Helping Users in Web Information Retrieval Via Fuzzy Association Rules

M.J. Martín-Bautista[1], D. Sánchez[1], J.M. Serrano[2], and M.A. Vila[1]

[1] University of Granada. Department of Computer Science and Artificial Intelligence. C/ Periodista Daniel Saucedo Aranda s/n, 18071 Granada, Spain
mbautis@decsai.ugr.es, daniel@decsai.ugr.es, vila@decsai.ugr.es
[2] University of Jaén. Department of Computer Science. Campus Las Lagunillas, 23071 Jaén, Spain
jschica@ujaen.es

Summary. We present an application of fuzzy association rules to find new terms that help the user to search in the web. Once the user has made an initial query, a set of documents is retrieved from the web. Representing these documents as text transactions, each item in the transaction means the presence of the term in the document. From the set of transactions, fuzzy association rules are extracted. Based on the thresholds of support and certainty factor, a selection of rules is carried out and the terms in those rules are offered to the user to be added to the query and to improve the retrieval.

1 Introduction

Finding information in the web is not so easy as users expect. Most of the documents retrieved as a result of a web search meet the search criteria but do not satisfy the user's preferences. Generally, this can be due to a not suitable formulation of the query, either because the query terms of the user does not match the indexed terms of the collection, or because the user does not know more vocabulary related to the search topic at the query moment.

To solve this problem, the query can be modified by adding or removing terms to discard uninteresting retrieved documents and/or to retrieve interesting documents that were not retrieved by the query. This problem has been named as *query refinement* or *query expansion* in the field of Information Retrieval [14].

In this work, we propose the use of mining techniques to solve this problem. For this purpose, we use fuzzy association rules to find dependence relations among the presence of terms in an initial set of retrieved documents. A group of selected terms from the extracted rules generates a vocabulary related to the search topic that helps the user to refine the query with the aim of improving the retrieval effectiveness. Data mining techniques have been applied

M.J. Martín-Bautista et al.: *Helping Users in Web Information Retrieval Via Fuzzy Association Rules*, StudFuzz **197**, 221–237 (2006)
www.springerlink.com

successfully in the last decade in the field of Databases, but also to solve some classical Information Retrieval problems such as document classification [26] and query refinement [35].

This paper is organized as follows: a survey of query refinement solutions found in the literature is given in Section 2. The concepts of association rules, fuzzy association rules and fuzzy transactions are presented briefly in Section 3, while the process to refine queries via fuzzy association rules is explained in Section 4. The obtention of the document representation and the extraction of fuzzy association rules are given in Section 5 and Section 6, respectively. Finally, some experimental examples are shown in Section 7 and the conclusions and future work are presented in 8.

2 Query Refinement

The query refinement process, also called query expansion, is a possible solution to the problem of dissatisfaction of the user with the answer of an information retrieval system, given a certain query. This problem is due, most of the times, to the terms used to query, which meet the search criteria, but do not reflects exactly what the user is really searching. This occurs, most of the times because the user does not know the vocabulary of the topic of the query, or the query terms do not come to user's mind at the query moment, or just because the vocabulary of the user does not match with the indexing words of the collection. This problem is even strong when the user is searching in the web, due to the amount of available information which makes that the user feels overwhelmed with the retrieved set of documents.

The process of query refinement solves this problem by modifying the search terms so the system results are more adequate to user's needs. There are mainly two different approaches in query refinement regarding how the terms are added to the query. The first one is called *automatic query expansion* [7], [18] and consist of the augmentation of query terms to improve the retrieval process without the intervention of the user. The second one is called *semi-automatic query-expansion* [30,37], where new terms are suggested to the user to be added to the original query in order to guide the search towards a more specific document space.

We can also distinguish different cases based on the source from which the terms are selected. By this way, terms can be obtained from the collection of documents [2,39], from user profiles [23], from user behavior [21] or from other users' experience [16], among others. If a document collection is considered as a whole from which the terms are extracted to be added to the query, the technique is called *global analysis*, as in [39]. However, if the expansion of the query is performed based on the documents retrieved from the first query, the technique is denominated *local analysis*, and the set of documents is called *local set*.

Local analysis can also be classified into two types. On the one hand, *local feedback* adds common words from the top-ranked documents of the local set. These words are identified sometimes by clustering the document collection [2]. In this group we can include the relevance feedback process, since the user has to evaluate the top ranked documents from which the terms to be added to the query are selected. On the other hand, *local context analysis* [39], which combines global analysis and context local feedback to add words based on relationships of the top-ranked documents. The calculus of co-occurrences of terms is based on passages (text windows of fixed size), as in global analysis, instead of complete documents. The authors show that, in general, local analysis performs better than global one.

