2006-11-13

Eliminating Redundant and Less-Informative RSS News Articles Based on Word Similarity and a Fuzzy Equivalence Relation

Ian Garcia

Yiu-Kai D. Ng
ng@cs.byu.edu

Eliminating Redundant and Less-Informative RSS News Articles Based on Word Similarity and a Fuzzy Equivalence Relation

Ian Garcia
Computer Science Department
Brigham Young University, Provo, Utah

Yiu-Kai Ng
Computer Science Department
Brigham Young University, Provo, Utah

Abstract

The Internet has marked this era as the information age. There is no precedent in the amazing amount of information, especially network news, that can be accessed by Internet users these days. As a result, the problem of seeking information in online news articles is not the lack of them but being overwhelmed by them. This brings huge challenges in processing online news feeds, e.g., how to determine which news article is important, how to determine the quality of each news article, and how to filter irrelevant and redundant information. In this paper, we propose a method for filtering redundant and less-informative RSS news articles that solves the problem of excessive number of news feeds observed in RSS news aggregators. Our filtering approach measures similarity among RSS news entries by using the Fuzzy-Set Information Retrieval model and a fuzzy equivalent relation for computing word/sentence similarity to detect redundant and less-informative news articles.

1 Introduction

During the past decades, besides abundant amounts of information on the Web, the way information is released has also been changed. "Just a decade ago, large-scale flows of information, such as news feeds, were owned, monitored, and filtered by organizations specializing in the provision of news. The Web has brought the challenges and opportunities of managing and absorbing news feeds to all interested users" [4]. The traditional way in which Internet users access news is by visiting a Web site, and revisiting the Web site to check for updates on any information. Typical Internet users are not interested in the information of only one Web site, instead they often visit several Web sites, diversified for various sources of information. If an Internet user wants to remain current with updates posted by those Web sites, (s)he would have to visit all the selected Web sites several times a day, which is a tedious and inefficient process. Since accessing online news is one of the favorite activities of Internet users[1], eliminating redundant and less-informative online news articles could help Internet users save time in locating useful information and avoid the frustrating process of filtering replicated information.

In March 1999, Dan Libby created a method for syndicating information (originated by UserLand Software in 1997) for use on the *My Netscape* portal, called RSS[2]. RSS is a series of XML formatted files for Web syndication which has become the de-facto standard for news portals and has been widely adopted to release other type of information, such as Weblogs, commercial Web sites, job listings, bug reports, and government information. As it has been claimed in *Feedster*[3], "Everything that is timely and valuable on the Web will be available as an RSS feed."

With the development of RSS, Internet users can (i) personalize the news they are interested in by including the headlines from the Web sites they access on a regular basis, and (ii) retrieve an RSS file containing not only the headlines, but a short description of each news and a link to the source of the news on one page, which is very useful. However, due to the large number of online news portals and huge amount of RSS news feeds, the user's challenge now is to process news articles in a timely manner. A possible solution to this problem is to deliver personalized news by removing redundant or less-informative news articles.

In this paper, we propose a filtering strategy for detecting and eliminating redundant and less-informative RSS news articles among the excessive number of news feeds entries. Our filtering approach measures similarity among RSS news entries by using the Fuzzy-Set Information Retrieval (IR) model and a word cluster for computing word/sentence similarity among news articles.

[1] According to the Newspaper Audience Database (NADbase), http://www.naa.org/nadbase, one in three Internet users (55 million) visited a news portal over the course of a month, which was incremented 21% from the year of 2005 to the first quarter of the year of 2006.

[2] The abbreviation is used to refer to the following standards: Rich Site Summary (RSS 0.91), RDF Site Summary (RSS 0.9 and 1.0), and Really Simple Syndication (RSS 2.0).

[3] Feedster (http://www.feedster.com) is one of the first Web sites to search, crawl, and index Weblogs.

Proceedings of the 18th IEEE International
Conference on Tools with Artificial Intelligence (ICTAI'06)
0-7695-2728-0/06 $20.00 © 2006 IEEE

We proceed to present our results as follows. In Section 2, we discuss related works in detecting similar documents. In Section 3, we introduce different word clusters and the Fuzzy-Set IR model for detecting similar and redundant RSS news articles. In Section 4, we present the detailed design on using a fuzzy equivalence relation for filtering less-informative RSS news articles. In Section 5, we give a concluding remark.

2 Related Work in Similarity Measures

Computing the similarity between documents has been extensively studied as an essential tool for applications such as text document searching [3], document clustering [16], copy or plagiarism detection [9, 12], text document retrieval, filtering, and categorization. Automatically determining whether two documents are similar and to what extent they are similar is a non-trivial problem.

The accuracy of similarity detection between two documents d_1 and d_2 relies heavily on computing the degree of similarity between d_1 and d_2, which can be determined by (i) the degree of lexical overlap in terms of the contents of d_1 and d_2, or (ii) the semantic contents, i.e., words/sentences, in d_1 and d_2. The semantic content approach goes beyond counting the number of words that appear in both d_1 and d_2, and the ability to assess the degree of semantic similarity between d_1 and d_2 automatically, scalably, and accurately is a key factor for justifying the effectiveness of information handling and decision support systems that detect similar text documents.

