equal to concept c_1 nor is it related to concept c_1, its value is decreased with 0.1. These constants were determined by experimenting with values ranging from 0 to 1 with a step of 0.1. Applying this procedure results in a matrix with rank values. The columns contain the items from the extended user profile (U_R) and the rows contain the items from the user profile (U). Table 1 shows a rank matrix, where $e_i \in U_R$ and $u_i \in U$. Summing the values of the cells in a column of the matrix, and repeating this process for each column, results in a vector with the final ranks for each concept, in the extended user profile.

Table 1: Rank matrix

	e_1	e_2	$\dots$	e_q
u_1	r_{11}	r_{12}	$\dots$	r_{11}
u_2	r_{21}	r_{22}	$\dots$	r_{2q}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
u_m	r_{m1}	r_{m2}	$\dots$	r_{mq}

The user might have read one or more articles about a concept. Logically, the user is presumed to be more interested in concepts that are found in several articles. The number of articles the user has read about concept u_i, is called the weight w_i,

$$W = \{w_1, w_2, \cdots, w_m\} \ . \quad (16)$$

Now we can calculate the value for each cell in the above matrix. This is done as follows:

$$r_{i,j} = w_i \times \begin{cases} +1.0 & \text{if } e_j = u_i \\ +0.5 & \text{if } e_j \neq u_i, e_j \in r(u_i) \\ -0.1 & \text{otherwise} \end{cases} \ . \quad (17)$$

The final rank for each concept from the extended user profile, can be computed by taking the sum of the values of the corresponding column in the matrix:

$$\text{Rank}(e_j) = \sum_{i=1}^{m} r_{ij} \ . \quad (18)$$

Those sums are stored in a vector V_U. Each concept in the extended user profile now has a rank. Before we can compare the user profile with an unread news article, we need to ensure that the range of the ranks is [0,1]. The normalization is done as follows:

$$V_U[v_i] = \frac{v_i - \min(v_u)}{\max(v_u) - \min(v_u)}, \quad (19)$$

where $v_i \in V_U$ and $v_u \in V_U$. With this normalization we can compare the extended user profile to a new article that

needs to be classified. The new article consists of a set of concepts, specified as A:

$$A = \{a_1, a_2, \cdots, a_t\} \ . \tag{20}$$

For this article we define a vector containing the ranks. This vector is defined as V_A:

$$V_A = (s_1, s_2, \cdots, s_t) \ , \tag{21}$$

$$s_i = \begin{cases} \text{Rank}(e_i) & \text{if } e_i \in A \\ 0 & \text{if } e_i \notin A \end{cases} \ . \tag{22}$$

Each concept from the extended user profile that appears in the article is assigned the same rank as the one in V_U. The remaining concepts are assigned zero. Concepts appearing in the article but not in the profile are ignored. In the current work we assume that all concepts found in a news item are equally important.

To compare the article with the user profile we propose to compute the extent to which the article fits the profile by dividing the sum of the ranks of concepts in the article by the sum of the ranks of the concepts in the user profile:

$$\text{Similarity}(V_A, V_U) = \frac{\sum_{v_a \in V_A} v_a}{\sum_{v_u \in V_U} v_u} \ . \tag{23}$$

The article with the highest similarity measure fits best with the user profile. The cut-off value for news item interestingness was fixed to 0.5, after experimenting with values ranging from 0 to 1 with a step of 0.1.

4. ATHENA IMPLEMENTATION

As Athena is an extension to the Hermes framework, it has been implemented as a plug-in to the existing implementation of the Hermes framework, the Hermes News Portal (HNP). The implementation of Athena is done in the same language as the HNP, Java. As a stemmer, for the content-based method, we have used the Krovetz Stemmer [9].

The user interface of Athena consists of 3 tabs: a browser for all news items, a tab for the recommendations, and a tab for evaluation purposes. The browser contains the news items sorted by date. Here, the user can browse through the news items instead of browsing through query results as in the HNP. Each item is presented with a title, summary, an image which is related to the news item, and the date published.