There are several approaches using different techniques to identify terms that should be added to the original query. The first group is based on their association relation by co-occurrence to query terms [36]. Instead of simply terms, in [39] the authors find co-occurrences of concepts given by noun groups with the query terms. Another approach based on the concept space approach is [8]. The statistical information can be extracted from a clustering process and a ranking of documents from the local set, as it is shown in [9] or by similarity of the top-ranked documents [28]. All these approaches where a co-occurrence calculus is performed has been said to be suitable to construct specific knowledge base domains, since the terms are related, but they can not be distinguished how [4].

In the second group of techniques, search terms are selected on the basis of their similarity to the query terms, by constructing a similarity term thesaurus [31]. Other approaches in this same group use techniques to find out the most discriminatory terms, which are the candidates to be added to the query. These two characteristics can be combined by first calculating the nearest neighbors and second, by measuring the discriminatory ability of the terms [30]. The last group is formed by approaches based on lexical variants of query terms extracted from a lexical knowledge base such as Wordnet [27]. Some approaches in this group are [38], and [4] where a semantic network with term hierarchies is constructed. The authors reveal the adequacy of this approach for general knowledge bases, which can be identified in general terms with global analysis, since the set of documents from which the hierarchies are constructed is the corpus, and not the local set of a first query. Previous approaches with the idea of hierarchical thesaurus can be also found in the literature, where an expert system of rules interprets the user's queries and controls the search process [18].

3 Association Rules and Fuzzy Association Rules

We use association rules and fuzzy association rules to find the terms to be added to the original query. In this section, we briefly review association rules

and some useful extensions able to deal with weighted sets of items in a fuzzy framework.

3.1 Association Rules

Given a database of transactions, where each transaction is an itemset, we can extract association rules [1]. Formally, let T be a set of transactions containing items of a set of items I. Let us consider two itemsets (sets of items) $I_1, I_2 \subset I$, where $I_1, I_2 \neq \varnothing$ and $I_1 \cap I_2 = \varnothing$. An association rule [1] $I_1 \Rightarrow I_2$ is an implication rule meaning that the apparition of itemset I_1 in a transaction implies the apparition of itemset I_2 in the same transaction. The reciprocal does not have to happen necessarily [22]. I_1 and I_2 are called antecedent and consequent of the rule, respectively. The rules obtained with this process are called boolean association rules or, in general, association rules since they are generated from a set of boolean or crisp transactions.

3.2 Fuzzy Association Rules

Fuzzy association rules are defined as those rules extracted from a set of fuzzy transactions FT where the presence of an item in a transaction is given by a fuzzy value of membership [3,10,19,24,25]. Though most of these approaches have been introduced in the setting of relational databases, we think that most of the measures and algorithms proposed can be employed in a more general framework. A broad review, including references to papers on extensions to the case of quantitative attributes and hierarchies of items, can be found in [11].

In this paper we shall employ the model proposed in [10]. This model considers a general framework where data is in the form of fuzzy transactions, i.e., fuzzy subsets of items. A (crisp) set of fuzzy transactions is called a FT-set, and fuzzy association rules are defined as those rules extracted from a FT-set. Fuzzy relational databases can be seen as a particular case of FT-set. Other datasets, such as the description of a set of documents by means of fuzzy subsets of terms, are also particular cases of FT-sets but fall out of the relational database framework.

Given a FT-set $\widetilde{T}$ on a set of items I and a fuzzy transaction $\tilde{\tau} \in \widetilde{T}$, we note $\tilde{\tau}(i)$ the membership degree of i in $\tilde{\tau}$ $\forall i \in I$. We also define $\tilde{\tau}(I_0) = \min_{i \in I_0} \tilde{\tau}(i)$ for every itemset $I_0 \subseteq I$.

With this scheme, we have a degree in $[0,1]$ associated to each pair $\langle \tilde{\tau}, I_0 \rangle$. Sometimes it is useful to see this information in a different way by means of what we call the *representation* of an itemset. The idea is to see an itemset as a fuzzy subset of transactions. The representation of an itemset $I_0 \subseteq I$ in a FT-set $\widetilde{T}$ is the fuzzy subset $\widetilde{I}_{I_0} \subseteq \widetilde{T}$ defined as

$$\widetilde{I}_{I_0} = \sum_{\tilde{\tau} \in \widetilde{T}} \tilde{\tau}\,(I_0)/\tilde{\tau} \tag{1}$$

On this basis, a fuzzy association rule is an expression of the form $I_1 \Rightarrow I_2$ that holds in a FT-set $\widetilde{T}$ iff $\widetilde{I}_{I_1} \subseteq \widetilde{I}_{I_2}$. The only difference with the definition of crisp association rule is that the set of transactions is a FT-set, and the inclusion above is the usual between fuzzy sets.

3.3 Measures for Association and Fuzzy Association Rules

There are two relevant aspects of association rules that we need to measure. On the one hand, an association rule can be interesting even if there are some exceptions to the rule in the set T, so we are interested in assessing the *accuracy* of the rule and to decide on its basis whether the rule is accurate or not. On the other hand, an accurate rule that holds in few transactions is not interesting since it is not representative of the whole data and its possible application is limited. Hence, we need to measure the amount of transactions supporting the rule and to decide on that basis whether the rule is important or not.