Many efforts have been made for computing the degree of similarity between two documents. From relatively simple programs, such as the *diff* command in UNIX/LINUX, which compares any two text documents in a line-by-line fashion and displays the contents and the line numbers where the two documents differ, through other more complex systems, such as COPS (COpy Protection System) [2], which is designed for detecting plagiarism. SIF [8] is one of the first copy-detection systems, which was intended not only for detecting similar text documents but binary documents as well. SIF, which considers the checksum of a file as its *fingerprint*, identifies similar files in a file system, and its approach is completely syntactic. COPS and SCAM (Stanford Copy Analysis Mechanism) [12] are two other copy-detection systems, which index a collection of documents by assigning hash values to sentences and paragraphs and comparing the hash values to determine the similarity among the corresponding documents. The main drawback of these copy detection approaches is the creation of a large number of collisions, same as other approaches that use hashing.

While SCAM is designed for comparing only small documents in a word-based fashion, document index graph (DIG) [5] is a document clustering system that uses a phrased-based matching model and an index model to detect similarity among documents. Even though *Diff, SIF, DIG, COPS*, and *SCAM* are different systems with different design goals, they all adopt lexical comparison approaches and thus fail to consider lexically different but semantically the same or similar documents. We further enhance the semantic comparison method in [10, 14], called Fuzzy-Set Information Retrieval (IR) model, for detecting similar, but not necessary the same, documents.

3 Word Clusters and the Fuzzy-Set IR Model

Detecting redundant and less-informative RSS news articles is a challenging task, since RSS news feeds are dynamic in nature. The technology of RSS allows Internet users to subscribe to Web sites that typically add or modify content regularly and rapidly. To use this technology, site owners create or obtain specialized software (such as a content management system), which is in the machine-readable XML format, and present new articles in a list, including a line or two of each article and a link to the full article. (See, as an example of, an RSS news feed file as shown in Figure 1.) One of the essential elements in an RSS file is *Channel*, which contains several sub-elements that describe the information contained in the file. Sub-elements of *Channel* include (i) *title*, which is similar to the *title* tag in an HTML file and is used for identifying the RSS news feed. Usually, the content of *title* is the name of the Web site that provides the RSS file. (ii) *Link*, which is the URL of the Web site from where the RSS file can be retrieved. (iii) *Description*, which includes a sentence that briefly describes what the "stories," i.e., news articles, contained in the file are about, such as politics, economics, sports, etc. (iv) *Item*, which is a story identified by an <item> tag. Several *items* can be specified in the *Channel* element. The most important sub-elements of an *item* are (a) *title*, which contains the headline of the story, (b) *link*, which is the URL where the story in full can be retrieved, (c) *description*, which contains a few lines about the story and many times it is the first sentences of the story, and (d) *pubDate*, which is the date and time when the story is posted. We treat an *item* as a tuple of an RSS news feed, which contains the elements in the <item> tag, i.e., *title*, *link*, *description*, and *pubDate*.

We propose a selective filtering approach using the Fuzzy-Set IR model, where RSS news entries to be filtered are the ones that provide less-information or possess information which is already included in another RSS news entries, i.e., redundant, from either a different or the same RSS news feed. This can be achieved by comparing the RSS news entries and determining the similarity among them.

3.1 The Fuzzy-Set IR Model

The fuzzy set theory relies on two main concepts: (i) sets are not well-defined, and (ii) an element (e.g., a word)

Proceedings of the 18th IEEE International
Conference on Tools with Artificial Intelligence (ICTAI'06)
0-7695-2728-0/06 $20.00 © 2006 IEEE

Figure 1. Portion of a RSS news feed file.

has a degree of membership to a set which falls within the range of the real interval $[0, 1]$ [6]. In [14], the Fuzzy-Set IR model is adopted to determine whether a keyword in a sentence belongs to a (fuzzy) set of words, which have certain degrees of similarity among themselves. The degrees of similarity, refereed as *correlation factors*, among words are given by a function which assigns a value in the range $[0, 1]$ to any two words. There are several methods to define the correlation factors among different words.

3.2 The Correlation Factor

The Fuzzy-Set IR model makes use of *correlation factors*, each of which is a similarity measure of any two words w_1 and w_2, i.e., the degree of similarity between w_1 and w_2. The correlation factor between w_i and w_j $(i, j \geq 1)$ is given in a symmetric, correlation matrix M, where each entry $m_{i,j} \in M$ is the correlation factor between w_i and w_j (also called *keywords* in [1]) and $m_{i,j} \in [0, 1]$.

In order to generate a correlation matrix of correlation factors, we must first collect a large number of "representative" English documents, which should be *unbiased* in terms of writing styles and *diversed* in contents, to calculate the correlation factors among distinct words according to their occurrences within each document of the collection. Some of the most popular sets of (archive) documents used in related projects are the TREC collection (http://trec.nist.gov/) and the Gutenberg project (http://www.gutenberg.org). These sets of documents, however, have major drawbacks. The Gutenberg project, which is a collection of books that is periodically augmented with new books, lacks a variety of topics, especially in science and technology. Even though the TREC collection includes a wide variety of topics, its public version has not been updated for several years. For these reasons, we have chosen the Wikipedia collection. Wikipedia [13] is a free online encyclopedia, which contains more than 930,000 articles and approximately 340 million words. The collection of Wikipedia articles, which are written by more than 89,000 volunteers, overcomes the drawbacks of the TREC and Gutenberg collections, since Wikipedia contains almost all possible topics in different areas of study, is constantly updated, and is unbiased in terms of writing styles and authorship. With the use of the Wikipedia documents to generate word-correlation factors, we can obtain a reliable similarity measures of different words according to their occurrences in various documents.

