Semantics-Based News Recommendation

Michel Capelle
michelcapelle@gmail.com

Marnix Moerland
marnix.moerland@gmail.com

Flavius Frasincar
frasincar@ese.eur.nl

Frederik Hogenboom
fhogenboom@ese.eur.nl

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

ABSTRACT

News item recommendation is commonly performed using the TF-IDF weighting technique in combination with the cosine similarity measure. However, this technique does not take into account the actual meaning of words. Therefore, we propose two new methods based on concepts and their semantic similarities, from which we derive the similarities between news items. Our first method, Synset Frequency – Inverse Document Frequency (SF-IDF), is similar to TF-IDF, yet it does not use terms, but WordNet synonym sets. Additionally, our second method, Semantic Similarity (SS), makes use of five semantic similarity measures to compute the similarity between news items for news recommendation. Test results show that SF-IDF and SS outperform the TF-IDF method on the F_1-measure.

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, Performance

Keywords

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

1. INTRODUCTION

Today's World Wide Web is a largely unstructured, enormous, and ever increasing collection of data, information, and knowledge. A specific and valuable type of information frequently found on the Web is news. Not all information can be classified as news: information is considered to be news if it was previously unknown to the viewer (i.e., an entity as simple as a human user or as complex as a reading agent or recommendation tool) and/or when the specific item reports on recent events. Often, news is organized in a few main categories, e.g., business, sports, politics, technology, etc. However, on Web sites, news is hardly ever organized to what might interest the reader and what might not. Lately, this resulted in a lot of research into news recommendation algorithms and systems.

Many algorithms exist for news item recommendation, of which in this paper, we discuss the most common ones and compare the results of these with one another. There are currently three different ways on how a news recommender can recommend news articles: a content-based recommendation, a collaborative filtering recommendation, and a hybrid recommendation.

Content-based recommendation focuses on existing user preferences and searches for news items which are similar to the user preferences. The collaborative recommendation focuses on other users which are similar to the person browsing for news items and what articles they found interesting. The hybrid method is a mix of the two previously mentioned recommendation systems and tries to combine the best of both worlds [1]. Some argue that within content-based recommending systems one could distinguish between two subtypes: a recommendation system that simply takes every word into account (without paying any attention to word meanings) on the one hand, and a semantic approach on the other hand, which tries to capture the meaning or sense of each word [13].

In our current endeavours, we focus both on the traditional content-based approach as well as the semantic approach to news recommendations. The collaborative recommendation and the hybrid recommendation are outside the scope of the research presented here. Our work builds on our earlier efforts in the field of news recommendation [6, 8, 9, 11, 13]. The focus of the work discussed in this paper is on employing new recommendation algorithms within the Hermes news personalization framework, which has been introduced in [8]. For this, we build the Ceryx framework, supporting many different kinds of recommendation systems, in order to compare the different methods and to find the best semantics-based method for news item recommendation.

WIMS'12, June 13–15, 2012, Craiova, Romania.

In this paper, we investigate how recommenders which focus on the meaning of a word perform when compared to one another as well as when compared to the traditional term-based recommender. The semantic recommenders we discuss are the frequently used Jiang & Conrath [16], Leacock & Chodorow [17], Lin [19], Resnik [22], and Wu & Palmer [29] similarity-based recommenders, as well as our own Synset Frequency-Inverse Document Frequency (SF-IDF) recommender. The traditional recommender uses the well-known TF-IDF algorithm in combination with the cosine similarity measure.

This paper is organized as follows. First, Section 2 discusses related work. Then, we present the Hermes News Portal (HNP), i.e., the implementation of the Hermes framework, and also the proposed Ceryx framework in Section 3. Subsequently, we explain our implementation of the Ceryx framework and we provide additional insights in the workings of the previously mentioned recommenders in Section 4. Next, we evaluate the results of our recommenders in Section 5. Last, we conclude our paper and propose future work in Section 6.

2. RELATED WORK

A lot of research has been done in the field of recommender systems, and more specifically much of the scientific effort has been focused on the development of similarity measures. In our introduction, we already briefly touched upon five semantic similarity measures [16, 17, 19, 22, 29], of which more details are given in Section 3. In [28], the authors compare the aforementioned similarity-based recommenders and use them to relate similar words with each other so that an ontology can be created.

Even though we use the similarity-based recommenders in order to measure the similarity between the different news items, the way the similarity between the words is being computed is the same as in the previously listed work. The main difference between the work of Warin and Volk and the work presented in this paper is that the authors of [28] do not take into consideration the actual meaning of the word in the sentence. For each word, the authors test every possible meaning, pick the one with the highest similarity to the ontology, and subsequently decide whether or not it is similar to another word based on a cut-off value. In contrast, in our efforts, we make use of the adapted Lesk method [15] in order to identify the specific meaning a word has in an arbitrary sentence, i.e., we perform word sense disambiguation. This choice is motivated by the fact that this gives us more certainty about the correct meaning of the word by taking into account its context.

