Techniques to Improve Multi-Agent Systems for Searching and Mining the Web

E. Herrera-Viedma[1], C. Porcel[1], F. Herrera[1], L. Martínez[2], and A.G. Lopez-Herrera[1]

[1] Dept. of Computer Science and Artificial Intelligence.
University of Granada, 18071 - Granada, Spain
viedma@decsai.ugr.es,cporcel@invest.ugr.es,herrera@decsai.ugr.es, agabriel@ugr.es

[2] Dept. of Computer Science.
University of Jaén, 23071 - Jaén, Spain
martin@ujaen.es

Summary. Nowadays, an abundant amount of information is created and delivered over electronic media. The information gathering in Internet is a complex activity and Internet users need tools to assist them to find the information required. Web multi-agent systems assist the users by gathering from Internet the information that best satisfies their specific needs.

In this paper, we analyze some techniques that applied together could provide major advances in the design of these Web multi-agent systems in order to improve their performance: i) information filtering tools and ii) the fuzzy linguistic modelling. Then, we present a model of a fuzzy linguistic multi-agent system for searching and mining the Web that is designed using some filtering tools and a particular fuzzy linguistic modelling, called multi-granular fuzzy linguistic modelling, which is useful when we have different label sets to assess the information.

Keywords: Web, fuzzy linguistic modelling, information filtering, information retrieval, intelligent agents.

1 Introduction

The exponential increase of Web sites and documents is contributing to that Internet users not being able to find the information they seek in a simple and timely manner. Users are in need of tools to help them cope with the large amount of information available on the Web [22, 23]. Therefore, techniques for searching and mining the Web are becoming increasing vital.

A multi-agent system is one in which a number of agents cooperates and interacts with each other in a distributed environment. On the Web the activity of a multi-agent system consists in to assist Internet users in information gathering processes by means of distributed intelligent agents in order to find the fittest information to their information needs. In a typical multi-agent system, the agents work together to achieve a global objective based on distributed data and control. Multi-agent systems have been widely used in Web applications [4, 24, 25].

In this paper we study two techniques that applied together can contribute to achieve major advances in the design of Web multi-agent system in order to improve their performance:

- *Information Filtering Tools:* A promising direction to improve the information access on the Web concerns the way in which it is possible to filter the great amount of information available across the Web. Information filtering is a name used to describe a variety of processes involving the delivery of information to people who need it. Operating in textual domains, *filtering systems* or *recommender systems* evaluate and filter the great amount of information available on the Web to assist people in their search processes [28].
- *Fuzzy Linguistic Modelling:* The great variety of representations and evaluations of the information in Internet is the main obstacle to the communication among the agents and between agents and user from what is very important the design of appropriate communication protocol. The problem becomes more noticeable when users take part in the process. This reveals the need of more flexibility in the communication among agents and between agents and users. To solve this problem we propose the use of *fuzzy linguistic modelling* [13, 14, 29] to represent and handle flexible information by means of linguistic labels.

Firstly, we revise the main aspects and models of information filtering tools, as for example, the content-based filtering tools and the collaborative filtering tools. Then, we revise different approaches of fuzzy linguistic modelling to represent the information in the information gathering process of a Web multi-agent system, as for example, the ordinal fuzzy linguistic modelling [13, 9], the 2-tuple fuzzy linguistic modelling [14, 16], the multi-granular fuzzy linguistic modelling [12, 15] and the unbalanced fuzzy linguistic modelling [10, 11]. And finally, we present a model of fuzzy linguistic multi-agent system which is designed using both information filtering tools and a multi-granular fuzzy linguistic modelling.

The paper is structured as follows. Section 2 revises the information filtering techniques. Section 3 analyzes different approaches of fuzzy linguistic modelling. Section 4 presents the new model of fuzzy linguistic multi-agent system for gathering information on the Web. Finally, some concluding remarks are pointed out.

2 Information Filtering Tools

Information gathering in Internet is a complex activity. Find the appropriate information, required for the users, on the World Wide Web is not a simple task. This problem is more acute with the ever increasing use of the Internet. For example, users who subscribe to internet lists waste a great deal of time reading, viewing or deleting irrelevant e-mail messages. To improve the information access on the Web the users need tools to filter the great amount of information available across the Web. *Information Filtering* (IF) is a name used used to describe a variety of processes involving the delivery of information to people who need it. It is a research area that offer tools for discriminating between relevant and irrelevant information by providing personalized assistance for continuous retrieval of information.

IF systems are characterized by [8]:

- are applicable for unstructured or semi-structured data (e.g. web documents, e-mail messages),
- are based on user profiles,
- handle large amounts of data,
- deal primarily with textual data and
- their objective is to remove irrelevant data from incoming streams of data items.

We can find some of the above features in Information Retrieval (IR) systems, but IF differs from traditional IR in that the users have long information needs that are described by means of user profiles, rather than ad-hoc needs that are expressed as queries posed to some IR system. Traditionally IR develops storage, indexing and retrieval technology for textual documents. A user describes his information need in the form of a query to the IR system and the system attempts to find items that match the query within a document store. The information need is usually very dynamic and temporary, i.e., a user issue a query describing an immediate need. Furthermore, information retrieval systems tend to maintain a relatively static store of information. Unlike IR systems, IF systems generally operate on continuous information streams, and always maintain a profile of the user interests needs throughout many uses of the system. As a result, IF systems tend to filter information based on more long-term interests.

Traditionally, these IF systems or recommender systems have fallen into two main categories [26]. *Content-based filtering systems* filter and recommend the information by matching user query terms with the index terms used in the representation of documents, ignoring data from other users. These recommender systems tend to fail when little is known about user information needs, e.g. as happens when the query language is poor. *Collaborative filtering systems* use explicit or implicit preferences from many users to filter and recommend documents to a given user, ignoring the representation of documents. These recommender systems tend to fail when little is known about a user, or when he/she has uncommon interests [26]. In these kind of systems, the users' information preferences can be used to define user profile that are applied as filters to streams of documents; the recommendations to a user are based on another users' recommendations with similar profiles. Many researchers think that the construction of accurate profiles is a key task and the system's success will depend to a large extent on the ability of the learned profiles to represent the user's preferences [27]. Several researchers are exploring hybrid content-based and collaborative recommender systems to smooth out the disadvantages of each one of them [1, 2, 7, 26].

2.1 Approaches in the Design of IF Systems

In this section we present two major approaches followed in the design and implementation of IF systems, that is, the *statistical approach* and *the knowledge based approach* [8].

2.1.1. Statistical Approach

This kind of IF systems represents the user profiles as weighted vector of index terms. To filter the information the system implements a statistical algorithm that computes the similarity of a vector of terms that represents the data item being filtered to a user's profile. The most common algorithm used is the Correlation or the Cosine measure between the user's profile and the document's vector.

The filtering activity is followed by a relevance feedback phase. Relevance feedback is a cyclic process whereby the user feeds back into the system decisions on the relevance of retrieved documents and the system then uses these evaluations to automatically update the user profile.

2.1.2 Knowledge Based Approach

IF systems that follow the knowledge based approach utilize Artificial Intelligence techniques, such as production rules, neural networks and evolutionary genetic algorithm, to represent user profiles and to implement the filtering and the learning (feedback) phases.

- **Rule based IF systems:** These IF systems use rules to represent user profiles, where each rule can represent a user information need or pattern of information filtering. For example, in e-mail messages, rules can be defined and applied to fields that appear in the message header (e.g. subject or sender). The rules may contain instructions on how to handle a message, depending on the values of these fields. The rules allow us either to filter out the data item or to treat it as relevant. For example, if the sender of an e-mail does not appear in a certain predefined list, the message gets a low relevance rank.

