Expert Systems with Applications 39 (2012) 4567–4581

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A Fuzzy Logic intelligent agent for Information Extraction: Introducing a new Fuzzy Logic-based term weighting scheme

Jorge Ropero *, Ariel Gómez, Alejandro Carrasco, Carlos León

Department of Electronic Technology, University of Seville, Av. Reina Mercedes s/n 41012, Spain

ARTICLE INFO

Keywords:
Information Retrieval
Information Extraction
Fuzzy Logic
Vector Space Model
Index terms
Term weighting
Intelligent agent

ABSTRACT

In this paper, we propose a novel method for Information Extraction (IE) in a set of knowledge in order to answer to user consultations using natural language. The system is based on a Fuzzy Logic engine, which takes advantage of its flexibility for managing sets of accumulated knowledge. These sets may be built in hierarchic levels by a tree structure. The aim of this system is to design and implement an intelligent agent to manage any set of knowledge where information is abundant, vague or imprecise. The method was applied to the case of a major university web portal, University of Seville web portal, which contains a huge amount of information. Besides, we also propose a novel method for term weighting (TW). This method also is based on Fuzzy Logic, and replaces the classical TF–IDF method, usually used for TW, for its flexibility.

1. Introduction

The abundant information due to the rise of Information Technology constitutes an enormous advantage for information searchers. Nevertheless, at the same time, a great problem arises as a result of this increase of data: the difficulty to distinguish the necessary information from the huge quantity of unnecessary data.

For this reason, Information Retrieval (IR) and Information Extraction (IE) have hit the scientific headlines strongly recently. Primarily, both were for document retrieval and extraction, but in the last years its use has been generalized for the search for other types of information, such as the one in a database, a web page or, in general, any set of knowledge. Especially, the so-called Vector Space Model (VSM) is much extended. VSM is based on the use of index terms. These index terms are associated with certain weights, which represent the importance of these terms in the considered set of knowledge. These techniques work reasonably well for IE and IR in many areas, but they have the disadvantage of not being so efficient when user queries are not very specific, or when there is an enormous and heterogeneous amount of information.

In this paper, we propose the development of an intelligent agent that is capable of answering to the needs of the users in their process of retrieving the desired information when it is enormous, heterogeneous, vague, imprecise, or not in order. The main contribution in this paper is the creation of a general method for retrieving and extracting information based on the use of Fuzzy Logic (FL). Fuzzy Logic is an ideal tool for the management of this kind of vague and heterogeneous information. Besides, this method has been implemented and validated for IE in web portals, where the information provided were imprecise and disordered.

Another important contribution in this paper deals with automatic term weighting for VSM. A novel Fuzzy Logic-based TW method is described. This method substitutes the TF-IDF term weighting classic method for its flexibility. In order to show the improvement caused by the new method, some tests have been held on the University of Seville web portal. Moreover, an intelligent agent based on our technology has been developed for the University of Seville and it will be functioning soon. Tests have shown an improvement with a better extraction of the requested information through the new method. Besides, the new method is also better for extracting related information, which might be of interest for users.

This paper has been organized in seven sections. Section 2 constitutes an introduction to IE, IR and Natural Language Processing (NLP). Bearing in mind that we have tested our method for a web portal, concepts like data mining and web mining (WM) are also introduced. Also, VSM is described as the intelligent agent which extracts relevant knowledge is based on it. Since this Agent interacts with users in Natural Language, it is also necessary to introduce the techniques for processing it, comparing the semantic approach with the vectorial one.

Section 3 introduces Fuzzy Logic and the state of the art of FL applications for IE and IR.

In Section 4, our concept of intelligent agent is presented. The analysis of current intelligent agent leads us to considering the major disadvantages derived from the current approach. The reasons for the use of FL for designing intelligent agents are also considered.

* Corresponding author. Tel.: +34 954554325.
E-mail address: jropero@dte.us.es (J. Ropero).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.10.009

Section 4 ends up introducing a FL based general method for knowledge extraction in noisy, disordered and imprecise environments. This method is validated in Section 5, by means of applying it for a web portal, where information has these features.

In Section 6, both TF–IDF TW classic method and a new FL based method are introduced. Besides, both methods are used for tests in the University of Seville web portal. A comparative analysis of the results is made.

Section 7 shows the main conclusions of our work.

2. Information Retrieval and Extraction: Natural Language Processing

The access to the contents of an extensive set of accumulated knowledge – a database, a summary of documents, web contents, goods in a store, pictures, etc – is an important concern nowadays. The users of these data collections may find important difficulties to find the required information. These needs become increased when the information is not in the form of text, the user in question is not habituated the matter, there are ambiguous contents, bad organization or, simply, complex topics or a great amount of information difficult to manage.

Section 2 shows how to find useful information in extensive sets of knowledge and different ways of confronting this problem. Given the need to extract information from the enormous quantity of available information, Section 2.1 introduces data mining, focusing on web mining, as we chose a web portal to validate our IE method. Sections 2.2 and 2.3 approach both IR and IE, respectively. Finally, Section 2.4 is dedicated to NLP, which has a cardinal importance in both tasks.

2.1. Web Mining, WM

Data Mining (DM) is an automatic process of analyzing information in order to discover patterns and to build predictive models (Klogsen & Zytkow, 2002) Applications of DM are numerous covering varied fields: e-commerce, e-learning and educational systems (Romero & Ventura, 2007), financial and marketing applications (Vercellis, 2009; Olson & Shi, 2007), problem solving (Liu & Ke, 2007), biology, medicine and bioengineering (Greenes, 2006), telecommunications (Pierre, 2002) Text Mining (Chakrabarti, 2000; Loh, Palazzo, De Oliveira & Gameiro, 2003) and Web Mining (Pal, Talwar, & Mitra, 2002; Kosala & Blockeel, 2000; Tao, Hong, & Su, 2008).

Nowadays the internet users provide enormous quantities of data sources of text and multimedia. The profusion of resources has caused the need to develop automatic technologies of data mining in the WWW, the so-called web mining (Pal et al., 2002).

Web mining may be divided into four different tasks, as it may be seen in Fig. 1 (Etzioni, 1996): IR, IE, generalization and analysis.

Of these tasks, we focus on Information Retrieval and Information Extraction.

2.2. Information Retrieval

Information Retrieval (IR) is the automatic search of the relevant information contained in a set of knowledge, guaranteeing at the same time that non-relevant retrieved information is as less as possible. The aim must be to reach an improvement in retrieval results according to two key concepts in IR: recall and precision. Recall bears in mind the fact that the most relevant objects for the user must be retrieved. Precision takes into account that strange objects must be rejected. (Ruiz & Srinivasan, 1998). An exact definition of recall and precision is given below.

$$\text{Recall} = \frac{\textit{retrieved relevant objects}}{\textit{total number of relevant objects}} \tag{1}$$

$$\text{Precision} = \frac{\textit{retrieved relevant objects}}{\textit{total number of retrieved objects}} \tag{2}$$

For instance, searching in a collection of 100 documents, in which only 20 are relevant for the user, if the search extracts 18 relevant documents and 7 non relevant ones, recall value is 18/20, that is, 90%, whereas precision value is 18/25 (72%).

IR has been widely used for text classification (Aronson, Ridflesch & Browne, 1994; Liu, Dong, Zhang, Li, & Shi, 2001) introducing approaches such as Vector Space Model (VSM), K nearest neighbor method (KNN), Bayesian classification model, neural networks and Support Vector Machine (SVM) (Lu, Hu, Wu, Lu, & Zhou, 2002). VSM is the most frequently used model. In VSM, a document is conceptually represented by a vector of keywords extracted from the document, with associated weights representing the importance of these keywords in the document. Eventually, these methods have been used not only for text classification but for managing a large amount of information of any kind.

In Vector Space Model (VSM), the content of a document is represented by a vector in a multidimensional space. Then, the corresponding class of the given vector is determined by comparing the distances between vectors. The procedure in VSM may be divided into three stages. The first stage consists of indexing the document, where most relevant terms are extracted from the text of the document. The second stage is based on the introduction of a weight for index terms, in order to improve the search of the relevant content for the user. The last stage classifies the document according to a measure of similarity (Raghavan & Wong, 1986).

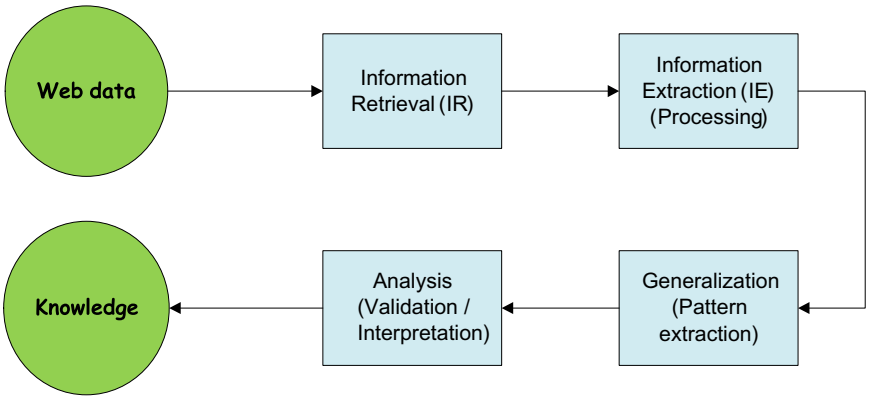

Fig. 1. Web mining tasks.

The most critical stage is the second one, usually called term weighting (TW). Associated weights represent the importance of these keywords in the document. Typically, the so-called TF–IDF method is used for determining the weight of a term (Lee, Chuang & Seamons, 1997). Term Frequency (TF) is the frequency of occurrence of a term in a document and Inverse Document Frequency (IDF) varies inversely with the number of documents to which the term is assigned (Salton, 1988). In Section 6, we discuss the TF–IDF method and we introduce a novel TW Fuzzy Logic based method, which improves the results for Information Extraction.