Lovelyn Rose and Chandran [20] aim to improve Web search results given a query statement by the user. In their attempt, the authors compare the different ranking methods with each other. Aside from using the five aforementioned methods, Lovelyn Rose and Chandran also make use of the adapted Lesk method as a way to find the similarity between words. While we have chosen to ignore the Lesk method for similarity ranking due to it being rather slow, the paper is similar in the way that it compares the results obtained using different similarity measures. However, we compare two complete texts with each other while this paper compares a query statement with a complete text.

Aside from the similarity measures already mentioned, we also use the SF-IDF technique in our paper. While this technique is much like the traditional TF-IDF weighing technique in combination with the cosine similarity measure described in [24], it is different in the sense that we consider a certain word and its synonyms, derived from a semantic lexicon like WordNet [7], as the same concept for counting purposes. There has been work on weighting techniques operating on (ontology) concepts instead of terms, i.e., Concept Frequency - Inverse Document Frequency (CF-IDF), which is a term coined in our previous work [11]. Although these techniques also take into account the meaning of words, they are rather limited in the sense that they rely on domain ontologies or other forms of concept specifications. As we employ a large semantic lexicon like WordNet, we avoid the need for regular knowledge base updating with new domain knowledge.

Getahun et al. [10] also use a semantics-based method for identifying news item similarity. Compared to [10], where the relatedness of two news items is computed by taking into account the global semantic neighborhood of a concept, in our approach, SF-IDF is combined with the cosine similarity measure. Furthermore, while we specifically focus on the similarity between two different news items, the authors of [10] aim to identify the relationship between two news items which can have disjointness, intersection, equality, inclusion, and oppositeness with each other. By purely focusing on similarities, we intend to analyze the effect of considering word senses for the recommendation of news items.

Additionally, in [5] a conceptual indexing method based on WordNet is proposed, which represents document contents by a semantic network [25]. The documents are mapped on the WordNet semantic network and converted from a set of terms to a set of concepts. Subsequently, extracted concepts are weighted using TF-IDF and Okapi BM25 [23]. 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 as present in our method.

3. CERYX FRAMEWORK

The Ceryx framework is an extension of Athena, which is a component of the Hermes framework for news recommendation. Its goal is to recommend news items based on a user behavior profile and a semantic similarity measure. This is being done in three steps. The first step is to construct a user profile which contains the user's preferences regarding previously browsed news items. Second, Hermes is employed for determining the senses of the words in the user profile and the unread news items. In the third and final step, the similarity between unread news items and the user profiles is measured by making use of either one of our two semantic similarity methods, or the already existing TF-IDF method. In the next subsections these three steps are described in more detail.

3.1 User Profile

A user profile consists of all the news items which the user has read so far. Once a user reads a previously unread news item, the user profile is updated by adding that news item to it. The set of news items in the user profile is defined as

$$P = \{p_1, p_2, \cdots, p_n\}, \tag{1}$$

where p_i represents a news item in the user profile and n is the number of news items in the user profile. The user profile gives us information about the preferences of the user, since the user primarily reads news items which he or she considers to be interesting. The user profile is the starting point from which interesting unread news items can be recommended to the user.

3.2 WordNet Synsets

The semantic similarity methods compare the words from the unread news items with the words from the news items in the user profile. To compare these words, the semantic similarity is measured by making use of the WordNet dictionary, which is an online lexical database for the English language that contains over 166,000 pairs of word forms and senses [7].

Each word form in WordNet comes with a set of senses, which is a set of possible meanings for that word form. An example of this, is the word form 'turkey' has associated the senses 'Meleagris gallopavo' (i.e., the animal) and 'Republic of Turkey'. Each word form with a sense is called a 'synset'. WordNet consists of four part-of-speeches 'noun', 'verb', 'adverb', and 'adjective'. These four different part-of-speeches are organized into sets of synonyms, each representing a lexicalized concept. These so-called synsets are linked to each other by making use of semantic pointers which refer from one synset to another synset with which it has a certain relation. The type of connection is given by the kind of semantic pointer. WordNet includes the semantic relations synonymy, antonymy, hyponymy, meronymy, troponymy, and entailment [7].

Ceryx compares the WordNet synsets from an unread news item and the WordNet synsets from the news items in the user profile. The synsets are obtained using a word sense disambiguation procedure [9]. A news item from the user profile can be described by a set of WordNet synsets as

$$S_p = \{s_1, s_2, \cdots, s_m\} , \tag{2}$$

where S_p is the set of WordNet synsets from news item p, s_i represents a WordNet synset in the news item, and m is the number of WordNet synsets in the news item.

We combine all the WordNet synsets from all of the news items in the user profile into one set. The union of all the WordNet synsets which appear in the user profile is then

$$R = \bigcup_{p \in P} S_p , \tag{3}$$

where S_p are the WordNet synsets from news item p, and P represents all previously read news items (the profile). The set of WordNet synsets from the unread news item is defined as

$$U = \{u_1, u_2, \cdots, u_k\} , \tag{4}$$

where u_i represents a WordNet synset in the unread news item, and k is the number of WordNet synsets in the unread news item. Next, we propose two methods to compare the words in the unread news item with all of the words in the user profile.