The Wikipedia articles are comprised in a single XML file of approximately 4.6Gb in size. We first filtered the content in the "title" and "text" tags of each Wikipedia article. The filtering process requires two steps: (i) removing stopwords [1] and (ii) stemming remaining words [11]. Stopwords, which are words that are very common, such as prepositions and demonstrative, interrogative, and indefinite pronouns, do not provide useful information to distinguish the contents of different documents. Stopword removal is accomplished by verifying if a word is contained in the stopwords hash table, which is constructed by using several widely used stopword lists. Once the stopwords are removed, we proceed to stem all the remaining words using the Porter algorithm [11]. Quoting Martin Porter himself [11]: "The Porter stemming algorithm (or Porter stemmer) is a process for removing the commoner morphological and inflexional endings from words in English. Its main use is as part of a term normalization process that is usually done when setting up Information Retrieval systems." Hereafter, we processed the remaining non-stop, stemmed words[4] and created a *word-frequency-location file* in which each record consists of (i) one of the words w, (ii) the document D (identified by document number) where w appears, (iii) the frequency of occurrence of w in D, and all the positions where w is present in D. (See Table 1 for portion of the word-frequency-location file.) The generated word-frequency-location file contains 144,048,788 records, and the number of non-stop, stemmed words in the file is 57,926. The corresponding symmetric matrix contains 1,677,681,775 = 57,926 $\times$ 57,925 / 2 entries and takes up 6.3Gb of disk space where each entry is stored as a 32-bit floating point number. Once the word-frequency-location file is generated, we can calculate a word-word correlation matrix by considering either (i) the number of documents (i.e., Wikipedia articles) in which two distinct terms w_1 and w_2 appear, which yields the *keyword-connection* correlation factors (matrix), (ii) the frequency of *co-occurrence* correlation factors of w_1 and w_2 in each document, which yields the co-occurrence matrix, and (iii) the distance correlation factors between w_1 and w_2 in each document, which yields the distance matrix.

[4]From now on, unless stated otherwise, whenever we use the term "word," we really mean "non-stop, stemmed word."

Keyword	Doc (D)	Frequency	Positions in D
computer	5	5	3, 21, 40, 32, 88
computer	12	3	14, 45, 100
comrade	1	2	30, 64
comrade	21	3	8, 30, 58

Table 1. Some keyword Frequencies and Positions in the word-frequency-location file

3.2.1 Keyword Connection

The *(key)word-connection* correlation factor, which has been used for comparing similarity among documents, calculates the correlation of any two words w_1 and w_2 by counting the number of documents in a collection C where both w_1 and w_2 appear together [1, 10]. In the keyword-connection matrix [10], each entry $m_{i,j}$, i.e., the *correlation factor*, for w_i and w_j is calculated as

$$c_{i,j} = \frac{n_{i,l}}{n_i + n_l - n_{i,l}} \tag{1}$$

where n_i (n_l, respectively) is the number of documents in C in which the keyword w_i (w_l, respectively) appears, and $n_{i,l}$ is the number of documents in C in where w_i and w_l appear. The correlation factors of different keywords computed by using the *keyword-connection* method follow the conjuncture that "The more documents in which two keywords occur, the more they relate to each other." [10]

The keyword-connection matrix is simpler to compute than the co-occurrence and the distance matrices; however, a major drawback of this simplicity is accuracy, since keyword-connection correlation factors do not consider other factors among different words, which include (i) the *frequency* of co-occurrence of any two words within a document and (ii) how *close* any two words appear together in a document. These factors are further considered by the co-occurrence frequency matrix (i.e., *association cluster* in [1]) and the distance matrix (i.e., *metric cluster* in [1]), respectively in computing the correlation factors of any two words.

3.2.2 Co-Occurrence Frequency

The *co-occurrence correlation factor* not only considers the number of documents in a collection where both words w_1 and w_2 appear, but it also considers the frequency of co-occurrence of both w_1 and w_2 in a document. In order to compute the co-occurrence factors, we obtain a frequency matrix m where each entry f_{i,d_u} is the frequency of word i in document d_u. The composition of m and its transpose, i.e., m^t, yields the matrix $c = m \cdot m^t$, where each entry $c_{i,j}$ of the matrix is the co-occurrence frequency factor of words w_i and w_j, i.e.,

$$c_{i,j} = \sum_{n=1}^{l} (f_{i,d_n} \times f_{j,d_n}) \tag{2}$$

where l is the total number of documents in a collection. We can normalize each correlation factor to limit the values in the interval [0, 1] as

$$c_{i,j_{norm}} = \frac{c_{i,j}}{c_{i,i} + c_{j,j} - c_{i,j}} \tag{3}$$

The frequency of co-occurrence of any two words in a document yields a more accurate correlation factor of the words than using the keyword-connection approach. Consider in an Wikipedia article about Shakespeare (http://en.wikipedia.org/wiki/ Shakespeare) in which *Shakespeare* appears sixty times, *english* appears twenty times, and *french* appears only one time. If we use the keyword-connection method, the correlation factor of *Shakespeare* and *english* will be the same as the correlation factor between *Shakespeare* and *french*. However, it is clear that the correlation factor between *Shakespeare* and *english* should be higher.

Even though the co-occurrence frequency matrix considers the co-occurrence frequency of any two words, it does not consider how close any two words are as they appear in a document, which is another important factor in providing an accurate correlation factor of any two words.

