

SABIO: SOFT AGENT FOR EXTENDED INFORMATION RETRIEVAL

Ariel Gómez, Carlos León, Jorge Roper, Alejandro Carrasco, and Joaquín Luque

Departamento de Tecnología Electrónica, Universidad de Sevilla, Sevilla, Spain

□ *In the current study, an integrated system called SABIO is presented. The current system applies Information Retrieval (IR) techniques developed for collections of textual documents to non-textual corpora. SABIO integrates a fuzzy logic-based procedure for IR. Its search algorithm improves the IR efficiency and decreases the computational burden by using a fuzzy logic-based procedure for IR. This procedure is integrated in a flexible and fault-tolerant, human-reasoning-based search algorithm. The Accumulated Knowledge Set (AKS) of the system is sorted in a hierarchic multilevel tree-structure-like ontology. The objects in the AKS are represented using a novel human-reasoning-based-method. This representation takes into account the occurrence of related terms. The system uses a novel fuzzy logic-based term-weighting (TW) method. The developed fuzzy logic method improves the classical term frequency-inverse document frequency (TF/IDF) method, generally used for TW. The abovementioned system is the core of a wizard for search into the website of the University of Sevilla, www.us.es, which is currently in testing.*

INTRODUCTION

The World Wide Web and the Internet allow users to access a wealth of information. This fact and the large quantity, and ever-growing, amount of information available make the demand for Information Retrieval (IR) techniques to increase (Aronson Rindfleisch, and Browne,1994; Liu et al. 2001). IR research deals mainly with documents. Achieving both high recall and precision in IR is one of its most important aims. IR has been widely used for text classification (Aronson et al. 1994; Liu et al. 2001) introducing approaches such as Vector Space Model (VSM), k-nearest neighbor method (KNN), Bayesian classification model and Support

Note: SABIO - Classified Information Automatic Retrieval System (“Sistema Automatizado de Búsqueda de Información Ordenada” in Spanish). “SABIO” means “wise” in Spanish.

Address correspondence to Ariel Gómez, Departamento de Tecnología Electrónica, Escuela Politécnica Superior de la Universidad de Sevilla. Calle Virgen de África, 7., 41011 Sevilla, Spain. E-mail: ariel@us.es

Vector Machine (SVM; Lu et al. 2002). In another vein, text mining (TM) techniques provide information derived as a result of the text document contents.

Due to the container environment, retrieved objects are textual (document, web pages, etc.). So, document retrieval systems are widely developed and applied for textual-type set search. Mainly, there are two approaches to the query: either the user provides a few keywords, or the user provides a document to use as a model. The second type of queries achieves a good degree of accuracy, but leads to an important computational load. This method is not suitable for large sets of accumulated knowledge. Furthermore, a keyword-based model has less computational burden and a similar structure to the question that a person would make. This feature is significant in systems where the man-machine interface is the natural language.

One of the most extended methods for keyword-based document content identification is the vector space model (VSM; Raghavan and Wong 1986). The method of representation of nontextual objects proposed in the current study is based on the VSM. In VSM, each document is represented by a set of words present in it (keywords). These keywords are chosen with the help of a stop list. The VSM rejects every matching word. Those remaining are called index terms and represent the document in the system. However, not all index terms are equally important for identifying the document they represent. So, it is necessary to add a factor to indicate its importance. This factor is known as the term-weight (TW).

One of the factors habitually used for term weighting in VSM is the so-called term frequency-inverse document frequency (TF-IDF; Lee, Chuang, and Seamons 1997). This scheme uses all the words present in any document representation as a system vocabulary. Term frequency (TF) is the number of occurrences of the index term in the represented document. Inverse document frequency (IDF) is related to the number of occurrences of the same index term in the other documents in the Accumulated Knowledge Set (AKS; Salton and Buckley 1996). Term weight is the product $tf \cdot idf$. With this method of document representing, the vector length depends on the number of words present in the document. This feature makes it difficult to compare the documents. Length normalization is applied to equalize the number of terms in all the vectors. However, the number of terms of a vector is usually quite large due to the vocabulary size. This feature makes the computational weight increase, and the method becomes impracticable for large corpora. The similarity between the objects is the distance between both the vectorial representations. One of the most used functions is the cosine similarity.

$$\text{similarity}(Q, D) = \frac{\sum_{k=1}^N w_{qk} \cdot w_{dk}}{\sqrt{\sum_{k=1}^N (w_{qk})^2 \cdot \sum_{k=1}^N (w_{dk})^2}} \quad (1)$$

Here, Q is the query representation, D is the document representation, N is the number of index terms available in both representations, w_{qk} is the weight coefficient associated to the k -th index term of the query, and w_{dk} is the weight coefficient associated to the k -th index term of the document.

It should be noted that in the VSM method (and others) objects are represented by parts of themselves, in other words, the words in the document. The main objective of the current study is to develop an information retrieval system that can manage information for any kind of knowledge (objects, experience, legislation, professional execution best practices, etc.) and not just in the textual form. In many cases, the representation of the object cannot be made up of parts of the object itself. Human knowledge is not achieved by incorporating parts of known reality. Humans translate the impulses that they perceive through senses. These are encoded in the protein chains that are stored in the brain (Kandel 2006; Hayashi and Yoshida 2004). All the visitors to the Giralda in Seville stored the data of the experience in their memories: architectural form, size, colors, history, location, and so forth. However, none of the physical components of the monument (bricks, marble piece, tile, plaster, or others) was added to the knowledge base of people who visited. The visitors created a representation of the monument that is stored in their memories. In the same way, in the proposed system, the representations of the real-world objects are built by attributes that are not a part of the objects themselves. Among the several possible ways of representing such objects, the system chooses natural language (NL), because the user query is made that way.

The retrieval effectiveness of an IR method is given by two factors: first, objects related with the query must be retrieved, and second, nonrelated objects must be rejected. The recall parameter is defined as an estimator of the first factor (Ruiz and Srinivasan 1998).

$$\text{recall} = \frac{\text{related retrieved objects}}{\text{total related objects in AKS}} \quad (2)$$

The precision parameter is defined as an estimator of the second factor.

$$\text{precision} = \frac{\text{related retrieved objects}}{\text{total retrieved objects}} \quad (3)$$

In the VSM approach and other IR methods, the query, or the document used as a model, is compared with every document in the

collection. In the proposed fuzzy IR method, the query is compared with only a few objects of the collection that represent the system knowledge.

To do this, the objects belonging to the AKS are grouped in a hierarchical tree structure like ontology. This proposed structure has multiple levels so that each set belonging to a level contains several sets of the lower level. Figure 1 shows the proposed structure. Hierarchical classification of AKS is detailed later in this article.

With the proposed four-level structure, it is simple to identify every object in the AKS by successive approximations without having to analyze all the objects of the knowledge. Figure 2 shows the presented system procedure for recovering the information.

Another feature in the procedures for IR and TW presented in the current study is that it takes into account the relationship between the terms (Gómez et al. 2008). In other IR methods, terms are managed independently from each other. This fact causes the loss of the information given by the compound terms. The fact that the representation of the document should correspond to the meaning must not be forgotten. Some authors include a procedure to take account of the syntax of the sentences in the methods (Chow et al. 2009; Song et al. 2008), others include concept networks to represent the knowledge base (Horng Chen, Chang, and Lee 2003; 2005). SABIO pays attention to this information during the process of rendering objects.

The outcome of the previous IR processes was documents, however, the goal of the TM techniques is to provide new information derived as a result of the contents of the text documents (Ben-Dov and Feldman 2010). This way, SABIO integrates the TM techniques with IR, as it finds new information—objects—derived from text-normalized objects.