3.3 Similarity Methods

Based on the WordNet synsets in news items, we propose two methods in order to compute the semantic similarity between two news items: Synset Frequency – Inverse Document Frequency (SF-IDF) and Semantic Similarity (SS). We compare both of them with the well-known Term Frequency – Inverse Document Frequency (TF-IDF) method. We will now describe how every method works in the following subsections.

3.3.1 TF-IDF

The TF-IDF method is a classical method which has been used many times in recommending news items and has many variants. We have chosen to use the traditional TF-IDF technique combined with the cosine similarity method, because these techniques are well-known, have been frequently applied, and have shown to produce good results [2]. The TF-IDF technique contains two elements. The first element is $tf(t, d)$, where t is the word that is being counted, and d the current document in which the word is being counted. $tf(t, d)$ returns a number indicating how many times word t appears in document d. The second element that plays a central role in the TF-IDF calculation is $idf(t, d)$, which is defined as

$$idf(t, d) = \log \frac{|D|}{|d \in D : t \in d|} , \tag{5}$$

where $|D|$ is the total number of documents in the collection of documents which is being compared, and $|d \in D : t \in d|$ is the number of documents in which word t appears. Since we cannot divide by zero, this number will always be 1 or higher. This makes sense if we assume we only look at the words which appear somewhere in the collection of documents.

By combining the two elements $tf(t, d)$ and $idf(t, d)$ through multiplication, we achieve the formula for TF-DF:

$$tf - idf(t, d) = tf(t, d) \times idf(t, d) . \tag{6}$$

After the TF-IDF vector for the current document has been computed, we compute the TF-IDF vector for the next document. Once every document has a TF-IDF vector associated, we are able to compute the cosine similarity.

The cosine similarity measure of vectors A and B can be described as:

$$similarity(A, B) = \frac{A \times B}{||A|| \cdot ||B||} , \tag{7}$$

where A is a vector containing all TF-IDF values for the different words from the user profile's news items and where B is a vector containing all TF-IDF values for the different words from an unread news item. Here, $||A||$ is the magnitude of vector A, and is defined as

$$||x|| = \sqrt{x_1^2 + \cdots + x_n^2} . \tag{8}$$

In (8), x is for example A, x_1 is the TF-IDF value for the ith encountered word in vector A, and n is the length of the vector. $||B||$ is calculated in the same way as $||A||$ using (8) with $x = B$.

The above similarity will be computed for every unread document, meaning that A and $||A||$ will remain the same as they represent the profile, but B and $||B||$ will be the vector and magnitude for the new unread news item, respectively. Afterwards, all of the results are sorted in a descending way, and those news items which have a similarity value higher than the cut-off value are recommended to the user.

3.3.2 SF-IDF

SF-IDF works in the same way as TF-IDF does, with the difference that t is now replaced by s, where s is not a word but a synset instead. This means that we consider two words with the same meaning as one and the same synset. The SF-IDF formula for similarity is:

$$sf - idf(s,d) = tf(s,d) \times idf(s,d) \ . \tag{9}$$

The above formula looks similar to the TF-IDF formula, except that s now takes the place of t. The cosine similarity measure is the same as (7). Once again, all of the results will be sorted from highest to lowest similarity to the profile, and those news items which have a similarity value higher than the cut-off value will be recommended to the user.

3.3.3 Semantic Similarity

Just like with the previous method, our Semantic Similarity (SS) method compares the WordNet synsets from the unread news items with the WordNet synsets from all the news items in the user profile. This is done by making pairs between the elements of the two sets with a common part-of-speech.

To measure the similarity a vector in the n-dimensional space is created of all the possible combinations of WordNet synsets from the unread news item on one hand and the union of WordNet synsets from the user profile on the other hand. The vector is defined as

$$V = (\langle u_1, r_1\rangle, \cdots, \langle u_k, r_l\rangle) \ \forall \ u \in U, \ r \in R \ , \tag{10}$$

where u_i represents a WordNet synset from the unread news item, r_j represents a WordNet synset from the user profile, k is the number of WordNet synsets in the unread news item, and l is the number of WordNet synsets in the user profile.

From that vector a subset is made that contains all the combinations which have a common part-of-speech. This subset can be described as

$$W \subseteq V \ \forall \ (u,r) \in W : POS(u) = POS(r) \ , \tag{11}$$

where $POS(u)$ and $POS(r)$ are defined as the part-of-speech of WordNet synset u in the unread news item or WordNet synset r in the user profile.

For every combination in the above subset, a similarity rank is computed with the selected semantic similarity measure. There are a total of five different semantic similarity measures from which can be chosen. These measures are: Jiang & Conrath [16], Leacock & Chodorow [17], Lin [19], Resnik [22], and Wu & Palmer [29]. Each of these measures will separately be discussed in more detail later in this section.

The final similarity rank of the unread news item is the sum of all the combination's similarities divided by the total number of combinations. This final rank is defined as

$$rank(newsitem) = \frac{\sum\limits_{(u,r)\in W} sim(u,r)}{|W|} \ , \tag{12}$$

where $sim(u,r)$ is the similarity rank between the WordNet synsets u and r, and $|W|$ is the number of combinations between the WordNet synsets from the unread news item and the user profile. As in TF-IDF and SF-IDF, the ranks which are higher than the cut-off value are recommended to the user. Next, the WordNet terminology is explained in order to be able to describe the different semantic similarity measures.