3.2.3 The Distance Correlation

The *distance correlation factor* between any two words w_1 and w_2 considers the frequency of occurrence, as well as the "distance," that is measured by the number of words, between w_1 and w_2 within a document as an additional factor to calculate the correlation factor of w_1 and w_2. In the distance correlation approach, it is assumed that keywords which appear closer together are more likely related than those that appear far apart in the same document. For example, in a document that discusses computer architecture, the two words "computer" and "architecture" are likely to appear closely together most of the times, and their correlation factor computed by using their distances would be higher if their positions are considered, along with their frequency of co-occurrence. However, if a document discusses the usage of computers by architects for creating the blue prints of home designs, the keyword-connection or co-occurrence correlation factor, could give the same correlation factor to "computer" and "architecture" in both documents, which is inaccurate, since it is clear that in the document talking about computer architecture, the correlation factor is higher between the two words, whereas in the document about architects using computers, the correlation factor of the same two words should be smaller.

The distance $d(w_i, w_j)$ between any two words w_i and w_j is defined by the (absolute) difference of the positions of any occurrence of w_i and w_j in a document, i.e., $d(w_i, w_j) = |Position(w_i) - Position(w_j)|$, and $d(w_i, w_j) = \infty$ when w_i and w_j do not appear in the same document. For each document d_u in a collection C, the distance among each occurrence of w_i and each occurrence of w_j in d_u is calculated, and the correlation factor of w_i and w_j, i.e., $c_{i,j}$, is computed as the sum of the inverse of the distances between any occurrence of w_i and w_j:

$$c_{i,j} = \sum_{w_i \in V(s_i)} \sum_{w_j \in V(s_j)} \frac{1}{d(w_i, w_j)} \quad (4)$$

where $V(S_i)$ ($V(S_j)$, respectively), denotes the sets of words that include w_i (w_j, respectively) and its respective stemmed words. We normalize the distance correlation factors in the interval [0, 1], as in the other correlation matrices, to redefined $c_{i,j}$ as

$$s_{i,j} = \frac{c_{i,j}}{|V(S_i)| \times |V(S_j)|} \quad (5)$$

Event though the calculation of the distance matrix is more complex than the keyword-connection and the co-occurrence matrices, the distance matrix is computed only *once* and captures the correlation factors of two words w_i and w_j more accurately than the other two, which will be verified in Section 3.3. We adopt the distance matrix approach in measuring the degrees of similarity among different words, which is used in the Fuzzy-Set IR model, for detecting redundant or less-informative RSS news entries.

3.3 Word-Sentence-Document Fuzzy Association

Once a word-to-word correlation matrix is calculated, we can define a fuzzy set association of each word and a sentence, paragraph, or document itself. Since a news article in RSS news feeds includes only a brief summary, i.e., the 3-4 line summary, in the Title and Description sections, we treat the Title and Description sections as the content of an RSS news article such that the degree of similarity between any two RSS news articles is determined by the correlation factors among the words in the respective Title and Description sections of the two articles[5].

The association between a keyword k_i and a document d_j (i.e., an RSS news article in this paper), referred in [10] as the word-document correlation factor $\mu_{i,j}$, is calculated as the complement of a negated algebraic product of all the correlations of the (key)word k_i and each (key)word $k_l \in d_j$, i.e.,

$$\mu_{i,j} = 1 - \prod_{w_k \in d_j} (1 - c_{i,k}) \quad (6)$$

The correlation value $\mu_{i,j}$ falls in the interval [0, 1] and reaches its maximum when $c_{i,k} = 1$, for any $k \in d_j$. In [14], the $\mu_{i,j}$ factor is modified to compute the *word-sentence* correlation factor between word i and sentence j, instead of computing the word-document correlation factor as defined in [10]. In this paper, we adopt the modification, i.e., $\mu_{i,j}$, as the correlation factor between keyword k_i and sentence S_j. Since each RSS news article contains at most three short sentences (on the average in its RSS file as explained earlier), we treat them as a *single* sentence.

The degree of similarity of sentence S_i (in an RSS news article) with respect to sentence S_j (in another RSS news article), denoted as $Sim_{i,j}$, is calculated as the average of all the values $\mu_{i,j}$, where $w_l \in S_i$ ($1 \leq l \leq n$), with respect to (all the words in) S_j as

$$Sim_{i,j} = \frac{\mu_{w_1,j} + \mu_{w_2,j} + \cdots + \mu_{w_n,j}}{n} \quad (7)$$

and $Sim_{i,j} \in [0, 1]$. It is important to note that in general, $Sim_{i,j} \neq Sim_{j,i}$. When $Sim_{i,j}$ = 0, it indicates that there are no words in sentence S_i that can be considered similar to any word in sentence S_j. If $Sim_{i,j}$ = 1, then either sentence S_i is (semantically) identical to sentence S_j, or S_i is subsumed by sentence S_j, i.e., all the words in S_i are (semantically) the same as (some of) the words in S_j.