The assessment of association rules is usually based on the values of *support* and *confidence*. We shall note $supp(I_k)$ the support of the itemset I_k. The support and the confidence of the rule $I_1 \Rightarrow I_2$ noted by $Supp\,(I_1 \Rightarrow I_2)$ and $Conf\,(I_1 \Rightarrow I_2)$, respectively. Support is the percentage of transactions containing an itemset, calculated by its probability, while confidence measures the strength of the rule calculated by the conditional probability of the consequent with respect to the antecedent of the rule. Only itemsets with a support greater than a threshold *minsupp* are considered, and from the resulting association rules, those ones with a confidence less than a threshold *minconf* are discarded. Both thresholds must be fixed by the user before starting the process.

To deal with the imprecision of fuzzy transactions, we need to obtain the support and the confidence values with alternative methods which can be found mainly in the framework of approximate reasoning. We have selected the the evaluation of quantified sentences presented in [40], calculated by means of method GD presented in [13].

Moreover, as an alternative to confidence, we propose the use of *certainty factors* to measure the accuracy of association rules, since they have been revealed as a good measure in knowledge discovery too [17]. Basically, the problem with confidence is that it does not take into account the support of I_2, hence it is unable to detect statistical independence or negative dependence, i.e., a high value of confidence can be obtained in those cases. This problem is specially important when there are some items with very high support. In the worst case, given an itemset I_C such that $supp(I_C) = 1$, every rule of the form $I_A \Rightarrow I_C$ will be strong provided that $supp(I_A) > minsupp$. It has been shown that in practice, a large amount of rules with high confidence are misleading because of the aforementioned problems.

The certainty factor (CF) of an association rule is defined as $I_1 \Rightarrow I_2$ based on the value of the confidence of the rule. If $Conf\,(I_1 \Rightarrow I_2) > supp\,(I_2)$ the

value of the factor is given by expression (2); otherwise, is given by expression (3), considering that if $supp(I_2) = 1$, then $CF(I_1 \Rightarrow I_2) = 1$ and if $supp(I_2) = 0$, then $CF\,(I_1 \Rightarrow I_2) = -1$

$$CF\,(I_1 \Rightarrow I_2) = \frac{Conf\,(I_1 \Rightarrow I_2) - supp\,(I_2)}{1 - supp\,(I_2)} \tag{2}$$

$$CF\,(I_1 \Rightarrow I_2) = \frac{Conf\,(I_1 \Rightarrow I_2) - supp\,(I_2)}{supp\,(I_2)} \tag{3}$$

4 Query Refinement via Fuzzy Association Rules

Besides the techniques explained in Section 2, we also consider fuzzy association rules (which generalize the crisp ones) as a way to find presence dependence relations among the terms of a document set. A group of selected terms from the extracted rules generate a vocabulary related to the search topic that helps the user to refine the query.

In a text framework, association rules can be seen as rules with a semantic of presence of terms in a group of documents (we explain it with detail in the following section). This way, we can obtain rules such as $t_1 \Rightarrow t_2$ meaning that the presence of t_1 in a document imply the presence of term t_2, but the opposite do not have to occur necessarily. This concept is different from the co-occurrences where, given an occurrence between t_1 and t_2, the presence of both terms is reciprocal, that is, if one occurs, the other also does [22]. Only when the association rule $t_1 \Rightarrow t_2$ and its opposite $t_2 \Rightarrow t_1$ are extracted, we can say there is a co-occurrence between t_1 and t_2. In query refinement association rules extend the use of co-occurrences since it allows not only to substitute one term by other, but also to modify the query making it more specific or more general.

The process occurs as follows: before query refinement can be applied, we assume that a retrieval process is performed. The user's initial query generates a set of ranked documents. If the top-ranked documents do not satisfy user's needs, the query improvement process starts. Since we start from the initial set of documents retrieved from a first query, we are dealing with a *local analysis* technique. And, since we just considered the top-ranked documents, we can classify our technique as a *local feedback* one.

From the initial retrieved set of documents, called *local set*, association rules are found and additional terms are suggested to the user in order to refine the query. As we have explained in Section 2, there are two general approaches to query refinement: automatic and semi-automatic. In our case, as we offer to the user a list of terms to add to the query, the system performs a semi-automatic process. Finally, the user selects from that list the terms to add to the query so the query process starts again. The whole process is summarized in the following:

Semi-automatic query refinement process using association rules

1. The user queries the system
2. A first set of documents is retrieved
3. From this set, the representation of documents is extracted and association rules are generated
4. Terms that appear in certain rules are shown to the user (Subsection 6.1)
5. The user selects those terms more related to her/his needs
6. The selected terms are added to the query, which is used to query the system again

Once the first query is constructed, and the association rules are extracted, we make a selection of rules where the terms of the original query appear. However, the terms of the query can appear in the antecedent or in the consequent of the rule. If a query term appears in the antecedent of a rule, and we consider the terms appearing in the consequent of the rule to expand the query, a generalization of the query will be carried out. Therefore, a generalization of a query gives us a query on the same topic as the original one, but looking for more general information. However, if query terms appear in the consequent of the rule, and we reformulate the query by adding the terms appearing in the antecedent of the rule, then a specialization of the query will be performed, and the precision of the system should increase. The specialization of a query looks for more specific information than the original query but in the same topic. In order to obtain as much documents as possible, terms appearing in both sides of the rules can also be considered.