- **Neural networks based IF systems:** A neural network is an interconnected assembly of simple processing elements, units or nodes, whose functionality is roughly based on the animal neuron. The processing ability of the network is stored in inter-unit connection weights, obtained by a process of adaptation to, or learning from, a set of training patterns. The weights are supposed to adapt when the net is shown examples from training sets. Neural networks can also be applied in IF systems, where a user profile is representing a user's concept with unseen associations, that adapts from training.
- **Evolutionary genetic algorithms based IF systems:** Evolutionary genetic algorithm based techniques borrow their model from the Darwinian concept of the natural process of survival. Nature selects the most fit individuals to survive, and genetic patterns are passed by the individuals down through generations. The changes take place by recombining the genetic codes of pairs of individuals. These features allow us to apply an evolutionary and genetic approach in IF systems. The analogy in information filtering makes use of the vector space model to represent documents. In this model, a gene would be represented as a term, an individual as a document in the vector space, and the community as a profile. An appropriate objective function is introduced as the survival process, to decide whether to update the profile.

2.2 On the Acquisition of User Data

Another topic that we must have in mind when we design a IF system is the method to gather user information. In order to discriminate between relevant and irrelevant information for a user, we must have some information about this user, i.e. we must know the user preferences. Information about user preferences can be obtained in two different ways [8], *implicit* and *explicit mode*, although these ways not be mutually exclusive.

The implicit approach is implemented by inference from some kind of observation. The observation is applied to user behavior or to detecting a user's environment (such as bookmarks or visited URL). The user preferences are updated by detecting changes while observing the user.

The other approach, the *explicit* approach, interact with the users by acquiring feedback on information that is filtered, that is, the users express some specifications of what they desire. This approach is the most common.

3 Fuzzy Linguistic Modelling

There are situations in which the information cannot be assessed precisely in a quantitative form but may be in a qualitative one. For example, when attempting to qualify phenomena related to human perception, we are often led to use words in natural language instead of numerical values, e.g. when evaluating the *comfort* or *design* of a car, terms like *good*, *medium* or *bad* can be used. In other cases, precise quantitative information cannot be stated because either it is unavailable or the cost for its computation is too high and an "approximate value" can be applicable, eg. when evaluating the speed of a car, linguistic terms like *fast*, *very fast* or *slow* can be used instead of numeric values.

The use of Fuzzy Sets Theory has given very good results for modelling qualitative information [29]. The *fuzzy linguistic modelling* is a tool based on the concept of *linguistic variable* [29] to deal with qualitative assessments in the problems. It has proven its useful in many problems, e.g., in decision making [13], quality evaluation [21], models of information retrieval [17, 18], etc.

In this section, we revise four different approaches of the fuzzy linguistic modelling which can provide a different support to represent the linguistic information managed in the communication processes developed by the multi-agent systems:

1. *Ordinal fuzzy linguistic modelling* [13, 9], which is defined to eliminate the excessive complexity of the traditional fuzzy linguistic modelling [29].
2. *2-tuple fuzzy linguistic modelling* [14, 16], which is is defined to improve the performance of the ordinal fuzzy linguistic approach.
3. *Multi-granular fuzzy linguistic modelling* [12, 15], which is defined to deal with situations in which the linguistic information is assessed on different label sets.
4. *Unbalanced fuzzy linguistic modelling* [10, 11], which is defined to deal with situations in which the linguistic information is assessed on an unbalanced label set, that is, a non-symmetrical and non-uniform label set.

3.1 The Ordinal Fuzzy Linguistic Modelling

The *ordinal fuzzy linguistic modelling* [13, 9] is a very useful kind of fuzzy linguistic approach proposed as an alternative tool to the traditional fuzzy linguistic modelling [29] which simplifies the computing with words process as well as linguistic aspects of problems. It is defined by considering a finite and totally ordered label set $S = \{s_i\}, i \in \{0, \ldots, g\}$ in the usual sense, i.e., $s_i \geq s_j$ if $i \geq j$, and with odd cardinality (7 or 9 labels). The mid term represents an assessment of "approximately 0.5", and the rest of the terms being placed symmetrically around it. The semantics of the label set is established from the ordered structure of the label set by considering that each label for the pair $(s_i, s_{\}-i})$ is equally informative. For example, we can use the following set of seven labels to represent the linguistic information:

$$S = \{s_0 = N,\ s_1 = VL,\ S_2 = L,\ s_3 = M,\ s_4 = H,\ s_5 = VH,\ s_6 = P\}.$$

Additionally, a fuzzy number defined in the $[0, 1]$ interval can be associated with each linguistic term. A way to characterize a fuzzy number is to use a representation based on parameters of its membership function. The linguistic assessments given by the users are just approximate ones, some authors consider that linear trapezoidal membership functions are good enough to capture the vagueness of such linguistic assessments. The parametric representation is achieved by the 4-tuple (a, b, c, d), where b and d indicate the interval in which the membership value is 1, with a and c indicating the left and right limits of the definition domain of the trapezoidal membership function. A particular case of this type of representation are the linguistic assessments whose membership functions are triangular, i.e., $b = d$, then we represent this type of membership functions by a 3-tuple (a, b, c). An example may be the following set of seven terms (Figure 1):

$s_0 = Null(N) = (0, 0, .17)$ $\quad s_1 = VeryLow(VL) = (0, .17, .33)$
$s_2 = Low(L) = (.17, .33, .5)$ $\quad s_3 = Medium(M) = (.33, .5, .67)$
$s_4 = High(H) = (.5, .67, .83)$ $\quad s_5 = VeryHigh(VH) = (.67, .83, 1)$
$s_6 = Perfect(P) = (.83, 1, 1)$.

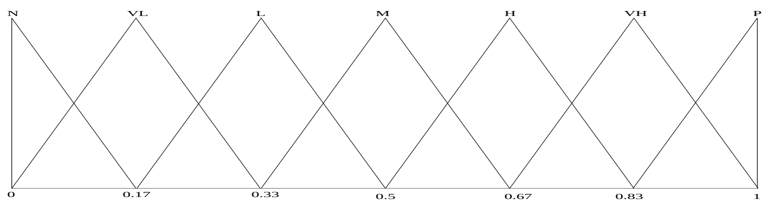

Fig. 1. A set of seven linguistic terms with its semantics

In any linguistic modelling we need management operators of linguistic information. An advantage of the ordinal fuzzy linguistic modelling is the simplicity and quickness of its computational model. It is based on the symbolic computation [13, 9] and acts by direct computation on labels by taking into account the order of such linguistic assessments in the ordered structure of labels. Usually, the ordinal fuzzy linguistic model for computing with words is defined by establishing i) a negation operator, ii) comparison operators based on the ordered structure of linguistic terms, and iii) adequate aggregation operators of ordinal fuzzy linguistic information. In most ordinal fuzzy linguistic approaches the negation operator is defined from the semantics associated to the linguistic terms as

$$NEG(s_i) = s_j \mid j = g - i;$$

and there are defined two comparison operators of linguistic terms:

1. *Maximization operator*: $MAX(s_i, s_j) = s_i$ if $s_i \geq s_j$; and
2. *Minimization operator*, $MIN(s_i, s_j) = s_i$ if $s_i \leq s_j$.

Using these operators it is possible to define automatic and symbolic aggregation operators of linguistic information, as for example the LOWA operator [13] and the LWA operator [9].

We must point out that in [5] we apply satisfactorily the ordinal fuzzy linguistic modelling to model the communication processes in the design of a Web multi-agent system.

3.2 The 2-Tuple Fuzzy Linguistic Modelling

The *2-tuple fuzzy linguistic modelling* [14, 16] is a kind of fuzzy linguistic modelling that mainly allows to reduce the loss of information typical of the ordinal fuzzy linguistic modelling. Its main advantage is that the linguistic computational model based on linguistic 2-tuples can carry out processes of computing with words easier and without loss of information. To define it we have to establish the 2-tuple representation model and the 2-tuple computational model to represent and aggregate the linguistic information, respectively.