2.3. Information Extraction

Once documents have been retrieved, the challenge is to extract the required information automatically. Information Extraction (IE) is the task of identifying the specific fragments of a document, which constitute its main semantic content. So far, IE methods involve writing wrappers (Kushmerick, 2002). Some examples of the use of wrappers for IE are STAVIES, which presents a fully automated IE method for web pages (Papadakis, Skoutas, Raftopoulos, & Varvarigou, 2005), or OMINI (Liu, Buttler, Caverlee, Pu, & Zhang, 2005), which introduces tags.

The problem, therefore, is the identification of the fragments of a text that answer to specific questions. Consequently, IE tries to extract new information from the retrieved documents taking advantage of the structure and the representation of the document. Meanwhile, IR experts see the text of a document as a bag of words and do not pay attention to the structure. Scalability is the biggest challenge for IE experts; it is not feasible to build scalable IE systems bearing in mind the size and the dynamism of the web. Therefore, due to the nature of the web, most of IE systems extract information focusing on specific web sites. Other systems use machine learning or data mining techniques for pattern and rule recognition and rules in documents in an automatic or semiautomatic way (Kushmerick, 2002). From this point of view, Web Mining would be part of the Web IE process. The results of this process might be presented in a structured database or as a summary of the texts or original documents.

2.4. Natural Language Processing

Natural Language Processing (NLP) techniques may be used for IR in several ways. As mentioned above, the main aim of using NLP for IR is to improve recall and precision. There are basically two approaches for NLP (Sparck-Jones99), (Aronson & Rindflesch, 1997), (Loh, Palazzo, De Oliveira, & Gameiro, 2003), (Larsen and Yager, 1993), (Berners-Lee and Miller, 2002):

- VSM approach. It is based in the introduction of index terms. An index term may be a keyword (a single word or a word root) or a join term: the latter can be a complex term or a related or similar term. In Fig. 2, different types of index terms are shown.
- Semantic based approach. Though NLP is not an easy task, its potential advantages for IR have made researchers to use both a syntactic approach and a semantic one (Aronson, Rindflesch, & Browne, 1994). It is based on the structure of the considered set of information. A key concept in this field is the concept of ontology (Berners-Lee & Miller, 2002), (Martin & Leon, 2010). Ontology is a common frame or a conceptual automatic and consensual structure to be able to retrieve the required information (Arano, 2003).

Therefore, it is necessary to choose the necessary approach for the web IE system. In the vectorial model, IR and IE are based on the *what* of the information. On the other hand, in semantic webs IR and IE are based on *how* this information is structured. The problem that arises is that, at present, the web does not still provide a great number of ontologies or schemes: only few are and in few matters. Besides, building an ontology from the start turns out to be a hard task and it depends very much on the knowledge engineer who develops it (Iannone, Palmisano, & Fanizzi, 2007). In our research we are inclined for a vectorial approach, though we consider the study of semantic webs a very interesting field of research.

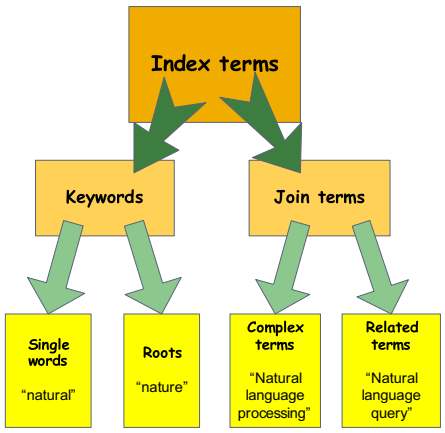

Fig. 2. Different types of index terms.

3. Computational intelligence for knowledge management

3.1. Introduction

It is necessary to consider that the aim of any knowledge access system is to satisfy the needs of the users who access information resources (Larsen, 1999). There are several problems in these knowledge-access systems:

- Information needs are vague or diffuse.
- Information needs change as the user receives this information during his query.
- Users are not conscious of their exact information needs.
- Asking the system about information needs is not usually easy.

Consequently, there is a need to look for a set of methodologies that reflect the notable aptitude of the human being for taking sensible decisions in an imprecise and uncertain environment. This set of methodologies is known as Soft Computing or Computational Intelligence (CI). The main CI tools are Artificial Neural Networks (ANN) and Fuzzy Logic (FL) (Zadeh, 1994).

3.2. Fuzzy Logic applications to knowledge discovery

Search engines, web portals and classic technologies for document retrieval usually consist in searching for keywords in the web. The result may be the finding of thousands of hits, with many of them being irrelevant or maybe not correct or applicable.

There are several approaches at the moment for information handling in an IR system. One of them is based on the Vector Space Model and the other one is related to the concepts of ontology and semantic web. Fig. 3, shows a conceptual scheme on FL applications for IR.

Among VSM based applications for IR, concepts such as queries, clustering, user profiles, and hierarchic relationships take importance (Haase, Steinmann, & Vejda, 2002; Cordon, de Moya, & Zarco, 2004; Mercier & Beigbeder, 2005; Friedman, Last, Zaafrany, Schneider, & Kandel, 2004; Subasic & Huettner, 2001; Horng, Chen,

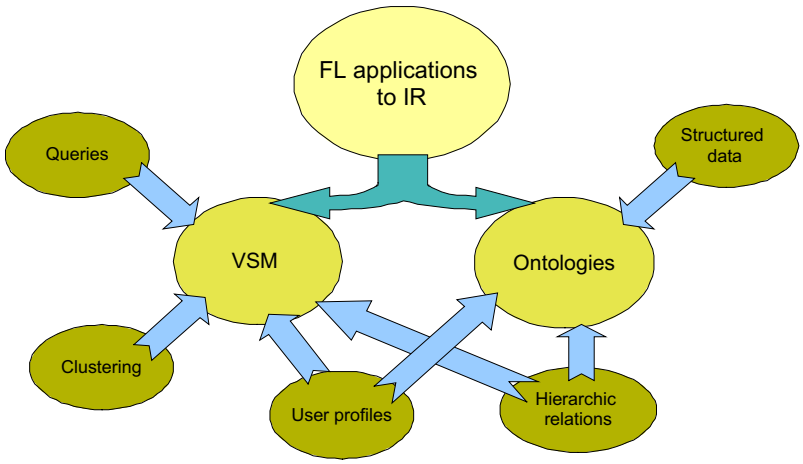

Fig. 3. Conceptual scheme on FL applications for IR.

Chang, & Lee, 2005; Ríos, Velásquez, Yasuda, & Aoki, 2006; Moradi, Ebrahim, & Ebadzadeh, 2008; Zhang & Zhang, 2003). Another possible way of using FL for IR is by means of ontologies. This processing model may help users to have access to the information stored in non-structured or semistructured documents or texts with certain efficiency. Structured data are combined with hierarchic relationships and user profiles for IR applications (Abulaish & Dey, 2005), (Quan, Hui, & Fong, 2006), (Zhai, Wang, & Lv, 2008), (Martin & Leon, 2009).

Anyway, all these applications have something in common with our proposed work on two basic aspects:

- The amount of information is too large to handle.
- The need for a hierarchic structure or the possibility of clustering the information.

Therefore, the ability of FL for the design of an intelligent agent to extract information from a web portal is beyond all doubt.

4. Fuzzy Logic-based intelligent agent

4.1. Intelligent agents for knowledge discovery

The approach to the contents of an extensive set of accumulated knowledge is an important concern nowadays. User needs become increased when the matter is not in the form of text, the user in question is not a habitual user of the matter, there are ambiguous contents, bad organization or, simply, complex topics or a great amount of information difficult to manage (Kwok, 1989). In many cases the solution is to seek some help from an expert on the topic. In fact the person asked to help is an interpreter who is able to generate a syntactically and semantically a correct search obtaining the desired answers. Consequently, there is the need for an agent to interpret the vague information we provide, giving us concrete answers related to the existing contents of the set of knowledge. This should be based on an estimation of the certainty of the relation between what we have expressed in natural language and the contents stored in the set of knowledge (Ropero, Gómez, León, & Carrasco, 2007).

Intelligent Agents, also known as Software Agents, Wizards or Multi-Agent Systems (Turban & Aronson, 2001), are programmed software entities that carry out a series of operations on behalf of a user or another program. They have some degree of independence or autonomy, using some knowledge or representation of the aims or desires of the user (Mengual et al., 2001). If an intelligent agent keeps any kind of conversation with the user, they are also known as Conversational Agents, bots or chatbots (Liao, 2005). At present there are several Intelligent conversational Agents for the most diverse applications, from e-commerce (Ajayi, Aderounmu, & Soriyan, 2009), (Garcia-Serrano, Martinez, & Hernandez, 2004) to virtual education (Kerly, Ellis, & Bull, 2007), (Wik & Hjalmarsson, 2009), or medical uses (Eisman, Lopez, & Castro, 2009), (Bickmore, Pfeifer, & Paasche-Orlow, 2009).

The main problem of most of current agents is, in general, their lack of flexibility. They react well to correct questions, but their answers are far of being too satisfactory when questions are vague or imprecise. And this is the main characteristic when the user is not an expert in the matter – where, in fact, an intelligent agent is more necessary. In addition and also related to this lack of flexibility, many of these agents do not provide more than one answer. It is essential that a user has the possibility of choosing among different chances, as there is a lot of related information in the Internet portals, which might be interesting for the user too.

4.2. Modeling the intelligent agent

4.2.1. Objectives of the intelligent agent

Keeping in mind the limitations of the current intelligent agents, we propose a general IE method using FL for an intelligent agent. An intelligent agent takes advantage of the flexibility the method provides. The method is described in this section. In Section 5 this method is applied to a web portal, using VSM and index terms, based on keywords. As said, above, the information contained in a web page is heterogeneous and vague in most cases, so FL is of great usefulness to find the required information. Besides, we propose a method of consultation based on FL by means of an interface in which it is possible to interact with in NL.