Each synset is represented by a node in the WordNet taxonomy. This taxonomy is a hierarchy of 'is-a' relationships between nodes. The similarity measures aim to explain that a synset is semantically closer to another synset, for example a 'turkey' is closer to 'animal' than that it is to 'boat'. The measures of Jiang & Conrath, Resnik, and Lin are based on the information content of the nodes, while Leacock & Chodorow and Wu & Palmer make use of the path length between the nodes.

The information content (IC) of a node is the negative logarithm of the sum of all probabilities of all the words in the synset. The probability that an instance x of synset occurs in a corpus is defined as $p(x)$. The information content can be written as

$$IC(s) = -\log \sum_{w \in s} p(w) \ , \tag{13}$$

with w representing a word in synset s, with the meaning given by s.

The path length is either the shortest path ($length$) between the two nodes or the maximum depth (D) from the lowest node to the top node. Further, the LCS between two nodes is the lowest common subsumer, which is the lowest node that dominates both nodes [22].

The Jiang & Conrath measure uses the information content of both the WordNet synsets and the lowest common subsumer. The similarity is

$$dist_{J\&C}(u,r) = IC(u) + IC(r) - 2 \times IC(LCS(u,r)) \ . \tag{14}$$

Since this measure returns low values for large similarity and high values for small similarity, the formula we use is the inverse of the original one, and is defined as

$$sim_{J\&C}(u,r) = \frac{1}{dist_{J\&C}(u,r)} \ . \tag{15}$$

Leacock & Chodorow's measure is, unlike Jiang & Conrath's measure, not based on the information content but on the path length between both nodes. The shortest path between the two nodes is divided by double the maximum depth, as is shown in the following equation:

$$sim_{L\&C}(u,r) = -\log \frac{length(u,r)}{2D} \ . \tag{16}$$

Lin's measure makes use of the logarithms of the chances of appearance of both nodes and the lowest common subsumer. Lin's measure is defined as

$$sim_L(u,r) = \frac{2 \times \log p(LCS(u,r))}{\log p(u) + \log p(r)} \ . \tag{17}$$

Resnik's measure maximizes the information content of the lowest common subsumer from both nodes, i.e.,

$$sim_R(u,r) = IC(LCS(u,r)) \ . \tag{18}$$

Similar to Leacock & Chodorow's measure, the Wu & Palmer's similarity measure is based on the path length between two nodes. It uses the depth of the lowest common subsumer of both nodes and the path length between them. The similarity is defined as

$$sim_{W\&P}(u,r) = \frac{2 \times depth(LCS(u,r))}{length(u,r) + 2 \times depth(LCS(u,r))} \ . \tag{19}$$

4. CERYX IMPLEMENTATION

The Ceryx implementation is an extension of the implementation of the Hermes framework, the Hermes News Portal (HNP) [9]. Ceryx is a plug-in for the existing HNP and an expansion of earlier work [6, 8, 9, 11, 13]. Because Hermes was written in Java, we have chosen to also write Ceryx in Java. In this section, we discuss the Ceryx plug-in, the user profile construction, the usage of WordNet, and last, the implementation of the three similarity methods being used in Ceryx: TF-IDF, SF-IDF, and SS.

4.1 The Ceryx Plug-in

The Ceryx plug-in has a tab in the user interface of HNP, entitled 'Semantic recommendations'. The tab consists of three sub-tabs, i.e., 'All News Items', 'Recommendations', and 'Test Results'.

The tab 'All News Items' allows browsing for all the news items in HNP, sorted by date. The list of news items is shown by pressing the 'Refresh' button. All items contain a title, a date, and an abstract. The user can browse through the list of items, instead of using the query function in Hermes [8, 9].

In the second tab, the 'Recommendations' tab, the user can get a news item recommendation, based on his user profile, which was already implemented in Hermes. In Hermes, the user can select a news item by clicking on one. The user profile is then updated with the read news item.

In Ceryx, a recommendation is made using a similarity method. In our current efforts, a comparison is made between the methods TF-IDF, SF-IDF, and the semantic similarity measures of the SS method. Abusing slightly the terminology, the selected similarity method (with the used similarity measure) is from now on called a 'recommender'. Once the user has selected a recommender, the recommendation can be made by clicking on the 'Get Latest Recommendations' button. All the recommenders use the same user profile, but each uses its own algorithm to calculate which news items are interesting for the user and which are not.

Once the recommendation is done, the user is shown a list with all of the recommended news items. Every item has a rank ranging from 0 to 100, where 100 is the highest recommendation and 0 the lowest. The list is sorted descending, which means that the most interesting unread news items are at the top and the least interesting unread news items at the bottom. For the words which are highlighted, the corresponding lemma appears at least once in the list of lemmas from the user profile. In this way, the user is shown exactly which words in the new news item can also be found in the list of words which make up the user profile.