An EQ function of $Sim_{i,j}$ and $Sim_{j,i}$ is defined below, which determines whether sentences S_i and S_j should be considered as (semantically) the same.

$$EQ(S_i, S_j) = \begin{cases} 1 & \text{if } Min(Sim_{i,j}, Sim_{j,i}) \geq \alpha \\ & \quad \wedge \mid Sim_{i,j} - Sim_{j,i} \mid \leq \epsilon \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

The parameter α is called the *permission threshold value*, and it determines the *minimum similarity* value for which S_i and S_j can be considered to be (semantically) the same. The second parameter ϵ is called the *variation threshold value*, which defines the *maximum difference* (in terms of the degree of similarity), between S_i and S_j, for which S_i and S_j can be considered to be (semantically) the same. The values for α and ϵ were empirically obtained using a data set consisting of twenty-six RSS news articles retrieved from different RSS news feeds, which include Associated Press (http://www.ap.com), Yahoo (http://news.yahoo.com), and Boston Globe (http://www.bostonglobe.com). The number of possible combinations of the matching news articles pairs is 26 × 25 ÷ 2 = 325. (We manually evaluated them and counted a total of 17 matching pairs among all the news articles.) We considered the word-correlation factors in each

[5] If the actual contents of two RSS news articles n_1 and n_2 are used instead of the summaries of n_1 and n_2, the accuracy of the degree of similarity of n_1 and n_2 could be further enhanced, assuming that the processing time is not a main concern.

Correlation Matrix	FP	FN	M	Accuracy %
Keyword-connection	2	9	8	47
Co-ocurrance	1	8	9	52
Distance	3	1	16	**94**

FP: F(alse)P(ositive); FN: F(alse)N(egative); M(atches)

Table 2. Experimental Results on twenty-six RSS news using the three different matrices

of the different matrices, i.e., the keyword-connection, co-occurrence, and distance matrices, that were previously created using the Wikipedia articles and tested for the degrees of similarity among the twenty-six articles. We counted (i) the number of pairs of RSS news entries that were considered *redundant* or *less-informative* when in fact they are not (i.e. *false positives*) and (ii) the number of pairs that were considered *different* but are in fact *redundant* or *less-informative* (i.e., *false negatives*).

The results we obtained by using each of the three matrices are shown in Table 2, which demonstrates that the correlation factors in the *distance matrix* provide the most accurate results among all the three matrices. These experimental results confirm the correct choice of the distance matrix and the EQ function for detecting redundant or less-informative news articles.

4 Fuzzy Equivalence Relation

By adopting the Fuzzy-Set IR model and the distance matrix, we can discard *redundant* RSS news articles, i.e., articles that do not provide new information. This task can be accomplished by eliminating any RSS news article s such that $Sim_{s,t}$ = 1, where t is another RSS news article, which implies that the information contained in s is entirely subsumed by t. Hereafter, we proceed to eliminate less-informative RSS news articles. This task can be accomplished by first generating clusters of all the RSS news articles that maintain certain degree of similarity. Note that news articles in the same cluster should be closely related, whereas distinct news articles belonged to different clusters should be significantly different in terms of their degrees of similarity. By deleting news articles in a non-singleton cluster C, which have the highest degree of similarity with respect to other news articles in C, we can eliminate less-informative news articles.

Clusters of news articles can be created by adopting a fuzzy equivalence relation that applies to the news articles to generate "crisp" subsets (i.e., clusters). News articles in each cluster C that are "less-informative" than others in C are discarded. A cluster of RSS news articles is defined as $C_\alpha = \{d \mid Sim_{d,e} \geq \alpha, \forall e \in C_\alpha\}$, where α is the *minimum degree of similarity* such that any two news articles in C_α must hold. We generate clusters of non-redundant RSS news articles collected from various RSS news feeds by considering a fuzzy equivalence relation as given in [6]. A fuzzy equivalence relation defines a "crisp" equivalence relation among the elements of a set, which have been widely studied to measure the degree of similarity among different elements in a set. A (similarity) relation R is a *fuzzy equivalence relation* if it is reflexive, symmetric, and $max\text{-}min$ transitive, i.e.,

$$R(x,x) = 1, \ \ \forall x \in R \tag{9}$$

$$R(x,y) = R(y,x), \ \ \forall x,y \in R \tag{10}$$

$$R(x,z) \geq \max_{y \in Y} \min\{R(x,y), R(y,z)\} \tag{11}$$

where Y is a fuzzy set and $x, y, z \in Y$.

The key for establishing a fuzzy equivalence relation is the definition of transitivity. The first definition for fuzzy transitivity was proposed by Zadeh [15], which is called the $max\text{-}min$ transitivity, as defined in Equation 11. However, the $max\text{-}min$ transitivity is known to be a restrictive constraint, which is not applicable to the similarity relation problem that we deal with. This is because in order to apply the $max\text{-}min$ transitivity to our similarity problem, it is required that for any two news articles (i.e., documents) d_x and d_z, there cannot exist another article d_y whose similarities with both d_x and d_z is greater than the similarity between d_x and d_z, i.e., the relation R is not $max\text{-}min$ transitive if given d_x and d_z, there exists d_y such that $R(d_x, d_z) < R(d_x, d_y)$ and $R(d_x, d_z) < R(d_y, d_z)$. Consider an example with the following sentences and the degrees of similarity $Sim_{1,2}$ = 0.20, $Sim_{1,3}$ = 0.80, $Sim_{2,1}$ = 0.33, $Sim_{2,3}$ = 1.00, $Sim_{3,1}$ = 0.50, and $Sim_{3,2}$ = 0.37:

S_1: Bush's proposal will benefit illegal immigrants.

S_2: The war in Iraq is not over said Bush.

S_3: Bush proposed a bill that will benefit immigrants and he said the war in Iraq will continue.