The main objective of the designed system must be to let the users find possible answers to what they are looking for in a huge set of knowledge. With this aim, the whole set of knowledge must be classified into different objects. These objects are the answers to possible user consultations, organized in hierarchic groups. One or more standard questions are assigned for every object, and different index terms from each standard question must then be selected in order to differentiate one object from the others. Finally, term weights are assigned to every index term for every level of hierarchy in a scheme based on VSM. These term weights are the inputs to a FL system. The system must return to the user the objects correspondent with the standard question, or questions that are more similar to the user consultation. The whole process,

together with other concepts defined below, is shown in Fig. 5 (Ropero et al., 2007).

4.2.2. Hierarchic Structure

Provided that the aim of the system is to find the possible answers to user consultations, returning not only the best answer, but also those that are related – user consultations are subject to possible imprecision – it is logical to establish a classification based on a certain criterion or group of criteria. This way, the user might obtain not only the object that is more fitted to his consultation but those that are more closely related.

For instance, in Section 5 we are considering a particular case of IE in a web portal. A hierarchic classification is completely appropriate, since a web portal also has a hierarchic structure. Consequently, it is necessary to identify a web page as an object. That is to say, every web page in a portal is considered to be an object and all these objects are grouped in a hierarchic structure. It is also possible to assign several objects to the same web page if the contained information is heterogenous enough. Likewise, it is necessary to store both the objects and the hierarchic structure of the set of knowledge in databases, as seen in Fig. 4.

4.2.3. Building the intelligent agent

To build the intelligent agent, it is first necessary to bear in mind that user consultations are in Natural Language (NL). We take advantage of this particularity to represent every object as one or several questions in NL, which we have called *standard questions*. Later, it is necessary to extract a series of index terms of the above mentioned standard questions. Finally, term weights must be assigned to these index terms according to the importance of them in the object they are representing. The process consists of two steps:

- The first step is to divide the whole set of knowledge into objects. One or more questions in NL are assigned to every object. The answer to this or these questions must represent the desired object. We have called these questions standard questions. The experience of the Engineer of Knowledge who defines these standard questions as for the jargon in the domain of the set of knowledge is important: the greater his knowledge, the more reliable are the proposed standard questions for the representation of the object. This is due to the fact that they may be more similar to possible user consultations. Nevertheless, it is possible to re-define representations of the object or to add new definitions that should analyze future user consultations and study their syntax and their vocabulary. Consequently, the system can refine his knowledge. In addition, the fact that the intelligent agent is based on FL will provide a greater flexibility.
- The second step is the selection of index terms, which are extracted from standard questions. Index terms represent the most related terms of standard questions with the represented object.

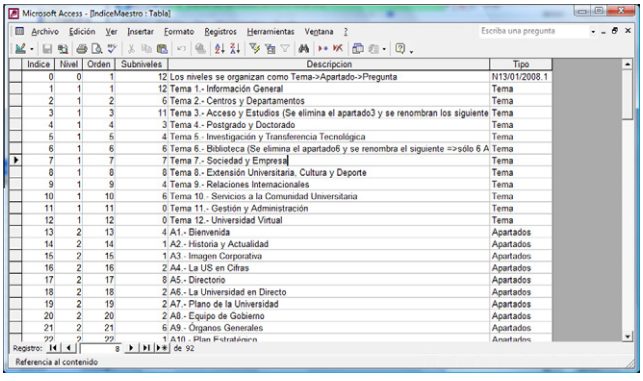

Fig. 4. Database containing the information in a web portal grouped hierarchically.

These index terms may be identified with keywords, but they may be compound terms, too. There exists the need of a series of coefficients associated with index terms whose values must be related somehow to the importance of an term index in the set of knowledge it is representing - it is to say, the importance of the term in every level of the hierarchic structure. These index terms must be stored in a database along with their corresponding weights, corresponding to each of the hierarchic levels. We may consider mainly two methods for term weighting (TW):

- Let an expert in the matter evaluate intuitively the importance of index terms. This method is simple, but it has the disadvantages of depending exclusively on the engineer of the knowledge, it is very subjective and it is not possible to automate.
- Automate TW by means of a series of rules.

Given the large quantity of information there is in a web portal, we choose the second option and we propose a VSM method for TW. The most widely used method for TW is the so-called TF–IDF method. Nevertheless, in this paper we also propose a modification of the method based on the use of FL. This method is described in Section 6. Every index term has an associated weight. This weight has a value between 0 and 1 depending on the importance of the term in every hierarchic level. The greater is the importance of the term in a level, the higher is the weight of the term. In addition, it is necessary to bear in mind that the term weight might not be the same for every hierarchic level, provided that the importance of a word to distinguish, for example, a section from another may be very different from its importance to distinguish between two objects. In short, the whole process of building of the intelligent agent is as summarized in Fig. 5.

For example, a web page may be divided in one o more objects according to the quantity of information it contains. Actually, every object is an answer to every possible user consultation. Since it is possible that several questions drive to the same answer, one or more standard questions may be defined for the same object. Once standard questions are defined, it is necessary to extract the index terms and to assign a weight to them. Index terms and their corresponding weights must be stored in respective databases that constitute a hierarchic structure.

4.3. Mode of operation of the intelligent agent

Once the intelligent agent has been built, it is necessary to know its mode of operation, that is to say, how it works when it receives a user consultation. Index terms are extracted by comparison with the contained ones in their corresponding database. The weights of these index terms for every level constitute the input to an FL system. At this point, the hierarchic structure of the system becomes important. The whole set of knowledge, which constitutes the hierarchic level 0, is divided into level 1 subsets. For each level 1 subset, index terms must have certain weights, which are the possible inputs to an FL engine. The FL engine provides an output for every subset. These outputs are called degrees of certainty. If the degree of certainty corresponding to a subset is lower than a predefined value, named threshold, the content of the corresponding subset is rejected. The aim of using a hierarchic structure is to make possible the rejection of a great amount of content, which will not have to be considered in future queries. For every subset that overcomes the threshold of certainty, the process is repeated. Now, the inputs to the FL engine are the level 2 weights for the corresponding index terms. For the outputs for level 2 subsets, those outputs with a degree of certainty that does not overcome a threshold are rejected

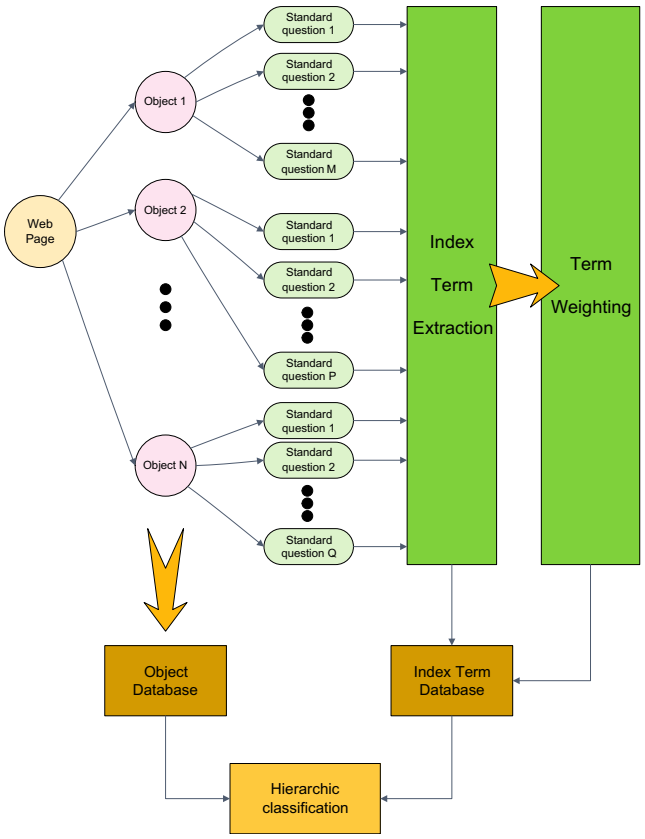

Fig. 5. Process of building an intelligent agent.

again. Otherwise, the process is repeated up to the last level. The final output corresponds to the elements of the last level – that is to say, the objects – whose degree of certainty overcomes the definitive threshold. There is the possibility of several answers. The more vague the queries are, the more answers are obtained. In Fig. 6, the complete process for a two-level hierarchic structure is shown. The whole set of knowledge – level 0 – is grouped in level 1 subsets and these are clustered in level 2 subsets. Since this is the last level, these subsets are own objects.

An application of this methodology for a web portal is described in Section 5 of this paper.

4.4. Fuzzy Logic system

The element of the intelligent agent which determines the degree of certainty for a group of index terms belonging or not to every of the possible subsets of the whole set of knowledge is the fuzzy inference engine. The inference engine has several inputs – the weights of selected index terms – and gives an output – the degree of certainty for a particular subset in a particular level. For the fuzzy engine, it is necessary to define:

- The number of inputs to the inference engine of inference. It depends on the extracted index terms, so it is variable. The inputs are the higher weights of the extracted index terms for every hierarchic level Likewise, it is suitable to define a maximum number of inputs to avoid too vague consultations and, therefore, retrieving too many objects.
- Input fuzzy sets: input ranges, number of fuzzy sets, and shape and range of membership functions.
- Output fuzzy sets: output ranges, number of fuzzy sets, and shape and range of membership functions.