The user profile is created from the articles the user has read. We define reading an article as opening it into the Web browser. After reading several articles, the user can select the recommendations tab in Athena. Here the user can choose a type of recommender, and get the recommended articles based on the user profile. Only one recommender can be chosen at a time. By clicking the refresh button, the recommender starts analyzing the user profile. After a short period of time, the recommender presents a list of news items that the user may find interesting. This list consists of the news items that the recommender ranked highest. Each news item is presented with its corresponding ranks. The user can browse through the results, and by double-clicking at a news item, it is registered in the user profile, whereafter the user's Web browser shows the concerning news article.

We also have included a concept list (similar to the well-known tag cloud), which displays all the concepts that have been stored in the user profile. When a concept is read in multiple articles, the font gets larger. Also, we have included a feature which highlights the concepts and related concepts found in the article in different colors.

Additionally, Athena provides a testing environment for evaluation purposes which will be discussed in section 5.

5. EVALUATION

Our research goal was to find whether ontology-based recommenders perform better than a classic recommender like TF-IDF. To evaluate our approach, we have developed a test method and built a test environment.

The testing method we have chosen, is based on supervised learning. First the user is shown a set of 300 news articles, assembled by the designer of the test. For each article the user has to read the title and the summary. Based on this, he should decide whether the article is interesting or not. For the experiments we have used 5 users, each user having different news interests than the other ones.

Subsequently, this set of articles, with the corresponding ratings by the user, is split randomly into two different sets, the training set (60%) and the validation set (40%). The two sets are filled with a relatively equal number of interesting items. The training set is used to create a user profile. Each item that is marked as interesting will be added to this profile. The validation set is used by each recommender to determine for each news item the similarity with the user profile. An article is considered to be interesting if the similarity to the user profile is higher than the predefined cut-off value, otherwise it is classified as not interesting.

To determine the performance of a recommender, measures like accuracy, precision, recall (sensitivity), and specificity are used. These measures are calculated by using a confusion matrix, which stores the number of true positives, false positives, false negatives, and true negatives, for each of the analyzed recommender systems. Based on these measures, in the rest of this section, we compare the performance of the ranked recommender with respect to the performance of the other considered recommender systems.

The results in Table 2 and Table 3 show that the ranked recommender scores better than TF-IDF for accuracy (94% vs. 90%), precision (93% vs. 90%), and recall (62% vs. 45%), and has the same high score for specificity (99%). For accuracy and precision, from all implemented methods, the ranked recommender scores best, closely followed (difference of 1%) by the Jaccard recommender. The recall of the ranked recommender (62%) is nevertheless lower than the recall of concept equivalence (98%), binary cosine (95%), and semantic relatedness (92%). The best specificity (99%) is for the ranked recommender, Jaccard, and TF-IDF.

The ranked recommender is able to propose interesting stories for the user, eliminating most uninteresting stories. Nevertheless, during the news filtering, news items deemed interesting by the user are also wrongly eliminated. However, the ranked recommender provides the user with more interesting news items relative to the total number of recommended new items than a traditional recommender system. The ranked recommender also suggests more interesting stories relative to the total number of recommended new items than the other considered semantic-based recommenders.

Table 2: Accuracy and Precision

Method	Accuracy	Precision
TF-IDF	90%	90%
Concept Equivalence	44%	22%
Binary Cosine	47%	23%
Jaccard	93%	92%
Semantic Relatedness	57%	26%
Ranked	94%	93%

Table 3: Sensitivity and Specificity

Method	Recall	Specificity
TF-IDF	45%	99%
Concept Equivalence	98%	32%
Binary Cosine	95%	36%
Jaccard	58%	99%
Semantic Relatedness	92%	47%
Ranked	62%	99%

6. CONCLUSION

This paper describes Athena, an extension to the Hermes framework that provides several methods for news item recommendation based on the user's interests. The system uses a user profile, news items, and several similarity measures.