5 Document Representation for Association Rule Extraction

From that initial retrieved set of documents, a valid representation for extracting the rules is needed. Different representations of text for association rules extraction can be found in the literature: bag of words, indexing keywords, term taxonomy and multi-term text phrases [12]. In our case, we use automatic indexing techniques coming from Information Retrieval [34] to obtain *word items*, that is, single words appearing in a document where stop-list and/or stemming processes can be applied. Therefore, we represent each document by a set of terms where a weight meaning the presence of the term in the document can be calculated. There are several term weighting schemes to consider [33]. In this work, we study three different weighting schemes [22]:

Boolean weighting scheme: It takes values {0,1} indicating the absence or presence of the word in the document, respectively.

Frequency weighting scheme: It associates to each term a weight meaning the relative frequency of the term in the document. In a fuzzy framework, the normalization of this frequency can be carried out by dividing the number of occurrences of a term in a document by the number of occurrences of the most frequent term in that document [6].

TFIDF weighting scheme: It is a combination of the within-document word frequency (TF) and the inverse document frequency (IDF). The expressions of these schemes can be found in [33]. We use this scheme in its normalized form in the interval [0, 1] according to [5]. In this scheme, a term that occurs frequently in a document but infrequently in the collection is assigned a high weight.

5.1 Text Transactions

Once we have the representation of the documents in a classical information retrieval way, a transformation of this representation into a transactional one is carried out. In a text framework, we identify each transaction with the representation of a document. Therefore, from a collection of documents $D = \{d_1, \ldots, d_n\}$ we can obtain a set of terms $I = \{t_1, \ldots, t_m\}$ which is the union of the keywords for all the documents in the collection. The weights associated to these terms in a document d_i are represented by $W_i = (w_{i1}, \ldots, w_{im})$. For each document d_i, we consider an extended representation where a weight of 0 will be assigned to every term appearing in some of the documents of the collection but not in d_i.

Considering these elements, we can define a *text transaction* $\tau_i \in T$ as the extended representation of document d_i. Without loosing generalization, we can write the set of transactions associated to the collection of document D as $T_D = \{d_1, \ldots, d_n\}$.

When the weights $W_i = (w_{i1}, \ldots, w_{im})$ associated to the transactions take values in {0,1}, that is, following the boolean weighting scheme of the former section, the transactions can be called boolean or crisp transactions, since the values of the tuples are 1 or 0 meaning that the attribute is present in the transaction or not, respectively.

Fuzzy Text Transactions

As we have explained above, we can consider a weighted representation of the presence of the terms in the documents. In the fuzzy framework, a normalized weighting scheme in the unit interval is employed. We call them *fuzzy weighting schemes*. Concretely, we consider two fuzzy weighting schemes, namely the frequency weighting scheme and the TFIDF weighting scheme, both normalized. Therefore, analogously to the former definition of text transactions, we can define a set of *fuzzy text transactions* $FT_D = \{d_1, \ldots, d_n\}$, where each document d_i corresponds to a fuzzy transaction $\tilde{\tau}_i \in FT$, and where the

weights $W = \{w_{i1}, \ldots, w_{im}\}$ of the keyword set $I = \{t_1, \ldots, t_m\}$ are fuzzy values from a fuzzy weighting scheme.

6 Extraction of Fuzzy Association Rules

As described in the previous subsection, we consider each document as a transaction. Let us consider $T_D = \{d_1, \ldots d_n\}$ as the set of transactions from the collection of documents D, and $I = \{t_1, \ldots, t_m\}$ as the text items obtained from all the representation documents $d_i \in D$ with their membership to the transaction expressed by $W_i = (w_{i1}, \ldots, w_{im})$. On this set of transactions we apply Algorithm 1 to extract the association rules. We must note that we do not distinguish in this algorithm the crisp and the fuzzy case, but we give general steps to extract association rules from text transactions. The specific cases will be given by the item weighting scheme that we consider in each case.

Algorithm 1 Basic algorithm to obtain the association rules from text

Input: a set of transactions $T_D = \{d_1, \ldots d_n\}$
a set of term items $I = \{t_1, \ldots, t_m\}$ with their associated weights $W_i = (w_{i1}, \ldots, w_{im})$ for each document d_i.
Output: a set of association rules.