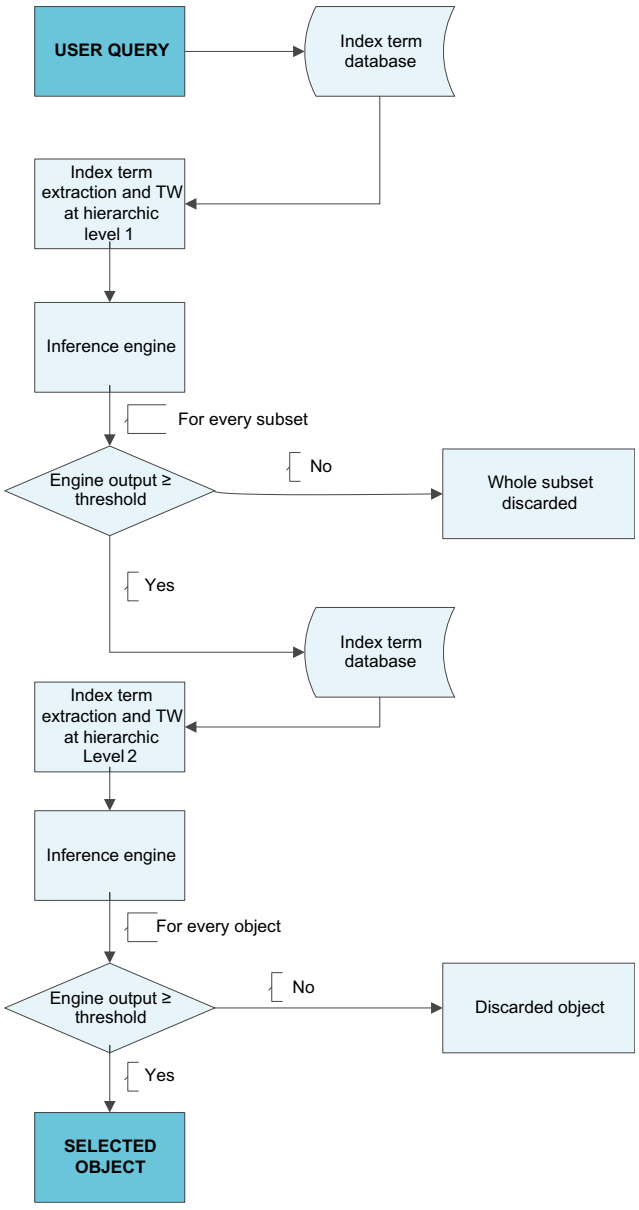

Fig. 6. The complete process of a two-level hierarchic structure.

- Fuzzy rules. They are of the IF . . . THEN type. Examples of these rules (81 in total for three inputs) are:
  - IF all inputs are LOW, THEN output is LOW (1 rule for three inputs).
  - IF one input is MEDIUM and the others are LOW, THEN output is MEDIUM-LOW (three rules for three inputs).
  - IF all inputs are MEDIUM or one input is HIGH, THEN output is HIGH (four rules for three inputs).
- Used methods for AND and OR operations and defuzzifying.

All these parameters must be taken into account to find the optimal configuration for the inference engine, core of the intelligent agent. The study of the suitable parameters in the case of a web portal is also described in Section 5.

5. Tests and results

So far, a general method for IR and IE has been proposed. Although we have stood out the method suitability for web applications, this section focuses on the use of this method for IE in web

portals, with the design of an intelligent agent for the web portal of the University of Seville, studying the best parameters for the FL system.

This web portal has 50,000 daily visits, which qualifies it into the 10% most visited university portals and is ranked 223 among more than 4,000 Universities in Webometrics rankings for Universities' web impact (Webometrics., 2009). Moreover, the intelligent agent based on our design is about to start functioning in the University of Seville web portal.

5.1. Set of Knowledge Structure

As 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.

As said above, to carry out IE, it is necessary to identify the web page and object, that is to say, every web page in a portal is considered an object. These objects are gathered in a hierarchical structure. Every object is accessible across a unique way of a classification tree. An object is classified under a unique criterion – or group of criteria – (Gómez, Ropero, León, & Carrasco, 2008).

Every object is represented by means of a set of questions, called *standard questions,* formulated in NL. The number of standard questions associated with every web page is variable, depending on the amount of information contained in every page, its importance and the number of index terms synonymous. Logically, system administrator's knowledge about the jargon of the related field is pretty important. The more knowledgeable he is, the higher the reliability of the proposed standard questions becomes, as they shall be more similar to possible user consultations. After all, users are the ones who extract the information.

5.2. Methodology of the intelligent agent

Our study was based both on the study of the web pages themselves and on previous consultations – University of Seville bank of questions. Once standard questions are defined, index terms are extracted from them. We have defined these index terms as words, though there also may be compound terms. Index terms are the ones that better represent a standard question. Every index term is associated with its correspondent term weight. This weight has a value between 0 and 1 and depends on the importance of the term in every hierarchic level. The higher the importance of a term in a level, the higher is the term weight. In addition, term weight is not constant for all levels, as the importance of a word to distinguish a topic from the others may be very different from its importance to distinguish between two objects. An example of the followed methodology is shown in Table 1.

On the other hand, the final aim of the intelligent agent must be to find the Object or Objects whose information is more similar to the requested user consultation. The process that the intelligent agent follows to extract the information related to the user consultation was described in detail in Section 4. To clarify further, we take up the example in Table 2. In the example, a user asks "Which services can I access as a virtual user at the University of Seville?", which corresponds to one of the defined standard questions. We show the process followed by the intelligent agent to extract the requested information and the related one.

Table 1
Example of the followed methodology.

Step	Example
Step 1: Web page identified by standard question/s	– Web page: www.us.es/univirtual/internetwww.us.es/univirtual/internet – Standard question: Which services can I access as a virtual user at the University of Seville?
Step 2: Locate standard question/s in the hierarchic structure.	Topic 12: Virtual University Section 6: Virtual User Object 2
Step 3: Extract index terms	Index terms: 'services', 'virtual', 'user'
Step 4: Term weighting	See Section 6

In this case, the requested information was retrieved, since the user consultation – Which services can I access as a virtual user at the University of Seville? – actually corresponds to a standard question. This standard question refers to Object 12.6.2 (abbreviated notation for the Object corresponding to Topic 12, Section 6, Object 2). In addition, other Objects overcome the defined threshold. Standard questions associated with these Objects are shown in Table 3.

As mentioned above, the first standard question corresponds to the desired Object. In addition, an important advantage is obtained: both the following retrieved standard questions are very much related to the desired Object – they are also related to the Virtual User –. So these Objects may be interesting for the user. The following standard questions are not so similar, but they are somehow related to the query. Our suggestion is to present the web page associated with the first Object to the user and, in another window, among three and five of the following retrieved options.

Moreover, the fact of retrieving other very much related Objects leads us to a conclusion: when the user consultation does not match exactly to any of the stored Objects, the system will try to find the most similar ones. This flexibility is one of the most important advantages of the use of FL.

5.3. Fuzzy Logic engine

As said in Section 4, the core of the intelligent agent is the FL system. For the FL system, we have to consider parameters such as the number of inputs and outputs, fuzzy sets and fuzzy rules.

To prove the efficiency of the proposed system and improve benefits, it was necessary to test the FL system in order to define the suitable parameters for a set of accumulated knowledge. As the portal of the University of Seville has a great amount of information, we tested our method with a more reduced set of knowledge. We used the bank of most frequent questions – answers of the University of Seville. This bank of questions – answers is considered our set of knowledge. It consists of 117 questions, and the results obtained from its use, due to the generality of the method, are applicable to any set of knowledge and, especially, to a web portal.

The first goal of these tests is to check that the system makes a correct identification of standard questions with an index of certainty higher than a certain threshold. The use of Fuzzy Logic makes it possible to identify not only the corresponding standard question but others as well. This is related to the concept of *recall*, though it does not match that exact definition (Ruiz & Srinivasan, 1998). The second goal is to check whether the required standard question is among the three answers with higher degree of certainty. These three answers should be presented to the user. The correct answer must be among these three options. This is related to *precision*, though it does not match that exact definition either.

To do the tests, the so-called standard questions were used as consultations in the Natural Language. The index terms for every standard question must be defined enough to identify the Object related to that standard question. Test results for standard question recognition fit into five categories:

Table 2
FL system response to a user consultation.

Step 1: User query in NL.

Which services can I access as a virtual user at the University of Seville?

Step 2: Index term extraction.

Index term	T1W	T2W	T3W	T4W	T5W	T6W
Services	0.14	0	0	0	0	0.16
User	0	0	0	0	0	0
Virtual	0	0	0.16	0	0	0
Index term	**T7W**	**T8W**	**T9W**	**T10W**	**T11W**	**T12W**
Services	0.16	0	0	0.14	0.16	0.15
User	0	0	0	0.29	0	0.6
Virtual	0	0	0	0	0.16	0.53

TiW = Term Weight Vector for Topic i.

Step 3: Weight vectors are taken as inputs to the fuzzy engine for every topic.

T1O	T2O	T3O	T4O	T5O	T6O
0.29	0.13	0.30	0.13	0.13	0.30
T7O	**T8O**	**T9O**	**T10O**	**T11O**	**T12O**
0.30	0.13	0.13	**0.43**	0.39	**0.62**

TiO = Fuzzy engine output for Topic i.

* Topics 10 and 12 are over the considered threshold – 0.4 in our case.

Step 4: Step 3 is repeated for the next hierarchic level – Sections of the selected Topics.

Index term	T12S1W	T12S2W	T12S3W	T12S4W	T12S5W	T12S6W
Services	0.37	0	0.16	0	0	0.12
User	0	0	0	0	0	0.6
Virtual	0.33	0	0	0.16	0.16	0.45

TiSjW = Term Weight Vector for Topic i, Section j.

T12S1O	T12S2O	T12S3O	T12S4O	T12S5O	T12S6O
0.51	0.13	0.30	0.30	0.30	**0.59**

TiSjO = Fuzzy engine output for Topic i, Section j.

* Topic 10 must also be considered, but we are considering only Topic 12 for simplicity

Step 5: Step 3 is repeated for the next hierarchic level – Objects of the selected Sections.

Index term	T12S6O1W	T12S6O2W	T12S6O3W
Services	0	0.4	0
User	0.57	0.52	0.52
Virtual	0.57	0.52	0.52

TiSjOkW = Term Weight Vector for Topic i, Section j, Object k.

T12S6O1O	T12S6O2O	T12S6O3O
0.6045	**0.7413**	**0.6005**

TiSjOkO = Fuzzy engine output for Topic i, Section j, Object k.