So, the current system applies the IR techniques developed for collections of textual documents to nontextual corpora. The current study develops a novel human reasoning-based method to represent

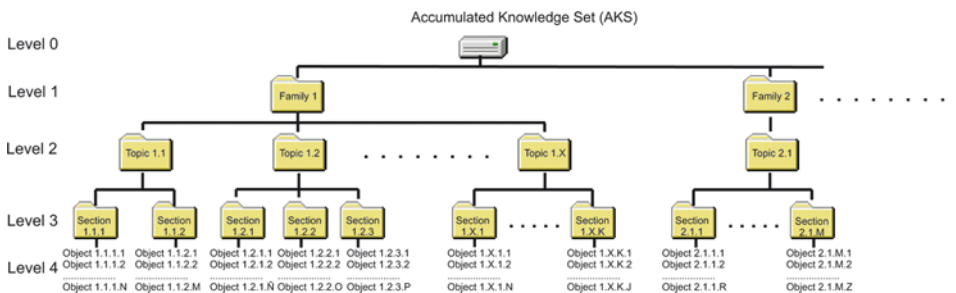


FIGURE 1 Hierarchical tree structure of the AKS. (Color figure available online.)

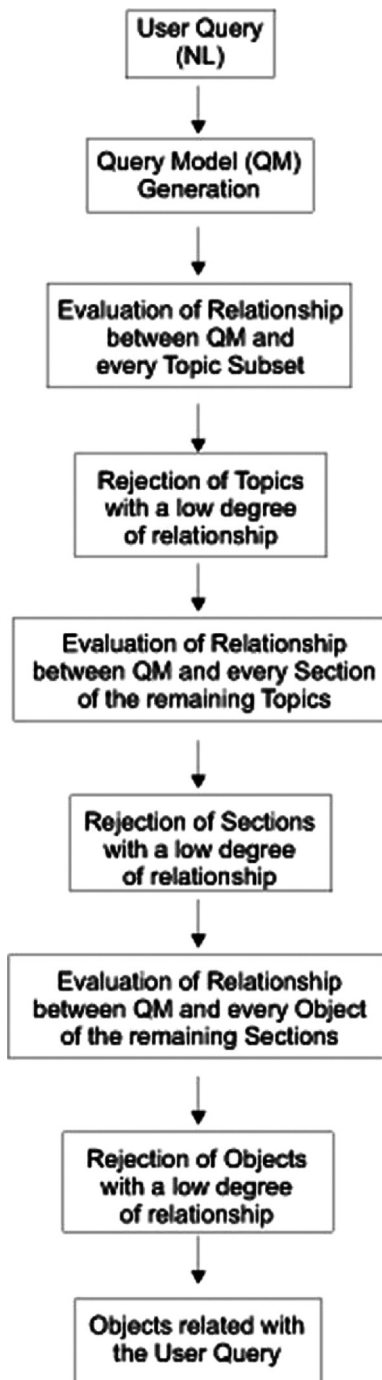


FIGURE 2 System procedure for recovering the information.

objects, taking into account the occurrence of related terms; proposes a fuzzy logic-based term-weighting method; structures the accumulated knowledge on several levels to improve the searches (Bathia and Deogun 1998) and decrease the computational burden; develops a fuzzy logic-based procedure to establish the similarity between a query and an object; and finally, proposes a flexible and fault-tolerant human reasoning-based search algorithm.

The next sections of the article are organized as follows. An accumulated knowledge objects normalization algorithm is introduced in “Accumulated Knowledge Objects Normalization Algorithm.” Hierarchical classification of the AKS is described in “Hierarchical Classification of the AKS.” The initial configuration of the fuzzy logic-based engine for retrieving the degree of certainty of relationships is explained in “Fuzzy Logic-Based Engine for Relationship Certainty Retrieval: Initial Configuration.” The proposed information retrieval algorithm is detailed in “Information Retrieval Algorithm.” Level weighting assignment procedure is detailed in “Level Term Weighting.” The realized tests, test-derivates system modifications, and model validation are detailed in “Tests, Modifications, and Model Validation.” The article ends with conclusions and future work proposals in “Conclusions.”

ACCUMULATED KNOWLEDGE OBJECTS NORMALIZATION ALGORITHM

In the above-mentioned IR methods, the objects of the AKS are usually documents. Their representations are built with parts from the objects themselves, in other words, the words contained in them. In SABIO, the AKS objects are not necessarily text-type. Thus, the existing object representation methods are not directly applicable. A general method should be proposed.

Just as the human brain transforms the received information by the senses and stores it permanently in the hippocampus and other structures (Kandel 2006; Hayashi and Yoshida 2004; Sato and Yamaguchi 2010) by using its own cells and proteins, SABIO builds the object representation using parts of the system itself. The bricks used by the system are the terms belonging to its vocabulary. The object representation is not complete without a term-weighting coefficient related to the importance of every word present into the object representation.

So, as object representation, the system uses a set of tuples $[a,b]$, where “a” is a word, and “b” is a related term-weighting coefficient. This transformation procedure is called object normalization. The normalized object representations are stored in a database for future retrieval.

Selection of Index Terms

In a general case of nontextual object, the process of choosing the NL terms to build its description cannot be made independently by direct analysis of the parts of the object itself. In this case, the choice of “a” terms to represent the object within the system is determined as follows.

The person who describes the objects in the set of knowledge is usually called the Knowledge Engineer (KE). The KE builds questions, which answer in NL to describe the object. This kind of sentence is named as a standard question. Another way of building a standard question is to just describe the object. The object representation will be built from a few standard questions.

From this set of standard questions, the KE must extract a few words, rejecting all the words that do not have a real relationship with the object. It should be noted that this fact excludes not only the stop words that were defined earlier, but also more words. A word can be very significant for the description of one object and nonrelevant for another one. This word is kept in the first case and is rejected in the second. The selected set of words that appears in any of the standard questions describing an object is called a set of index terms. Each of these words constitutes the “a” elements of the tuple array that represents the object to the system. In Figure 3 an index-terms selection for object normalization is described.

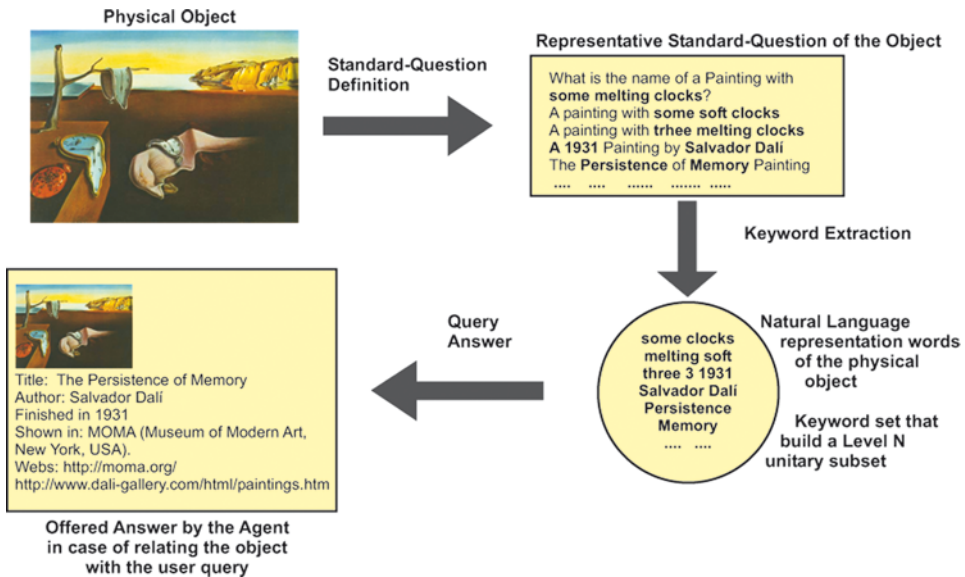


FIGURE 3 Normalization process of the objects in AKS. (Color figure available online.)