Let $S = \{s_0, ..., s_g\}$ be a linguistic term set with odd cardinality ($g + 1$ is the cardinality of S), where the mid term represents an assessment of approximately 0.5 and with the rest of the terms being placed symmetrically around it. We assume that the semantics of labels is given by means of triangular membership functions represented by a 3-tuple (a, b, c) and consider all terms distributed on a scale on which a total order is defined $s_i \leq s_j \iff i \leq j$. In this fuzzy linguistic context, if a symbolic method [13, 9] aggregating linguistic information obtains a value $\beta \in [0, g]$, and $\beta \notin \{0, ..., g\}$, then an approximation function is used to express the result in S.

Definition 1. [14] *Let β be the result of an aggregation of the indexes of a set of labels assessed in a linguistic term set S, i.e., the result of a symbolic aggregation operation, $\beta \in [0, g]$. Let $i = round(\beta)$ and $\alpha = \beta - i$ be two values, such that, $i \in [0, g]$ and $\alpha \in [-.5, .5)$ then α is called a Symbolic Translation.*

The 2-tuple fuzzy linguistic approach is developed from the concept of symbolic translation by representing the linguistic information by means of 2-tuples (s_i, α_i), $s_i \in S$ and $\alpha_i \in [-.5, .5)$:

- s_i represents the linguistic label of the information, and
- α_i is a numerical value expressing the value of the translation from the original result β to the closest index label, i, in the linguistic term set ($s_i \in S$).

This model defines a set of transformation functions between numeric values and 2-tuples.

Definition 2. [14] *Let $S = \{s_0, ..., s_g\}$ be a linguistic term set and $\beta \in [0, g]$ a value representing the result of a symbolic aggregation operation, then the 2-tuple that expresses the equivalent information to β is obtained with the following function:*

$$\Delta : [0, g] \longrightarrow S \times [-0.5, 0.5)$$

$$\Delta(\beta) = (s_i, \alpha), \; with \; \begin{cases} s_i & i = round(\beta) \\ \alpha = \beta - i & \alpha \in [-.5, .5) \end{cases}$$

where $round(\cdot)$ is the usual round *operation, s_i has the closest index label to "β" and "α" is the value of the symbolic translation.*

For all Δ there exists Δ^{-1}, defined as $\Delta^{-1}(s_i, \alpha) = i + \alpha$. On the other hand, it is obvious that the conversion of a linguistic term into a linguistic 2-tuple consists of adding a symbolic translation value of 0: $s_i \in S \Longrightarrow (s_i, 0)$.

The 2-tuple linguistic computational model is defined by presenting the comparison of 2-tuples, a negation operator and aggregation operators of 2-tuples.

1. Comparison of 2-tuples. The comparison of linguistic information represented by 2-tuples is carried out according to an ordinary lexicographic order. Let (s_k, α_1) and (s_l, α_2) be two 2-tuples, with each one representing a counting of information:

- If $k < l$ then (s_k, α_1) is smaller than (s_l, α_2).
- If $k = l$ then
  1. if $\alpha_1 = \alpha_2$ then (s_k, α_1) and (s_l, α_2) represent the same information,
  2. if $\alpha_1 < \alpha_2$ then (s_k, α_1) is smaller than (s_l, α_2),
  3. if $\alpha_1 > \alpha_2$ then (s_k, α_1) is bigger than (s_l, α_2).

2. Negation operator of 2-tuples: $Neg((s_i, \alpha)) = \Delta(g - (\Delta^{-1}(s_i, \alpha)))$.

3. Aggregation operators of 2-tuples. The aggregation of information consists of obtaining a value that summarizes a set of values, therefore, the result of the aggregation of a set of 2-tuples must be a 2-tuple. In the literature we can find many aggregation operators which allow us to combine the information according to different criteria. Using functions Δ and Δ^{-1} that transform without loss of information numerical values into linguistic 2-tuples and viceversa, any of the existing aggregation operator can be easily extended for dealing with linguistic 2-tuples. Some examples are:

Definition 3. (Arithmetic Mean). *Let $x = \{(r_1, \alpha_1), \ldots, (r_n, \alpha_n)\}$ be a set of linguistic 2-tuples, the 2-tuple arithmetic mean $\overline{x}^e$ is computed as,*

$$\overline{x}^e[(r_1, \alpha_1), \ldots, (r_n, \alpha_n)] = \Delta(\sum_{i=1}^{n} \frac{1}{n} \Delta^{-1}(r_i, \alpha_i)) = \Delta(\frac{1}{n} \sum_{i=1}^{n} \beta_i).$$

Definition 4. (Weighted Average Operator). *Let $x = \{(r_1, \alpha_1), \ldots, (r_n, \alpha_n)\}$ be a set of linguistic 2-tuples and $W = \{w_1, ..., w_n\}$ be their associated weights. The 2-tuple weighted average $\overline{x}^w$ is:*

$$\overline{x}^w[(r_1, \alpha_1), \ldots, (r_n, \alpha_n)] = \Delta(\frac{\sum_{i=1}^{n} \Delta^{-1}(r_i, \alpha_i) \cdot w_i}{\sum_{i=1}^{n} w_i}) = \Delta(\frac{\sum_{i=1}^{n} \beta_i \cdot w_i}{\sum_{i=1}^{n} w_i}).$$

Definition 5. (Linguistic Weighted Average Operator). *Let* $x = \{(r_1, \alpha_1), \ldots, (r_n, \alpha_n)\}$ *be a set of linguistic 2-tuples and* $W = \{(w_1, \alpha_1^w), ..., (w_n, \alpha_n^w)\}$ *be their linguistic 2-tuple associated weights. The 2-tuple linguistic weighted average* $\overline{x}_l^w$ *is:*

$$\overline{x}_l^w[((r_1, \alpha_1), (w_1, \alpha_1^w))...((r_n, \alpha_n), (w_n, \alpha_n^w))] = \Delta(\frac{\sum_{i=1}^n \beta_i \cdot \beta_{W_i}}{\sum_{i=1}^n \beta_{W_i}}),$$

with $\beta_i = \Delta^{-1}(r_i, \alpha_i)$ *and* $\beta_{W_i} = \Delta^{-1}(w_i, \alpha_i^w)$.

We must point out that in [6] we apply the 2-tuple fuzzy linguistic modelling in the design of a Web multi-agent system as a way to overcome the problems of loss of information observed in the Web multi-agent system presented in [5].

3.3 The Multi-Granular Fuzzy Linguistic Modelling

In any fuzzy linguistic approach, an important parameter to determinate is the "granularity of uncertainty", i.e., the cardinality of the linguistic term set S used to express the linguistic information. According to the uncertainty degree that an expert qualifying a phenomenon has on it, the linguistic term set chosen to provide his knowledge will have more or less terms. When different experts have different uncertainty degrees on the phenomenon, then several linguistic term sets with a different granularity of uncertainty are necessary (i.e. multi-granular linguistic information) [12, 15, 20]. The use of different label sets to assess information is also necessary when an expert has to assess different concepts, as for example it happens in information retrieval problems, to evaluate the importance of the query terms and the relevance of the retrieved documents [19]. In such situations, we need tools for the management of multi-granular linguistic information, i.e., we need to define a *multi-granular fuzzy linguistic modelling*. In [12] we define a proposal of multi-granular fuzzy linguistic modelling based on the ordinal fuzzy linguistic modelling and in [15] we define other one based on the 2-tuple fuzzy linguistic modelling. In this paper, we follow that defined in [15] which uses the concept of the *Linguistic Hierarchies* to manage the multi-granular linguistic information.

A *linguistic hierarchy* is a set of levels, where each level is a linguistic term set with different granularity from the remaining of levels of the hierarchy [3]. Each level belonging to a linguistic hierarchy is denoted as *l(t,n(t))*, being t a number that indicates the level of the hierarchy and *n(t)* the granularity of the linguistic term set of the level *t*.

Usually, linguistic hierarchies deal with linguistic terms whose membership functions are triangular-shaped, symmetrical and uniformly distributed in [0,1]. In addition, the linguistic term sets have an odd value of granularity representing the central label the value of *indifference*.

