News Personalization using the CF-IDF Semantic Recommender

Frank Goossen
frank.goossen@xs4all.nl

Wouter IJntema
wouterijntema@gmail.com

Flavius Frasincar
frasincar@ese.eur.nl

Frederik Hogenboom
fhogenboom@ese.eur.nl

Uzay Kaymak
kaymak@ese.eur.nl

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

ABSTRACT

When recommending news items, most of the traditional algorithms are based on TF-IDF, i.e., a term-based weighting method which is mostly used in information retrieval and text mining. However, many new technologies have been made available since the introduction of TF-IDF. This paper proposes a new method for recommending news items based on TF-IDF and a domain ontology. It is demonstrated that adapting TF-IDF with the semantics of a domain ontology, resulting in Concept Frequency - Inverse Document Frequency (CF-IDF), yields better results than using the original TF-IDF method. CF-IDF is built and tested in Athena, a recommender extension to the Hermes news personalization framework. Athena employs a user profile to store concepts or terms found in news items browsed by the user. The framework recommends new articles to the user using a traditional TF-IDF recommender and the CF-IDF recommender. A statistical evaluation of both methods shows that the use of an ontology significantly improves the performance of a traditional recommender.

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

Algorithms, Design, Performance

Keywords

Content-based recommender, News personalization, Ontology, Recommender systems, Semantic Web, User profiling

1. INTRODUCTION

There are three basic types of recommendation systems: content-based recommenders, which recommend news items based on their content, collaborative filtering recommenders, which recommend news items by means of user similarity, and hybrid recommenders, that combine the previous two approaches. In this paper we focus on content-based recommenders. We analyze two types of content-based recommenders: traditional, which are term-based, and semantic, which are concept-based.

TF-IDF [32] is a well-known method for assigning an importance weight to a term in a document. Combined with the vector space model [33], TF-IDF can be used to recommend news items to a specific user. When employing user profiles that describe users' interest based on the previously browsed items, these can be translated into vectors of TF-IDF weights. With a measure like cosine similarity, one can calculate how interesting a new item might be based on user profiles. For this, TF-IDF weights are computed on every term within a document. Since the last decade, methods have been developed to find key concepts in a text. A framework which implements this kind of methods is the news personalization framework called Hermes [10, 17, 34], which uses an ontology to store concepts and their relations.

This paper proposes a new method for recommending news items, i.e., CF-IDF (Concept Frequency - Inverse Document Frequency) weighting. This method is based on TF-IDF, but instead of using all the terms of a text, this method only looks at the key concepts found in this text. In order to test this new method, we implement it in Athena [21], which is an extension to the Hermes framework.

Athena is able to observe user browsing behavior and generate recommendations based on this behavior. In order to recommend news items, first the user's browsing behavior is modeled. 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' (unread) news items that are possibly interesting to the user. Athena already implements several recommendation algorithms using various similarity measures: TF-IDF weights combined with cosine similarity, concept equivalence similarity, binary cosine similarity, Jaccard coefficient, concept neighborhood, and ranked semantic similarity [21]. However, in our current endeavors we solely focus on the existing TF-IDF recommender and the newly created CF-IDF recommender. Using the latter two recommenders we are able to compare 'new' news items with user profiles. The news items that have the highest similarity with the user profile are recommended to the user.

The main contribution of this paper is twofold. Firstly, this paper proposes a new method for recommending news items, i.e., CF-IDF weighting. Secondly, we present a comparison of the performance with the TF-IDF traditional recommender through evaluation of the results of our implementation, Athena, i.e., an extension to the Hermes framework.

The structure of this paper is as follows. Section 2 presents related work. Subsequently, Sect. 3 elaborates on the Hermes framework and its implementation, the Hermes News Portal (HNP). Next, the Athena framework and its traditional and semantic recommendation algorithms are discussed in Sect. 4. Then, Sect. 5 gives an overview of our implementation of Athena in HNP. Section 6 presents the results of our evaluation, and last, conclusions and future work directions are presented in Sect. 7.

2. RELATED WORK

This paper focuses primarily on the semantic extension of a TF-IDF recommendation approach. First we introduce different term/concept weighting methods and then we show how these have been applied in existing recommender systems.

2.1 TF-IDF

There are many term weighting methods available, such as for example probabilistic weighting, term frequency (TF) weighting, inverse document frequency (IDF) weighting, TF-IDF weighting, variations of TF-IDF weighting, etc. [32]. The main term weighting method that is focused on specifically in the work presented in this paper is the traditional TF-IDF weighting scheme. A classic approach in comparing documents is the use of TF-IDF together 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.

Before calculating the TF-IDF values, the stop words are being filtered from the document. After stop word removal, the remaining words are stemmed by a stemmer. There are multiple stemmers available like the Krovetz [23], Lovins [24] and the Porter stemmer [29]. A stemmer reduces words back to their root word, for example the words 'processor' and 'processing' are reduced to '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 TF-IDF weight is the multiplication of TF and IDF.

2.2 CF-IDF

There has been some previous work on the use of TF-IDF with concepts (similar to CF-IDF). In [5] a conceptual indexing method based on WordNet [16], a large lexical database, is proposed. This approach represents document contents by the semantic network [37] called document semantic core. The documents are mapped on the WordNet semantic network and converted from a set of terms to a set of concepts. After that, the extracted concepts are weighted like in the classical index term case, using the weighting schema's TF-IDF and Okapi BM25 [31]. This method differs from ours in the detection of concepts. It does not take into account synonyms and it lacks a word sense disambiguation procedure present in our method [10]. Furthermore we do a more thorough comparison with TF-IDF as we perform an extensive evaluation including the Student t-test, the Area Under the Curve (AUC) and Cohen's Kappa coefficient. Yan and Li [38] propose a Word Sense Disambiguation (WSD) [1] method called Term Co-Occurrence Graph (TCOG), which uses WordNet to create a text representation model. In order to represent a text, a set of WordNet concepts with a CF-IDF weight is used. This idea is extended by taking into account also higher level concepts (e.g., concept 'boat' has as higher level concepts 'vessel' and 'vehicle'). The authors compare their method against TF-IDF and an adapted Lesk algorithm [4]. The main difference between our approach and TCOG is the purpose of the research. While we focus on a recommendation system, TCOG is meant for text classification with respect to topics. Furthermore, as is the case with [5], we perform a more thorough evaluation.

2.3 Content-Based Recommenders

In content-based approaches to news recommending, articles are recommended according to a comparison between their contents and the user profiles. The user profiles contain information about the users' content-based preferences. Both of these components have data-structures which are created using features extracted from text. A weighting scheme is often used to assign high weights to the most discriminating features/preferences, and low weights to the less informative ones.

2.3.1 Traditional Content-Based Approaches

Traditional content-based approaches are purely content-based without any semantics. Concepts get weights assigned that are obtained without semantic knowledge of underlying relations between the concepts. User interests are often measured with machine learning algorithms, like Nearest Neighbor or Naive Bayes.

In the traditional content-based approaches we review, articles are processed with TF-IDF by taking all terms (but the stop words) into account. The article is stored in a weighted vector of terms, and compared with a user profile by using a similarity measure. The main difference between the related approaches and the method proposed in this paper, CF-IDF, is the way how we represent an article. CF-IDF considers a news item as a weighted vector of key concepts instead of terms. This makes it a more 'intelligent' recommender: since it already knows the most important terms in the document, there are no 'noise' terms which can pollute the outcome. The similarity measure used for comparing the article with the user profile is the cosine similarity.