It should be noted that using the complete set of index terms for the IR does not make sense, as there is no request containing all the defined index terms. It is obvious that not all the selected index terms have the same relevance in the description of the object. So it is necessary to assign a term weight “b” to each index term “a” to describe this feature (relationship). All the index term sets build the vocabulary of the system. TW normalization is not possible because the representation of the object is not unique.

Term-Weight “b” Purpose

The strength of the relationship between a keyword “a” and the object is expressed by a coefficient. In the current system, this coefficient can take values between 0 and 1. Assigning a 0.00 value means no index-term relationship with the object, whereas assigning a value of 1.00 indicates the highest possible degree of relationship. This concept is related with the TW methods previously described as VSM. “Level Term Weighting” details the algorithm for calculating the “b” component of the tuple. Thus, each object of the AKS is represented by a set of tuples [a,b], where “a” is an index term, and “b” is the value representative of the degree of affinity between the term “a” and the object.

Normalization of the Query by the System

As mentioned previously, the SABIO Human Machine Interface (HMI) is natural language. So, information retrieval is made by a user query in NL. This query will not match exactly with one of the system-defined standard questions to extract object representations. Thus, it must be processed by the system. It is therefore necessary to model the received query but not to classify it, as in Chali (2009), because the answer isn’t a document. This modeling process is called query normalization. The representation of the query is the set of words present in the query that match any of those belonging to the vocabulary of the system. SABIO considers only the words that make sense to wake up its memory. It may be noticed that the index terms in a query do not have any associated “b” coefficients.

HIERARCHICAL CLASSIFICATION OF THE AKS

Once all of the objects in the AKS are normalized, there is a whole bag of tuples. Each of them represents an object. When the system receives a query, it must establish the relationship with every object in its AKS. If the system behaved as the current IR methods do, it should make an estimate for every object of AKS. However, if the objects were previously

grouped (by some suitable criterion), the system could determine the affinity of the query with some of the elements of each group, by a single estimate. This argument has two flaws. If small groups are formed, then the procedure is not effective, and if groups are big, the precision of the IR is poor. However, this method should be useful to exclude many objects not related to the received query by a single estimate. This feature causes the rejection of a significant number of objects, reducing the computational burden and processing time for IR. The objects belonging to the subsets not previously rejected could be treated as in the previous case. For this purpose, all objects in a subset should be grouped into smaller subsets defining a second level of grouping. Every set in the second subdivision contains fewer objects than those of the previous level. So fewer objects will be rejected if no relationship with the query is established, but the precision improves. If the last level of aggregation contains singletons, each set corresponds to a single object and the recursive application of this method identifies the objects in the AKS by successive approximations.

It is necessary to find the suitable number of levels of grouping objects so that the identification process provides advantages over the existing ones. It is also necessary to define a clustering approach. Another aspect to determine is the representation format of every group of objects.

Suitable Number of Hierarchic Levels

SABIO proposes to group objects into different sets for every considered level. The common feature for the objects belonging to a set is the existence of the same or similar index terms in their representations in NL. Every set is represented by the union of the NL representations of their component objects. This grouping provides a level of classification with a lower resolution than the previous one.

“Conclusions” validates that grouping the AKS objects in a three-level structure (called topic, section, and object) is enough to improve the efficiency of the subsequent information retrieval about a specific area. The addition of a fourth level (family) of classification of objects should be necessary when the system needs to extract knowledge from significantly different areas.

Representation of the Subsets

In the level structure described, Level N is the highest level (object representation). At this level all the subsets are singletons and the representation is the bag of tuples for every object of the AKS. The next

level (Level N-1) groups, from the previous level, objects that have some common properties. Level N-1 is called the *section level*. Each subset in this section level must have a representation in order to allow the system to determine the relationship with the query. In order to use the same method to establish the relationship to the query, representation of each section subset must have the same structure as that of the objects.

Therefore, the representation of each section consists of an array of tuples $[a_s, b_s]$ (another bag of tuples) The terms “ a_s ” correspond to union of the the terms “ a ” present in the representations of the objects belonging to the section. The terms “ b_s ” establish the relationship of each term “ a_s ” with the objects included in the section. The number of tuples in the array is determined by the union of the terms “ a ” of representations of objects belonging to the section. The term “ b_s ” associated with each indexterm “ a_s ” is determined by the values “ b ” of the representations of objects in which the term “ a ” appears. Level weighting is detailed in “Level Term Weighting.”

The process of grouping several subgroups of AKS in sets containing more objects can be repeated as many times as necessary. At the end of the overall process, the AKS is clustered in different ways at the various levels. Overlapping levels have a pyramid shape. The number of objects in each level is always the same, but the number of subsets grows when closer to the level N. This structure improves the retrieval procedure.

The relationship between an index term “ a ” with a subset of the AKS is not the same for all the levels because the relevance of the term “ a ” varies according to the subset in which representation appears. Thus, the “ b ” term weighing will be different for each level, and the tuples are not the same for every level of the hierarchically structured AKS.

Clustering Criteria

Grouping procedure involves two revisions over the objects of the AKS. The first one is top-down made. Objects are grouped by thematic affinities: topic and section. Once this provisional classification is made, a second bottom-up step is done. Refining criterion is used to put those objects together with the maximum number of common “ a ” terms in their representations. This criterion is applied only to the grouping of the N-1 level. Most objects are well grouped after the first revision because the common topic usually implies the presence of similar terms. However, at this second step, some objects could be moved from one group to another.

FUZZY LOGIC-BASED ENGINE FOR RELATIONSHIP CERTAINTY RETRIEVAL: INITIAL CONFIGURATION

The aim of the developed system is to answer queries from users without an extensive knowledge of any subject. Therefore, in some cases consultations are expected to be vague and/or nonspecific. Fuzzy logic techniques are suitable for managing this kind of information (Yager and Larsen 1993). The system core is a fuzzy logic engine (FE). FE establishes the degree of relationship between a query and an object or a set of objects in the AKS. The FE receives as input the “b” term of each tuple belonging to the representation of the object to be related; which term “a” matches with any word belonging to the query representation. This process is also applied not only for objects, but for every level of knowledge.

As said in “Term-Weight “b” Purpose,” “b” coefficient represents the strength of the relationship between the term “a” and the objects belonging to a certain set. The term “b” is transformed into a linguistic variable that expresses the degree of membership of the term “a” with respect to the subset evaluated. This variable can take three linguistic values: low, medium, and high. Figure 4 shows the aspect of the universe of discourse of this variable.

To answer a user query, the system needs to determine the relationship between the query and one of the objects present in its corresponding AKS. For this task, the system has an FE capable of establishing the degree of certainty for the relationship between the query and one object in the AKS. The determining FE parameters are the number of inputs, the number of outputs, the inference rules, the type of fuzzyficator, and the type of defuzzyficator. The different processes involved in the determination of these parameters are described in the following section.

Fuzzy Logic Rules

To oversimplify, the methodology for determining the degree of relationship between the query and an object in the AKS is based on the values of the chosen “b” terms. Frequently, the higher the “b” terms, the higher the degree of certainty.

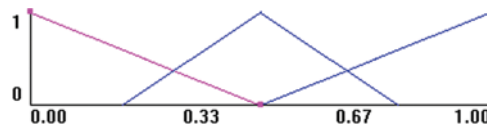


FIGURE 4 Universe of discourse of the input. (Color figure available online.)

The inference rules determine the degree of certainty in the relationship between the query and the object. Obviously, this way of describing the degree of relationship determines fuzzy logic as the better way to solve the proposed task. Thus, the rules that decide the relationship degree between the query and the object will be expressed as: “IF... THEN...” sentences. The generally used criterion for defining the rules is as follows: the more inputs with high values, the higher the value of the relationship. The deployment of this approach results in a number of rules depending on the number of FE inputs.