In order to establish a fuzzy equivalence relation that satisfies the $max\text{-}min$ transitivity for S_1, S_2, and S_3, it is necessary to define a relation (function) R of the similarity values among S_1, S_2, and S_3 such that

$$f(0.20, 0.33) \geq \min\{f(0.80, 0.50), f(1.0, 0.37)\} \tag{12}$$
$$f(0.80, 0.50) \geq \min\{f(0.20, 0.33), f(1.0, 0.37)\} \tag{13}$$
$$f(1.00, 0.37) \geq \min\{f(0.80, 0.50), f(0.33, 0.2)\} \tag{14}$$

We consider a function that takes two input values, which are the similarity measures between two documents i and j and returns a high value if the two input values are high. The function f as shown in Equations 12, 13, and 14 does not satisfy this criteria, since f in each of the inequality equations yields a high value when the given values are low.

Proceedings of the 18th IEEE International
Conference on Tools with Artificial Intelligence (ICTAI'06)
0-7695-2728-0/06 $20.00 © 2006 IEEE

There exists another fuzzy transitivity relation, the max-$prod$ transitivity [6] (as defined below), which can be adopted to establish a fuzzy equivalence relation.

$$R(x,z) \geq \max_{y \in Y}\{R(x,y) \times R(y,z)\} \quad (15)$$

where x, y, z, and Y are as defined in Equation 11.

The max-$prod$ transitivity is not as restrictive as the max-min transitivity and can be more easily satisfied by a function R whose values fall in the interval $[0, 1]$, since the product of two numbers $x, y \in [0, 1]$ in the max-$prod$ transitivity is smaller than x and y, i.e., if $x, y \in [0, 1]$, then $x \geq x \times y$ and $y \geq x \times y$. For this reason, we consider the max-$prod$ transitivity instead of the max-min transitivity.

In order to adopt the fuzzy equivalence relation with max-$prod$ transitivity for eliminating less-informative RSS news articles, it is necessary to define a function that combines the similarity measures of two news articles d_i and d_j, i.e., $Sim_{i,j}$, and $Sim_{j,i}$, into a single one. We consider several functions and choose Q in Equation 16 that satisfies the max-$prod$ transitivity property as the desired function.

4.1 Combination Functions

One of the most commonly used combination equations is *average*. However, the average of two pairs of different similarity values, e.g., (0.5, 0.5) and (0.9, 0.1), can yield the same result, e.g., (0.5 + 0.5)/2 = (0.9 + 0.1)/2. In addition, the average function is not max-$prod$ transitive.

In [14], an $EQ(S_i, S_j)$ function of the similarity values of two documents d_i and d_j, i.e., $Sim_{i,j}$ and $Sim_{j,i}$, is defined (as given in Equation 8), which determines whether d_i and d_j should be considered as the same. The EQ function, which is a discrete function, assigns the values 1 or 0, and is symmetric and reflexive; however, EQ is neither max-min transitive nor max-$prod$ transitive. Furthermore, one of the drawbacks of EQ is that its two estimated threshold parameter values, i.e., α and ϵ, must be established before EQ can be adopted.

In [7], two combination equations that combine two values, e.g., $Sim_{i,j}$ and $Sim_{j,i}$, are defined as follows:

$$Q(Sim_{i,j}, Sim_{j,i}) = \frac{Sim_{i,j} + Sim_{j,i}}{1 - min(Sim_{i,j}, Sim_{j,i})} \quad (16)$$

$$Q_1(Sim_{i,j}, Sim_{j,i}) = (Sim_{i,j} + Sim_{j,i}) - (Sim_{i,j} \times Sim_{j,i}) \quad (17)$$

Both functions Q and Q_1 are simple to compute; however, Q_1 has the similar drawback as the average function, i.e., it yields the same result to different pairs of values. For example, both (similarity) value pairs (0.9, 0.9) and (0.99, 0.1) are assigned the same value 0.99 by Q_1. In contrast, Q assigns a high value only when both similarity measures of two documents d_i and d_j are high. Furthermore, function Q in Equation 16 is fuzzy-symmetric, however, Q is neither fuzzy-reflexive nor fuzzy-transitive. We modify Q so that the modified Q function, i.e., E (given below), is a fuzzy equivalence relation.

$$E(d_i, d_j) = \begin{cases} 1 & \text{if } i = j \\ 0.0001 & \text{if } Q(d_i, d_j) < 0.0001 \\ \frac{Q(X,Y)}{\max(Q(W,Z))} & \text{otherwise} \end{cases} \quad (18)$$

where X and Y are $Sim_{i,j}$ and $Sim_{j,i}$, respectively, and W and Z are $Sim_{l,k}$ and $Sim_{k,l}$, respectively for any d_k and d_l in the set of documents which includes d_i and d_j.

The first condition in Equation 18 is introduced to satisfy *reflexivity*, whereas the third condition is the normalized Q function, which restricts the values of E to the interval [0, 1]. The second condition guarantees max-$prod$ *transitivity*.

We have experimentally verified that the E function is max-$prod$ transitive, using twelve sets of news articles S from different RSS news feeds and evaluating the max-$prod$ transitivity inequality (as given in Equation 15) for each set of news articles. We consider every possible pair of news articles d_x and d_z in each RSS news feed and compute every possible $E(d_x, d_y) \times E(d_y, d_z)$ value to verify that $E(d_x, d_z) \geq E(d_x, d_y) \times E(d_y, d_z)$, for every d_y. The paramenter value 0.0001, which was obtained empirically using S, limits the similarity value between any two news articles so that the max-$prod$ transitivity is satisfied. It is essential to understand that the value 0.0001 is not simply set up to satisfy the max-$prod$ transitivity. In fact, given any two documents d_1 and d_2 that are similar, if there exists another document d_3 such that d_1 and d_3, as well as d_3 and d_2, are similar, then the max-$prod$ transitivity using 0.0001 always holds for d_1, d_2, and d_3.