* Topic 12, Section 1 must also be considered, but we are considering only Topic 12, Section 6 for simplicity

1. The correct question is the only one found or the one that has the highest degree of certainty.
2. The correct question is one between the two with the highest certainty or is the one that has the second highest degree of certainty.
3. The correct question is one among the three with the highest degree of certainty or is the one that has the third highest certainty.
4. The correct question is found but not among the three with the highest degree of certainty.
5. The correct question is not found.

These tests are useful to determine the ideal parameters of the FL system for IE. These parameters are described in the following sections.

5.3.1. I/O variables

As said above, the intelligent agent must extract the index terms during a user consultation. The N index terms with a higher weight for evey level of the hierarchy are chosen as inputs to the FL inference engine. Therefore, the first item to do is to determine the suitable number of inputs to the system. The fact of organizing the content hierarchically avoids the need for consulting for the Objects one by one. This is due to the fact that subsets of knowledge whose correspondent output – the output for the FL engine – is lower than a certain threshold are discarded.

In tests, we have considered thresholds of 0.5 for all levels, although these can be modified to obtain better results. In addition, fuzzy sets were defined for inputs and outputs, together with fuzzy rules. The way they were defined is explained in next section.

Table 3
Associated standard questions for the objects retrieved in the example.

Position	Object	Certainty (%)	Associated standard question
1	12.6.2	74.13	Which services can I access as a virtual user at the University of Seville?
2	12.6.1	60.45	I would like to request for an account as a virtual user at the University of Seville
3	12.6.3	60.05	I do not remember my Virtual User password at the University of Seville
4	12.1.5	54.07	I would like to access the Economic Services at the Virtual Secretariat of the University of Seville
5	12.1.6	54.07	I would like to access the management services at the virtual secretariat of the University of Seville
6	10.4.9	48.96	What services does the service of computers and communications offer?
7	12.1.1	41.04	How can I access the virtual secretariat at the University of Seville?

As for the input to the FL system, one to five index terms can be extracted from a consultation. We consider that more than five index terms may not be relevant for IE, so two fuzzy engines were defined: a three-input fuzzy engine and a five-input one. Tests show that using few inputs to a fuzzy engine causes a rapid saturation of the system. This is a great disadvantage for precision: 90% of the correct Objects are detected but only half of them are the first option, as may be seen in Table 4, where results for a three-input fuzzy engine are shown, among the results for other configurations of the engine.

Nevertheless, when a five-input fuzzy engine is used, there are very low values in the degree of certainty. Precision rises to 55%, but recall decreases, as shown in Table 4.

Therefore, we concluded that a low number of inputs affect precision in a negative way, whereas a high number of inputs affect recall. Nevertheless, some improvements may take effect ifa variable number of inputs are used. This point is explained later. In addition, from the analysis of unsuccessful results, it was observed most of the times, the desired Object was not retrieved because the output was below the fixed threshold. There is the possibility of lowering the thresholds of certainty to accept the result as correct. However, this modification takes many erroneous answers as valid, spoiling part of the previous results. The proposed solution is to modify the procedure so that the intelligent agent lowers automatically the fixed threshold only in case that no result overcomes it. With this method, results improve remarkably, as may also be seen in Table 4.

In summary, if the three most probable Objects are retrieved for the user, the desired Object is retrieved 88% of times, and it is the first option 70% of the times.

As for the number of inputs, it is necessary to bear in mind that sometimes it is better to use the three-input fuzzy engine and sometimes the five-input one. We propose a commitment using an input number variable engine dependent on the number of index terms extracted from the user consultation. In the case among one and three extracted index terms, the three-input engine is used. Otherwise, the five-input engine is utilized. Results are shown in Table 4. If the three most probable Objects are retrieved for the user, the desired Object is retrieved 97% of the times, and 77% of the times it is the first option. We consider then that the best choice for I/O parameters is the use of a fuzzy engine with variable output thresholds and a variable number of inputs. This number depends on the number of extracted index terms from a user consultation.

Table 4
Results for different engine configurations. Cat, category.

	Cat1	Cat2	Cat3	Cat4	Cat5
Three-input fuzzy engine results	45%	24%	9%	12%	10%
Five-input fuzzy engine results.	55%	12%	3%	1%	29%
Five-input fuzzy engine results with variable output thresholds.	70%	14%	3%	1%	12%
Five-input fuzzy engine results with variable output thresholds and variable input number fuzzy engine.	77%	16%	4%	1%	2%

5.3.2. I/O Fuzzy set definition

Input range corresponds to weight range for every index term so it is between 0.0 and 1.0. We considered three fuzzy sets represented by the values LOW, MEDIUM and HIGH. Likewise, for simplicity, all of them are kept as triangular although sets were modified later in order to find their ideal shape. The output, which gives the degree of certainty, is also in the 0–1 range, where 0 is the minimum certainty and 1 is the maximum one. Output may be LOW, MEDIUM-LOW, MEDIUM-HIGH and HIGH. These values correspond to output fuzzy sets. The fact that the input takes these three values – LOW, MEDIUM and HIGH – is due to the fact that the number of fuzzy sets is enough so that results are coherent and there are not so many options to let the number of rules increase considerably – in next section this feature is commented, but it seems to be clear that, the higher the number of values, the number of rules defined must be higher. In fact, the outputs were also defined this way – three fuzzy sets – in the beginning, but we introduced one more set for the outputs as we observed a considerable improvement in the tests we made with this modification. The range of values for every input fuzzy set is as follows (Fig. 7):

- LOW, from 0.0 to 0.4 centered in 0.0.
- MEDIUM, from 0.2 to 0.8 centered in 0.5.
- HIGH, from 0.6 to 1.0 centered in 1.0.

The range of values for every output fuzzy set is:

- LOW, from 0.0 to 0.4 centered in 0.0.
- MEDIUM-LOW, from 0.1 to 0.7 centered in 0.4.
- MEDIUM-HIGH, from 0.3 to 0.9 centered in 0.6.
- HIGH, from 0.6 to 1.0 centered in 1.0.

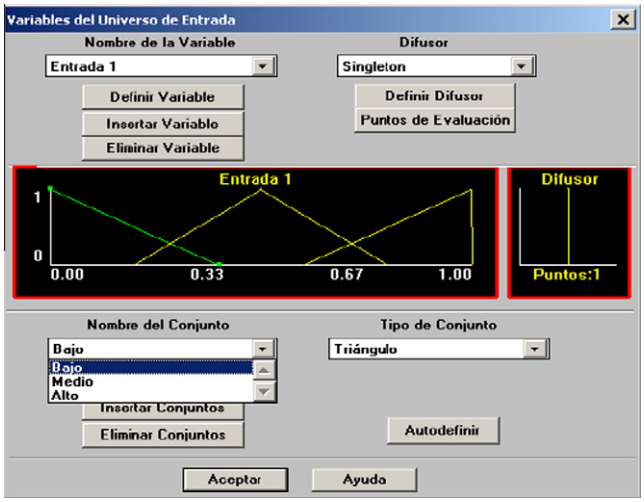

Fig. 7. Fuzzy input set definition.

Table 5
Fuzzy rules for a three input engine.

Rule number	Rule definition	Output
R1	IF one or more inputs = HIGH	HIGH
R2	IF three inputs = MEDIUM	HIGH
R3	IF two inputs = MEDIUM and one input = LOW	MEDIUM–HIGH
R4	IF one input = MEDIUM and two inputs = LOW	MEDIUM–LOW
R5	IF all inputs = LOW	LOW

Eventually, after tests, we came to the conclusion that he best option is the use of:

- Triangular fuzzy sets.
- Singleton fuzzifier.
- Center of gravity defuzzifier.

5.3.3. Rule definition

Once the number of inputs and fuzzy sets has been defined, it is necessary to define the rules for the inference fuzzy engine. Previously, we had established the suitability of implementing a variable input number engine according to the number of index terms extracted from a user consultation. In practice, this causes the implementation of two different inference engines:

- If three or less index terms are extracted, a three input engine is used.
- If more than three index terms are to be extracted, a five input engine is used – if more than five index terms are extracted, only the most significant ones are considered.

Besides, three fuzzy sets had been defined for every input. For a three input engine of three $3^3 = 27$ fuzzy rules were defined, whereas for the five input engine it is necessary to define $3^5 = 243$ rules. This is one of the reasons why more fuzzy sets were not defined: with only one more fuzzy set, it would be necessary define $4^5 = 1024$ rules for the five input engine.

As an example, fuzzy rules defined for the three input engine may be seen in Table 5. These rules cover 27 possible combinations.

6. Fuzzy Logic-based term weighting scheme

Term weighting (TW) is one of the major challenges in IE and IR. The most extended model for IR and IE, as was mentioned in Section 2, is VSM. In VSM, the importance of a term in a subset of knowledge is given by a certain associate weight (Lee et al., 1997). In Section 6.1, there is a brief introduction to TW. In Section 6.2, classic method for TW, so-called TF–IDF is analyzed, and in Section 6.3, we introduce the novel proposed method, based on FL. The values of the weights must be related somehow to the importance of an index term in its corresponding set of knowledge – in our case, Topic, Section or Object. We may consider two options to define these weights:

- An expert in the matter should evaluate intuitively the importance of the index terms: This method is simple, but it has the disadvantage of depending exclusively on the knowledge engineer. It is very subjective and it is not possible to automate the method.
- The generation of automated weights by means of a set of rules: The most widely used method for TW is the TF–IDF method, but we propose a novel Fuzzy Logic based method, which achieves better results in IE.

When a large amount of information needs to be managed, the first option is unfeasible, for it is tedious, dense, and a high level of mastery is necessary on the part of the engineer of knowledge in charge of this task. It is necessary, so, to automate TW.

6.1. The TF–IDF method

Although it was in the late 1950s when the idea of automatic text retrieval systems based on the identification of text content and associated identifiers originated, it was Gerard Salton in the late 1970s and the 80s who laid the foundation for the existing relation between these identifiers and the texts they represent (Salton & Buckley, 1996). Salton suggested that every document D could be represented by term vectors t_k and a set of weights w_{dk}, which represent the weight of the term t_k in document D, that is to say, its importance in the document.