The third tab, 'Test Results', is a testing environment. When the user has loaded a test file, which consists of a user profile, all the recommenders are tested on the measures accuracy, sensitivity, precision, and specificity. The user can set the random seed, the cut-off value, and the number of tests. The random seed is used to divide the news items used for testing in a training set and a test set. The cut-off value decides from which point on the threshold from which the unread news item is considered 'interesting'. The number of tests determines how many times you split the set of items in different training and test sets. Figure 1 depicts the user interface after an arbitrary test run within the testing environment.

4.2 User Profile Construction

A user profile consists of all the news items which the user has read. In Hermes, a user is able to click on a news item, resulting in a Web browser that is opened, showing the full news item. Additionally, the user profile is updated with the selected news item. In HNP, each news item has its own unique identifier, a Uniform Resource Identifier (URI).

4.3 WordNet

When comparing news items with each other using a semantic recommender, WordNet is used for word comparison in the texts. In order to retreive the WordNet synset from a word in a news item, the word form, the part-of-speech, and the word sense are needed to specify exactly which meaning it represents. Ceryx uses several steps to come to the correct WordNet word. These steps are described below.

First, all of the words in the text receive a part-of-speech. The Stanford Log-linear Part-of-Speech Tagger from the Stanford Natural Language Processing Group is used to determine the part-of-speech of every word [27].

The second step is to remove all stop words. Stop words have little to no meaning to the similarity and mostly only act as noise, things which cause the news recommender to make a faulty conclusion, when trying to compare texts. A list of the words which we consider to be stop words can be found at [18]. The lemma of a word is determined using JAWS [26].

As a third step, the right word sense id determined using the adapted Lesk algorithm for word sense disambiguation. Since a word can have multiple meanings, the correct meaning should be selected if we want to be able to precisely compare different news items with each other. By using every lemma without the stop words and with their corresponding parts-of-speech as inputs for the adapted Lesk algorithm, we retrieve the unique synsets for the words that best match the input [4]. The used implementation of adapted Lesk is provided by the Denmark Technical University (DTU) Similarity application [14].

Now that all of the words in the news items have been checked for stop words, and each word has its lemma, part-of-speech, and correct word sense determined, all of the information needed in order to compute the similarity is looked up in the WordNet dictionary and stored inside the news item.

4.4 Similarity Methods

To make a comparison between news items, Ceryx makes use of the TF-IDF method, and our two methods for measuring semantic similarity: SF-IDF and SS. We will discuss the implementation of these three methods in this order.

4.4.1 TF-IDF

Without making use of WordNet, the TF-IDF method simply goes through all of the words in the user profile and makes a list of all the different words with the number of times each word appears in the user profile. Once the list of all the different words has been created for the user profile, a list of all different words will be created for every unread news item. Now that the term frequency for each word in every news item has been determined, the next step is to calculate the inverse document frequency, thus effectively calculating the TF-IDF value. For every word in the current news item, the other documents are checked whether or not

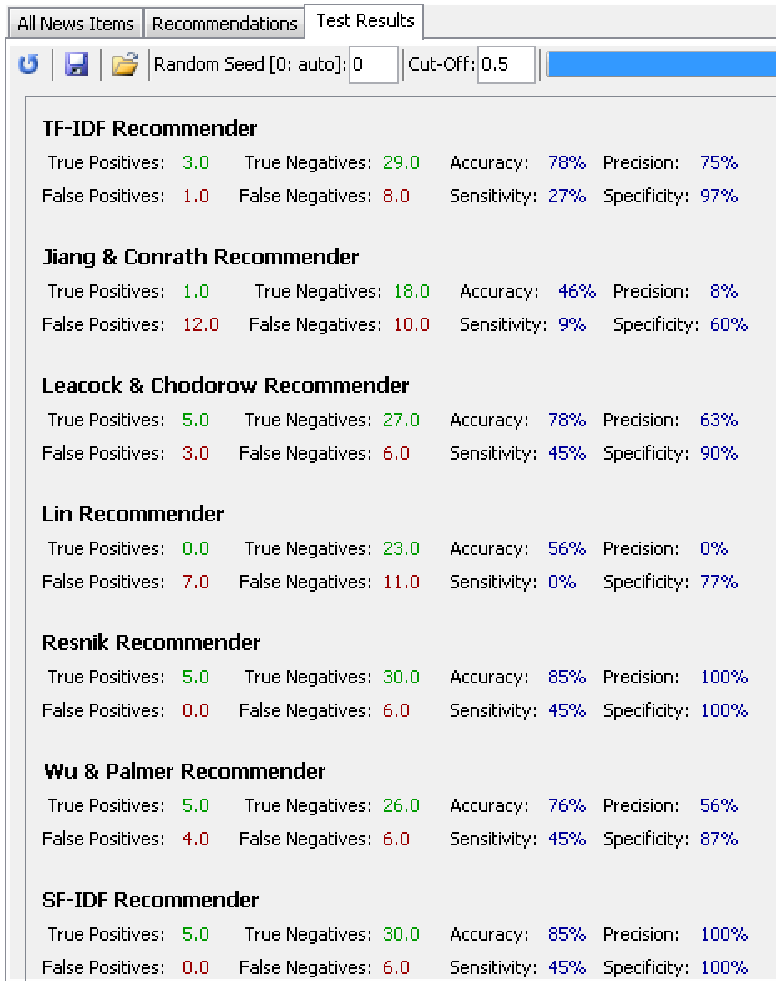

Figure 1: Performance measures within the test environment in Ceryx (for one run).

they contain that word. After this, the TF value and the IDF value of that word are multiplied by each other, and we obtain the TF-IDF value of that word in the first news item. This is done for every word in every news item.