At the heart of Athena is the ontology provided by the Hermes framework. This ontology contains the domain concepts and the relationships between the concepts. With these relationships, more information about each concept is available than only the concept itself. This allows Athena to consider different articles interesting than by using existing technologies that employ content-based methods, like TF-IDF, because it does not only consider the concepts that appear in the article, but also the ones that are related to them.

We have described different methods to employ ontologies in comparing the user profile with a new article. We started with a content-based method that employs TF-IDF and the cosine similarity measure, followed by three basic semantic-based methods. Concept equivalence is a simple, intuitive method that looks for articles that contain at least one of the concepts from the profile. This method does not take into account the number of concepts found in the news article. In order to take into account these concepts, we have used binary cosine and Jaccard. Those methods compute the similarity between the article and the profile.

A more advanced method also takes into account the semantic relatedness between different concepts, which are provided by the underlying ontology. A weight is assigned to each concept based on its neighborhood and the enclosure similarity. This method, referred as semantic relatedness, is based on linguistic relationships. Finally, we presented a new method, called ranked recommender, which also uses the ontology relationships between the concepts. It takes the concepts from the user profile and combines these with the related concepts to create the extended user profile.

In this paper, we have shown that the ranked recommender, our ontology-based recommender, performs better than a traditional recommender systems based on TF-IDF for accuracy, precision, and recall, and equally good for specificity. It also performs better, or equally good, with respect to accuracy, precision, and specificity than the other considered ontology-based recommenders. Nevertheless, the recall is lower than some of the implemented ontology-based recommenders.

The knowledge base that is used, is partly created by a domain expert and takes a lot of effort. Future research should focus on automatically creating and maintaining such a knowledge base to support ontology-based recommendation methods. Besides the improvement of the knowledge base, the algorithm can be improved as well. In our approach we have focused on a limited number of relations between concepts, for instance only the direct relations. However, concepts might be related to each other on different levels, i.e., concepts might not be directly related to each other but there might exist a relation with one or more concepts between them. Additionally, we would like, in the future, to take into account the importance of a concept in a news item.

7. REFERENCES

[1] J. Ahn, P. Brusilovsky, J. Grady, D. He, and S. Y. Syn. Open User Profiles for Adaptive News Systems: Help or Harm? In *16th International Conference on World Wide Web*, pages 11–20. ACM, 2007.

[2] D. Billsus and M. J. Pazzani. A Personal News Agent that Talks, Learns and Explains. In *The Third Annual Conference on Autonomous Agents*, pages 268–275. ACM, May 1999.

[3] T. Bogers and A. van den Bosch. Comparing and Evaluating Information Retrieval Algorithms for News Recommendation. In *ACM Conference On Recommender Systems*, pages 141–144. ACM, 2007.

[4] H. Cunningham. GATE, a General Architecture for Text Engineering. *Computers and the Humanities*, 36:223–254, 2002.

[5] P. De Bra, A. T. M. Aerts, G. J. Houben, and H. Wu. Making General-Purpose Adaptive Hypermedia Work. In *WebNet 2000 Conference*, pages 117–123. AACE, 2000.

[6] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.

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

[8] F. Getahun, J. Tekli, C. Richard, M. Viviani, and K. Yetongnon. Relating RSS News/Items. In *9th International Conference on Web Engineering*, pages 442–452. Springer, 2009.

[9] S. Guzman-Lara. KStem Java Implementation. University of Massachusetts Amherst, 2007. `http://ciir.cs.umass.edu/cgi-bin/downloads/downloads.cgi`.

[10] S. E. Middleton, N. R. Shadbolt, and D. C. D. Roure. Ontological User Profiling in Recommender Systems. *ACM Transactions on Information Systems*, 22(1):54–88, 2004.

[11] G. Salton and C. Buckley. Term Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[12] A. Singhal, G. Salton, M. Mitra, and C. Buckley. Document Length Normalization. *Information Processing and Management*, 32(5):619–633, 1996.