A TW system should improve efficiency in terms of two main factors, recall and precision, as it was mentioned in Section 2. Recall bears in mind the fact that the most relevant objects for the user must be retrieved. Precision takes into account that strange objects must be rejected (Ruiz & Srinivasan, 1998). Recall improves if high-frequency terms are used, because such terms will make it possible to retrieve many objects, including the relevant ones. Precision improves if low-frequency terms are used, as specific terms will isolate the relevant objects from the non-relevant ones. In practice, compromise solutions are used, using terms which are frequent enough to reach a reasonable level of recall without producing a too low precision.

Therefore, terms that are mentioned often in individual objects, seem to be useful to improve recall. This suggests the utilization of a factor named term frequency (TF). Term frequency (TF) is the frequency of occurrence of a term. On the other side, a new factor should be introduced. This factor must favor the terms concentrated in a few documents of the collection. The inverse frequency of document (IDF) varies inversely with the number of objects (n) to which the term is assigned in an N-object collection. A typical IDF factor is log (N/n) (Salton & Buckley, 1996). A usual formula to describe the weight of a term j in document i is given in Eq. 3.

$$W_{ij} = tf_{ij} x idf_j. \tag{3}$$

This formula has been modified and improved by many authors to achieve better results in IR and IE (Lee et al., 1997), (Liu & Ke, 2007), (Zhao & Karypis, 2002), (Lertnattee & Theeramunkong, 2003).

6.2. The FL-based method

The TF-IDF method works reasonably well, but it has the disadvantage of not considering two key aspects for us, as it was explained in ref. (Ropero et al., 2009)

- The first parameter is the degree of identification of the object if only the considered index term is used. This parameter has a strong influence on the final value of a term weight if the degree of identification is high. The more a keyword identifies an object, the higher the value for the corresponding term weight. Nevertheless, this parameter creates two disadvantages in terms of practical aspects when it comes to carrying out a term weight automated and systematic assignment. On the one hand, the degree of identification is not deductible from any characteristic of a keyword, so it must be specified by the System Administrator. The assigned values could be neither univocal nor systematic. On the second hand, the same keyword may have a different relationship with different objects.

- The second parameter is related to join terms. In the index term 'term weighting', this expression would constitute a join term. Every single term in a join term has a lower value than it would have if it did not belong to it. However, if we combine all the single terms in a join term, term weight must be higher. A join term may really determine an object whereas the appearance of only one of its single terms may refer to another object.

The consideration of these two parameters together with classical TF and IDF determines the weight of an index term for every subset in every level. The FL-based method gives a solution to both the problems and also has two main advantages. The solution to both problems is to create a table with all the keywords and their corresponding weights for every object. This table will be created in the phase of keyword extraction from standard questions. Imprecision practically does not affect the working method due to the fact that both term weighting and Information Extraction are based on Fuzzy Logic, which minimizes possible variations of the assigned weights. The way of extracting information also helps to successfully overcome this imprecision. In addition, the two important advantages are the term weighting is automated; and the level of required expertise for an operator is lower. This operator would not need to know anything about the FL engine functioning, but would know only how many times does a term appear in any subset and the answer to these questions:

- Does a keyword undoubtedly define an object by itself?
- Is a keyword tied to another one?

In our case, the application of this method to a web portal, the web portal developer himself may define simultaneously the standard questions and index terms associated with the object – a web page – and the response to the questions mentioned above.

6.3. Implementation of both methods

This section shows how the TF–IDF method and the FL-based method were implemented in practice, in order to compare both methods applying them to the University of Seville web portal.

As mentioned in previous sections, a reasonable measure of the importance of a term may be obtained by means of the product of TF and IDF (TF × IDF). However, this formula has been modified and improved by many authors to achieve better results in IR and IE. Eventually, the chosen formula for our tests was the one proposed by Liu et al. (2001)

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

Here tf_{ik} is the ith term frequency of occurrence in the kth subset – Topic/Section/Object – n_i is the number of subsets to which the term T_i is assigned in a collection of N objects. Consequently, it is taken into account that a term might be present in other sets of the collection.As an example, we are using the term 'virtual', as used in the example in Section 5.

At Topic level:

- 'Virtual' appears 8 times in Topic 12 ($tf_{ik} = 8$, $K = 12$).
- 'Virtual' appears twice in other Topics ($n_k = 3$)
- There are 12 Topics in total ($N = 12$) – for normalizing, it is only necessary to know the other tf_{ik} and n_k for the Topic –.
- Substituting, $W_{ik} = 0.20$.

At Section level:

- 'Virtual' appears 3 times in Section 12.6 ($tf_{ik} = 3$, $K = 6$)
- 'Virtual' appears 5 times in other Sections in Topic 12 ($n_k = 6$)
- There are 6 Sections in Topic 12 ($N = 6$).
- Substituting, $W_{ik} = 0.17$.

At Object level:

- 'Virtual' appears once in Object 12.6.2 ($tf_{ik} = 1$, $K = 2$). – Logically a term can only appear once in an Object -.
- 'Virtual' appears twice in other Topics ($n_k = 3$)
- There are 3 Objects in Section 12.6 ($N = 3$).
- Substituting, $W_{ik} = 0.01$. In fact, 'virtual' appears in all the Objects in Section 12.6, so it is irrelevant to distinguish the Object.

Consequently, 'virtual' would be relevant to find out that the Object is in Topic 12, Section 6, but irrelevant to find out the definite Object, which should be found according to other terms in a user consultation.

However, TF-IDF has the disadvantage of not considering the degree of identification of the object if only the considered index term is used and the existence of tied keywords. Like TF-IDF method, it is necessary to know TF and IDF, and also the answer to the questions mentioned above. FL-based term weighting method is defined below. Four questions must be answered to determine the Term Weight of an Index Term:

- Question 1 (Q1): How often does an index term appear in other subsets? – Related to IDF.
- Question 2 (Q2): How often does an index term appear in its own subset? – Related to TF.
- Question 3 (Q3): Does an index term undoubtedly define an object by itself?
- Question 4 (Q4): Is an index term tied to another one?

The answer to these questions gives a series of values which are the inputs to a Fuzzy Logic system, called Weight Assigner. The output of the Weight Assigner is the definite weight for the correspondent index term. The followed scheme may be observed in Fig. 8.

Subsequently, the way of defining input values associated with each of four questions is described.

6.3.1. Question 1

Term weight is partly associated with the question 'How often does an index term appear in other subsets?'. It is given by a value between 0 – if it appears many times – and 1 – if it does not appear in any other subset. To define weights, we are considering the times that the most used terms in the whole set of knowledge appear. The list of the most used index terms is as follows:

1. Service: 31 times.
2. Services: 18 times.
3. Library: 16 times.
4. Research: 15 times.
5. Address: 14 times.

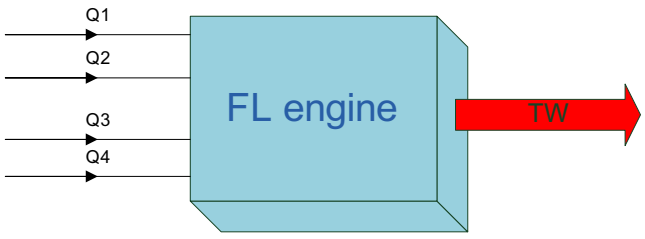

Fig. 8. TW generation method scheme.

+. Student: 14 times
7. Mail: 13 times.
+. Access: 13 times.
9. Electronic: 12 times.
+. Computer: 12 times.
+. Resources: 12 times.
12. Center: 10 times.
+. Education: 10 times.
+. Registration: 10 times.
+. Program: 10 times.

Provided that there are 1114 index terms defined in our case, we think that 1% of these words must mark the border for the value 0 (11th word). Therefore, whenever an index term appears more than 12 times in other subsets, we will give it the value of 0. Values for every Topic are defined in Table 6. Appearing between 0 and 3 times – approximately a third of the possible values – we consider that an index term belongs to the so-called HIGH set. Therefore, it is defined in its correspondent fuzzy set with uniformly distributed values between 0.7 and 1. Analogously, we may distribute all values uniformly according to different fuzzy sets. Fuzzy sets are triangular, on one hand for simplicity and on the other hand because we tested other more complex types of sets (Gauss, Pi type, etc) and the results did not improve at all.

Provided that different weights are defined in every hierarchic level, we should consider other scales to calculate them. As for the Level Topic we were considering the immediately top level – the whole set of knowledge; for the Section level we should consider the times that an index term appears in a certain Topic. The list of the most used index terms in a unique Topic is the following:

1. Service: Topic 10, 16 times.
2. Address: Topic 1, 10 times.
+. Library: Topic 6, 10 times.
+. Registration: Topic 3, 10 times.
5. Mail: Topic 1, 9 times.
+. Electronic: Topic 1, 9 times.
7. Virtual: Topic 12, 8 times.
8. Computer: Topic 10, 7 times.
+. Services: Topic 1, 7 times.
10. Education: Topic 1, 5 times.
+. Resources: Topic 12, 5 times.

In the same way, at the level of Topic, term weight has a value between 0 – if it appears many times – and 1 – if it does not appear in any other subset. We again consider that 1% of these words must mark the border for the value 0 - 11 words – so whenever a term appears more than 5 times in other subsets, its weight takes the value 0 at the Section level.

Possible term weights for the level of Section are shown in Table 7. The method is analogous and considers the definition of the fuzzy sets. At the level of Object, term weights are shown in Table 8.

6.3.2. Question 2

To find out the term weight associated with question 2 – Q2. How often does an index term appear in its own subset? – the reasoning is analogous. However, we have to bear in mind that it is necessary to consider the frequency inside a unique set of knowledge, thus the number of appearances of index terms decreases considerably. The list of the most used index terms in a Topic must be considered again. It also must be born in mind that the more an index term appears in a Topic or Section, the higher the value for an index term is. Q2 is senseless at the level of Object. The proposed values are given in Table 9.

Table 7
Term weight values for every Section for Q1.