Inputs Number of the Fuzzy Logic Engine

The inputs to the FE are the “b” terms extracted from the query. Ideally, the query should correspond exactly with any of the KE defined standard-questions. Thus, the FE input number is conditioned by the number of words appearing in the representation of objects. This should be sufficient to consider for calculating all of the “b” terms of the tuples of the representation, for every standard question. In most cases, 3 to 5 words are extracted from each defined standard question. Therefore, the use of a three-input engine was initially proposed to assess the certainty of the relationship between an object and the query.

The FE final configuration and the reasoning for it are detailed in “Conclusions.”

Fuzzy Logic Engine Output

As described in “Effect of threshold value,” the FE has to show a single output: the degree of certainty of the relationship between the query and an object of the AKS. By the nature of the fuzzy rules, the output is a fuzzy value. Thus, its expression is provided by a linguistic expression. The linguistic variable called “certainty of relationship” can take four linguistic values: Low, Medium-Low, Medium-High, and High. Graphically, the shape of this output is shown in the Figure 5.

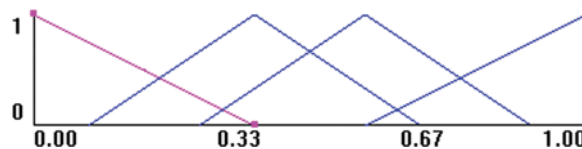


FIGURE 5 Shape of certainty degree of relationship. (Color figure available online.)

Fuzzyfication and Defuzzyfication Methods

Once the input and output numbers and the inference rules are specified, the fuzzyfication and defuzzyfication methods remain to be chosen. Because the output value has to be related to the set of all inputs, and not only to a dominant one, the center of gravity (COG), and mean of maximum (MOM) are considered as the defuzzyfication methods. Initially, the chosen method is COG. For the fuzzyfication method, a generic singleton is elected. “Conclusions” details the tests that lead to the final configuration of the system.

Relationship Certainty Retrieval Algorithm

Determination of the certainty of relationship between the received query and the corresponding subset of the AKS algorithm includes the following steps:

1. Query normalization as described in. “Normalization of the Query by the System” In the end, the query is represented by a word array.
2. Selection of tuples belonging to the evaluated subset representation that term “a” matches with any of the terms of the query representation. If the query representation involves more tuples than FE inputs, those tuples whose “b” terms are lower are rejected. This condition occurs when the number of selected tuples is higher than the FE inputs number.
3. The “b” terms of the selected tuples are the input values to the FE. If any FE input has no associated “b” value, 0.00 is taken as the associated input value. This condition is presented when the number of selected tuples is lower than the FE inputs number.
4. In general, the returned value by the FE is the degree of certainty associated with the relationship of the query with any of the objects contained in the considered subset. Figure 6 shows the described procedure.

INFORMATION RETRIEVAL ALGORITHM

At this point, there is a hierarchically structured AKS. It is structured in several levels, considering three levels for the example given following. There is also a system capable of evaluating the certainty degree of the relationship between a query and a subset of the AKS. There is a need to describe the full information retrieval algorithm used by the system intelligently. The main goal of the algorithm is to be able to discard many of the

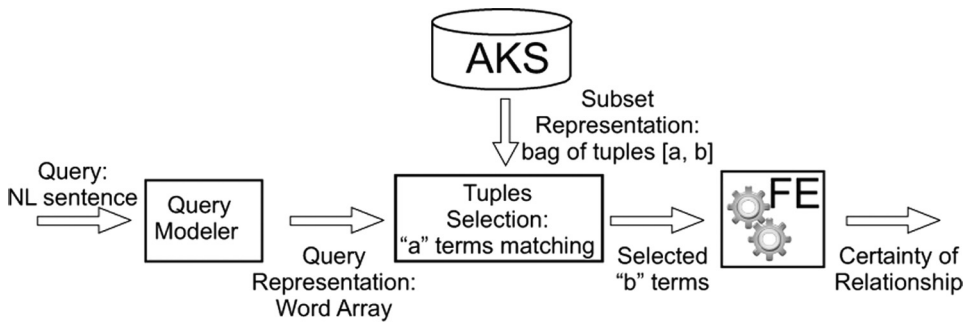


FIGURE 6 Certainty of relationship retrieval algorithm.

objects in the AKS in the early steps. To do this task, the system begins evaluating the certainty degree of the relationship between the query and every first-level subset to see which is the largest by applying the algorithm described in “Relationship Certainty Retrieval Algorithm.”

A threshold value is established for every level. The purpose of this threshold is to reject those subsets with a lower certainty of relationship obtained in the previous step. In this manner, many objects are rejected by only one estimation. Thus, the computational efficiency of the algorithm increases. Now, the process is applied again to those subsets that obtained a degree of certainty higher than the threshold, but using the next level of classification. The aim of the process is to approach the query-related objects without evaluating every object of the AKS. This search refines the results using the subsets present in the following levels. Only subsets corresponding to the accepted sets of the previous level are considered. Figure 7 illustrates the procedure, considering a three-level structured AKS.

The first step is to normalize the query. The representation of the query consists of a word array without the associated coefficients instead of a set of

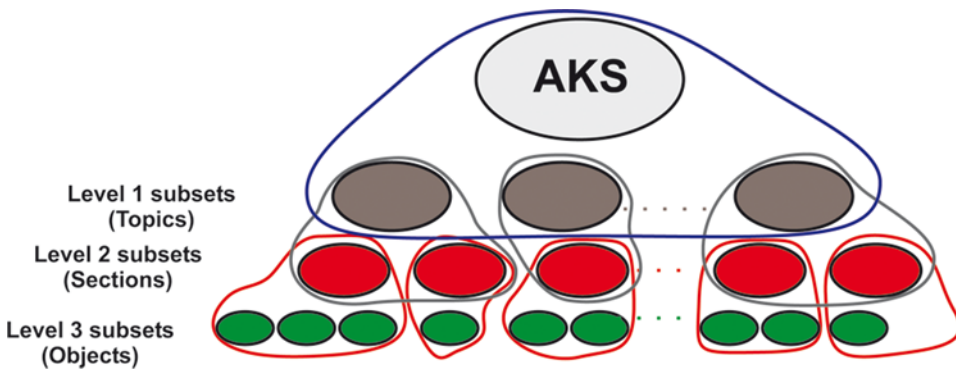


FIGURE 7 Three-level structured AKS. (Color figure available online.)

tuples. As mentioned previously, selected words should be contained in the system vocabulary.

The second step is to determine the degree of certainty that the query is related to any of the objects in first-level subset (Topic). For every subset in this level, the system takes the set of tuples representing the subset for which the relationship is being evaluated. The system selects those tuples whose “a” term matches any word present in the representation of the query.

In the next step, the FE inputs are fed with the associated “b” terms of the previously selected tuples. FE output is the degree of certainty of the relationship between the query and some of the objects in the specific subset. This procedure is applied one by one to every first-level subset. At the end, the system has a certainty value associated with each first-level subset.

The last step for this level is to reject those subsets whose associated certainty value is lower than a predetermined threshold. In Figure 8, only the first and the last subsets are above the threshold. Thus, the remaining subsets are rejected.

The same procedure is applied to each Level-2 subset belonging to those Level-1 subsets for which the certainty value was higher than the threshold. As a result, a new array of certainty values decides which Level-2 subsets are rejected. Only those Level-2 subsets whose associated value is greater than the Level-2 threshold will remain. Note that the threshold for Level-1 does not have to be the same as that of Level-2.

In Figure 9, only the second and the third Level-2 subsets are accepted. Those remaining are rejected.