1. Construct the itemsets from the set of transactions T.
2. Establish the threshold values of minimum support *minsupp* and minimum confidence *minconf*
3. Find all the itemsets that have a support above threshold *minsupp*, that is, the *frequent itemsets*
4. Generate the rules, discarding those rules below threshold minconf

We must point out that, as it has been explained in [15], [32], in the applications of mining techniques to text, documents are usually categorized, in the sense of documents which representation is a set of keywords, that is, terms that really describe the content of the document. This means that usually a full text is not considered and its description is not formed by all the words in the document, even without stop words, but also by keywords. The authors justify the use of keywords because of the appearing of useless rules. Some additional commentaries about this problem regarding the poor discriminatory power of frequent terms can be found in [30], where the authors comment the fact that the expanded query may have worst performance than the original one due to the poor discriminatory ability of the added terms.

Therefore, the problem of selecting good terms to be added to the query has two faces. On the one hand, if the terms are not good discriminators, the expansion of the query may not improve the result. But, on the other hand,

in dynamic environments or systems where the response-time is important, the application of a pre-processing stage to select good discriminatory terms may not be suitable. In our case, since we are dealing with a problem of query refinement in Internet, information must be shown on-line to the user, so a time constraint is present.

Solutions for both problems can be given. In the first case, discriminatory schemes almost automatic can be used alternatively to a preprocessing stage for selecting the most discriminatory terms. This is the case of the *TFIDF* weighting scheme (see Section 5). In the second case, when we work in a dynamic environment, we have to remind that to calculate the term weights following the *TFIDF* scheme, we need to know the presence of a term in the whole collection, which limits in some way its use in dynamic collections, as usually occurs in Internet. Therefore, instead of improving document representation in this situation, we can improve the rule obtaining process. The use of alternative measures of importance and accuracy such as the ones presented in Section 3 is considered in this work in order to avoid the problem of non appropriate rule generation.

Additionally to the representation of the documents by terms, an initial categorization of the documents can be available. In that case, the categories can appear as items to be included in the transactions with value $[0, 1]$ based on the membership of the document to that category. This way, the extracted rules not only provide additional terms to the query, but also information about the relation between terms and categories.

6.1 The Selection of Terms for Query Refinement

The extraction of rules is usually guided by several parameters such as the minimum support *(minsupp)*, the minimum value of certainty factor *(mincf)*, and the number of terms in the antecedent and consequent of the rule. Rules with support and certainty factor over the respective thresholds are called *strong rules*.

Strong rules identify dependence in the sense of nontrivial inclusion of the set of transactions where each itemset (set of terms in this case) appears. This information is very useful for us in order to refine the query. First, the minimum support restriction ensures that the rules apply to a significant set of documents. Second, the minimum accuracy restriction, though allowing for some exceptions, ensures that the inclusion holds to an important degree.

Once the strong association rules are extracted, the selection of useful terms for query refinement depends on the appearance in antecedent and/or consequent of the terms. Let us suppose that *qterm* is a term that appears in the query and let $term \in S$, $S_0 \subseteq S$. Some possibilities are the following:

- Rules of the form $term \Rightarrow qterm$ such that $qterm \Rightarrow term$ has low accuracy. This means that the appearance of *term* in a document "implies" the appearance of *qterm*, but the reciprocal does not hold significantly, i.e.,

$\Gamma_{term} \subseteq \Gamma_{qterm}$ to some extent. Hence, we could suggest the word *term* to the user as a way to restrict the set of documents obtained with the new query.

- Rules of the form $S_0 \Rightarrow qterm$ with $S_0 \subseteq S$. We could suggest the set of terms S_0 to the user as a whole, i.e., to add S_0 to the query. This is again uninteresting if the reciprocal is a strong rule.
- Rules of the form $qterm \Rightarrow term$ with $term \in S$ and $term \Rightarrow qterm$ a not strong rule. We could suggest the user to replace *qterm* with *term* in order to obtain a set of documents that include the actual set (this is interesting if we are going to perform the query again in the web, since perhaps *qterm* is more specific that the user intended).
- Strong rules of the form $S_0 \Rightarrow qterm$ or $term \Rightarrow qterm$ such that the reciprocal is also strong. This means co-occurrence of terms in documents. Replacing *qterm* with S_0 (or *term*) can be useful in order to search for similar documents where *qterm* does not appear. These rules can be interesting if we are going to perform the query again in Internet, since new documents not previously retrieved and interesting for the user can be obtained by replacing *qterm* with *term*.

7 Experimental Examples

The experiments have been carried out in the web with the search engine *Google (http://www.google.es)*. Three different queries have been submitted to the engine, with the search and results in English, namely: *networks*, *learning* and *genetic*. The purpose of our system is to find additional terms that can modify the query but narrow the set of retrieved documents in most of the cases, and/or improve the retrieval effectiveness. Therefore, if the user has the intention of searching for documents about *genetic* with a Computer Science and an Artificial Intelligence meaning, but she/he does not know more vocabulary related to that concept, the resulting rules can suggest her/him some terms to add to the query. This new query can discard the documents related to other meanings (always that the additional terms are not in the vocabulary of the other meanings).