News Dude [7] is a personal news recommending agent that uses TF-IDF in combination with the Nearest Neighbor algorithm and uses the full text of an article. News Dude first considers the short-term interests to look for similar items and if this does not return satisfiable results, long-term interests are considered.

The next related work is Daily Learner [8]. This is an adaptive news service which allows users to personalize the news to their own taste. First a user gives his preferences of what type of news he is interested in. Based on this user profile, the system then delivers those stories that best match this user's interests. A new article is processed with TF-IDF, and represented as a vector. Then this article is compared with the user profile (also a vector with TF-IDF weights), using cosine similarity. Finally, the user explicitly provides feedback using four ratings (interesting, not interesting, more information, already known). Short-term interests are determined by analyzing the N most recently rated stories, based on the Nearest Neighbor Algorithm. Long-term interests are modeled with the Naive Bayes Classifier.

YourNews [2] is another example of a content-based news recommendation system. It is a personalized news system, which intends to increase the transparency of adapted news delivery. It allows the user to view and edit his interest profile. To support this, YourNews highlights the key terms in news items. The news items are represented as weighted vectors of terms. The weight of each term is calculated using TF-IDF. Before creating those vectors, the text is filtered from stop words and each word is reduced to its stem using a Krovetz Stemmer [23]. The user profile is represented as a weighted vector of terms extracted from the user's view history and similarities between user profiles and news articles are computed using the cosine similarity measure.

Personalized Recommender System [26] (PRES) is a news personalization system that applies content-based filtering. PRES also uses the combination of TF-IDF and the cosine similarity measure. Every time a new news item is browsed, the system updates existing weights assigned to terms using a certain diminishing factor. This way PRES aims to keep the interests up-to-date, allowing changes over time. The diminishing factor is determined via experimentation.

Traditional TF-IDF recommending approaches consider the full text of the news articles. However, as the authors of [9] made a comparison with different lengths of documents, the performance decreases as documents get larger. CF-IDF does not consider the full text, but only the concepts that exist in the knowledge base. With the semantic knowledge about the concepts it is possible to consider more than just the text at hand. The strength of the CF-IDF algorithm depends on the quality of the knowledge base.

2.3.2 Semantic Content-Based Approaches

Semantic content-based approaches aim to recommend news items by combining content-based techniques with domain semantics. Weights for concepts take into account the semantic knowledge about these concepts. Each of the reviewed recommenders has a different approach of applying the semantic knowledge provided by the ontology. The CF-IDF recommender only records the concepts to calculate weights. The approach proposed in [21], which was created in the same environment as our CF-IDF approach, calculates a similarity based on not only the concepts themselves but also based on the directly and indirectly related concepts, which are described in an ontology.

OntoSeek [20] is a content-based approach which aims to retrieve information from online yellow pages and product catalogs. It matches content with the help of the large Sensus [22] ontology, which comprises a simple taxonomic structure of approximately 70,000 nodes. OntoSeek does not employ a user profile. Instead, OntoSeek uses lexical conceptual graphs to represent queries and resource descriptions, i.e., a tree structure where nodes are nouns from the descriptions and arcs are concepts inferred by the corresponding nouns. The ontology is used for classifying items, and to match an item with a query. The user is required to disambiguate the meaning of his queries. This process is performed by the user interface that tries to identify the concept provided and asks the user to choose between potential solutions.

Quickstep [27] is one of two proposed recommendation approaches [28] for online academic publications where user profiling is based on an external research paper topic ontology. The papers are represented using term vectors. All the terms in the text are considered and stemmed using the Porter stemmer [29]. After this processing, the term vector weights are computed using the term frequency weighting method (TF). The classification of papers is done using a k-Nearest Neighbor type classifier and a boosting algorithm. The user profile is created automatically and real-time, based on the vector representations of papers downloaded by a user. Finally, Quickstep generates recommendations by calculating the correlation (recommendation confidence) between the users' current field of interest and the papers which are classified to be in this field of interest. Recommendations are presented to the user sorted by the recommendation confidence. Similar to Quickstep, our approach CF-IDF is maintaining the user profile by observing the user real-time, providing an up-to-date profile. Another similarity with Quickstep is the use of the vector space model to compare news items and the user profile. The most important difference between our approach and Quickstep is the essence of our approach: the use of key concepts for representing news items.

The authors of [12] propose News@hand, a news-based recommendation system which uses Semantic Web technologies to describe and relate news items and user preferences in order to recommend items to a user. To represent news contents and user preferences the authors make use of concepts which appear in a set of domain ontologies. News@hand looks very similar to the Hermes News Personalization framework. Both approaches classify news items to gain key concepts, and work with a domain ontology. For recommending, News@hand makes use of 3 different semantic methods for recommendations: content-based, collaborative filtering, and a hybrid approach [11, 13]. The latter two are not discussed since this paper focuses primarily on content-based approaches. The semantic content-based recommendation approach employs a certain similarity measure that utilizes the semantic preferences (weighted concepts gained by observing and profiling user behavior) of the user and the semantic annotations (the key concepts weighted by the classification) of an item. In our approach we follow similar procedures, i.e., we create a vector of the user profile by computing the CF-IDF weights of all distinct concepts found in all read news items. Subsequently, we create a vector of CF-IDF weights belonging to the concepts found in a 'new' news item and we compare the two vectors using cosine similarity. The main difference between the semantic content-based approaches and our approach is the aim of the approach itself. CF-IDF is mainly created to proof that a term-based recommender can be significantly improved with the help of the semantic annotations, whereas the content-based approach in News@hand is mainly used for comparison with the hybrid recommendation approach.

3. HERMES NEWS PERSONALIZATION

Athena [21] is an existing extension to the Hermes news personalization framework [10, 17, 34], which allows for experimenting with different content-based recommenders. The Hermes News Portal (HNP) is an implementation of the Hermes framework, and Athena has been implemented as a plug-in for the HNP. The Hermes framework is utilized for building a news personalization service, and we can be structure its core functionalities in three main processes, i.e., input, internal processing, and output. The framework's input consists of user-defined RSS feeds of news items. Subsequently, the processing takes place, in which news items from the RSS feeds are classified using concepts from an internal knowledge base. Finally, the output is the set of personalized news items based on the selected concepts.

3.1 Hermes Domain Ontologies

The Hermes news personalization framework provides a semantic-based approach for retrieving news items related, directly or indirectly, to the concepts of interest from a domain ontology. Hermes stores its knowledge base and the list of classified news items in separate ontologies. The knowledge base describes the general domain the user is interested in. It is used for several functionalities: the classification of 'new' news items and the graphical representation used for selecting concepts of interest, i.e., the search graph, which is used in the HNP. The knowledge base is maintained using discovered information in news items [34].

3.2 Classification

The classification of news items is done after loading news from RSS feeds. Hermes employs an advanced Natural Language Processing (NLP) engine that uses techniques like tokenization, part-of-speech tagging, word sense disambiguation, gazetteering, etc. This prepares the news item for querying. During the classification, the system searches through the news item to find all knowledge base concepts. The news item is then stored with annotations in a news ontology, so that future queries can be done in a fast and direct way, without going through the NLP steps again. All concepts of interest are stored in the domain ontology (knowledge base) and are represented by ontology concepts. An ontology concept is a class or instance from the domain ontology. Each concept has a lexical representation derived from a semantic lexicon like WordNet [16].