At this point, only three objects of the AKS would be related with the query. To determine which are finally related, the above procedure is applied one more time to the objects belonging to the remaining subsets. This time, Level-3 subsets are the object representations themselves. All

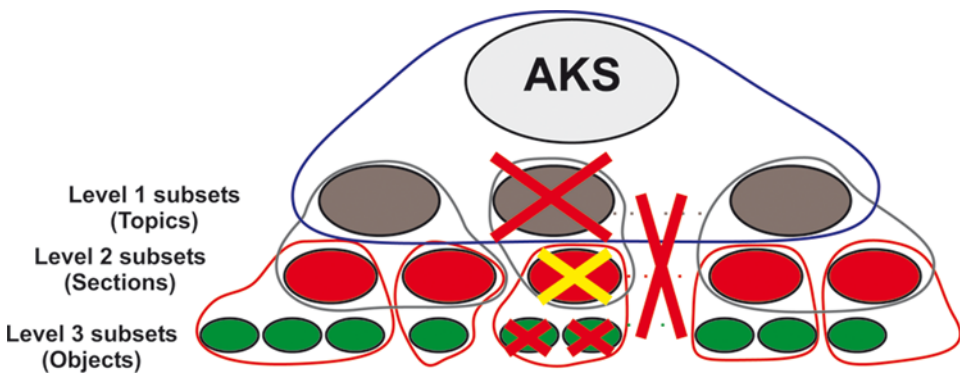


FIGURE 8 Example of first-level evaluation. (Color figure available online.)

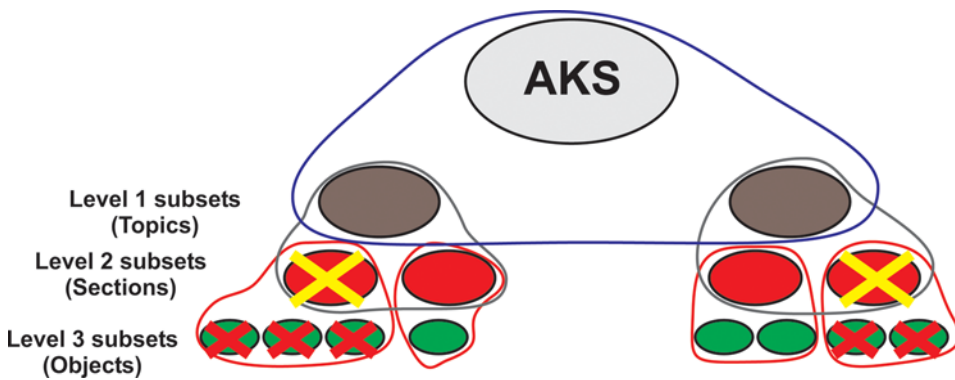


FIGURE 9 Example of second-level evaluation. (Color figure available online.)

those objects whose value of certainty exceeds the Level-3 threshold are identified as related to the query.

Effect of Threshold Value

The threshold value can significantly affect the operation of the system. If a high value is set, only the objects that are strongly related to the query are considered. If the setting is low, fewer objects are rejected, thereby decreasing the efficiency and precision parameters. Thus, it might seem that high values would improve the efficiency of the system. Nevertheless, it should not be forgotten that the HMI is the natural language, and the user is not necessary skilled in the art. If the query is vague or imprecise, the recall parameter would decrease significantly. The test results showed 0.5, 0.55, and 0.65 threshold values for Level 1, Level 2, and Level 3, respectively, which is a configuration that leads to good results when the users have some knowledge about the subject being queried. Fixing the threshold value to 0.5 for all the levels is a more general configuration that also shows good results. In this case, no special requirements are needed from the users.

Another problem related to the thresholds occurs when none of the subsets of a given level has a certainty value higher than the required threshold. In this case, the system response would be: none object related. However, the real problem could be a too-high threshold value, a weak relation, or a missed coefficient in the database.

In order to avoid quitting the procedure in the early stages, the system applies a human reasoning so-called *inkling theory* (Gabora 2000). When a person is asked about something and is obviously reminiscent, a relationship between the question and the memories can be established with great certainty. However, when a person is unable to relate any memories so

strongly, an attempt is made to link weak memories. In fact, the followed process reduces the level of demand to the certainty of the relationship between the question and the related memory. The analogous process followed in the system is to automatically lower the threshold value at the level where none of the subsets takes a value of certainty sufficient to reach the initially set threshold. When none of the subsets in a level take a value of certainty to reach the threshold, SABIO decreases the threshold value on 0.05 steps. This reduction takes place until one of the subsets obtains a value that reaches the new threshold. It should be noted that the other threshold levels remain unchanged, and the modified threshold assumes its original value for new queries. Often, more than one object exceeds the reduced threshold. All of them are accepted and the others are rejected.

LEVEL TERM WEIGHTING

As a result of the procedure described in “Information Retrieval Algorithm,” every object representation needs a “b” coefficient for every defined level. So, for each level, a “b” coefficient associated to the “a” terms belonging to each subset must be calculated. This calculation is one of the most important tasks for IR. To solve the problem, many authors have considered VSM and, specifically, have used the TF-IDF method. In this section, a novel alternative fuzzy logic-based method for TW has been proposed.

In the current proposal, not just the statistical parameters are included in the weighting calculation, but the meaning of the term “a” and the possibility of it being part of a compound term is also taken into account. Thus, TW includes the influence of the affinity between the meaning of the index term and the object itself. The proposed TW scheme is a fuzzy logic-based product of the two meaning-based parameters mentioned earlier, plus two other TF-IDF-based statistical parameters.

Therefore, in the proposed weighting method, the assigned value for the coefficient “b” is related to four parameters that can take values between 0.00 and 1.00. The four proposed parameters and their influence are detailed in the next subsection.

Weighting Related Parameters

The first and most significant parameter is the degree to which the term “a” undoubtedly identifies the object without any other term present in the query. The more identification, the higher is the parameter value. This parameter is a new approach for introducing semantic information in the object representation. An expert in the matter should intuitively

evaluate the importance of the “a” terms. This method is simple, but it has the disadvantage of depending exclusively on the KE. It is very subjective and not possible to completely automate the method. This parameter has no correspondence to any previous method in IR.

The parameter value is given by Table 1.

The second parameter depends on the frequency of occurrence of the term “a” in the representations of the other subsets at the same level in the AKS. The higher the frequency, the lower is the parameter value. This parameter is related with the classical VSM concept of IDF but, in the current case, the assigned value is obtained through a table, and not by any of the usual formulae.

For the construction of the table it was considered that 1% of the most-frequently used words present in the vocabulary define the border for the value 0.00. The most-frequently used words should be understood as those belonging to a higher number of other subset representations. This ranking is made for every considered level. For example, if the vocabulary is 1000 words in size, the one that ranks tenth in the number of appearances in the other subsets of a specific level indicates the number of occurrences for which the parameter value is 0.0 for the considered level.

Continuing the example, it is assumed that the tenth word belongs to the representation of thirteen subsets. An “a” frequency of occurrence greater than or equal to 13 leads to a 0.0 value for this second parameter. This parameter is easily computable by the system, so the table will have 13 columns and 2 rows. The number of occurrences is in the first row, while the associated second parameter value is in the second one. The 0.00 to 1.00 range is divided among the thirteen possible values. The values for the considered example are shown in Table 2.

The third parameter depends on the number of object representations belonging to the same subset where the “a” term appears. The more objects in a set an “a” term belongs to, the higher the value for the corresponding parameter value. This parameter is related with the classical VSM concept of TF but, in our case, the assigned value is again obtained through Table 3.

In the same manner, 1% of the most-often used words define the boundary value 1.00. Consider the same example given previously. In the new most-often used word list, the tenth one sets the number of occurrences from which the parameter value is 1.00. Now, the most-often used

TABLE 1 First Weighting Parameter Value

Does this “a” term undoubtedly define the object by itself?	Yes	Rather	No
1st parameter value	1.0	0.5	0.0

TABLE 2 Second Weighting Parameter Value

Subsets representation to which “a” belongs	0	1	2	3	4	5	6	7	8	9	10	11	12	≥13
2nd Parameter Value	1.00	0.90	0.80	0.70	0.64	0.59	0.53	0.47	0.41	0.36	0.30	0.20	0.10	0.00

words should be intended as those belonging to a higher number of object representations, in the same subset, for the considered level. If the tenth “a” term belongs to five object representations, any “a” term representing six or more objects in a subset takes a value of 1.00 for this parameter.