The levels belonging to a linguistic hierarchy are ordered according to their granularity, i.e., for two consecutive levels *t* and *t+1*, $n(t+1) >$ *n(t)*. Therefore, each level $t + 1$ provides a linguistic refinement of the previous level t.

A linguistic hierarchy, *LH*, is defined as the union of all levels *t*: $LH = \bigcup_t l(t, n(t))$. To build LH we must keep in mind that the hierarchical order is given by the increase of the granularity of the linguistic term sets in each level. Let $S^{n(t)} = \{s_0^{n(t)}, ..., s_{n(t)-1}^{n(t)}\}$ be the linguistic term set defined in the level t with $n(t)$ terms, then the building of a linguistic hierarchy must satisfy the following linguistic hierarchy basic rules [15]:

1. To preserve all *former modal points* of the membership functions of each linguistic term from one level to the following one.
2. To make *smooth transactions between successive levels*. The aim is to build a new linguistic term set, $S^{n(t+1)}$. A new linguistic term will be added between each pair of terms belonging to the term set of the previous level t. To carry out this insertion, we shall reduce the support of the linguistic labels in order to keep place for the new one located in the middle of them.

Generically, we can say that the linguistic term set of level *t+1*, $S^{n(t+1)}$, is obtained from its predecessor level *t*, $S^{n(t)}$ as: $l(t, n(t)) \rightarrow l(t+1, 2 \cdot n(t) - 1)$. Table 1 shows the granularity needed in each linguistic term set of the level t depending on the value *n(t)* defined in the first level (3 and 7 respectively).

	Level 1	Level 2	Level 3
l(t,n(t))	l(1,3)	l(2,5)	l(3,9)
l(t,n(t))	l(1,7)	l(2,13)	

Table 1. Linguistic Hierarchies.

A graphical example of a linguistic hierarchy is shown in figure 2:

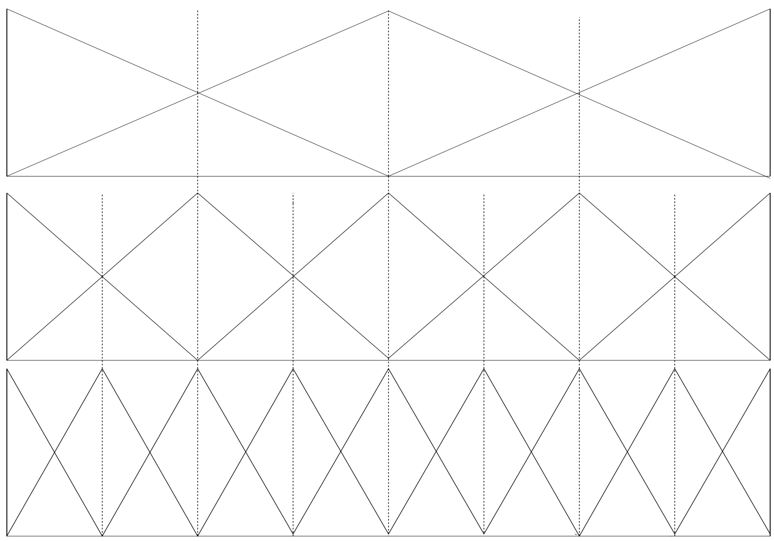

Fig. 2. Linguistic Hierarchy of 3, 5 and 9 labels

In [15] was demonstrated that the linguistic hierarchies are useful to represent the multi-granular linguistic information and allow to combine multi-granular linguistic information without loss of information. To do this, a family of transformation functions between labels from different levels was defined:

Definition 6. *Let* $LH = \bigcup_t l(t, n(t))$ *be a linguistic hierarchy whose linguistic term sets are denoted as* $S^{n(t)} = \{s_0^{n(t)}, ..., s_{n(t)-1}^{n(t)}\}$*. The transformation function between a 2-tuple that belongs to level* t *and another 2-tuple in level* $t' \neq t$ *is defined as:*

$$TF_{t'}^{t} : l(t, n(t)) \longrightarrow l(t', n(t'))$$

$$TF_{t'}^{t}(s_i^{n(t)}, \alpha^{n(t)}) = \Delta\left(\frac{\Delta^{-1}(s_i^{n(t)}, \alpha^{n(t)}) \cdot (n(t') - 1)}{n(t) - 1}\right)$$

As it was pointed out in [15] this family of transformation functions is biyective.

3.4 The Unbalanced Fuzzy Linguistic Modelling

In any problem that uses linguistic information the first goal to satisfy is the choice of the linguistic terms with their semantics, for establishing the label set to be used in the problem. In the literature, we can find two different possibilities for choosing the linguistic terms and their semantics:

- We can assume that all the terms of the label set are equally informative, i.e., symmetrically distributed as it happens in the above fuzzy linguistic modelling.
- We can assume that all the terms of the label set are not equally informative, i.e., not symmetrically distributed. In this case, we need an *unbalanced fuzzy linguistic modelling* [10, 11] to manage the linguistic term sets with different discrimination levels on both sides of the mid term (see Figure 3). As was known in [10], in the information retrieval systems the use of unbalanced linguistic term sets seems more appropriate than the use of symmetrical linguistic term sets, as to express the importance weights in the queries as to represent the relevance degrees of the documents.

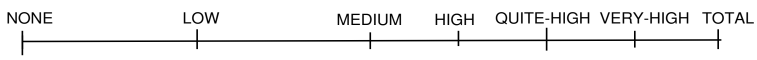

Fig. 3. Unbalanced Linguistic Term Set of 7 Labels

To manage unbalanced linguistic term sets we propose a method based on the 2-tuple fuzzy linguistic modelling. Basically, this method consists of representing unbalanced linguistic terms from different levels of an LH, carrying out computational operations of unbalanced linguistic information using the 2-tuple computational model. The method consists of the following steps:

1. Represent the unbalanced linguistic term set $\mathcal{S}$ by means of a linguistic hierarchy, LH.
   1.1. Chose a level t^- with an adequate granularity to represent using the 2-tuple representation model the subset of linguistic terms of $\mathcal{S}$ on the left of the mid linguistic term.
   1.2. Chose a level t^+ with an adequate granularity to represent using the 2-tuple representation model the subset of linguistic terms of $\mathcal{S}$ on the right of the mid linguistic term.
2. Define an unbalanced linguistic computational model.
   2.1. Choose a level $t' \in \{t^-, t^+\}$, such that $n(t') = max\{n(t^-), n(t^+)\}$.
   2.2. Define the comparison of two 2-tuples $(s_k^{n(t)}, \alpha_1)$, $t \in \{t^-, t^+\}$, and $(s_l^{n(t)}, \alpha_2)$, $t \in \{t^-, t^+\}$, with each one representing a counting of unbalanced information. Its expression is similar to the usual comparison of two 2-tuples but acting on the values $TF_{t'}^t(s_k^{n(t)}, \alpha_1)$ and $TF_{t'}^t(s_l^{n(t)}, \alpha_2)$. We should point out that using the comparison of 2-tuples we can easily define the comparison operators Max and Min.
   2.3. Define the negation operator of unbalanced linguistic information. Let $(s_k^{n(t)}, \alpha)$, $t \in \{t^-, t^+\}$ be an unbalanced 2-tuple then:

$$\mathcal{NEG}(s_k^{n(t)}, \alpha) = Neg(TF_{t''}^t(s_k^{n(t)}, \alpha)),\ t \neq t'',\ t'' \in \{t^-, t^+\}.$$

   2.4. Define aggregation operators of unbalanced linguistic information. This is done using the aggregation processes designed in the 2-tuple computational model but acting on the unbalanced linguistic values transformed by means of $TF_{t'}^t$. Then, once it is obtained a result, it is transformed to the correspondent level t by means of $TF_{t'}^t$ to express the result in the unbalanced linguistic term set.

Assuming the unbalanced linguistic term set shown in Figure 3 and the linguistic hierarchy shown in Figure 2, in Figure 4 we show how to select the different levels to represent the unbalanced linguistic term set.