Once the queries have been submitted to the search engine for the first time, an initial set of documents is retrieved, from which we take the first 100 top-ranked documents. Since we start from the initial set of documents retrieved from a first query, we are dealing with a *local analysis* technique. And, since we just considered the top-ranked documents, we can classify our technique as a *local feedback* one. From this *local set*, a document representation is obtained as in classical information retrieval, and a transformation of this representation into a transactional one is carried out. These transactions are mined for each query to obtain a set of association rules so additional terms can be offered to the user to refine the query. The number of results in each query, the number of text transactions and the number of terms (items)

Table 1. Queries with number of results, transactions and terms

Query	N. Results	N. Transactions	N. Terms
networks	94.200.000	100	839
learning	158.000.000	100	832
genetic	17.500.000	100	756

can be seen in Table 1. It must be remarked the difference in the length of the dimensions of the set of transactions obtained. In traditional data mining, the number of transactions is usually greater while the number of items is lower. In our case it is the opposite, although the goodness of the rules has not to be affected.

The weighted schemes considered are those proposed in Section 5, that is, the boolean, the frequency and the TFIDF weighting scheme. We must point out that the first one is crisp, while the other two are fuzzy values. The threshold of support is established to 2% for the crisp and the frequency case, while for the TFIDF we decide to remove the threshold, since no rules appear with more than a 2% for all the queries. For the obtention of the rules, we have established a level of the rule of 5, which implies that the number of components appearing in the rule (antecedent and consequent) can not be more than 5 adding both sides of the rule). The number of rules obtained for each weighting scheme with these thresholds can be seen in Table 2. In this table, we can observe the main advantages of the fuzzy weighting schemes against the crisp case. We must remember that the boolean scheme assigns 0 if the term does not appear in the document, and 1 if the terms appears, no matter how many times. This implies that the importance of a term will be 1 either if the term appears 1 or 20 times in the same document, which does not reflect the real presence of a term in a document. From the point of view of rules, this generates a huge number of them which give not very realistic presence relations among the terms, so they are not very useful for the user.

In the case of the TFIDF case, this scheme assigns a low weight to those items appearing very frequently in the whole collection. When the TFIDF scheme is used, the term query, for instance, *networks* is assigned a weight of 0, since it appears in all the documents of the collection. This means that no rule with the term *networks* will appear in the set of extracted rules in this case. This effect is the same that is obtained with the selection of rules, where

Table 2. Number of rules for each query with different weighting schemes

Query	Boolean	Norm. Freq.	TFIDF
networks	1118	95	56
learning	296	73	10
genetic	233	77	10

high frequent terms are not considered since they do not give new information. However, this lack of new information does not mean that the terms appearing in the same rule as the query term do not help to refine the query to decrease the number of retrieved documents and increase the satisfaction of the user.

The best scheme to analyze cases is the normalized frequency scheme. This scheme assigns a weight to a term meaning the normalized relative frequency of the term in the document, which is more realistic than the boolean scheme but less discriminatory than the TFIDF one. For instance, in the document set retrieved as the answer of query *genetic*, there are documents related to Biology and to Computer Science. If a novel user does not know the vocabulary of the topic, and the intention of the search is looking for *genetic* in the field of Computer Science, rules such as $programming \Rightarrow genetic$, can suggest to the user a new term, *programming*, in order to add it to the query so the results of the refined query are more suitable to user's needs. This case is of type $term \Rightarrow qterm$, where the rule $programming \Rightarrow genetic$ holds with a certainty factor of 1 while the opposite rule $genetic \Rightarrow programming$ holds with a certainty factor of 0.013.

Other example in this case is related to the query *learning*. Let us suppose that the user has the intention of searching about learning and the new technologies, but only use the query term *learning* so millions of documents are retrieved by the search engine. Some interesting rules obtained in this case related learning and new technologies are shown in Table 3, where the terms appearing in the antecedent of the rules are shown in the left column and the terms appearing in the consequent of the rules are shown in the first row of the table.

Table 3. Confidence/Certainty Factor values of some rules with the normalized frequency weighting scheme for the query *learning*

	learning	technology	web	online
learning	–	0.04/0.01	0.06/0.01	0.15/0.03
technology	0.94/0.94	–	–	–
web	0.8/0.79	–	–	–
online	0.79/0.76	–	–	–

We can also observe a case of substitution of terms when both $term \Rightarrow qterm$ and its reciprocal are strong rules. For instance, with the query of *networks*, the rule $rights \Rightarrow reserved$ and its reciprocal $reserved \Rightarrow rights$, appears with a support of 2.3% and a certainty factor of 1. This means that these two terms are equivalent to be used as additional terms to refine the query.