This parameter is easily computable by the system. Note that this parameter is senseless at the level of the object because all of the subsets are singletons.

The fourth parameter is related to the possibility that the term “a” belongs to a compound term, (i.e., web, mail, and web mail). This parameter increases the semantics precision of the representative ghost of the object. Four cases are considered and the corresponding parameter value is shown in Table 4.

A different approach of including the related terms effect can be found in Chow, Zhang, and Rahman (2009). The system related in the current study is not capable of evaluating this parameter by itself because of the nature of the representation of the objects. Because the relationship between the value of the four parameters involved in the TW task and the final value of “b” term weight coefficient is difficult to express numerically, it seems more appropriate to use a fuzzy reasoning. So, the FE described in “Accumulated Knowledge Objects Normalization Algorithm” is adapted to determine the “b” term value using the four described parameter values as inputs.

Fuzzy Weighting Rules

Now, the FE inputs are the four weighting-parameter values, and the possible input values are High (H), Medium (M), and Low (L). The new output is the “b” value. The possible output values of “b” are High (H), Medium-High (MH), Medium-Low (ML), and Low (L).

TABLE 3 Third Weighting Parameter Value

Object representation to which “a” belongs	1	2	3	4	5	≥6
3rd Parameter Value	0.00	0.30	0.45	0.60	0.70	1.00

TABLE 4 Fourth Weighting Parameter Value

Number of tied "a" terms to the considered one	0	1	2	>2
4th Parameter value	1.00	0.70	0.30	0.00

Another set of rules is defined for the new purpose. Table 5 summarizes the rules.

A system prototype was created to test the performance of the weighting method proposed. This prototype was implemented using Borland C++Builder.

Reduction of Human Dependence

The first and fourth parameters described in "Weighting Related Parameters" require the intervention of a person, preferably a KE, to assign a specific value to them. To avoid this dependence as much as possible and minimize the qualification of the person in charge, the specification requirement is reduced to answering two questions in NL. The first question is: "Does this "a" term undoubtedly define the object by itself?" The response has only three possible values: Yes, Rather, or No. Those values correspond to inputs High, Medium, or Low, respectively. Table 1 shows the possible numerical values for this parameter.

The second question is: "Is this "a" term tied to another one?" The response has only four possible values: "to none," "to another one," "to another two," or "to more than 2." Table 4 shows the possible numerical values for this parameter. These questions are easy enough for anyone introducing knowledge into the system to be able to answer without any special requirements. The goal is to answer these two questions when the object is added to the AKS system as an integral part of the process of adding new objects. With these two parameters, the system has all the data to

TABLE 5 TW Rules

Rule n°	Rule definition	Output
R1	IF P2=H, AND P3 ≠ L	At least MH
R2	IF P2=M, AND P3=H	At least MH
R3	IF P2=L, AND P3=L	Depends on other questions
R4	IF P2=H, AND P3=H	Depends on other questions
R5	IF P1=H	At least MH
R6	IF P4=L	Descends a level
R7	IF P4=M	If the output is ML, it descends to L
R8	IF (R1 and R2) OR (R1 and R5) OR (R2 and R5)	H
R9	Any other case	ML

determine by itself the last level of “b” term values. For higher levels of the AKS, the value taken by the “b” term is the average of the lower levels.

Additionally, there are two important advantages for the new method. On the one hand, TW is close to being automatic, whereas on the other hand, the level of required expertise is much lower. This is because there is no need for an operator to know much about the way FE works, but only to know how many times a keyword appears in every set and the answer to two simple questions: “Does a keyword undoubtedly define an object by itself?” and “Is a keyword tied to another one?”

In “Term Weighting Test,” a test comparing the TF-IDF method and the fuzzy logic-based one was performed.

TESTS, MODIFICATIONS, AND MODEL VALIDATION

A desktop application was created to test the performance of the whole proposed method. This prototype was implemented using Borland C++Builder. Figures 10 and 11 shows its main windows.

The prototype can generate a report detailing the reasoning followed by the system, as shown in Figure 12. This feature has proven to be very helpful in debugging the faults found by determining the failure causes and correcting them.

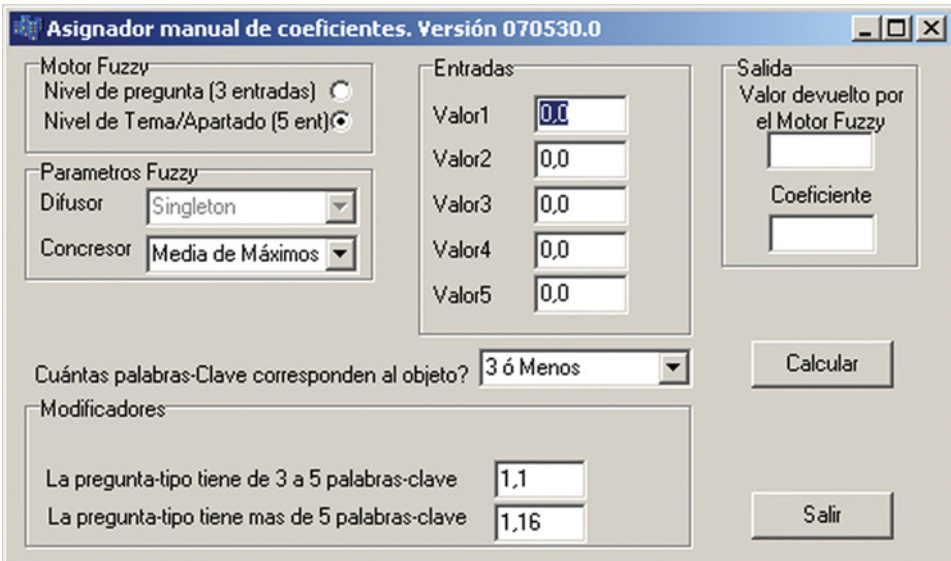


FIGURE 10 Main window application for testing the proposed TW method. (Color figure available online.)

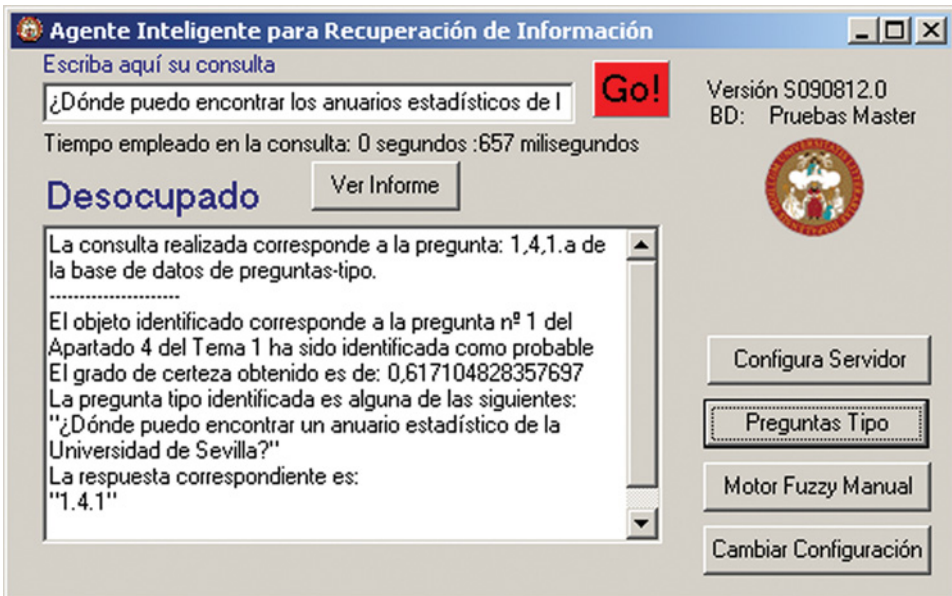


FIGURE 11 Main window of application for testing whole proposed method. (Color figure available online.)