4 A Model of Multi-Granular Fuzzy Linguistic Multi-Agent System Based on Filtering Techniques

In this section we present a new model of Web multi-agent system that combines both techniques aforementioned to improve their information gathering processes on Internet, i.e., it is designed using information filtering techniques and assumes a multi-granular fuzzy linguistic modelling.

As it is known, a promising direction to improve the effectiveness of search engines concerns the way in which it is possible to "filter" the great amount of information available across the Internet. Then, this new model incorporates in its activity the two more important existing filtering techniques, i.e. content-based filtering and collaborative filtering [26, 28]. On the other hand,

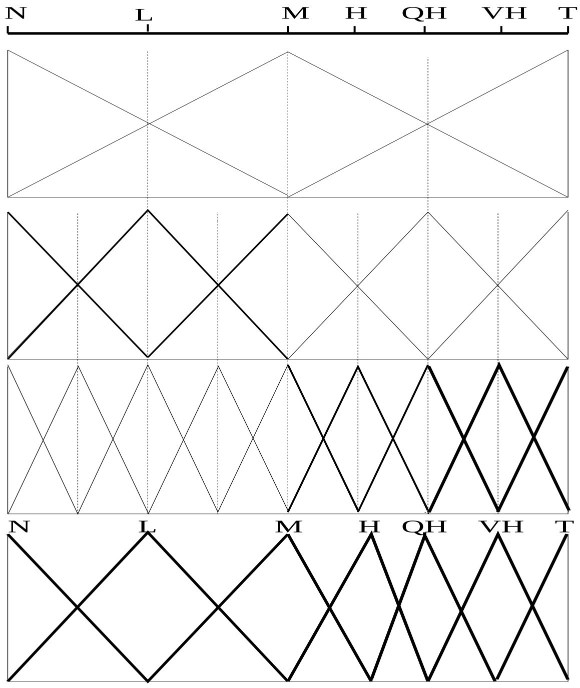

Fig. 4. Unbalanced Linguistic Term Set of 7 Labels

our fuzzy linguistic multi-agent models defined in [5, 6] present the following drawback: we assume the use of the same label set to represent the different linguistic information that appear in the communication processes developed in the multi-agent system, as for example, when user weighted queries and the relevance degrees of the retrieved documents are assessed using a same label set, although they represent different concepts. In this model the communication among the agents of different levels and between users and agents is carried out by using different label sets, i.e. working with multi-granular linguistic information, in order to allow a higher flexibility in the processes of communication of the system.

In what follows, we present the architecture of this multi-agent model and its operation.

4.1 Architecture

In [5, 6] were defined two fuzzy linguistic distributed multi-agent models that use linguistic information to carry out the communication processes among the agents. The architecture of these models is hierarchical and is composed of five action levels: *Internet Users, Interface Agents, Task Agents, Information Agents and Information Sources.* The architecture of our new multi-agent model must allow the application of the content-based and collaborative filtering tools. To do that, we incorporate in its architecture two new action levels: the level of the *content-based filtering agents* and the level of *collaborative filtering agent*. Therefore, this model presents a hierarchical architecture that contains seven activity levels: *Internet Users, Interface Agents, Collaborative Filtering Agent, Task Agents, Content-based Filtering Agent, Information Agents and Information Sources*. Furthermore, it works assuming a 2-tuple based multi-granular fuzzy linguistic modelling, that is, it uses different label sets ($S_1,\ S_2,\ S_3...$) to represent the different concepts to be assessed in its retrieval activity. These label sets S_i are chosen from those label sets that composes a LH, i.e., $S_i \in LH$. For example, we can use the LH shown in Figure 2. We should point out that the number of different label sets that we can use is limited by the number of levels of LH, and therefore, in many cases different the label sets S_i and S_j can be associated to a same label set of LH but with different interpretations depending on the concept to be modelled.

- **Level 1:** *Internet user*, which expresses his/her information needs by means of a linguistic multi-weighted query. Each term of a user query can be weighted simultaneously by two linguistic weights. The first weight is associated with a classical threshold semantics and the second one with a relative importance semantics. Then, the user makes a query to look for those documents related to the terms $\{t_1, t_2, ..., t_m\}$, which are weighted by a linguistic degree of threshold $\{p^1_1, p^1_2, ..., p^1_m\}$ with $p^1_i \in S_1$, and by a linguistic degree of relative importance $\{p^2_1, p^2_2, ..., p^2_m\}$ with $p^2_i \in S_2$. The user also expresses an information need category $\mathcal{A}_i$ chosen from a list of information need categories $\{\mathcal{A}_i, \ldots, \mathcal{A}_l\}$ provided by the system, and the user's identity $\mathcal{ID}$. All this information is given by the user to the *interface agent*.
- **Level 2:** *Interface agent* (one for user), that communicate the user's weighted query, the information need category and the user identity to the collaborative filtering agent, and filters the retrieved documents from collaborative filtering agent to give to the users those that satisfy better their needs. Finally, informs the collaborative filtering agent on set of documents used by user to satisfy his/her information needs DU.
- **Level 3:** *Collaborative filtering agent* (one for interface agent), that communicates the user multi-weighted query to the task agent, receives the more relevant documents chosen by the task agent, retrieves the recommendations on such documents from a collaborative recommendation system using the information need category expressed by the user $RC^{\mathcal{A}_i} = \{RC^{\mathcal{A}_i}_1, ..., RC^{\mathcal{A}_i}_v\}\ RC^{\mathcal{A}_i}_j \in S_3 \times [-0.5, 0.5)$, filters the documents by recalculating their relevance using these recommendations, and communicates these documents together with their new relevance degrees to the interface agent. Later, it carries out the tasks to update in the collaborative recommendation system the recommendations on the documents used by the user, i.e., it invites user to provide a recommendation rc_y on

each chosen document $d_y^U \in DU$ and this recommendation is stored in the collaborative recommendation system together with the recommendations provided by other users that used d_y^U.

- **Level 4:** *Task agent* (one for interface agent, generally), that communicate the terms of user query to the content-based filtering agents, and filters those documents from every content-based filtering agent that fulfills better the query.
- **Level 5:** *Content-based filtering agent* (one for agent information). Each content-based filtering agent communicates the terms of user query to its respective information agent and filters the relevant documents provided by its information agent by recalculating their relevance using the threshold weights. Then, the task agent receives from every content-based filtering agent h a set of documents and their relevance (D^h, RN^h), where every document d_h^h has associated a linguistic degree of relevance expressed in linguistic 2-tuples $rn_j^h \in S_4 \times [-0.5, 0.5)$ $(j = 1, ..., Card(D^h))$. It also receives a set of linguistic degrees of satisfaction $C^h = \{c_1^h, c_2^h, ..., c_m^h\}$, $c_i^h \in S_5 \times [-0.5, 0.5)$ of this set of documents D^h with regard to every term of the query t_i.
- **Level 6:** *Information agents*, which receive the terms of user query from the content-based filtering agents and look for the documents in the information sources. Then, each content-based filtering agent h receives from its respective information sources h the set of relevant documents that it found through information sources D^h and their relevance R^h, where every document d_j^h has an associated degree of relevance $r_j^h \in S_4 \times [-0.5, 0.5)$ $(j = 1, ..., Card(D^h))$.
- **Level 7:** *Information sources*, consisting of all data sources within the Internet, such as databases and information repositories.

This structure is presented in Figure 5.

4.2 Operation of the Model

The activity of this multi-agent model is composed of two phases:

1. *Retrieval phase*: This first phase coincides with the information gathering process developed by the multi-agent model itself, i.e., this phase begins when a user specifies his/her query and finishes when he/she chooses his/her desired documents among the relevant documents retrieved and provided by the system.
2. *Feedback phase*: This second phase coincides with the updating process of collaborative recommendations on desired documents existing in the collaborative recommender system, i.e., this phase begins when the *interface agent* informs the documents chosen by the user to the *collaborative filtering agent* and finishes when the recommender system recalculates and updates the recommendations of the desired documents.