Since we now have all of the TF-IDF values, we compute cosine similarity between two news items, or in our case between one news item and a collection of news items that represent the user profile. By computing the dot product and the magnitude of the unread news item and the user profile, we end up with a score indicating how similar the two are. The final step is to compute this for every unread news item so that we can order the news items based on their similarity rank. Every unread news item which has a similarity rank above the cut off value is suggested to the user.

4.4.2 SF-IDF

Once every word in the text has been identified and has been assigned a corresponding WordNet synset (synonym set), a list with all of the synsets in the user profile is created. Then, the same thing is being done for all of the unread news items in the database. One of the unread news items is then being compared to the user profile using the SF-IDF vectors computed in a similar way as TF-IDF vectors, words being replaced by synsets in the computations. The cosine similarity measure is computed and the resulting value is assigned to the specific unread news item. After this has been done for every unread news item, the values assigned to the unread news items are sorted descendingly. The items with the highest scores are the ones most similar to the user profile, and are thus suggested to the user in that order.

4.4.3 Semantic Similarity

Before measuring the final similarity rank of an unread news item, a set of all of the WordNet synsets from the unread news item and a set of all of the WordNet synsets from all the read news items in the user profile are collected. Then, all of the combinations between synsets from those two sets which have a common part-of-speech are selected.

With the selected combinations, a similarity rank is computed with a Java implementation of similarities [12] of the application written originally in Perl, as described in [21]. The word sense and the lemma of both WordNet words, the common part-of-speech, and the similarity measure are then given as input to the similarity computation. The similarity computation returns the rank of semantic similarity between the WordNet synset from the unread news item and the WordNet synset from the user profile.

When the similarity is known for all the combinations, the final similarity for the unread news item is computed by summing up every combination's similarity and dividing it by the total number of combinations possible. When all similarities are known, all the unread news item are sorted in a descending way.

5. EVALUATION

In order to evaluate the performance of our two recommendation methods, i.e., the SF-IDF and SS approaches, compared to the more traditional TF-IDF method, as performance measures we make use of accuracy, precision, recall, and specificity. Using the precision and recall, we calculate the F_1-measure, i.e., the harmonic mean between precision and recall. A more detailed description of our measures can be found in general works on information extraction, e.g., [3].

5.1 Experimental Setup

In our experiments, a Web site is utilized, which shows 100 news items one by one to users. Each user has to indicate whether the news item is interesting to him/her or not. A set of read items is subsequently constructed, consisting of the news item's URIs and whether it is interesting. Ceryx subsequently loads this data.

We use 19 participants, ranging in age from 19 to 24. All participants are (mainly male) students in the field of Informatics and Economics, and are familiar with the area related to a predefined profile that is used in the test. Note that due to the limited number of participants, each of the participants is given the same profile in order to avoid user bias. The profile that is given to the users beforehand is constructed in such a way that it is assumed the user is interested in *articles that are related to Microsoft, and its products and competitors.*

We use supervised learning for processing the test results. A random split is made into a training set and a test set. Of our data, 60 percent is assigned to the training set, while 40 percent is assigned to the test set. The training and test set have a proportional part of interesting and uninteresting news items.

Using the training set a user profile is constructed, to which all interesting news items are added. The value for each similarity measure is subsequently computed. If this similarity exceeds a predefined cut-off value, the news item is recommended. We use a cut-off value of 0.5, as this has proven to perform best, with respect to the F_1-measure, in initial experiments.

In order to measure the performance of the different similarity measures, confusion matrices are created, which are used for the calculation of the performance measures. These measures yield insights into the performance of the various similarity measures. Multiple runs are made, for each run a random division between the test and training set is made. We use five runs for each user in our experiment.

We investigate whether a recommender performs statistically better than another recommender. The test data for five tests (averaged) for each of the 19 users is gathered. The performance of the F_1-measure is tested in a one-tailed two-sample paired Student t-test. With a level of 95% significance, the null and alternative hypotheses are defined as

$$H_0 : \mu_1 < \mu_2 \ , \ H_1 : \mu_1 > \mu_2 \ , \alpha = 0.05 \ , \qquad (20)$$

where μ_1 is the mean performance on the F_1-measure of the first recommender and μ_2 is the mean performance on the F_1-measure of the second recommender.

5.2 Experimental Results

After processing the test data from the 19 participants, the performance measures are calculated. Table 1 shows the performance of TF-IDF and SF-IDF, while Table 2 displays the performance measures of the similarity measures used in the SS method. As shown in the table, SF-IDF (46.8%) outperforms TF-IDF (32.0%) on the F_1-measure. Also, the performance of the SS recommenders is better than TF-IDF, except for the Lin recommender. The best method is SS with the Wu & Palmer recommender (47.1%), which is slightly better than SF-IDF (46.8%).