4.2 Clustering and Discarding Less-Informative News Articles

After the fuzzy equivalence relation E is established, we can apply E to determine the equivalence classes (clusters) of news articles from different RSS news feeds by setting an α-cut value [6]. As the value of α increases, the number of equivalence classes of the α-cut also increases, whereas the size of each equivalence class is reduced. An α-cut value guarantees that every pair of news articles in the same cluster has the degree of similarity at least equal to α. We discard one or more (but not all) news articles from each cluster. We do not consider clusters with only one article, since they include the only news article which is dissimilar to other news articles in other clusters, assuming that our equivalence class generation approach is correct.

Since the same news article may appear in different clusters[6], we cannot consider each cluster separately while se-

[6] A fuzzy equivalence relation based on the max-$prod$ transitivity does

lecting less-informative news articles to discard. Instead, we rank the news articles in different clusters generated by an α-cut value and discard those that have higher rankings, i.e., articles that are highly similar to others and thus are "less-informative," in the same cluster. While discarding news articles from different clusters, we cannot rank the news articles simply based on their similarity values, since we could discard all the news articles in a cluster. Consider the set of news articles $C = \{a, b, c, d, e, f\}$ and their similarity values: $Sim_{a,b}$ = 0.8, $Sim_{b,a}$ = 0.85, $Sim_{c,d}$ = 0.6, $Sim_{d,c}$ = 0.7, $Sim_{e,f}$ = 0.61, and $Sim_{f,e}$ = 0.5, whereas all the other similarity values are 0.2, except the similarity value of the same pair of documents, which is 1.0. Using Equation 18, the E value of each pair of news articles can be computed by using their similarity values, i.e., $E(a, b)$ = 0.055, $E(c, d)$ = 0.021, $E(e, f)$ = 0.014, whereas all the other possible pairs yield the same E value, i.e., 0.0025. If we set α = 0.1, then three clusters $C_1 = \{a, b\}$, $C_2 = \{c, d\}$, and $C_3 = \{e, f\}$ are generated. Suppose we need to discard two news articles. If we simply discard the top two articles with the highest similarity values, then we would discard a and b in C_1, which is an undesirable result.

4.3 Different Ranking Approaches

We compare three different approaches in ranking the news articles in each cluster generated by an α-cut value. In the first approach, we consider the *entropy value* of each news article. The entropy in information theory is an alternative way to describe how much information is carried in a document and thus can be used to determine the relative degrees of similarity among different news articles. We measure the entropy of each news article by using

$$H(x) = -\sum_{i=1}^{N} p(i) \log_2 p(i) \tag{19}$$

where x represents a non-redundant news article and N is the number of clusters to where x belongs.

We have chosen the Q function in Equation 16 to compute probability p in Equation 19 for each news article. Table 3 shows the ranking according to the entropy values of a set of news articles (collected from various RSS news feeds) in different clusters generated by an α-cut value. By using the number of occurrences of each news article in different clusters, we realize that there is a clear bias towards those articles that have higher frequency of occurrences, which is not a desired behavior, since news articles that appear several times in different clusters are not necessarily the ones that have the higher degree of similarity with a particular news article than the remaining news articles.

not always yield disjoint equivalence classes, which is different from the fuzzy equivalence relation based on the max-min transitivity.

Average M			**Entropy** (H)			**Average** E		
Art.	F	Rank	Art.	F	Rank	Art.	F	Rank
204	1	0.93	36	4	0.27	222	1	0.06
41	1	0.93	67	3	0.24	218	1	0.06
299	2	0.87	137	1	0.15	204	1	0.02
116	1	0.86	299	2	0.13	324	1	0.02
36	4	0.76	89	2	0.10	89	2	0.01
271	3	0.72	324	1	0.10	49	1	0.01
67	3	0.71	415	1	0.09	299	2	0.01
...	...	...	...	...	...	...	...	...

Art(icle): identified by its article number; F(requency)

Table 3. Article rankings using Equations 20, 19, and 18.

We have considered another ranking, which is the *average* of the E values (in Equation 18) of news articles. The average includes all the E values obtained between a document d_i and each of the other documents d_j that appear in each of the clusters to where d_i belongs. However, the average of the E values does not yield a good ranking either, since the average of the E values assigns a low ranking to articles 41 and 36. Article 41 should have a high ranking and be discarded, since it is very similar to article 28, and article 36 should also receive a high ranking, since (i) it appears in several clusters and (ii) has high similarity values with those news articles in the clusters. Instead of adopting the entropy or the average E values, we use the function M in Equation 20 to rank news articles in all clusters that include at least two news articles generated by an α-cut value. M provides the *average* of the maximum similarity values of a news article d_i with another news article d_j in each cluster where d_i appears.

$$M(d_i) = \frac{\sum_{k=1}^{N} \max_{C_k}\{Sim_{i,j}\}}{N} \tag{20}$$

The ranking based on the average of H values considers the trade off between the (i) the *frequency* of a news article in clusters generated by an α-cut value and (ii) the *similarity values* of a news article with respect to other news articles in various clusters, which yields the most accurate ranking on news articles according to the conducted experiments. According to the rankings, the top n ($n \geq 1$) ranked (i.e., less-informative) news articles, along with all the redundant ones that have been detected earlier, are discarded, assuming that their frequencies of occurrence is greater than 1, i.e., they occur in non-singleton clusters.