In the following subsections, we explain both phases.

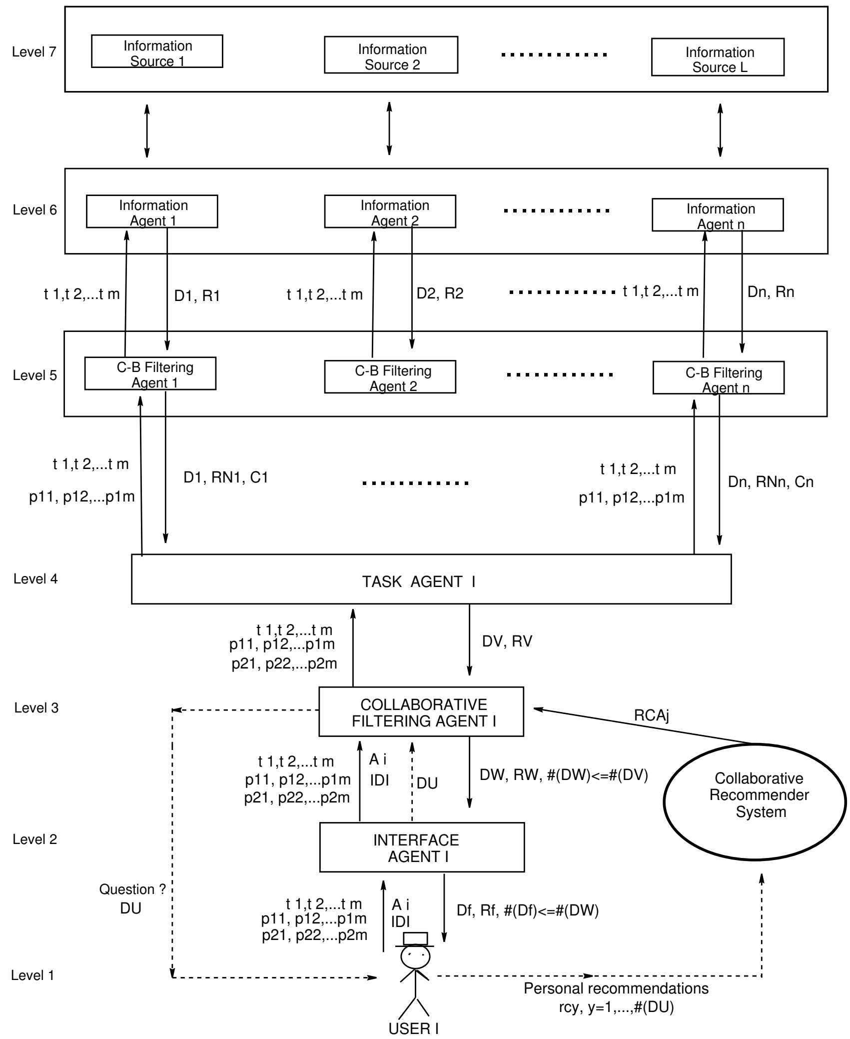

Fig. 5. Structure of Model of Web Multi-Agent System

4.2.1 Retrieval Phase

The description of the information gathering process of multi-agent model is as follows:

- **Step 1:** An *Internet user* expresses his/her information needs by means of a linguistic multi-weighted query $\{(t_1, p_1^1, p_1^2),\ (t_2, p_2^1, p_2^2), \ldots,\ (t_m, p_m^1, p_m^2)\}$, with $p_i^1 \in S_1$ and $p_i^2 \in S_2$, and an information need category $\mathcal{A}_i$ chosen from a list of information need categories $\{\mathcal{A}_1, \ldots, \mathcal{A}_l\}$ provided by the system. The system also requires the user's identity $\mathcal{ID}$. All this information is given by the user to the *interface agent*.

- **Step 2:** The *interface agent* gives the query together with the information need category ($\mathcal{A}_i$) to the *collaborative filtering agent*.
- **Step 3:** The *collaborative filtering agent* gives the terms and their importance weights to the *task agent*.
- **Step 4:** The *task agent* communicates the terms of the query and their importance weights to all the *content-based filtering agents* to which it is connected.
- **Step 5:** Each *content-based filtering agent h* makes the query to its respective *information agent h* and gives it the terms of the query $\{t_1, t_2, ..., t_m\}$.
- **Step 6:** All the *information agents* that have received the query, look for the information that better satisfies it in the *information sources*, and retrieve from them the documents. We assume that the documents are represented in the *information sources* using an index term based representation as in Information Retrieval [17, 18]. Then, there exists a finite set of index terms $T = \{t_1, ..., t_l\}$ used to represent the documents and each document d_j is represented as a fuzzy subset

$$d_j = \{(t_1, F(d_j, t_1)), ..., (t_l, F(d_j, t_l))\}, F(d_j, t_i) \in [0, 1],$$

  where F is any numerical indexing function that weighs index terms according to their significance in describing the content of a document. $F(d_j, t_i) = 0$ implies that the document d_j is not at all about the concept(s) represented by index term t_i and $F(d_j, t_i) = 1$ implies that the document d_j is perfectly represented by the concept(s) indicated by t_i.
- **Step 7:** Each *content-based filtering agent h* receives from its respective *information agent h* a set of documents and their relevances (D^h, R^h) ordered decreasingly by relevance. Every document d_j^h has an associated linguistic degree of relevance $r_j^h \in S_4 \times [-0.5, 0.5)$ which is calculated as

$$r_j^h = \overline{x}^e[\Delta(g \cdot F(d_j^h, t_1)), \ldots, \Delta(g \cdot F(d_j^h, t_m))] = \Delta(g \cdot \sum_{i=1}^{m} \frac{1}{m} F(d_j^h, t_i)),$$

  being $g+1$ the cardinality of S_4. Each *content-based filtering agent h* filters documents received from its respective *information agent h* by recalculating their relevance by means of a linguistic matching function

$$e_h : (S_4 \times [-0.5, 0.5)) \times S_1 \to S_4 \times [-0.5, 0.5),$$

  which is defined to model the semantics of threshold weights associated with the query terms. This linguistic matching function requires a previous transformation of threshold weights expressed in labels of S_1 that must be transformed in labels of S_4; to make uniform the multi-granular linguistic information, we chose the linguistic term set used to express the relevance degrees. We use the transformation function viewed in definition 6 ($TF^t_{t'}$), to transform the linguistic labels in level S_1 (t) to labels in level S_4 (t'):

$$TF^{S_1}_{S_4}(s_i^{n(S_1)}, \alpha^{n(S_1)}) = \Delta(\frac{\Delta^{-1}(s_i^{n(S_1)}, \alpha^{n(S_1)}) \cdot (n(S_4)-1)}{n(S_1)-1})$$

obtaining the new linguistic threshold weights $\{p_1^{1'}, p_2^{1'}, ..., p_m^{1'}\}$, $p_i^{1'} \in S_4$ for the terms $\{t_1, t_2, ..., t_m\}$. Different *content-based filtering agents* can have different threshold matching functions. For example, some linguistic matching functions that we can use are:

1. $e^1(\Delta(g \cdot F(d_j, t_i)), p_i^{1'}) = \begin{cases} (s_g, 0) \text{ if } \Delta(g \cdot F(d_j, t_i)) \geq (p_i^{1'}, 0) \\ (s_0, 0) \text{ otherwise.} \end{cases}$
2. $e^2(\Delta(g \cdot F(d_j, t_i)), p_i^{1'}) = \begin{cases} \Delta(g \cdot F(d_j, t_i)) \text{ if } \Delta(g \cdot F(d_j, t_i)) \geq (p_i^{1'}, 0) \\ (s_0, 0) \qquad\qquad \text{otherwise.} \end{cases}$
3. $e^3(\Delta(g \cdot F(d_j, t_i)), p_i^{1'}) = \begin{cases} \Delta(min\{g, 0.5 + g \cdot F(d_j, t_i)\}) \text{ if } \Delta(g \cdot F(d_j, t_i)) \geq (p_i^{1'}, 0) \\ \Delta(max\{0, g \cdot F(d_j, t_i) - 0.5\}) \text{ otherwise.} \end{cases}$