In order to evaluate the significance of the test results, the p-values of a one-tailed two-sample paired Student t-test are determined for the F_1-measure. The hypothesis is defined in (20) for the results of five tests for the means of the 19 users (19 F_1-measure values. The recommenders for which the p-values are smaller than 0.05 performed statistically better than the other recommender. As shown in Table 3, with a 95% significance, the SF-IDF and Semantic Similarity with the similarity measures of Leacock & Chodorow, Resnik, and Wu & Palmer perform statistically better than TF-IDF. Significant results are also obtained when we compare the F_1-measure of Wu & Palmer with Jiang & Conrath and Lin, showing the superiority of the first approach compared to the other two approaches.

6. CONCLUSION

In an attempt to improve the Term Frequency – Inverse Document Frequency (TF-IDF) weighting technique that is traditionally used for news item recommendation, in this paper we presented two approaches that take into account the meaning of words. Our methods are based on concepts and their semantic similarities, from which we derive the similarities between news items. Our first method, Synset Frequency – Inverse Document Frequency (SF-IDF), is similar to TF-IDF, yet it does not use terms but WordNet synonym sets. Additionally, our second method, Semantic Similarity (SS), makes use of five semantic similarity measures to compute the similarity between news items for recommending.

The proposed approaches to news item recommendation have been implemented as Ceryx, an extension to the Hermes News Portal news personalization service. Ceryx makes use of the SF-IDF and SS methods, a user profile, and a set of news items in order to suggest interesting unread news articles to the user. Before using our semantic methods, news items are preprocessed by first identifying the part-of-speech, lemma, and the sense of every word. By comparing the F_1-measure, accuracy, precision, recall, and specificity

Table 1: Average test results for 19 users (5 tests per user) for TF-IDF and SF-IDF.

Performance measure	TF-IDF	SF-IDF
Accuracy	78.2%	80.1%
Precision	77.4%	77.8%
Recall	22.0%	35.9%
Specificity	97.2%	94.7%
F_1-measure	32.0%	46.8%

Table 2: Average test results for 19 users (5 tests per user) for Semantic Similarity recommenders: Jiang & Conrath (J&C), Leacock & Chodorow (L&C), Lin (L), Resnik (R), and Wu & Palmer (W&P).

Performance measure	J&C	L&C	L	R	W&P
Accuracy	78.3%	59.5%	38.1%	74.5%	58.5%
Precision	64.2%	33.7%	19.9%	56.4%	35.3%
Recall	29.3%	63.5%	49.7%	40.0%	73.6%
Specificity	94.6%	57.9%	34.0%	86.3%	52.6%
F_1-measure	38.4%	43.2%	27.7%	42.8%	47.1%

Table 3: Significance of test results for 19 users (5 tests per user) for the F_1-measure for the recommenders: TF-IDF, SF-IDF, Jiang & Conrath (J&C), Leacock & Chodorow (L&C), Lin (L), Resnik (R), and Wu & Palmer (W&P) (reject $\mu_{row} < \mu_{column}$).

p (one-tailed)	TF-IDF	SF-IDF	J&C	L&C	L	R	W&P
TF-IDF		0.006	0.076	0.009	0.159	0.011	0.001
SF-IDF	0.006		0.093	0.203	0.002	0.207	0.468
J&C	0.076	0.093		0.165	0.006	0.136	0.009
L&C	0.009	0.203	0.165		0.000	0.465	0.106
L	0.159	0.002	0.006	0.000		0.000	0.000
R	0.011	0.207	0.136	0.465	0.000		0.062
W&P	0.001	0.468	0.009	0.106	0.000	0.062	

of the various recommenders against TF-IDF, we observe that the SF-IDF and Leacock & Chodorow, Resnik, and Wu & Palmer SS methods outperform the traditional TF-IDF method.

For future work, we suggest that the SF-IDF method should consider other lexical semantic connections into account apart from just the synonyms. Examples of these other lexical semantic connections are antonyms, hypernyms, and hyponyms. The assumption is that if an article discusses subjects which are using the previous relations to the subjects discussed in the user profile, the user might be interested in that article as well. Furthermore, it might be interesting to analyze whether the SF-IDF technique would still perform better than the TF-IDF technique if the texts are written in a different language than English (e.g., Dutch).

Acknowledgment

The authors are partially sponsored by the NWO Physical Sciences Free Competition project 612.001.009: Financial Events Recognition in News for Algorithmic Trading (FERNAT) and the Dutch national program COMMIT.