3.3 News Querying

News querying is done by expressing the topics of interest using concepts from the domain ontology. The user can select these concepts with the help of the ontology graph. The user has the additional possibility to express constraints that the timestamps belonging to news items need to satisfy. The news querying step consists of two parts: query formulation, i.e., supporting the user to build queries, which uses the ontology graph, and query execution, i.e., computing the results of query evaluation. The results returned from the query are presented in the order of their relevance for the user query. For this purpose, for each returned news item a relevance degree is computed based on all the hits between the news item and the query concepts.

3.4 Hermes News Portal

The Hermes News Portal (HNP) is the implementation of the utilized Hermes framework, which allows the user to query the news and view the knowledge base. The domain ontology is represented in OWL [6] and news querying is done by means of a SPARQL [30] variant, tSPARQL [17], which extends SPARQL by offering a wider range of time-related functionalities. The domain ontology graph is maintained using SPARQL Update [35]. Finally, classification of news items is done using GATE [15] and the WordNet [16] semantic lexicon.

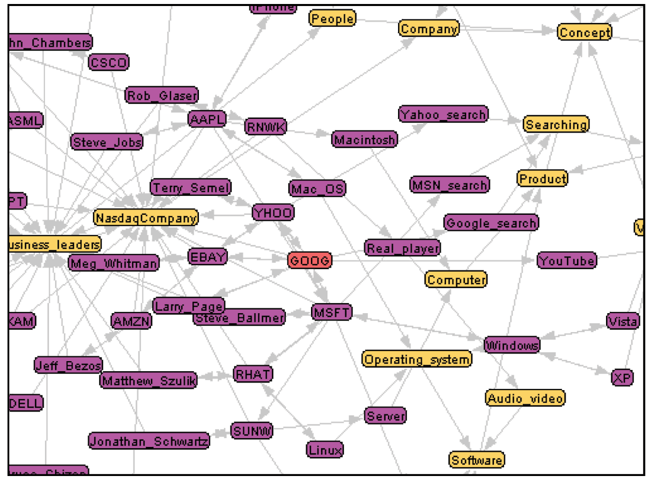

Figure 1: Example conceptual graph in HNP

In [34] the quality of semantic annotations was investigated. The obtained precision and recall for domain ontology concept identification were 86% and 81% respectively. This precision is measured as the number of concepts correctly classified in the news items divided by the total number of concepts classified in the news items. The recall is defined as the number of concepts correctly classified in the news items divided by the total number of concepts that is present in the news items according to domain experts.

The programming language Java [19] has been chosen since many libraries for manipulating, reasoning with, querying, and visualizing ontologies are available. Also, it combines well with GATE, which we use for most of the Natural Language Processing, since both GATE and its components are programmed in Java. Jena [25] is used for manipulating and reasoning with ontologies. It allows users to specify queries for the concepts of interest and temporal constraints, and retrieve the corresponding news items.

The user can do a search query with the help of the graphical tab of the HNP which is shown in Fig. 1. This is the graphical interpretation of the knowledge base. When the user commits its search query, the HNP translates this query into a SPARQL query. This SPARQL query will return only those news items which are most related to the users' concepts of interest. Finally, the HNP will present the most relevant news items to the user. The HNP includes the possibility to add plug-ins, an example of an existing plug-in is a tab which allows loading news items from an RSS feed into the news ontology. A plug-in can be built as a new tab into the HNP. Athena is an example of an existing plug-in.

4. ATHENA: THE FRAMEWORK

Athena [21] is a recommender framework which builds on the Hermes news personalization platform. Athena allows for news recommendation to the users based on user behavior profiles. The recommendations for news items are made in three steps. Firstly, a user profile is constructed based on the articles the user has read. Secondly, the user profile and the articles need to be represented in a uniform way, and for this purpose the vector space model [33] is used. Finally, the similarity between the user profile and a new article is computed.

Athena already implements several recommendation algorithms. One of these recommenders is the TF-IDF recommender. However, we will review this recommender again briefly in order to generate a better understanding of our CF-IDF recommender. The CF-IDF algorithm is a new algorithm implemented in Athena. The other several recommendation algorithms discussed in [21] will not be used in this research, as they are outside the scope of this paper.

4.1 User Profile Construction

Recommending news items starts with building a user profile. Building a user profile can be defined as 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 can be constructed in different ways. The TF-IDF recommender analyses every term (but the stop words) in a news item, so the profile the TF-IDF recommender needs consists of a list of news items which it can process. The CF-IDF recommender uses the same way of gathering information from a user profile. The main difference is that this recommender does not take all the text in a news item into account, but only the domain concepts found in it.

4.2 Vector Space Model

Each semantic recommender in Athena utilizes vectors of concepts and weights. Athena uses a domain ontology that is also employed within the Hermes Framework. This ontology contains a set of i concepts and their relations, i.e.,

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

Subsequently, the user profile consists of j concepts, which can be represented as

$$U = \left\{c^u_1, c^u_2, c^u_3, \cdots, c^u_j\right\}, \text{where } c^u \in C \ . \tag{2}$$

Here, the concept c^u is associated with k news articles a_i in which it is found. The concept in the user profile (which stores all browsed news), c^u, can be represented as

$$c^u = \{a_1, a_2, a_3, \cdots, a_k\} \ . \tag{3}$$

The semantic recommenders in Athena consider an article to be a set of l elements representing the number of concepts c that appear in the article a:

$$a_{Semantic} = \{c^a_1, c^a_2, c^a_3, \cdots, c^a_l\}, \text{where } c^a \in C \ . \tag{4}$$

The TF-IDF recommender has a different interpretation of a news article, as it considers the article a to be a set of m terms t, and hence

$$a_{TF-IDF} = \{t^a_1, t^a_2, t^a_3, \cdots, t^a_m\} \ . \tag{5}$$

Each recommender in Athena has its own interpretation of the vector space model. The TF-IDF recommender takes into account all articles read by the user. Then the TF-IDF recommender calculates a TF-IDF weight for each distinct term t^a_i in all articles. The resulting TF-IDF weight w^u for each term is stored in vector V^u_{TF-IDF}. This vector, which is defined as

$$V^u_{TF-IDF} = [\langle t^u_1, w^u_1\rangle, \cdots, \langle t^u_m, w^u_m\rangle] \ , \tag{6}$$

will be used as the user profile. The CF-IDF recommender receives the set of j concepts U from the user profile. Then the CF-IDF recommender calculates a CF-IDF weight w^u for all distinct concepts in U, and stores these weights in vector V^u_{CF-IDF}:

$$V^u_{CF-IDF} = [\langle c^u_1, w^u_1\rangle, \cdots, \langle c^u_j, w^u_j\rangle] \ . \tag{7}$$

When the TF-IDF recommender starts recommending, it compares a new (unread) article a with the user profile. Then it creates a new TF-IDF vector V^a_{TF-IDF} for this article, containing a TF-IDF weight w^u for all m terms t^u:

$$V^a_{TF-IDF} = [\langle t^u_1, w^u_1\rangle, \cdots, \langle t^u_m, w^u_m\rangle] \ . \tag{8}$$

The TF-IDF recommender then compares the vector V^a_{TF-IDF} with the user profile vector V^u_{TF-IDF}. This comparison is done using the cosine similarity measure.