Then, each *content-based filtering agent h* calculates a new set of relevance degrees $RN^h = \{rn_j^h, j = 1, ..., card(D^h)\}$ characterizing the documents D^h, which is obtained as

$$rn_j^h = \overline{x}^e[e_h(\Delta(g \cdot F(d_j^h, t_1)), p_1^{1'}), ..., e_h(\Delta(g \cdot F(d_j^h, t_m)), p_m^{1'})] =$$

$$\Delta(\sum_{i=1}^{m} \frac{1}{m} \Delta^{-1}(e_h(\Delta(g \cdot F(d_j^h, t_i)), p_i^{1'}))).$$

- **Step 8:** The *task agent* receives from every *content-based filtering agent* a set of documents and their new relevance (D^h, RN^h). It also receives a set of linguistic degree of satisfaction $C^h = \{c_1^h, c_2^h, ..., c_m^h\}$, $c_i^h \in S_3 \times [-0.5, 0.5)$ of D^h with regard to every term of the query as

$$c_i^h = \overline{x}^e[e_h(\Delta(g \cdot F(d_1^h, t_i)), p_i^{1''}), ..., e_h(\Delta(g \cdot F(d^h_{card(D^h)}, t_i)), p_i^{1''})] =$$

$$\Delta(\sum_{j=1}^{card(D^h)} \frac{1}{card(D^h)} \Delta^{-1}(e_h(\Delta(g \cdot F(d_j^h, t_i)), p_i^{1''}))).$$

where the $p_i^{1''}$ are the p_i^1 expressed in the set S_5, using the transformation function $TF^{S_1}_{S_5}$ viewed in definition 6.

Then, the *task agent* selects the number of documents to be retrieved from each *content-based filtering agent h.* To do so, it applies the following three steps:

– **Step 8.1:** The *task agent* orders D^h with respect to the new relevance RN.
– **Step 8.2:** The *task agent* aggregates both linguistic information weights, the satisfactions of the terms of the query from every *information agent*, $(c_i^h, \alpha_i), c_i^h \in S_5$, and the importance weights that the user assigned to these terms, $(p_i^2, \alpha_i), p_i^2 \in S_2$, using the aggregation process for multi-granular linguistic information presented in [15]:
  1. *Normalization Phase*: the linguistic term set used to express the relevance is chosen to make uniform the multi-granular linguistic information. Then, all the information are expressed in that linguistic term set by means of 2-tuples.
  2. *Aggregation Phase*: through a 2-tuple aggregation operator the information is aggregated. In this paper we use the 2-tuple linguistic weighted average operator, $\overline{x}_l^w$, for combining the satisfactions of the terms of the query and the importance weights.

  Let $\{[(p_1^2, \alpha_1), (c_1^h, \alpha_1^w)], ..., [(p_m^2, \alpha_m), (c_m^h, \alpha_m^w)]\}$, $p_i^2 \in S_2$ *and* $c_i^h \in S_5$ be the set of pairs of linguistic 2-tuples of importance and satisfaction to be aggregated by the task agent for every information agent h. Then, for combining them first the linguistic values $(p_i^2, \alpha_i), p_i^2 \in S_2$ and $(c_i^h, \alpha_i^w), c_i^h \in S_5$ are transformed in the linguistic term set used to express the relevance degrees, in this case S_4, obtaining their corresponding values $(p_i^{2'}, \alpha_i')$, $p_i^{2'} \in S_4$ and $(c_i^{h'}, \alpha_i^{w'})$, $c_i^{h'} \in S_4$. Once the multi-granular information has been unified according to the 2-tuple linguistic weighted average operator definition, the aggregation of the pair associated with every term is obtained as:

$$\lambda^h = \overline{x}_l^w([(p_1^{2'}, \alpha_1'), (c_1^{h'}, \alpha_1^{w'})], ..., [(p_m^{2'}, \alpha_m'), (c_m^{h'}, \alpha_m^{w'})])$$

– **Step 8.3:** To gather the better documents from *content-based filtering agents*, the *task agent* selects a number of documents $k(D^h)$ from every *content-based filtering agent* h being proportional to its respective degree of satisfaction λ^h:

$$k(D^h) = round(\frac{\sum_{i=1}^n card(D^i)}{n} \cdot P_s^h),$$

  where $P_s^h = \frac{\Delta^{-1}(\lambda^h)}{\sum_{i=1}^n \Delta^{-1}(\lambda^h)}$ is the probability of selection of the documents from *content-based filtering agent* h.

- **Step 9:** The *collaborative filtering agent* receives from the *task agent* a list of documents $DV = \{d_1^V, ..., d_v^V\}$ ordered with respect to their relevance RV, such that:
  1. $r_j^V \geq r_{j+1}^V$,
  2. for a given document $d_j^V \in DV$ there exists a h such that $d_j^V \in D^h$ and $r_j^V \in RN^h$, and

3. $card(DV) = v \leq \sum_{i=1}^{n} k(D^i)$.

Then, *collaborative filtering agent* filters the documents provided by the *task agent* using the recommendations on such documents provided by other users in previous searches which are stored in a *collaborative recommender system*. This is done in the following steps:

- **Step 9.1:** The *collaborative filtering agent* asks *collaborative recommender system* the recommendations existing on DV associated with the information need category $\mathcal{A}_i$ expressed by the user and retrieves them,

$$RC^{\mathcal{A}_i} = \{RC_1^{\mathcal{A}_i}, ..., RC_v^{\mathcal{A}_i}\},\ RC_j^{\mathcal{A}_i} \in S_3 \times [-0.5, 0.5).$$

- **Step 9.2:** The *collaborative filtering agent* filters the documents by recalculating their relevance using these recommendations $RC^{\mathcal{A}_i}$. Then, for each document $d_j^V \in DV$ a new linguistic relevance degree r_j^{NV} is calculated from r_j^V and $RC_j^{\mathcal{A}_i}$ by means of the 2-tuple weighted operator $\overline{x}^w$ defined in Definition 4:

$$r_j^{NV} = \overline{x}^w(r_j^V, RC_j^{\mathcal{A}_i}),$$

  using for example the weighting vector $W = [0.6, 0.4]$.

- **Step 10:** The *interface agent* receives from the *collaborative filtering agent* a list of documents $DW = \{d_1^W, ..., d_w^W\}$ ordered with respect to their relevance RW, such that:
  1. $r_j^W \geq r_{j+1}^W$,
  2. for a given document $d_j^W \in DW$ there exists a i such that $d_j^W = d_i^V$ and $r_j^W = r_i^{NV}$, and
  3. $card(DW) = w \leq v = card(DV)$.

  Then, the *interface agent* filters these documents in order to give to the user only those documents that fulfill better his/her needs, which we call D_f. For example, it can select a fixed number of documents K and to show the K best documents.

4.2.2 Feedback Phase

This phase is related to the activity developed by the *collaborative recommender system* once user has taken some of documents retrieved by the multi-agent system. In the collaborative recommender systems the people collaborate to help one another to perform filtering by recording their reactions to documents they read [21, 28]. In our multi-agent model this feedback activity is developed in the following steps:

- **Step 1:** The *interface agent* gives the user's identity $\mathcal{ID}$ (usually his/her e-mail) together with the set of documents $DU = \{d_1^U, ..., d_u^U\},\ u \leq card(D_f)$ used by the user to the *collaborative filtering agent*.