Regarding the information retrieval effectiveness values, as we add terms to the query, in our experiments the precision increases while the recall decreases. For instance, let us suppose again the example of the user looking

for documents related to *genetic* in the field of Computer Science. If the user submit the query with only the term *genetic*, the recall value is 1 while the precision value is of 0.16 in the top-ranked first 100 documents. As the rule *programming* $\Rightarrow$ *genetic* has a support of 6% and a certainty factor of 1, it will be selected to show to the user the term *programming* to be added to the query. With the refined query, the recall decreases to 0.375, but the precision increases to 1.

8 Conclusions and Future Work

We have presented a possible solution to the Information Retrieval problem of query refinement in the web by means of fuzzy association rules. The fuzzy framework allows to represent documents by terms with an associated weight of presence. This representation improves the traditional ones based on binary presence/ausence of terms in the document, since it allows to distinguish between terms appearing in a document with different frequencies. This representation of documents by weighted terms is transformed into a transactional one, so text rules can be extracted following a mining process. From all the extracted rules, a selection process is carried out, so only the rules with a high support and certainty factor are chosen. The terms appearing in these rules are shown to the user, so a semi-automatic query refinement process is carried out. As it has been shown in the experimental examples, the refined queries reflect better the user's needs and the retrieval process is improved. The selection of rules and the chance to make the query more general, specific or to change the terms with the same meaning in order to improve the results lead us to consider this approach an useful tool for query refinement.

In the future, we will study the extraction of rules over a collection of documents as in the global analysis techniques, so we could compare with the actual system.

Acknowledgements

This work is supported by the research project Fuzzy-KIM, CICYT TIC2002-04021-C02-02.

References

1. Agrawal, R., Imielinski, T. & Swami, A. "Mining Association Rules between Set of Items in Large Databases". In *Proc. of the 1993 ACM SIGMOD Conference*, 207–216, 1993.
2. Attar, R. & Fraenkel, A.S. "Local Feedback in Full-Text Retrieval Systems". *Journal of the Association for Computing Machinery 24(3)*:397–417, 1977.

3. Au, W.H. & Chan, K.C.C. "An effective algorithm for discovering fuzzy rules in relational databases". In *Proc. Of IEEE International Conference on Fuzzy Systems*, vol II, 1314–1319, 1998.
4. Bodner, R.C. & Song, F. "Knowledge-based approaches to query expansion in Information Retrieval". In McCalla, G. (Ed.) *Advances in Artificial Intelligence*:146–158. New-York, USA: Springer Verlag, 1996.
5. Bordogna, G., Carrara, P. & Pasi, G. "Fuzzy Approaches to Extend Boolean Information Retrieval". In Bosc., Kacprzyk, J. *Fuzziness in Database Management Systems*, 231-274. Germany: Physica Verlag, 1995.
6. Bordogna, G. & Pasi, G. "A Fuzzy Linguistic Approach Generalizing Boolean Information Retrieval: A Model and Its Evaluation". *Journal of the American Society for Information Science 44(2)*:70–82, 1993.
7. Buckley, C., Salton. G., Allan, J. & Singhal, A. "Automatic Query Expansion using SMART: TREC 3". *Proc. of the 3rd Text Retrieval Conference*, Gaithersburg, Maryland, 1994.
8. Chen, H., Ng, T., Martinez, J. & Schatz, B.R. "A Concept Space Approach to Addressing the Vocabulary Problem in Scientific Information Retrieval: An Experiment on the Worm Community System". *Journal of the American Society for Information Science 48(1)*:17–31, 1997.
9. Croft, W.B. & Thompson, R.H. "I^3R: A new approach to the design of Document Retrieval Systems". *Journal of the American Society for Information Science 38(6)*, 389–404, 1987.
10. Delgado, M., Marín, N., Sánchez, D. & Vila, M.A. "Fuzzy Association Rules: General Model and Applications". *IEEE Transactions on Fuzzy Systems 11*:214–225, 2003a.
11. Delgado, M., Marín, N., Martín-Bautista, M.J., Sánchez, D. & Vila, M.A. "Mining Fuzzy Association Rules: An Overview". *2003 BISC International Workshop on Soft Computing for Internet and Bioinformatics"*, 2003b.
12. Delgado, M., Martín-Bautista, M.J., Sánchez, D. & Vila, M.A. "Mining Text Data: Special Features and Patterns". In *Proc. of EPS Exploratory Workshop on Pattern Detection and Discovery in Data Mining*, London, September 2002a.
13. Delgado, M., Sánchez, D. & Vila, M.A. "Fuzzy cardinality based evaluation of quantified sentences". *International Journal of Approximate Reasoning 23*:23–66, 2000c.
14. Efthimiadis, E. "Query Expansion". *Annual Review of Information Systems and Technology 31*:121–187, 1996.
15. Feldman, R., Fresko, M., Kinar, Y., Lindell, Y., Liphstat, O., Rajman, M., Schler, Y. & Zamir, O. "Text Mining at the Term Level". In *Proc. of the 2nd European Symposium of Principles of Data Mining and Knowledge Discovery*, 65–73, 1998.
16. Freyne J, Smyth B. (2005) Communities, collaboration and cooperation in personalized web search. In Proc. of the 3rd Workshop on Intelligent Techniques for Web Personalization (ITWP'05). Edinburgh, Scotland, UK
17. Fu, L.M. & Shortliffe, E.H. "The application of certainty factors to neural computing for rule discovery". *IEEE Transactions on Neural Networks 11(3)*:647–657, 2000.
18. Gauch, S. & Smith, J.B. "An Expert System for Automatic Query Reformulation". *Journal of the American Society for Information Science 44(3)*:124–136, 1993.