The CF-IDF recommender does the weighting in a similar way. It creates a CF-IDF weight vector V^a_{CF-IDF} with a CF-IDF weight w^u for each term in article a. The weight vector is defined as

$$V^a_{CF-IDF} = [\langle c^u_1, w^u_1\rangle, \cdots, \langle c^u_l, w^u_l\rangle] \ , \tag{9}$$

and contains weights for all l concepts c in article a. The CF-IDF recommender subsequently measures the similarity between the vector V^a_{CF-IDF} and the vector V^u_{CF-IDF}. Just like the TF-IDF recommender, this comparison is done using the cosine similarity measure.

4.3 TF-IDF

Although the TF-IDF technique is well-known, the basics of calculating the weight are briefly reviewed for better understanding of the modifications made for the CF-IDF technique.

The *Term Frequency* is the occurrence of a term t_i in a document d_j, $n_{i,j}$, divided by the total number of occurrences of all terms in document d_i, $n_{k,j}$. Hence,

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \ . \tag{10}$$

The *Inverse Document Frequency* can be calculated by taking the total number of documents $|D|$, and dividing it by the number of documents in which the term t_i appears, then taking the logarithm of this division. Thus, the Inverse Document Frequency can be defined as

$$idf_i = \log \frac{|D|}{|\{d : t_i \in d\}|} \ . \tag{11}$$

Finally, *tf* is multiplied with *idf*, forming the weight for term t_i of document d_j, i.e.,

$$tf\text{-}idf_{i,j} = \mathrm{tf_{i,j}} \times \mathrm{idf_i} \ . \tag{12}$$

4.4 CF-IDF

Essentially, the CF-IDF recommender is based on one modification of the TF-IDF recommender. The CF-IDF recommender primarily uses a vector for each item, and calculates a CF-IDF weight for each concept, instead of going through all the terms. Then, in the same way as the TF-IDF recommender, it stores the calculated weights (together with the corresponding terms) of a news item in a vector. The user profile is also a vector of CF-IDF weights, which can be compared with a news item CF-IDF vector by using cosine similarity. Below we describe how to compute the CF-IDF weights.

First we calculate the *Concept Frequency*, $cf_{i,j}$, which is the occurrence of a concept c_i in document d_j, $n_{i,j}$, divided by the total number of occurrences of all concepts in document d_i, $n_{k,j}$:

$$cf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \; . \tag{13}$$

Subsequently, we calculate the *Inverse Document Frequency*. We take the total number of documents, $|D|$, and divide it by the number of documents in which the concept c_i appears, then taking the logarithm of this division, i.e.,

$$idf_i = \log \frac{|D|}{|\{d : c_i \in d\}|} \; . \tag{14}$$

Finally, cf is multiplied with idf, forming the weight for concept c_i of document d_j. Hence,

$$cf\text{-}idf_{i,j} = \mathrm{cf_{i,j}} \times \mathrm{idf_i} \; . \tag{15}$$

This change causes the recommender to deal with only 'important' terms, making it more effective. Another advantage of this method is that the CF-IDF recommender can process the news items much faster than the TF-IDF recommender. The traditional TF-IDF processes all words (but the stop words) in the text. The CF-IDF recommender uses only the concepts found in the text.

The goal of this research is to find whether a CF-IDF based recommender performs better than a TF-IDF based recommender. Intuitively, one should expect that the CF-IDF recommender will be more effective, working only with the main concepts without all surrounding noisy terms. This will be investigated and statistically tested in Sect. 6.

5. ATHENA: THE IMPLEMENTATION

As Athena is an extension to the Hermes framework, it has been implemented as a plug-in in 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.

5.1 The Athena Plug-in

The user interface of Athena consists of three tabs: a browser for all news items, a tab for the recommendations (as depicted in Fig. 2), and a tab for evaluation purposes. The browser contains a number of news items sorted by date. This allow the user to browse freely through the news items, instead of browsing through query results as in the Hermes News Portal. Each item is presented with a title, abstract, and the date published of a news item.

Athena provides an expandable platform for testing recommendation algorithms. As the user browses through items, these items are stored in the user profile. Each recommender works with the same profile, only the way of interpretation is different. It is also possible to add new recommendation algorithms to Athena. In this case, the CF-IDF recommender has been added to the existing recommenders. In this paper, we will only test the TF-IDF and the CF-IDF recommenders.

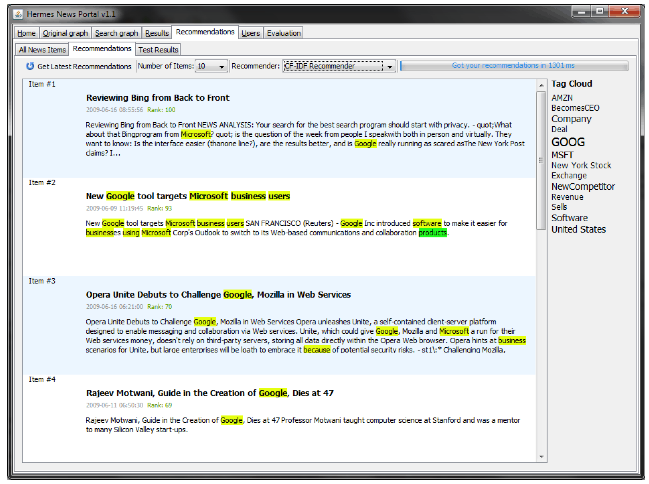

Figure 2: Recommendations in Athena

Athena also provides an evaluation environment, which is shown in Fig. 3. This environment is used to load 'test user profiles' for statistics of the performance of recommenders, which is explained in more details in Sect. 6.

5.2 User Profile Construction

User profiles are created from news articles read by the individual users. We consider articles to be read whenever they have been opened into the Web browser. Athena shows the user a list of news items, which can be clicked. When the user clicks on an article, this article opens in the Web browser. At this moment, Athena registers this item in the user profile.

The Hermes News Portal stores the news articles in a separate ontology. In this ontology each individual has a unique identifier, i.e., a Uniform Resource Identifier (URI). In the Hermes News Portal, this identifier is a hexadecimal repre-

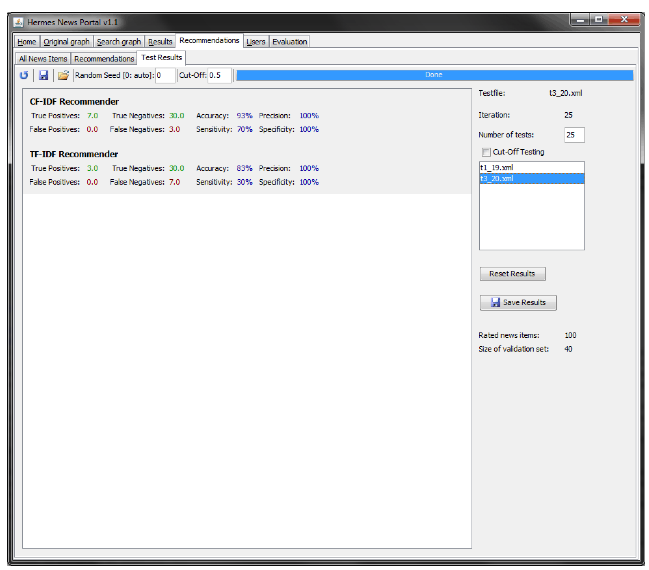

Figure 3: Testing environment provided by Athena

sentation of the characters in the title and publishing date of an article. This unique identifier enables us to store a minimal amount of information that identifies the read articles. Storing only the URI makes it possible to lookup any information about the article, i.e., the title, date, and content (abstract). Besides a minimal need in storage capacity, the user profile also increases the flexibility of the system, because each recommender needs different information.