7. REFERENCES

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

[2] Ahn, J., Brusilovsky, P., Grady, J., He, D., Syn, S.Y.: Open User Profiles for Adaptive News Systems: Help or Harm? In: Williamson, C.L., Zurko, M.E., Patel-Schneider, P.F., Shenoy, P.J. (eds.) 16th International Conference on World Wide Web (WWW 2007). pp. 11–20. ACM (2007)

[3] Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison Wesley Longman, 2 edn. (2011)

[4] Banerjee, S., Pedersen, T.: An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In: Gelbukh, A.F. (ed.) 4th International Conference on Computational Linguistics and Intelligent Text Processing (CICLING 2002). pp. 136–145. Springer-Verlag (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] Borsje, J., Levering, L., Frasincar, F.: Hermes: a Semantic Web-Based News Decision Support System. In: 23rd Annual ACM Symposium on Applied Computing (SAC 2008). pp. 2415–2420. ACM (2008)

[7] Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press (1998)

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

[9] Frasincar, F., IJntema, W., Goossen, F., Hogenboom, F.: Business Intelligence Applications and the Web: Models, Systems and Technologies, chap. A Semantic Approach for News Recommendation, pp. 102–121. IGI Global (2011)

[10] Getahun, F., Tekli, J., Chbeir, R., Viviani, M., Yetongnon, K.: Relating RSS News/Items. In: Gaedke, M., Grossniklaus, M., Díaz, O. (eds.) 9th International Conference on Web Engineering (ICWE 2009). pp. 442–452. Springer-Verlag (2009)

[11] Goossen, F., IJntema, W., Frasincar, F., Hogenboom, F., Kaymak, U.: News Personalization using the CF-IDF Semantic Recommender. In: Akerkar, R. (ed.) International Conference on Web Intelligence, Mining and Semantics (WIMS 2011). ACM (2011)

[12] Hope, D.: Sussex University: NLP Lab, Homepage.

http://www.cogs.susx.ac.uk/users/drh21/ (2011)
[13] IJntema, W., Goossen, F., Frasincar, F., Hogenboom, F.: Ontology-Based News Recommendation. In: Daniel, F., Delcambre, L.M.L., Fotouhi, F., Garrigós, I., Guerrini, G., Mazón, J.N., Mesiti, M., Müller-Feuerstein, S., Trujillo, J., Truta, T.M., Volz, B., Waller, E., Xiong, L., Zimányi, E. (eds.) International Workshop on Business intelligencE and the WEB (BEWEB 2010) at Thirteenth International Conference on Extending Database Technology and Thirteenth International Conference on Database Theory (EDBT/ICDT 2010). ACM (2010)
[14] Jensen, A.S., Boss, N.S.: Dty similarity. http://damn.dk/similarity/javadoc/model/similarity/Lesk.html (2008)
[15] Jensen, A.S., Boss, N.S.: Textual Similarity: Comparing Texts in Order to Discover How Closely They Discuss the Same Topics. Bachelor's Thesis, Technical University of Denmark (2008), http://orbit.dtu.dk/getResource?recordId=220969&objectId=1&versionId=1
[16] Jiang, J.J., Conrath, D.W.: Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In: 10th International Conference on Research in Computational Linguistics (ROCLING 1997). pp. 19–33 (1997)
[17] Leacock, C., Chodorow, M.: WordNet: An Electronic Lexical Database, chap. Combining Local Context and WordNet Similarity for Word Sense Identification, pp. 265–283. MIT Press (1998)
[18] Lextek: Onix Text Retrieval Toolkit – API Reference. http://www.lextek.com/manuals/onix/stopwords1.html (2011)
[19] Lin, D.: An Information-Theoretic Definition of Similarity. In: Shavlik, J.W. (ed.) 15th International Conference on Machine Learning (ICML 1998). pp. 296–304. Morgan Kaufmann (1998)
[20] Lovelyn Rose, S., Chandran, K.R.: Web knowledge and Wordnet based Automatic Web Query Classification. International Journal of Computer Applications 17(7), 23–28 (2011)
[21] Pedersen, T., Patwardhan, S., Michelizzi, J.: WordNet::Similarity – Measuring the Relatedness of Concepts. In: McGuinness, D.L., Ferguson, G. (eds.) 19th National Conference on Artificial Intelligence (AAAI 2004). pp. 1024–1025. AAAI Press / MIT Press (2004)
[22] Resnik, P.: Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In: 14th International Joint Conference on Artificial Intelligence (IJCAI 1995). pp. 448–453. Morgan Kaufmann (1995)
[23] 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)
[24] Salton, G., Buckley, C.: Term-Weighting Approaches in Automatic Text Retrieval. Information Processing and Management 24(5), 513–523 (1988)
[25] Sowa, J.F.: Semantic Networks. In: Shapiro, S.C. (ed.) Encyclopedia of Artificial Intelligence, vol. 2. Wiley (1992)
[26] Spell, B.: Java API for WordNet Searching (JAWS). http://lyle.smu.edu/~tspell/jaws/index.html (2011)
[27] Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In: Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLTNAACL 2003). pp. 252–259 (2003)
[28] Warin, M.: Using WordNet and Semantic Similarity to Disambiguate an Ontology. Tech. rep., Stockholms Universitet – Institutionen för Lingvistik (2004), http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.95.9729&rep=rep1&type=pdf
[29] Wu, Z., Palmer, M.S.: Verb Semantics and Lexical Selection. In: 32nd Annual Meeting of the Association for Computational Linguistics (ACL 1994). pp. 133–138. Association for Computational Linguistics (1994)