- **Step 2:** The *collaborative filtering agent* asks user his/her opinion or evaluation judgements about DU, for example by means of an e-mail.
- **Step 3:** The *Internet user* communicates his/her linguistic evaluation judgements to the *collaborative recommender system*, rc_y, $y = 1, ..., card(DU)$, $rc_y \in S_3$.
- **Step 4:** The *collaborative recommender system* recalculates the linguistic recommendations of set of documents DU by aggregating again the opinions provided by other users together with those provided by the Internet user. This can be done using the 2-tuple aggregation operator $\overline{x}^e$ given in Definition 3. Then, given a chosen document $d_y^U \in DU$ that receives a recommendation or evaluation judgement rc_y from the Internet user, and supposing that in the collaborative recommender system there exists a set of stored linguistic recommendations $\{rc_1, ..., rc_M\}$, $rc_i \in S_3$ associated with d_y^U for the information need category $\mathcal{A}_i$, which were provided by M different users in previous searches, then a new value of recommendation of d_y^U is obtained as

$$RC_y^{\mathcal{A}_i} = \overline{x}^e[(rc_1, 0), ...(rc_M, 0), (rc_y, 0)].$$

5 Concluding Remarks

Nowadays Internet users need tools to assist them in his/her processes of information gathering because of the large amount of information available on the Web. We have presented two techniques that could contribute to solve this problem, the information filtering tools and the fuzzy linguistic modelling. Then, we have defined a new model of fuzzy linguistic Web multi-agent system using both techniques. In particular, this new model of Web multi-agent system is based as on content-based filtering tools as on collaborative filtering tools and on the multi-granular fuzzy linguistic modelling. Some advantages of this model are the following:

- We improve the search and mining processes on the Web and this could increase the users' satisfaction degrees.
- The use of the multi-granular linguistic information allows a higher flexibility and expressiveness in the communication among the agents and between users and agents in the information gathering process.
- The use of the multi-granular linguistic information does not decrease the precision of system in its results.
- The use of IF techniques allow to filter the information and so to improve the retrieval process.

References

1. C. Basu, H. Hirsh, W. Cohen, Recommendation as classification: Using social and content-based information in recommendation, *Proc. of the Fifteenth National Conference on Artificial Intelligence,* 1998, pp. 714-720.
2. M. Claypool, A. Gokhale, T. Miranda, Combining content-based and collaborative filters in an online newpaper, *Proc. of the ACM SIGIR Workshop on Recommender Systems-Implementation and Evaluation.*
3. O. Cordón, F. Herrera and I. Zwir. *Linguistic modelling by hierarchical systems of linguistic rules.* IEEE Transactions on Fuzzy Systems, 10 (1) (2001) 2-20.
4. M. Chau, D. Zeng, H. Chen, M. Huang, D. Hendriawan, Design and evaluation of a multi-agent collaborative Web mining system, Decision Support Systems 35 (2003) 167-183.
5. M. Delgado, F. Herrera, E. Herrera-Viedma, M.J. Martín-Bautista, M.A. Vila, Combining linguistic information in a distributed intelligent agent model for information gathering on the Internet, in P.P. Wang, Ed., *Computing with Words*, (John Wiley & Son, 2001) 251-276.
6. M. Delgado, F. Herrera, E. Herrera-Viedma, M.J. Martín-Bautista, L. Martínez, M.A. Vila. A communication model based on the 2-tuple fuzzy linguistic representation for a distributed intelligent agent system on Internet, *Soft Computing*, 6 (2002) 320-328.
7. N. Good, J.B. Shafer, J.A. Konstan, A. Borchers, B.M. Sarwar, J.L. Herlocker, J. Riedl, Combining collaborative filtering with personal agents for better recommendations, *Proc. of the Sixteenth National Conference on Artificial Intelligence,* 1999, 439-446.
8. U. Hanani, B. Shapira, P. Shoval. Information Filtering: Overview of Issues, Research and Systems. *User Modeling and User-Adapted Interaction* 11: 203-259, 2001.
9. F. Herrera, E. Herrera-Viedma, Aggregation operators for linguistic weighted information, *IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems*, 27 (1997) 646-656.
10. F. Herrera, E. Herrera-Viedma, L. Martínez. An Information Retrieval System with Unbalanced Linguistic Information Based on the Linguistic 2-tuple Model. 8th *International Conference on Information Processing and Management of Uncertainty in Knowledge-Bases Systems (IPMU'2002).* Annecy (France) 23-29.
11. F. Herrera, E. Herrera-Viedma, L. Martínez, P.J. Sanchez. A Methodology for Generating the Semantics of Unbalanced Linguistic Term Sets. 9th International Conference on Fuzzy Theory and Technology, Florida, 2003, 151-154, 2003.
12. F. Herrera, E. Herrera-Viedma, L. Martínez. A Fusion Approach for Managing Multi-Granularity Linguistic Term Sets in Decision Making, *Fuzzy Sets and Systems*, 114 (2000) 43-58.
13. F. Herrera, E. Herrera-Viedma, J.L. Verdegay, Direct approach processes in group decision making using linguistic OWA operators, *Fuzzy Sets and Systems*, 79 (1996) 175-190.
14. F. Herrera, L. Martínez, A 2-tuple fuzzy linguistic representation model for computing with words, *IEEE Transactions on Fuzzy Systems*, 8 (6) (2000) 746-752.

15. F. Herrera, L. Martínez, A model based on linguistic 2-tuples for dealing with multigranularity hierarchical linguistic contexts in multiexpert decision-making, *IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics*, 31(2) (2001) 227-234.
16. F. Herrera, Martínez, The 2-tuple linguistic computational model. Advantages of its linguistic description, accuracy and consistency, *Int. J. of Uncertainty , Fuzziness and Knowledge-Based Systems*, 9 (2001) 33-48.
17. E. Herrera-Viedma, Modeling the retrieval process of an information retrieval system using an ordinal fuzzy linguistic approach, *J. of the American Society for Information Science and Technology*, 52(6) (2001) 460-475.
18. E. Herrera-Viedma, An information retrieval system with ordinal linguistic weighted queries based on two weighting elements, *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9 (2001) 77-88.
19. E. Herrera-Viedma, O. Cordón, M. Luque, A.G. López, A.M. Muñoz, A Model of Fuzzy Linguistic IRS Based on Multi-Granular Linguistic Information, *International Journal of Approximate Reasoning*, 34 (3) (2003) 221-239.
20. E. Herrera-Viedma, L. Martínez, F. Mata, F. Chiclana. A Consensus Support System Model for Group Decision-making Problems with Multi-granular Linguistic Preference Relations, *IEEE Trans. on Fuzzy Systems* 2005. To appear.
21. E. Herrera-Viedma, E. Peis, Evaluating the informative quality of documents in SGML-format using fuzzy linguistic techniques based on computing with words, *Information Processing & Management*, 39(2) (2003) 195-213.
22. M. Kobayashi, K. Takeda, Information retrieval on the web, *ACM Computing Surveys*, 32(2) (2000) 148-173.
23. S. Lawrence, C. Giles, Searching the web: General and scientific information access, *IEEE Comm. Magazine*, 37 (1) (1998) 116-122.
24. H. Lieberman, Personal assistants for the Web: A MIT perspective. In M.Klusch(Ed.), *Intelligent Information Agents* (Springer-Verlag, 1999) 279-292.
25. A. Moukas, G. Zacharia, P. Maes, Amalthaea and Histos: Multiagent systems for WWW sites and representation recommendations, in M. Klusch(Ed.), *Intelligent Information Agents* (Springer-Verlag, 1999) 293-322.
26. A. Popescul, L.H. Ungar, D.M. Pennock, S. Lawrence, Probabilistic models for unified collaborative and content-based recommendation in sparce-data environments. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI)*, San Francisco, (2001) 437-444.
27. L.M. Quiroga, J. Mostafa, An experiment in building profiles in information filtering: the role of context of user relevance feedback, *Information Processing and Management* 38 (2002) 671-694.
28. P. Reisnick, H.R. Varian, Recommender Systems. Special issue of Comm. of the ACM, 40 (3) (1997) 56-59.
29. L.A. Zadeh, The concept of a linguistic variable and its applications to approximate reasoning. Part I, *Information Sciences*, 8 (1975) 199-249, Part II, *Information Sciences*, 8 (1975) 301-357, Part III, *Information Sciences*, 9 (1975) 43-80.