As explained in Sect. 4, the profile is a set of concepts from the articles the user has read. A concept is given by the Hermes knowledge base. Each news article contains zero or more concepts. The user's interest can therefore be determined from the visited news items. Therefore the user profile consists of concepts from read articles.

The user profile is stored in OWL format, because of OWL's flexibility and clear structure. An example of a user profile used in Athena is shown in Fig. 4. The TF-IDF and the CF-IDF recommender make use of the URIs of previously browsed news items. The TF-IDF recommender computes the weights for all the words (but stop words) appearing in a news item, while the CF-IDF recommender focuses on concepts.

5.3 TF-IDF Implementation: Stemmer

The TF-IDF recommender uses a Java implementation of the Krovetz [23] stemmer. This stemmer first removes the suffixes from the words, and then (by checking a dictionary for any recordings) returns the stem to a word. The removing of the inflectional suffixes takes place in 3 steps. First it transforms plurals to their single form (e.g. '-ies', '-es' and '-s'). Then it transforms past to present tense (e.g. 'ed') and finally it removes the 'ing' parts from words.

6. EVALUATION

In order to evaluate the performance of the ontology-based CF-IDF recommender compared to the term-based TF-IDF recommender, we construct a test method and test environment. The results are evaluated by means of hypothesis testing, Receiver Operating Characteristic (ROC) and Precision-Recall (PR) curves, and the Kappa statistic [14].

6.1 Experimental Setup

For the test environment, we develop a Web site where 100 news items are displayed one at a time. For each article the user has to indicate whether it is interesting or not. The experiment requires the user to keep a clear profile in mind, which is *all articles that are related to Microsoft, and its products and competitors*. After the user finishes the test, the resulting set of articles is created, consisting of URI's of the 100 articles and their rating ('interesting' or 'uninteresting'). Athena is then able to process this result set.

The group of participants consists of 19 students (age ranging from 19 to 23, as shown in Fig. 5) in the field of Informatics and Econometrics. All students are familiar with the area related to the profile used in the test. From the 19 participants, 18 are male and 1 is female.

The processing of the result set is based on supervised learning. The result set is randomly split into two different sets, the training set (60%) and the test set (40%). The two sets are filled with a relatively equal number of items rated as interesting. The training set is used to create a user profile. Each item that is marked as interesting will be

```
<?xml version="1.0?>
<!DOCTYPE rdf:RDF [
   <!ENTITY up "http://www.hermes.com/up.owl#" >
   <!ENTITY news "http://www.hermes.com/news.owl#" >
   <!ENTITY kb "http://www.hermes.com/kb.owl#" >
   <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
   <!ENTITY owl2xml "http://www.w3.org/2006/12/owl2-xml#" >
   <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
   <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
   <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
]>
<rdf:RDF xmlns="http://www.hermes.com/up.owl#"
     xml:base="http://www.hermes.com/up.owl"
     xmlns:up="http://www.hermes.com/up.owl#"
     xmlns:news="http://www.hermes.com/news.owl#"
     xmlns:kb="http://www.hermes.com/kb.owl#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#">

   <owl:ObjectProperty rdf:about="#foundIn">
      <rdfs:domain rdf:resource="#Concept"/>
      <rdfs:range rdf:resource="&rdfs;Resource"/>
   </owl:ObjectProperty>

   <owl:ObjectProperty rdf:about="#inNews">
      <rdfs:domain rdf:resource="&rdfs;Resource"/>
      <rdfs:range rdf:resource="#NewsItem"/>
   </owl:ObjectProperty>

   <owl:DatatypeProperty rdf:about="#hits">
      <rdfs:domain rdf:resource="&rdfs;Resource"/>
      <rdfs:range rdf:resource="&rdfs;Literal"/>
   </owl:DatatypeProperty>

   <owl:Class rdf:about="#Concept">
      <rdfs:subClassOf rdf:resource="&owl;Thing"/>
   </owl:Class>

   <owl:Class rdf:about="#NewsItem">
      <rdfs:subClassOf rdf:resource="&owl;Thing"/>
   </owl:Class>

   <Concept rdf:about="&kb;Company">
      <foundIn rdf:parseType="Resource">
         <inNews rdf:resource="&news;fb1dad00"/>
         <hits>3</hits>
      </foundIn>
      <foundIn rdf:parseType="Resource">
         <inNews rdf:resource="&news;ff93d1d90"/>
         <hits>1</hits>
      </foundIn>
   </Concept>

   <Concept rdf:about="&kb;Networking">
      <foundIn rdf:parseType="Resource">
         <inNews rdf:resource="&news;fa3b5f68"/>
         <hits>2</hits>
      </foundIn>
      <foundIn rdf:parseType="Resource">
         <inNews rdf:resource="&news;ff93d1d90"/>
         <hits>1</hits>
      </foundIn>
   </Concept>

   <NewsItem rdf:about="&news;fa3b5f68"/>
   <NewsItem rdf:about="&news;fb1dad00"/>
   <NewsItem rdf:about="&news;ff93d1d90"/>
</rdf:RDF>
```

Figure 4: Example of a user profile

added to this profile. The test set is then used by both recommenders to determine for each news item the similarity with the user profile. An article is considered to be interesting if the similarity is higher than a predefined cutoff value,

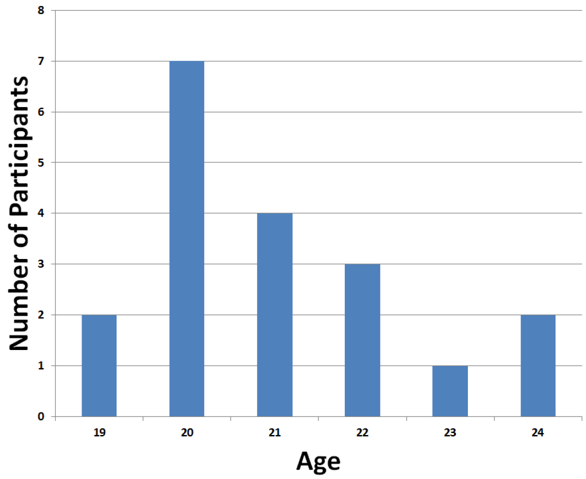

Figure 5: Age distribution of participants

otherwise it is classified as not interesting. The effect of the cutoff value can be seen in Fig. 6, where the F_1-measure is plotted against a varying cutoff value. We have chosen to use several cutoff values to obtain a objective overview of the performances. The cutoff values for testing used are 0.4, 0.5 and 0.6.

In order to be able to determine the performance of a recommender, several measures are calculated using a confusion matrix, as shown in Table 1. This confusion matrix can be used to compute several statistical measures. The evaluation focuses on the following measures: *accuracy*, *precision*, *recall*, *specificity* and the (traditional) F_1*-measure*. The definitions of these measures can be found in [3]. With these measures we are able to gather sophisticated information about the two recommendation methods. For both methods we can determine whether they perform good or bad with respect to a certain measure.

One run might yield unreliable results, because of the random value for splitting the result set. Therefore, Athena performs the testing process for 100 iterations with different random values for splitting the result set, and calculates the average performance of the recommenders.