FIGURE 12 Report of the reasoning applied. (Color figure available online.)

For the initial tests, a FAQ set was taken as an AKS system. Every question was treated as an object. All of these objects were normalized, and many standard questions were added to take into account other same-meaning expressions. For this initial test, the TW assignment was manually made, and all the coefficients were kept in a Microsoft Access data base.

The AKS system was built in a three-level structure: Topic, Section, and Object. The system consists of 12 Topics. Every topic is divided into a number between 3 and 12 Sections, and each Section contains between one and eight Objects.

The used validation test was called a “self-test.” It consists of feeding the system with its own standard questions. Although potential users probably would not use the exact same standard questions, the aim of this test is obvious: the system must identify the normalized representation of each object in the AKS. Thus, the system’s standard questions will be used as a query. Moreover, the certainty of the recognition should have a high value, over 0.7.

The prototype is provided with an interface to change the relevant parameters of its configuration. Thus, when the partial test result was admissible, an entire self-test was made.

The position in which the correct answer appeared among the retrieved answers is considered in order to compare the self-test results. The results are grouped into five categories. If the test result belongs to one of the first four (a, b, c, d), it is considered satisfactory. On the contrary, if it belongs to the last (e), the result is considered unsatisfactory. The meaning of each category is shown in Table 6.

For the self test, the configuration of the FE was as follows: input number–3; output number–1; fuzzyficator–singleton; defuzzyficator–COG; thresholds–0.5, 0.5, 0.5 fixed. The obtained results for the first test are shown in Table 7. The test results show a good performance of the method when the object is represented from two to four tuples. For the objects represented by more tuples, the system displays a tendency to consider them related, even when they are not related or are just nearly related.

TABLE 6 Possible Test Results Grouped by Categories

Category	Meaning
a	The correct answer is retrieved as the only answer or it is the one that has a higher degree of certainty among the answers retrieved by the system.
b	The correct answer is retrieved among the two with a higher degree of certainty—excluding the previous case.
c	The correct answer is retrieved among the three with a higher degree of certainty—excluding the previous case.
d	The correct answer is retrieved, but not among the three with a higher degree of certainty.
e	The correct answer is not retrieved by the system.

TABLE 7 Categorized Results of Self-Tests

Category	a (%)	b (%)	c (%)	d (%)	e (%)
1st test	43.51	24.22	8.59	11.72	10.16
2nd test	54.89	12.03	3.01	0.75	29.32
3rd test	69.93	14.29	3.00	0.75	12.03
4th test	77.44	15.79	4.51	0.75	1.51

The test was repeated using a five-input FE and the same settings for the rest of the parameters. The obtained results for this second test are also shown in Table 7. A positive observed effect is that the “a” category improved their performance, which means that the precision increases. On the other hand, the “e” category increases their matching, thereby indicating that recall gets worse. Analyzing the reasoning reports of the system results belonging to the category “e,” it becomes clear that the failure appears because none of the subsets exceed the required threshold in some stage of the procedure. This fact encourages changing the algorithm for determining the relationship. According to this, if no subset has a relationship certainty above the required threshold, the threshold value is decreased in 0.05 steps until any subset exceeds it.

A third test was done using the adaptive threshold algorithm and the same configuration as above. The test results are also shown in Table 7. Many more answers were observed in the “a” category, whereas many fewer answers were observed in the “e” category. This means that the introduced changes to the algorithm improve recall and precision. Analyzing the reasoning report provided by the system, applied to the objects of the category “e,” it is clear that the procedure assigns a lower value for the certainty of relationship in a query when the object is represented by three or fewer tuples. This finding encourages a new change in the IR procedure. According to this fact, if the object representation has three or fewer tuples, a three-input FE is used to determine the degree of certainty of relationship in a query. Otherwise, a five-input FE is used. This new modification is related in the same manner with the VSM normalization concept. However, the proposed scheme is significantly simpler and does not require a recalculation of all the coefficients in the case of a change in the vocabulary of the system. In both cases the adaptive thresholds algorithm is applied.

A fourth test—another autotest—was made using the last procedure modification. The configuration of the FE was as follows: number of inputs—either three or five, depending on the index terms extracted from a query; number of outputs—1; fuzzyficator—singleton; defuzzyficator—COG; thresholds—0.5, 0.5, 0.5 adjustable. The obtained results are shown in Table 7.

The obtained results with this last configuration show significant improvement over any one of the earlier configurations. An increase in both recall (98.49%) and precision (77.44%) was observed. Therefore, it is considered that the results validate the algorithm for determining the degree of certainty of the relationship with the query, and the proposed IR procedure.

Fuzzy Logic Engine Optimization

Once the IR process is set and validated, it is desirable to optimize the FE core. A battery of tests is specified to determine the best fuzzyficator, defuzzyficator, and the most suitable type of universe for the inputs and the output. Table 8 shows the settings of the six proposed self-tests.

The autotests results are shown in Table 9.

The analysis of these results shows the following:

- The couple triangle fuzzyficator and COG defuzzyficator obtain more “e” category results regardless of the type of universe of the inputs and the output.
- The couple singleton and COG obtain more “a” category results in the curved universe for the inputs and the output.
- The couple singleton and COG obtain more “a” + “b” + “c” categories results in the orthogonal universe for the inputs and the output.
- The couple singleton and COG obtain fewer “d” + “e” categories results in the orthogonal universe for the inputs and the output.

Thus, it was concluded that the optimal configuration uses a singleton fuzzyficator, a COG defuzzyficator, a straight input universe, and a straight output universe.

Term Weighting Test

To validate the usefulness of the proposed fuzzy logic-based weighting method, a comparative test between the classical TF-IDF method and the proposed one was suggested. Some of these results were presented in

TABLE 8 Proposed Self-Test to Improve the FE Core Performance

Test n°	Fuzzyficator	Defuzzyficator	Input universe	Output universe
1	Singleton	COG	Straight	Straight
2	Triangle	COG	Straight	Straight
3	Singleton	MOM	Straight	Straight
4	Singleton	COG	Curved	Curved
5	Triangle	COG	Curved	Curved
6	Singleton	MOM	Curved	Curved

TABLE 9 Categorized Results of Improvement of the FE Core Performance Self-Tests

Category test n ^o	a	b	c	d	e
1	77.44	15.79	4.51	0.75	1.51
2	69.17	18.05	3.76	5.26	3.67
3	68.42	15.04	6.77	7.16	2.26
4	75.94	15.79	4.51	1.50	1.50
5	84.21	8.21	1.50	2.26	3.76
6	65.41	18.78	6.02	8.27	1.50

advance in Ropero et al. (2009). A new AKS was built using the objects belonging to the web portal of the University of Seville. This web portal has 50,000 daily visits, which qualifies it into the 10% most visited university portals, and it is ranked 190 among more than 20,000 Universities in the Webometrics rankings for Universities' web impact (Webometrics 2011). Because the information in the university web portal is abundant, 253 objects grouped in 12 topics were defined. All these groups were made up of a variable number of sections and objects. 2107 standard questions surged from these 253 objects. However, slightly more than half of these questions were eliminated for these tests because of being very similar to others. Eventually, the tests consisted of 914 possible user queries.