Times appearing	0	1	2	3	4	5	⩾6
Value	1	0.7	0.6	0.5	0.4	0.3	0

Table 8
Term weight values for every Object for Q1.

Times appearing	0	1	2	⩾3
Value	1	0.7	0.3	0

6.3.3. Question 3

In the case of question 3 – Q3. Does a term define undoubtedly a standard question? – the answer is completely subjective and we propose the answers 'Yes', 'Rather' and 'No'. Term weight values for this question are shown in Table 10.

6.3.4. Question 4

Finally, question 4 – Q4. Is an index term tied to another one? – deals with the number of index terms tied to another one. We propose term weight values for this question in Table 11. Again, the values 0.7 and 0.3 are a consequence of considering the border between fuzzy sets.

After considering all these factors, fuzzy rules for Topic and Section levels are defined in Table 12. These rules cover all the 81 possible combinations. Note that, apart from the three input sets mentioned in previous sections, four output sets have been defined - HIGH, MEDIUM-HIGH, MEDIUM-LOW and LOW. At the level of Object, we must discard question 2 and rules change.

The only aspect which has not been defined yet is multiple appearances in a Topic or Section. For example, it is possible that

Table 9
Term weight values for every Topic and Section for Q2.

Times appearing	0	1	2	3	4	5	⩾6
Value	1	0.7	0.6	0.5	0.4	0.3	0

Table 10
Term weight values for Q3.

Answer to Q3: Does a term define undoubtedly a standard question?	Yes	Rather	No
Value	1	0.5	0

Table 11
Term weight values for Q4.

Number of index terms tied to another index term	0	1	2	⩾3
Value	1	0.7	0.3	0

Table 6
Term weight values for every Topic for Q1.

Times appearing	0	1	2	3	4	5	6	7	8	9	10	11	12	⩾13
Value	1	0.9	0.8	0.7	0.64	0.59	0.53	0.47	0.41	0.36	0.3	0.2	0.1	0

Table 12
Rule definition for Topic and Section levels.

Rule number	Rule definition	Output
R1	IF Q1 = HIGH and Q2 ≠ LOW	At least MEDIUM–HIGH
R2	IF Q1 = MEDIUM and Q2 = HIGH	At least MEDIUM–HIGH
R3	IF Q1 = HIGH and Q2 = LOW	Depends on other Questions
R4	IF Q1 = HIGH and Q2 = LOW	Depends on other Questions
R5	IF Q3 = HIGH	At least MEDIUM–HIGH
R6	IF Q4 = LOW	Descends a level
R7	IF Q4 = MEDIUM	If the Output is MEDIUM–LOW, it descends to LOW
R8	IF (R1 and R2) or (R1 and R5) or (R2 and R5)	HIGH
R9	In any other case	MEDIUM–LOW

the answer to question 3 is 'Rather' in one case 'No' in another one. In this case, a weighted average of the corresponding term weights is calculated.

An example of all the processes is shown below

Example.

Object 12.6.2 is defined by the following standard question:

Which services can I access as a virtual user at the University of Seville?

If we consider the term 'virtual':

- At Topic level:
- 'Virtual' appears twice in other Topics in the whole set of knowledge, so that the value associated with Q1 is 0.80.
- 'Virtual' appears 8 times in Topic 12, so that the value associated with Q2 is 1.
- The response to Q3 is 'Rather' in 5 of the 8 times and 'No' in the other three, so that the value associated with Q3 is a weighted average: $(5*0.5 + 3*0)/8 = 0.375$.
- Term 'virtual' is tied to one term 7 times and it is tied to two terms once. Therefore, the average is 1.14 terms. A linear extrapolation leads to a value associated with Q4 of 0.65.
- With all the values as inputs for the Fuzzy Logic engine, we obtain a term weight of 0.53.
- At Section level:
- 'Virtual' appears 5 times in other Sections corresponding to Topic 12, so that the value associated with Q1 is 0.30.
- 'Virtual' appears 3 times in Topic 12, so that the value associated with Q2 is 0.45.
- The response to Q3 is 'Rather' in all cases, so that the value associated with Q3 is 0.5.
- Term 'virtual' is tied to term 'user' so that the value associated with Q4 is 0.7.
- With all the values as inputs for the Fuzzy Logic engine, we obtain a term weight of 0.45.
- At Object level:
- 'Virtual' appears twice in other Objects corresponding to Section 12.6, so that the value associated with Q1 is 0.30.
- The response to Q3 is 'Rather', so that the value associated to Q3 is 0.5.
- Term 'virtual' is tied to term 'user' so that the value associated with Q4 is 0.7.
- With all the values as inputs for the Fuzzy Logic engine, we obtain a term weight of 0.52. We can see the difference with the corresponding term weight obtained with the TF-IDF method, but it is exactly what we are looking for: not only the desired object but the most closely related to it must be retrieved.

To compare results, we considered the position in which the correct answer appeared among the retrieved answers, according to fuzzy engine outputs. For it, the first necessary step to follow is to define the overcoming thresholds for the fuzzy engine. This way, Topics and Sections that are not related with the Object to identify are eliminated. We also have to define low enough thresholds, in order to be able to obtain related Objects also. We suggest presenting between 1 and 5 answers, depending on the number of related Objects. As explained in previous sections, term weights are lower for TF-IDF method, due to normalization. For this reason, thresholds were fixed to 0.2 to overcome the level of Topic and 0.3 to overcome the level of Section for the method TF–IDF. Meanwhile, both thresholds have a value of 0.4 for the FL-based method.

Table 13
Comparison between TF-IDF classic method and the novel FL-based method.

	Cat1	Cat2	Cat3	Cat4	Cat5	Total
TF-IDF Method	466 (50.98%)	223 (24.40%)	53 (5.80%)	79 (8.64%)	93 (10.18%)	914
FL Method	710 (77.68%)	108 (11.82%)	27 (2.95%)	28 (3.06%)	41 (4.49%)	914

The results of the consultation were sorted in 5 categories:

- Category Cat1: 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.
- Category Cat2: The correct answer is retrieved between the 3 with a higher degree of certainty -excluding the previous case.
- Category Cat3: The correct answer is retrieved among the 5 with a higher degree of certainty - excluding the previous cases.
- Category Cat4: The correct answer is retrieved, but not among the 5 with a higher degree of certainty.
- Category Cat5: The correct answer is not retrieved by the system.

The ideal situation comes when the desired Object is retrieved as Cat1, though Cat2 and Cat3 would be reasonably acceptable. The results obtained in the tests are shown in Table 13. Though the obtained results with the TF-IDF method are quite reasonable, 81.18% of the objects being retrieved among the first 5 options - and more than as Cat1, the FL 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.

More detailed tests were made, according to the type of standard questions and the number of standard questions defined for every Object. We came to the conclusion that the more intricate, disordered and confused the information is, the better the FL TW method is, compared with the classic TF-IDF one. This makes its application ideal for the case of an intelligent agent for a web portal, where the information has these features and users may carry out inaccurate or disoriented consultations.

7. Conclusions

In this paper, we present a novel general method for IE and IR by means of the use of an intelligent agent based on FL. Given the lack of flexibility of most of intelligent agents when information is abundant, confused, vague or heterogeneous, we propose an IE method based on the VSM and FL. A set of knowledge is divided in different hierarchic levels up to a level where the instances or Objects are extracted. A series of standard questions are assigned to every Object, based on the possible consultations from a user in Natural Language. These questions drive to the extraction of index terms.

Index terms have associated term weights, according to their importance in their correspondent subset of knowledge. Given

Fig. 9. Prototype of the intelligent agent developed for University of Seville.

the need to automate TW we propose a novel TW method based on the use FL. This method replaces the classic method, the so-called TF-IDF method.

This method has been applied in development of the University of Seville intelligent agent, which is to be functioning soon. An image of the prototype is shown in Fig. 9.

We also propose some future lines of investigation. First of all, the study of the ontology based instead of the vectorial approach. The fact that there is difficulty in using ontologies does not mean that we should not consider this quite an interesting field of investigation. Secondly, CI techniques, other than FL, can be applied to build intelligent agents. Specifically, neuro-fuzzy techniques are a very interesting possibility, as they combine the human reasoning of FL with the neural connection based structure of the ANN, taking advantage of both techniques.

References

Abulaish, M., & Dey, L. (2005). Biological ontology enhancement with fuzzy relations: A text-mining framework. In *Proceedings of the 2005 IEEE/WIC/ACM international conference on web intelligence, France* (pp. 379–385).

Ajayi, A. O., Aderounmu, G. A., & Soriyan, H. A. (2009). An adaptive fuzzy Information Retrieval model to improve response time perceived by e-commerce clients. *Expert Systems with Applications (ESWA), 37*(1), 82–91.

Arano, S. (2003). La ontología: una zona de interacción entre la Lingüística y la Documentación. *Hipertext.net*, No. 2.

Aronson, A. R., & Rindflesch, T. C. (1997). Query expansion using the UMLS Metathesaurus. In *Proceedings of the 1997 AMIA Annual Fall Symposium* (pp. 485–489).

Aronson, A. R., Rindflesch, T. C., & Browne, A. C. (1994). Exploiting a large thesaurus for information retrieval. In *Proceedings of RIAO* (pp. 197–216).

Berners-Lee, T., & Miller, E (2002). The Semantic Web lifts off. *ERCIM News No. 51. Special Semantic Web.*

Bickmore, T. W., Pfeifer, L. M., & Paasche-Orlow, M. K. (2009). Using computer agents to explain medical documents to patients with low health literacy. *Patient Education and Counseling, 75*(3), 315–320.

Chakrabarti, S. (2000). Data Mining for hypertext: A tutorial survey. *ACM SIGKDD Explorations, Newsletter of the Special Interest Group on Knowledge Discovery and Data Mining..*

Cordon, O., de Moya, F., & Zarco, C. (2004). Fuzzy logic and multiobjective evolutionary algorithms as soft computing tools for persistent query learning in text retrieval environments. In *Proceedings of the IEEE international conference on fuzzy systems* (Vol. 1, pp. 571–576).