6.2 Experiment

Our conducted experiment aims to investigate whether the ontology-based TF-IDF recommendation approach performs statistically better than the term-based TF-IDF approach. For each participant we gather the relevant news items. After Athena processes the news items, the resulting performance measures are saved. This data is used for statistical purposes.

This experiment includes hypothesis testing, ROC curves, precision recall graphs and the Kappa statistic. Only the same performance measures are compared. One sample consists of 19 values, each representing the average measured performance of a recommender on the result set of one user. In the test, the average of this sample (the average of 100 testing iterations) is taken as mean performance. We compare the means as follows:

$$\begin{aligned} \mu_1&: \text{Mean performance CF-IDF recommender} , \\ \mu_2&: \text{Mean performance TF-IDF recommender} . \end{aligned} \qquad (16)$$

Here, performance is performance defined as accuracy, precision, recall, or specificity. With a level of significance of 95%, the corresponding hypotheses are as follows. First the two-sided hypothesis:

$$H_{0_a} : \mu_1 = \mu_2, H_{1_a} : \mu_1 \neq \mu_2, \text{with } \alpha = 0.05 . \qquad (17)$$

And, in case we can reject H_{0_a}, the one-sided hypothesis is denoted as:

$$H_{0_b} : \mu_1 < \mu_2, H_{1_b} : \mu_1 > \mu_2, \text{with } \alpha = 0.05 . \qquad (18)$$

Because there are two related samples, we employ the paired Student's t-test for significance evaluation.

6.3 Experimental Results: Averages

After processing the results of the 19 participants, we are able to create an overview on how the recommenders performed. Table 2 shows all the average performances of both recommenders. The averages indicate that CF-IDF seems to outperform TF-IDF slightly on accuracy and precision. Note that a recommender performing better on average does not necessarily perform better in a t-test. The difference between CF-IDF and TF-IDF regarding recall and the F_1-measure is exceptionally large. A high recall means that according to the averages CF-IDF performs better in classifying all interesting items as interesting. The F_1-measure is positively related with the recall. Based on these results we can conclude that the CF-IDF recommender performs notably better on accuracy, precision, recall and the F_1-measure. Whether these findings are significant will be further investigated using the Student t-test in Sect. 6.4.

6.4 Experimental Results: Significance

In order to be able to evaluate the significance of the results presented in Sect. 6.3, we have to determine the p-values by means of the Student t-test. The decision rule for the first test (two-sided) becomes: *Reject* H_{0_a} *if* $p_a < 0.05$. The decision rule for the second (one-sided) test becomes: *Reject* H_{0_b} *if* $p_b < 0.05$.

All the p-values calculated are listed in Table 3. Values which reject both H_{0_a} and H_{0_b} are printed bold. Values which do not reject H_{0_a} are printed in a normal font.

As can be seen in Table 3, the CF-IDF recommender performs statistically better than the TF-IDF recommender on accuracy, recall and the F_1-measure. Both recommenders are statistically similar regarding the precision and specificity.

The higher accuracy means that the CF-IDF recommender is performing better in classifying items correctly, both interesting as uninteresting items. Also on recall, the number of interesting news items being classified as interesting, the CF-IDF recommender performs significantly better. Although CF-IDF performs not significantly different than TF-IDF on precision, the precision is relatively high compared to the recall. Recall and precision are usually negatively related.

Table 1: Confusion Matrix

		Actual	
		Interesting	Not Interesting
Predicted	Interesting	TP	FP
	Not Interesting	FN	TN

Table 2: Average test results for various cutoff values

	Cutoff 0.4			Cutoff 0.5			Cutoff 0.6		
Measure	CF-IDF	TF-IDF	Diff.	CF-IDF	TF-IDF	Diff.	CF-IDF	TF-IDF	Diff.
Accuracy	87.4%	83.2%	4.2%	85.9%	81.2%	4.7%	83.5%	79.7%	3.8%
Precision	83.1%	79.6%	3.5%	83.4%	82.5%	0.9%	86.0%	83.7%	2.3%
Recall	72.1%	48.1%	14.0%	59.9%	35.5%	24.4%	47.4%	25.6%	21.8%
Specificity	93.5%	96.7%	-3.2%	96.0%	98.0%	-2.0%	97.0%	98.6%	-1.6%
F_1-measure	75.5%	56.4%	19.1%	68.0%	46.1%	21.9%	58.5%	37.1%	21.4%

Table 3: Significance of results for various cutoff values

	Cutoff 0.4		Cutoff 0.5		Cutoff 0.6	
Measure	p (two-tailed)[1]	p (one-tailed)	p (two-tailed)	p (one-tailed)	p (two-tailed)	p (one-tailed)
Accuracy	**0.018**	**0.009**	**0.004**	**0.002**	**0.012**	**0.006**
Precision	0.166	0.083	0.804	0.402	0.576	0.288
Recall	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
Specificity	0.168	0.084	0.107	0.054	0.098	0.049
F_1-measure	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**

[1] A two-tailed test is harder to pass then a one-tailed test.

This is also shown in Fig. 8 where the precision and recall are plotted against each other. It is up to the end-user to determine what is more important, a high precision, with a clean result with a low number of uninteresting items, or a high recall, with the insurance that every interesting item will be in the result set, at the cost of more uninteresting items in the result set.

The specificity is not significantly different for the recommenders. As shown in Table 2, the measure scores very high for both recommenders, which means that both recommenders classify most of the uninteresting items as uninteresting. However, recall (sensitivity) is relatively low, corresponding to the number of interesting items classified as interesting. This means that both recommenders do succeed in getting a relatively clean result, but still need improvement in classifying interesting items as interesting. The recall and specificity are both strongly influenced by the cutoff value.

As shown in Table 3, CF-IDF performs significantly better than TF-IDF on the F_1-measure. The F_1 measure indicates the performance of a recommender regarding balance between recall and precision. Figure 6 depicts the combined curve of the F_1-measure for both recommenders, as the cutoff value was varied from 0.000 to 1.000. This figure shows that overall, the CF-IDF recommender scores higher than the TF-IDF recommender. The standard cutoff value used for testing is set to 0.4, where CF-IDF has the highest F_1-score.

6.5 Experimental Results: ROC - PR Curves

When evaluating the Receiver Operating Characteristic (ROC) curves, we plot recall against the False Positive Rate (FPR), i.e., the percentage of uninteresting rated articles classified interesting by the recommender, *1 - specificity*, while varying the discrimination threshold.