The formula to obtain the TW coefficient using the TF-IDF product has been modified and improved by many authors to achieve better results in IR and IE. Eventually, the chosen formula for the current tests was the one proposed by Liu et al. (2001)


$$W_{ik} = \frac{tf_{ik} \times \log(N/n_k + 0.01)}{\sqrt{\sum_{k=1}^m (tf_{ik} \times \log(N/n_k + 0.01))^2}}, \quad (4)$$


where tf_{ik} is the i th term frequency of occurrence in the k th subset— Topic/ Section/Object—and n_k is the number of subsets to which the term T_k is assigned in a collection of N objects. Consequently, it has been taken into account that a term might be present in other sets of the collection.

It was suggested to present between 1 and 5 answers, depending on the number of related Objects. The results of the consultation were sorted in

TABLE 10 Categorized Results of TF-IDF vs. SABIO Self-Test


Category	a	b	c	d	e
TF-IDF	466 (50.98%)	223 (24.40%)	53 (5.80%)	79 (10.18%)	93 (10.18%)
SABIO	710 (77.68%)	108 (11.82%)	27 (2.95%)	28 (3.06%)	41 (4.49%)





UNIVERSIDAD DE SEVILLA

Virtual Assistant



Make your question

Q:: bye!
A:: It was a pleasure to help you. See you soon!

Q:: I want information about a scholarship in North America
A:: Hi. In the following web sites all the necessary information is provided in order to apply for a scholarship in North America, as well as a list of the required paperwork
<http://www.internacional.us.es/bostarado-eeuu>

How would you rate my answer?

Good Acceptable Bad

[Close](#) [Legal notice](#)

FIGURE 13 Wizard for navigation on the website of the University of Seville. (Color figure available online.)

the same five categories as those in Table 6, titled “Possible Test Results Grouped by Categories.” The ideal situation comes when the desired Object is retrieved as “a,” though “b” and “c” would be reasonably acceptable. The obtained results are shown in Table 10.

Although the obtained results with the TF-IDF method are quite reasonable, 81.18% of the objects being retrieved among the first five options and more as “a” category, the fuzzy logic-based method turns out to be clearly better, with 92.45% of the desired objects retrieved and more than three-quarters as the first option.

CONCLUSIONS

The current study presents an Information Retrieval system that is able to manage information relating to any kind of knowledge (objects, experience, legislation, professional execution best practices, etc.), and not only to textual knowledge. The human-system interface is natural language.

The hierarchical structure for information classification and storage proposal, in conjunction with the retrieval procedure of the objects related to the query, leads to a lower required computational load, unlike most of the existing procedures.

A novel fuzzy logic-based algorithm for determining the certainty of the relationship between a query and its corresponding subset of the AKS is developed.

The article also presents a novel fuzzy logic-based term weighting algorithm. This novel TW algorithm is easy to use and requires no specialized knowledge. Tests show that this novel algorithm improves the performance when compared to the widely spread classical TF-IDF.

The system described in the current study is being implemented in the development of a Wizard of contents for the website of the University of Seville. At the present time, the Wizard is in a state of internal testing and will shortly be put into production. Figure 13 shows the appearance of the prototype of the application. In the same manner, the presented system can be used to manage information relating to any matter if queries utilize natural language.

The system presented can also be integrated in a multiagent system (MAS) environment in order to manage more complex knowledge. To achieve this goal, complex knowledge has to be able to be split into several simple components parts. Once the complex information is split into several simple faces, the MAS dedicates a soft agent to manage every simple aspect of the whole knowledge. The MAS system should be provided with a special agent to manage and split the received user query. Other special agents in charge of composing the received simple information should also

exist. A more complex answer must be built from the received information from the several existing soft agents.

REFERENCES

- Aronson, A. R., T. C. Rindflesch, and A. C. Browne. 1994. Exploiting a large thesaurus for information retrieval. In *Proceedings of RIAO*, New York, 197–216.
- Bathia, S. K., and J. S. Deogun. 1998. Conceptual clustering in information retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 28 (3): 427–436.
- Ben-Dov, M., and R. Feldman. 2010. Text mining and information extraction. *Data mining and knowledge discovery handbook* doi: 10.1007/978-0-387-09823-4_42.
- Chali, Y. 2009. Question answering using question classification and document tagging. *Journal of Applied Artificial Intelligence* 23 (6): 500–521.
- Chow, T. W. S., H. Zhang, and M. K. M. Rahman. 2009. A new document representation using term frequency and vectorized graph connectionists with application to document retrieval. *Expert Systems with Applications* 36:12,023–12,035.
- Gabora, L. 2000. Toward a theory of creative inklings. In *Art, technology, and consciousness*, ed. R. Ascott, 159–164. Oxford: Intellect Press.
- Gómez, A., J. Ropero, C. León, and A. Carrasco. 2008. A novel term weighting scheme for a fuzzy logic based intelligent web agent. In *ICEIS 2008—Proceedings of the 10th international conference on enterprise information systems*, Barcelona: AIDSS, 496–499.
- Hayashi, H., and Y. Motoharu. 2004. A memory model based on dynamical behavior of the hippocampus. In *Lecture notes in computer science* 3213/2004:967–973, doi: 10.1007/978-3-540-30132-5_130.
- Hornig, Y. J., S. M. Chen, Y. C. Chang, C. H. Lee. 2003. Automatically constructing multirelationship fuzzy concepts networks for document retrieval. *Journal of Applied Artificial Intelligence* 17 (4): 303–328.
- Hornig, Y. J., S. M. Chen, Y. C. Chang, C. H. Lee. 2005. A new method for fuzzy information retrieval based on fuzzy hierarchical clustering and fuzzy inference techniques. *IEEE Transactions on Fuzzy Systems* 139 (2): 216–228.
- Kandel, E. R. 2006. *In search of memory: The emergence of a new science of mind*. New York, NY: W.W. Norton.
- Lee, D. L., H. Chuang, and K. Seamons. 1997. Document ranking and the vector-space model. *IEEE Software* 14 (2): 67–75.
- Liu, S., M. Dong, H. Zhang, R. Li, and Z. Shi. 2001. An approach of multi-hierarchy text classification. In *Proceedings of the international conferences on info-tech and info-net* 3:95–100. Beijing.
- Lu, M., K. Hu, Y. Wu, Y. Lu, and L. Zhou. 2002. SECTCS: Towards improving VSM and naive Bayesian classifier. *IEEE international conference on systems, man and cybernetics*, Hammamet, Tunisia, 5:5.
- Raghavan, V. V., and S. K. Wong. 1986. A critical analysis of vector space model for information retrieval. *Journal of the American Society for Information Science* 37 (5): 279–87.
- Ropero, J., A. Gómez, C. León, and A. Carrasco. 2009. Term weighting: Novel fuzzy logic based method vs. classical tf-idf method for web information extraction. In *ICEIS 2009—Proceedings of the 11th international conference on enterprise information systems*, AIDSS, Milan, Italy, 130–137.
- Ruiz, M., and P. Srinivasan. 1998. Automatic text categorization using neural networks. In *Advances in classification research 8: Proceedings of the 8th ASIS SIG/CR Classification Research Workshop*, ed. E. Efthimiadis, 59–72. Medford, NJ: Information Today.
- Salton, G. and C. Buckley. 1996. Term weighting approaches in automatic text retrieval. *Information Processing and Management* 32 (4): 431–443.
- Sato, N., Yamaguchi, Y. 2010. Simulation of human episodic memory by using a computational model of the Hippocampus. *Advances in Artificial Intelligence* 2010, Article ID 392868, 10 pages, doi: 10.1155/2010/392868.
- Song, Y. I., K. S. Han, S. B. Kim, S. O. Park, and H. C. Rim. 2008. A novel retrieval approach reflecting variability of syntactic phrase representation. *Journal of Intelligent Information Systems* 31:265–286, doi 10.1007/s10844-007-0045-0.
- Webometrics <http://www.webometrics.info/index.html>. 2011.
- Yager, R., and H. Larsen. 1993. Retrieving information by fuzzification of queries. *Journal of Intelligent Information Systems* 2:421–441.