Example 1 Consider a set of ten RSS news articles that were extracted from various RSS news feeds as shown in Table 4 and the non-singleton clusters that were generated as shown in Figure 2. The clusters and the computed rankings of the ten articles can be used to determine which one(s) of the ten articles should be discarded.

Proceedings of the 18th IEEE International
Conference on Tools with Artificial Intelligence (ICTAI'06)
0-7695-2728-0/06 $20.00 © 2006 IEEE

ID	Title and URL
a_0	*House Considers Bill to Boost Refineries*. abcnews.go.com/Business/wireStory?id=1191725&amp
a_1	*Intruder Gains Front Lawn of White House*. abcnews.go.com/Politics/wireStory?id=1823761
a_2	*Screaming Intruder Jumps White House Fence*. www.examiner.com/a-73142
a_3	*Saudi Ambassador: Iraq Invasion Helped Spread Terrorism.* www.foxnews.com/story/0,2933,178188,00.html
a_4	*Bush says Iraqi parliamentary elections "a major step forward"*. english.people.com.cn/200512/16/eng20051216
a_5	*Powerball Winners Donate to Homeless*. www.foxnews.com/story/0,2933,190670,00.html
a_6	*mSleep disorders affect millions of Americans*. english.people.com.cn/200604/05/eng20060405_256141.html
a_7	*Millions of Americans suffer sleep disorders*. news.xinhuanet.com/english/2006-04/05/content_4386829.htm
a_8	*Gasoline price drop may only be temporary*. www.boston.com/news/nation/washington/articles/2005/09/19/
a_9	*House considers bill to boost refineries*. abcnews.go.com/Business/wireStory?id=1191983&page=1

Table 4. Ten RSS news articles downloaded from various RSS news feeds and their URLs

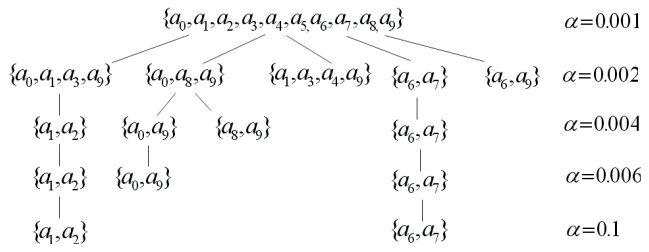

Figure 2. Clusters generated on the set of 10 RSS news articles as shown in Table 4 according to different α-cut values

The number of "less-informative" news articles to be discarded can be determined by using (i) the average number of new articles that are accessed by an individual user, and (ii) the number of (new or updated) articles posted by each individual RSS news feed that the user accesses on a regular basis.

5 Conclusions

In this paper, we propose a solution to the RSS news information overflow problem by providing a filtering approach which selectively eliminates redundant or less-informative RSS news feeds entries. The proposed filtering approach, which adopts the Fuzzy-Set IR model, a distance matrix, and a fuzzy equivalence relation with max-$prod$ transitivity, can detect redundant and less-informative RSS news articles accurately. Furthermore, we have fully implemented our filtering approach in PERL and JAVA.

For future work, we would like to investigate using natural language processing, user personal profiles, and unsupervised machine learning approaches to further enhance our filtering method by taking into consideration the user preference and other semantic constraints detected from source documents.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[2] S. Brin, J. Davis, and H. Garcia-Molina. Copy detection mechanisms for digital documents. In *Processings of ACM SIGMOD*, pages 398–409, 1995.

[3] S. Chien and N. Immorlica. Semantic similarity between search engine queries using temporal correlation. In *Processings of the World Wide Web Conf.*, pages 2–11, 2005.

[4] E. Gabrilovich, S. Dumais, and E. Horvitz. Newsjunkie: Providing personalized newsfeeds via analysis of information novelty. In *Processings of the World Wide Web Conference*, pages 482–490, 2004.

[5] K. Hammouda and M. Kamel. Efficient phrase-based document indexing for web document clustering. *IEEE TKDE*, 16(10):1279–1296, 2004.

[6] G. Klir, U. St. Clair, and B. Yuan. *Fuzzy Set Theory, Foundations and Applications*. Prentice Hall, 1997.

[7] G. Luger. *Artificial Intelligence, Structures and Strategies for Complex Problem Solving*. Addison Wesley, 1997.

[8] U. Manber. Finding similar files in large file system. In *Processings of the USENIX Winter Technical Conf.*, 1994.

[9] H. Nevin. Scalable document fingerprinting. In *Processings of the 2^{nd} USENIX Workshop on Electronic Commerce*, pages 191–200, 1996.

[10] Y. Ogawa, T. Morita, and K. Kobayashi. A fuzzy document retrieval system using the keyword connection matrix and a learning method. *Fuzzy Sets & Systems*, 39:163–179, 1991.

[11] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[12] N. Shivakumar and H. Garcia-Molina. The scam approach to copy detection in digital libraries. *Diglib Magazine*, 1995.

[13] Wikipedia. http://wikipedia.org/.

[14] R. Yerra and Y.-K. Ng. Detecting similar html documents using a fuzzy set information retrieval approach. In *Processings of IEEE International Conference on Granular Computing (GrC'05)*, pages 693–699, 2005.

[15] L. Zadeh. Similarity relations and fuzzy orderings. *Information Sciences*, 3:177–200, 1970.

[16] Y. Zhao and G. Karypis. Topic-driven clustering for document datasets. In *Processings of SIAM International Conference on Data Mining*, pages 358–369, 2005.