Figure 7 shows that the CF-IDF recommender performs overall slightly better than the TF-IDF recommender regarding the balance between recall and false positives. Especially at the left side of the graph, until the FPR of 30%, the difference in performance is significant. For example at

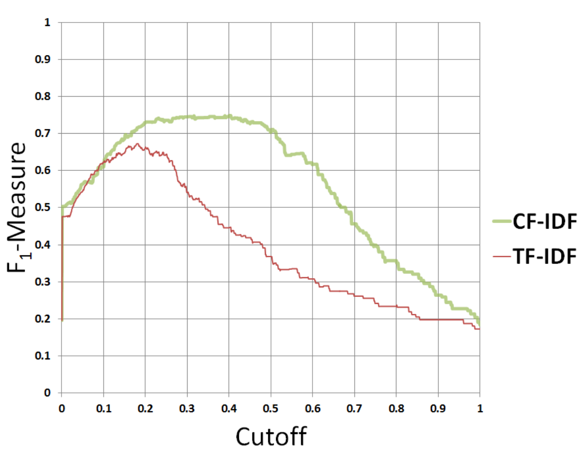

Figure 6: F_1-scores for the CF-IDF and TF-IDF recommenders for various cutoff values

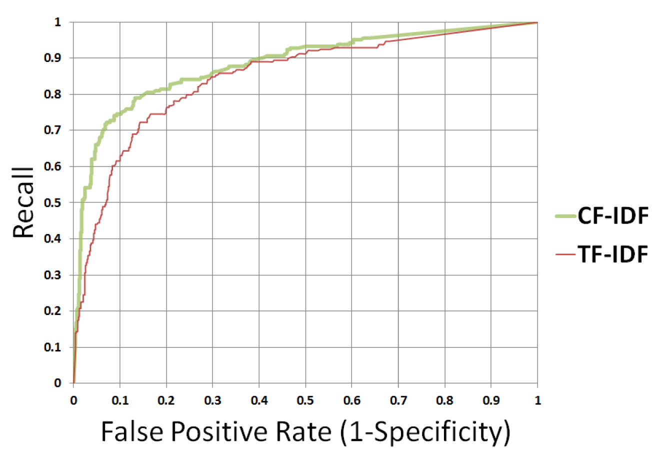

Figure 7: ROC curves for the CF-IDF and TF-IDF recommenders

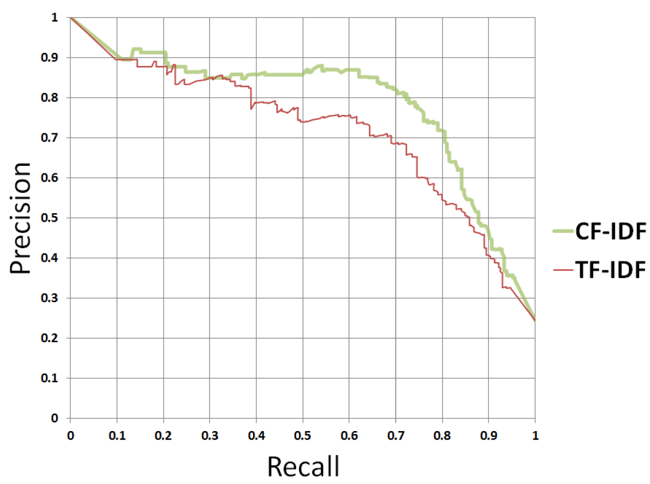

Figure 8: PR curves for the CF-IDF and TF-IDF recommenders

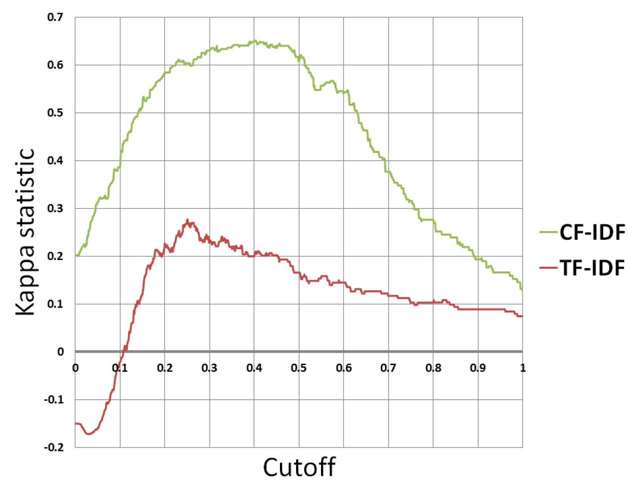

Figure 9: Kappa statistic for various cutoff values for the CF-IDF and TF-IDF recommenders

the FPR of 5%, the recall of CF-IDF is about 65%, where the recall of TF-IDF is about 45%. At the right hand of the graph the difference is less significant, although CF-IDF still outperforms TF-IDF. This is supported by the AUCs (Area Under the Curve) for both ROC curves in Fig. 7. For TF-IDF, the AUC is equal to 0.847, whereas for CF-IDF the AUC is 0.884.

The second curve we use for analyzing the test results is the Precision-Recall (PR) curve. This curve shows the relation between the precision and the recall of a recommender. Usually, obtaining a higher precision results in a lower recall. This is because a cleaner result with few false positives (high precision) will usually lead to more false negatives (low recall). This holds for both directions.

In Fig. 8 the PR curves for both recommenders are plotted. CF-IDF outperforms TF-IDF significantly according to this graph. CF-IDF manages to gain a higher precision a corresponding recall than TF-IDF. If we consider the recall of 70%, the CF-IDF recommender scores a precision about 82%, whereas the TF-IDF recommender scores about 68% for precision. The CF-IDF recommender maintains a high precision when the recall becomes larger, where TF-IDF loses precision faster. We may conclude that the CF-IDF recommender has a better balance between precision and recall than the TF-IDF recommender.

6.6 Experimental Results: Kappa Statistic

In Fig. 9, the results of the Kappa statistic [14] for the various cutoff values are shown. The Kappa statistic measures whether the proposed classifications are better than random guessing. The closer to 1, the more classification power a recommender has. Figure 9 shows that for each cutoff value, the CF-IDF recommender scores a higher Kappa statistic than the TF-IDF recommender. This means that the CF-IDF recommender seems to have more classification power than the TF-IDF recommender. One can note that the negative values for the TF-IDF recommender mean that the TF-IDF recommender performed worse than the expected performance with random guessing for low cut-off values.

7. CONCLUSION

This paper focuses on the improvement of a TF-IDF recommendation approach by using the knowledge of an ontology. By employing the knowledge base provided by the Hermes news personalization framework, a new recommendation approach, CF-IDF, has been created. This approach is based on both TF-IDF and concepts, provided by the domain ontology of the Hermes framework. The recommenders have been implemented, tested, and evaluated in Athena. Athena is a news recommendation plug-in built for the Hermes News Portal, which is the implementation of the Hermes Framework. Athena offers an environment where new recommendation approaches for news items can be implemented and tested. Both the traditional TF-IDF approach and the new CF-IDF approach have been tested in Athena.

The performances of the recommenders have been compared using the Student t-test, ROC curves, PR curves and the Kappa statistic. As the experimental results show, the CF-IDF recommender outperforms the TF-IDF approach on several measures. The CF-IDF recommender scores significantly higher compared the TF-IDF recommender on accuracy, recall, and the F_1-measure. The CF-IDF recommender also outperforms the TF-IDF recommender regarding the AUC and the Kappa statistic. On two measures, precision and specificity, both recommenders perform statistically the same. Overall, one may conclude that according to these results there is a certain benefit of using semantic techniques for a recommendation system. The use of key concepts instead of all terms does seem to yield a significant improvement over the traditional recommender. In other words: *less is more*.

However, the validity of this research needs some discussion. One methodological weakness is the quality of preprocessing the news items for the TF-IDF recommender. This can have a significant effect on the performance of TF-IDF itself. In this research we used the Krovetz [23] stemmer, but for sake of generalization it may be interesting for future work to also consider alternatives (Lovins [24], Porter [29], etc.) and compare them with the ontology-based approach.

Furthermore, since the ontology of Hermes News Portal is based mainly on business-related news, we have only processed news items in that area for both recommenders. The main weakness of the CF-IDF approach is the large dependency on the quality and completeness of the knowledge base used.