19. Hong, T.P., Kuo, C.S. & Chi, S.C. "Mining association rules from quantitative data." *Intelligent Data Analysis 3*:363–376, 1999.
20. Jiang, M.M., Tseng, S.S. & Tsai, C.J. "Intelligent query agent for structural document databases." *Expert Systems with Applications 17*:105–133, 1999.
21. Kanawati R., Jaczynski M., Trousse B., Andreoli J.M. (1999) Applying the Broadway recommendation computation approach for implementing a query refinement service in the CBKB meta search engine. In Proc. of the French Conference of CBR (RaPC99), Palaiseau, France
22. Kraft, D.H., Martín-Bautista, M.J., Chen, J. & Sánchez, D. "Rules and fuzzy rules in text: concept, extraction and usage". *International Journal of Approximate Reasoning 34*, 145–161, 2003.
23. Korfhage R.R. (1997) Information Storage and Retrieval. John Wiley & Sons, New York
24. Kuok, C.-M., Fu, A. & Wong, M.H. "Mining fuzzy association rules in databases," *SIGMOD Record 27(1)*:41–46, 1998.
25. Lee, J.H. & Kwang, H.L. "An extension of association rules using fuzzy sets". In *Proc. of IFSA'97*, Prague, Czech Republic, 1997.
26. Lin, S.H., Shih, C.S., Chen, M.C., Ho, J.M., Ko, M.T., Huang, Y.M. "Extracting Classification Knowledge of Internet Documents with Mining Term Associations: A Semantic Approach". In *Proc. of ACM/SIGIR'98*, 241–249. Melbourne, Australia, 1998.
27. Miller, G. "WordNet: An on-line lexical database". *International Journal of Lexicography 3(4)*:235–312, 1990.
28. Mitra, M., Singhal, A. & Buckley, C. "Improving Automatic Query Expansion". In *Proc. Of ACM SIGIR*, 206–214. Melbourne, Australia, 1998.
29. Moliniari, A. & Pasi, G. "A fuzzy representation of HTML documents for information retrieval system." *Proceedings of the fifth IEEE International Conference on Fuzzy Systems, vol. I*, pp. 107–112. New Orleans, EEUU, 1996.
30. Peat, H.P. & Willet, P. "The limitations of term co-occurrence data for query expansion in document retrieval systems". *Journal of the American Society for Information Science 42(5)*, 378–383, 1991.
31. Qui, Y. & Frei, H.P. "Concept Based Query Expansion". In *Proc. Of the Sixteenth Annual International ACM-SIGIR'93 Conference on Research and Development in Information Retrieval*, 160–169, 1993.
32. Rajman, M. & Besançon, R. "Text Mining: Natural Language Techniques and Text Mining Applications". In *Proc. of the 3rd International Conference on Database Semantics (DS-7)*. Chapam & Hall IFIP Proceedings serie, 1997.
33. Salton, G. & Buckley, C. "Term weighting approaches in automatic text retrieval". *Information Processing and Management 24(5)*, 513–523, 1988.
34. Salton, G. & McGill, M.J. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.
35. Srinivasan, P., Ruiz, M.E., Kraft, D.H. & Chen, J. "Vocabulary mining for information retrieval: rough sets and fuzzy sets". *Information Processing and Management 37*:15–38, 2001.
36. Van Rijsbergen, C.J., Harper, D.J. & Porter, M.F. "The selection of good search terms". *Information Processing and Management 17*:77–91, 1981.
37. Vélez, B., Weiss, R., Sheldon, M.A. & Gifford, D.K. "Fast and Effective Query Refinement". In *Proc. Of the 20th ACM Conference on Research and Development in Information Retrieval (SIGIR'97)*. Philadelphia, Pennsylvania, 1997.

38. Voorhees, E. "Query expansion using lexical-semantic relations. *Proc. of the 17th International Conference on Research and Development in Information Retrieval (SIGIR)*. Dublin, Ireland, July, 1994.
39. Xu, J. & Croft, W.B. "Query Expansion Using Local and Global Document Analysis". In *Proc. of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 4–11, 1996.
40. Zadeh, L.A. "A computational approach to fuzzy quantifiers in natural languages". *Computing and Mathematics with Applications 9(1)*:149–184, 1983.