Eisman, E. M., Lopez, V., & Castro, J. L. (2009). Controlling the emotional state of an embodied conversationalagent with a dynamic probabilistic fuzzy rules based system. *Expert Systems with Applications, 36*(6), 9698–9708.

Etzioni, O. (1996). The World Wide Web: Quagmire or gold mine. *Communicational ACM, 39*(11), 65–68.

Friedman, M., Last, M., Zaafrany O., Schneider, M., & Kandel, A. (2004). A new approach for fuzzy clustering of web documents. In *Proceedings of the IEEE international conference on fuzzy systems* (Vol. 1, pp. 377–381).

Garcia-Serrano, A., Martinez, P., & Hernandez, J. (2004). Using AI techniques to support advanced interaction capabilities in a virtual assistant for e-commerce". *Expert Systems with Applications, 26*(3), 413–426.

Gómez, A., Ropero, J., León, C., & Carrasco, A. (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, AIDSS* (pp. 496–499).

Greenes, R. A. (2006). *Clinical decision making - The road ahead*. Elsevier.

Haase, V. H., Steinmann, C., & Vejda, S. (2002). Access to knowledge: better use of the internet. In *IS2002 Proceedings of the informing science + IT education conference, Cork, Ireland* (pp. 618–627).

Horng, Y. J., Chen, S. M., Chang, Y. C., & Lee, C. H. (2005). A new method for fuzzy information retrieval based on fuzzy hierarchical clustering and fuzzy inference techniques. *IEEE T. Fuzzy Systems, 2*, 216–228. 13.

Iannone, L., Palmisano, I., & Fanizzi, N. (2007). An algorithm based on counterfactuals for concept learning in the semantic web. *Applied Intelligence, 26*(2), 139–159, ISSN 0924-669X 2007, Springer.

Kerly, A., Ellis, R., & Bull, S. (2007). CALMsystem: A Conversational Agent for Learner Modelling. *Knowledge-Based Systems, 21*(3), 238–246.

Klogsen, W., & Zytkow, J. (2002). *Handbook of data mining and knowledge discovery*. New York: Oxford University Press.

Kosala, R., & Blockeel, H. (2000). Web mining research: A survey. , *SIGKDD: SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining.* ACM (Vol. 2).

Kushmerick, N. (2002). Gleaning answers from the web. In *Proceedings of the AAAI spring symposium on mining answers from texts and knowledge bases, Palo Alto* (pp. 43–45).

Kwok, K. L. (1989). A neural network for probabilistic information retrieval. In *Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval, Cambridge, Massachusetts, United States*

Larsen, H. L. (1999). An approach to flexible information access systems using soft computing. In *Proceedings of the 32nd annual Hawaii international conference on system sciences, HICCS99* (pp. 5–8).

Larsen, H., & Yager, R. (1993). The use of fuzzy relational thesauri for classificatory problem solving in information retrieval and expert systems. In *Proceedings of the IEEE transactions on systems, man, and cybernetics* (Vol. 23, pp. 31–41).

Lertnattee, V, & Theeramunkong, T. (2003). Combining homogenous classifiers for centroid-based text classification. In *Proceedings of the 7th international symposium on computers and communications* (pp. 1034–1039).

Lee, D. L., Chuang, H., & Seamons, K. (1997). Document ranking and the vector-space model. In *IEEE Software* (Vol. 14, pp. 67-75).

Liao, S. H. (2005). Expert system methodologies and applications—a decade review from 1995 to 2004. *Expert Systems with Applications, 28*(1), 93–103.

Liu, D. R., & Ke, C. K. (2007). Knowledge support for problem-solving in a production process: A hybrid of knowledge discovery and case-based reasoning. *Expert Systems with Applications., 33*(1), 147–161.

Liu, L., Buttler, D., Caverlee, J., Pu, C., & Zhang, J. (2005). A methodical approach to extracting interesting objects from dynamic web pages. *International Journal of Web and Grid Services, 1*(2), 165–195.

Liu, S., Dong, M., Zhang, H., Li, R., & Shi, Z. (2001). An approach of multi-hierarchy text classification. In *Proceedings of the international conferences on info-tech and info-net, Beijing* (Vol. 3, pp. 95–100).

Loh, S., Palazzo, J., De Oliveira, M., & Gameiro, M. (2003). Knowledge discovery in texts for constructing decision support systems. *Applied Intelligence, New York, NY, USA, 18*(3), 357–366.

Lu, M., Hu, K., Wu, Y., Lu, Y., & Zhou, L. (2002). SECTCS: towards improving VSM and Naive Bayesian classifier. *IEEE International Conference on Systems, Man and Cybernetics, 5*, 5.

Martin, A., & Leon, C. (2009). Intelligent retrieval in a digital library using semantic web. *IADAT Journal of Advanced Technology on Educaction, 3*(3), 427–429.

Martin, A., & Leon, C. (2010). Expert knowledge management based on ontology in a digital library. In *ICEIS 2010 12th international conference on enterprise information systems, Madeira (Portugal)* (pp. 291–298).

Mengual, L., Barcia, N., Bobadilla, J. Jimenez, R., Setien, J., & Yaguez, J. (2001). Arquitectura multi-agente segura basada en un sistema de implementación

automática de protocolos de seguridad. I Simposio Español de Negocio Electrónico.
Mercier, A., & Beigbeder, M. (2005). Fuzzy proximity ranking with Boolean queries. In *Proceedings of the 14th text retrieval conference (TREC), Gaithersburg, Maryland, USA*.
Moradi, P., Ebrahim, M., & Ebadzadeh, M. M. (2008). Personalizing results of information retrieval systems using extended fuzzy concept networks. In *3rd International conference on information and communication technologies: From theory to applications, ICTTA* (pp. 1–7).
Olson, D., & Shi, Y. (2007). *Introduction to business data mining*. McGraw-Hill.
Pal, S. K., Talwar, V., & Mitra, P. (2002). Web Mining in Soft Computing Framework: Relevance, State of the Art and Future Directions. *IEEE Transactions on Neural Networks, 13*(5), 1163–1177.
Papadakis, N. K., Skoutas, D., Raftopoulos, K., & Varvarigou, T. A. (2005). STAVIES: A System for Information Extraction from Unknown Web Data Sources through Automatic Web Wrapper Generation Using Clustering Techniques. *IEEE Transactions on Knowledge and Data Engineering, 17*(12), 1638–1652.
Pierre, S. (2002). *Intelligent and Heuristic Approaches and Tools for the Topological Design of Data Communication Networks. Data Communication Network Techniques and Applications*. New York: Academic Press, pp. 289–326.
Quan, T. T., Hui, S. C., & Fong, A. C. M. (2006). Automatic fuzzy ontology generation for semantic help-desk support. *Industrial Informatics, IEEE Transactions on, 2*(3), 155–164.
Raghavan, V. V., & Wong, S. K. (1986). A critical analysis of Vector Space Model for information retrieval. *Journal of the American Society for Information Science, 37*(5), 279–287.
Ríos, S. A., Velásquez, J. D., Yasuda, H., & Aoki, T. (2006). Improving the web site text content by extracting concept-based knowledge. *Lecture Notes in Artificial Intelligence, 1*, 371–378. 4252.
Romero, C., & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications., 33*(1), 135–146.
Ropero, J., Gómez, A., León, C., & Carrasco, A. (2007). *Information Extraction in a set of knowledge using a Fuzzy Logic based intelligent agent. Lecture Notes in Computer Science* (Vol. 47). LNCS, part 3, pp. 811–820.
Ropero, J., Gomez, A., Leon, C., & Carrasco, A (2009). Term weighting: Novel fuzzy logic based method Vs. classical TF-IDF method for Web information extraction. *Proceedings of the 11th international conference on enterprise information systems* (pp. 130-137).
Ruiz, M., & Srinivasan, P. (1998). Automatic text categorization using neural networks . In E. Efthimiadis (Ed.), *Advances in classification research, vol. 8: Proceedings of the 8th ASIS SIG/CR classification research workshop*. New Jersey: Information Today, Medford (pp. 59–72).
Salton, G. (1988). *Automatic Text Processing*. Addison-Wesley Publishing Company.
Salton, G., & Buckley, C. (1996). Term weighting approaches in automatic text retrieval. *Technical report TR87-881, Department of Computer Science, Cornell University, 1987. Information Processing and Management, 32*(4), 431–443.
Subasic, P., & Huettner, A. (2001). Affect analysis of text using fuzzy semantic typing. *IEEE Transactions on Fuzzy Systems* [Special issue].
Tao, Y. H., Hong, T. P., & Su, Y. M. (2008). Web usage mining with intentional browsing data. *Expert Systems with Applications., 34*(3), 1893–1904.
Turban, E., & Aronson, J. E. (2001). *Decision support systems and intelligent systems* (6th ed.). Hong Kong: Prentice Internacional Hall.
Vercellis, C. (2009). *Business Intelligence: Data Mining and Optimization for Decision Making*. Wiley Publishing.
Webometrics. (2009). http://www.webometrics.info/index.html.
Wik, P., & Hjalmarsson, A. (2009). Embodied conversational agents in computer assisted language learning. *Speech Communication, 51*(10), 1024–1037.
Zadeh, L. A. (1994). Fuzzy logic, neural networks and soft computing. *Communications of the ACM, 3*(3), 77–84.
Zhai, J, Wang, Q., & Lv, M. (2008). Application of fuzzy ontology framework to information retrieval for SCM. In *Proceedings of ISIP08, International symposiums on information processing* (pp. 173–177).
Zhang, R., & Zhang, Z. (2003). Addressing CBIR efficiency, effectiveness, and retrieval subjectivity simultaneously. In *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval, MIR*. New York, NY, USA: ACM Press (pp. 71–78).
Zhao, Y., & Karypis, G. (2002). Improving precategorized collection retrieval by using supervised term weighting schemes. In *Proceedings of the international conference on information technology: coding and computing* (pp. 16–21).