For future work it would be interesting to compare and, if possible, combine our CF-IDF recommender with other semantic approaches proposed in [18, 21]. The recommenders in [21] are also based on key concepts and the relations between them. An interesting research direction would be to make use of the different concept relationship types during user profile/news item representation (e.g., concepts related to the found ones in the user profile/news item should be treated differently based on the involved relationship types). Alternatively, the TF-IDF method has known limitations for which many variations have been proposed that try to account for these [31, 32, 36]. Hence, one could also investigate how the CF-IDF approach can be improved using these TF-IDF variations.

8. REFERENCES

[1] Agirre, E., Edmonds, P.G. (eds.): Word Sense Disambiguation: Algorithms and Applications. Springer (2006)

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

[3] Baeza-Yates, R., Ribeiro, B., Ribeiro-Neto, B.: Modern Information Retrieval. ACM Press (1999)

[4] Banerjee, S., Pedersen, T.: An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In: Gelbukh, A.F. (ed.) 3rd International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2002). Lecture Notes in Computer Science, vol. 2276, pp. 136–145. Springer (2002)

[5] Baziz, M., Boughanem, M., Traboulsi, S.: A Concept-Based Approach for Indexing Documents in IR. In: Actes du XXIIIème Congrès Informatique des Organisations et Systèmes d'Information et de Décision (INFORSID 2005). pp. 489–504. HERMES Science Publications (2005)

[6] Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A.: OWL Web Ontology Language Reference - W3C Recommendation 10 February 2004 (2004), `http://www.w3.org/TR/owl-ref/`

[7] Billsus, D., Pazzani, M.J.: A Personal News Agent that Talks, Learns and Explains. In: 3rd Annual Conference on Autonomous Agents (AGENTS 1999). pp. 268–275. ACM (1999)

[8] Billsus, D., Pazzani, M.J.: User Modeling for Adaptive News Access. User Modeling and User-Adapted Interaction 10(2), 147–180 (2000)

[9] Bogers, T., van den Bosch, A.: Comparing and Evaluating Information Retrieval Algorithms for News Recommendation. In: Konstan, J.A., Riedl, J., Smyth, B. (eds.) ACM Conference on Recommender Systems. pp. 141–144. ACM (2007)

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

[11] Cantador, I., Bellogín, A., Castells, P.: A Multilayer Ontology-Based Hybrid Recommendation Model. AI Communications 21(2–3), 203–210 (2008)

[12] Cantador, I., Bellogín, A., Castells, P.: News@hand: A Semantic Web Approach to Recommending News. In: 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2008). pp. 279–283. Springer-Verlag, Berlin, Heidelberg (2008)

[13] Cantador, I., Bellogín, A., Castells, P.: Ontology-Based Personalised and Context-Aware Recommendations of News Items. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT 2008). vol. 1, pp. 562–565. IEEE Computer Society (2008)

[14] Cohen, J.: A Coefficient of Agreement for Nominal Scales. Educational and Psychological Measurement 20(1), 37–46 (1960)

[15] Cunningham, H.: GATE, a General Architecture for Text Engineering. Computers and the Humanities 36(2), 223–254 (2002)

[16] Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge, MA (1998)

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

[18] Frasincar, F., IJntema, W., Goossen, F., Hogenboom, F.: A Semantic Approach for News Recommendation. In: M.E. Zorrilla and J.-N. Mazón and Ó. Ferrández and I. Garrigós and F. Daniel and J. Trujillo (ed.) Business Intelligence Applications and the Web: Models, Systems and Technologies. IGI Global (2011), To appear

[19] Gosling, J., Joy, B., Steele, G., Bracha, G.: The Java Language Specification, Third Edition. Addison-Wesley (2005)

[20] Guarino, N., Masolo, C., Vetere, G.: OntoSeek: Content-Based Access to the Web. IEEE Intelligent Systems 14(3), 70–80 (1999)

[21] IJntema, W., Goossen, F., Frasincar, F., Hogenboom, F.: Ontology-Based News Recommendation. In: International Workshop on Business intelligence and the Web (BEWEB 2010), EDBT/ICDT Workshops. pp. 16:1–16:6. ACM, New York, USA (2010)

[22] Knight, K., Luk, S.: Building a large knowledge base for machine translation. In: 12th National Conference on Artificial Intelligence (AAAI 1994). pp. 773–778. AAAI (1994)

[23] Krovetz, R.: Viewing Morphology as an Inference Process. In: 26th ACM Conference on Research and Development in Information Retrieval (SIGIR 1993). pp. 191–202. ACM (1993)

[24] Lovins, J.B.: Development of a Stemming Algorithm. Mechanical Translation and Computational Linguistics 11, 22–31 (1968)

[25] McBride, B.: Jena: A Semantic Web Toolkit. IEEE Internet Computing 6(6), 55–59 (2002)

[26] van Meteren, R., van Someren, M.: Using Content-Based Filtering for Recommendation. In: Potamias, G., Moustakis, V., van Someren, M. (eds.) ECML/MLnet Workshop on Machine Learning in the New Information Age (MLNIA 2000). pp. 47–56 (2000)

[27] Middleton, S.E., Roure, D.D., Shadbolt, N.R.: Ontology-Based Recommender Systems. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, pp. 577–498. International Handbooks on Information Systems, Springer (2004)

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

[29] Porter, M.: An Algorithm For Suffix Stripping. Program 14(3) (1980)

[30] Prud'hommeaux, E., Seaborne, A.: SPARQL Query Language for RDF - W3C Recommendation 15 January 2008 (2008), http://www.w3.org/TR/rdf-sparql-query/

[31] Robertson, S.E., Walker, S., Hancock-Beaulieu, M., Gull, A., Lau, M.: Okapi at TREC. In: 1st Text Retrieval Conference (TREC 1992). pp. 21–30. National Technical Information Service (1992)

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

[33] Salton, G., Wong, A., Yang, C.S.: A Vector Space Model for Automatic Indexing. Communications of the ACM 18(11), 613–620 (1975)

[34] Schouten, K., Ruijgrok, P., Borsje, J., Frasincar, F., Levering, L., Hogenboom, F.: A Semantic Web-Based Approach for Personalizing News. In: Shin, S.Y., Ossowski, S., Schumacher, M., Palakal, M.J., Hung, C.C. (eds.) 25th Symposium On Applied Computing (SAC 2010). pp. 854–861. ACM (2010)

[35] Seaborne, A., Manjunath, G., Bizer, C., Breslin, J., Das, S., Davis, I., Harris, S., Idehen, K., Corby, O., Kjernsmo, K., Nowack, B.: SPARQL/Update: A Language for Updating RDF Graphs - W3C Member Submission 15 July 2008 (2008), http://www.w3.org/Submission/SPARQL-Update/

[36] Singhal, A., Buckley, C., Mitra, M.: Pivoted Document Length Normalization. In: Frei, H.P., Harman, D., Schäuble, P., Wilkinson, R. (eds.) 19th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval (SIGIR 1996). pp. 21–29. ACM (1996)

[37] Sowa, J.F.: Semantic Networks. In: Shapiro, S.C. (ed.) Encyclopedia of Artificial Intelligence. Wiley, 2 edn. (1992)

[38] Yan, L., Li, C.: A Novel Semantic-based Text Representation Method for Improving Text Clustering. In: Prasad, B. (ed.) 3rd Indian International Conference on Artificial Intelligence (IICAI 2007). pp. 1738–1750 (2007)