Searching Web Documents Using a Fuzzy-Based Method

Enrico Fischetti[1,*], Aniello Nappi[2]

[1]Dipartimento di Ingegneria Informatica (DIEM)
University of Salerno, Italy

[2]MANUCOR spa
Caserta, Italy

[*]*Corresponding author's email: efischetti [AT] unisa.it*

ABSTRACT--- Web searching could be more fruitful if a user easily found documents which satisfy his/her needs in terms of structure, format and contents. In this paper first the state of the art is briefly reviewed and then the solution proposed uses a fuzzy linguistic description of the documents, a linguistic variant of standard metadata types. Linguistic expressions are used to qualitatively represent both meta-information and user needs and a matching system is developed to select the most compatible documents with the user profile. So the documents retrieved by a web search engine are organized in clusters and ordered in each of them.

Keywords – Web documents, Search algorithms, Type-2 fuzzy sets, Linguistic variables, User modeling

1. INTRODUCTION

The amount of information on the Web is growing up with an impressive speed and involves any knowledge and information area; this growth has lead it to become a huge repository of documents extremely different in terms of structure, format and contents. On one side, this situation is undoubtedly positive in terms of wealth of information, on the other side it may create enormous problems when these data are to be retrieved. The current Web, in fact, has not a well-defined structure, it makes impossible to adapt the search to user needs and it obliges him/her to long and sometimes unproductive sessions of searching activity. This problem is avowedly recognized as a major one and several papers tackle it and suggest solutions [5, 7, 8, 10, 11].

More specifically, in [5] an architecture for semantic based information retrieval is proposed, in which plain text is red semantically and the extracted metadata is stored and later used for semantic search. In turn, the paper [7] presents an algorithm to improve a web search query based on the feedback on the viewed documents. In [8] a summarization method is illustrated to enhance the current web-search approaches by offering a summary of each clustered set of web-search results with contents addressing the same topic, which should allow the user to quickly identify the information covered in the clustered search results. In [10] the documents are modeled as a set of structures that describe relationships among the entities mentioned in the text and in [11] the search logs in the distributed search servers are treated as footprints and an adaptive method is proposed to support effective searching over large-scale web documents.

A very good solution would be to redesign the web as a a semantic structured web, even diversified in the concepts , like suggested by most important organizations for standardization such as W3C, IEEEE [3, 4, 15]. The declared common objective of this line of study is to give a formal semantics through the use of a standardized metadata structure, realized by the XML language (frequently identified as the most valid instrument for this goal). Interesting results have been attained [6, 12, 13, 14] and intensive research activity is under way.

More specifically, in [13] an approach established on the basis of the new concept of section-semantic relation structure is presented. "section" is defined as a block of media that contains a single "atom" of information, and "semantic relation" is defined as the relationship between two sections. In turn, in [14] a formal ontology is presented which not only allows for representing the structure of multimedia documents but also to connect with arbitrary background knowledge on the web.

This paper presents a fuzzy-based approach to the above mentioned problem different from other authors' [2, 9]. In particular, the paper [2] presents a method based on the term frequency–inverse document frequency to auto-extract the keywords in the patent literature. In [9] the case is investigated which includes the documents that belong to more than one category and a similar document search system that uses fuzzy clustering is illustrated. The method illustrated in this paper presents two relevant features: the use of the natural language to characterize a document through qualitative information (in order to help the authors in developing metadata) and the definition of a matching, during a search session, between the user profile and the document he/she is looking for.

Through the use of type-2 fuzzy sets and a linguistic variable (lv, for short), more expressivity to document metadata can be given, thanks to linguistic terms that reflect the imprecision of a characterization. With these fuzzy metadata, one can express user preferences, and the results of a search can be organized on the basis of their compatibility with the user profile.

The basic idea is that a document that perfectly matches the user needs can be "useful enough" also for other users with different preferences: this vague relationship can be represented and managed in this fuzzy-based model.

This paper is organized as follows: in Section 2 some basic concepts regarding fuzzy sets are recalled; then in Section 3: 1) a suitable linguistic approximation algorithm is illustrated, 2) a type-2 resemblance index is presented, 3) a measure for inequality is introduced. Section 4 deals with formalized documents, user profiles and linguistic attributes. Section 5 illustrates the selection algorithm and the next section discusses a meaningful case study. Finally, concluding remarks sketch some aspects deserving further investigation.

2. BASIC CONCEPTS

Given a nonempty classical crisp set *X*, a *fuzzy set* on *X* [16] is a function *A*: X $\rightarrow$ [0, 1]. The number X(x) is interpreted as the membership degree of the element x of X to the fuzzy set *A*.

If X is a finite set (whose cardinality is n), the following notation is used: $A = a_1/x_1 + a_2/x_2 + ... + a_n/x_n$, where $A(x_i) = a_i$.

If *A* is a triangular function, as presented in Fig. 1, then *A* is said a *fuzzy triangular number*. It is singled out through three parameters and represented as [a, d, c], with a≤d≤c.

The following operations can be easily extended to *fuzzy triangular numbers:*

[a, b, c] + [a', b', c'] [a+a', b+b', c+c'] (*sum*)

[a, b, c] $\underline{+}$ [a', b', c'] [min(1, a+a'), min(1, b+b'), min(1, c+c')] (*limited sum in [0,1]*)

[a, b, c] - [a', b', c'] = [a-a', b-b'), c-c'] (*difference*)

[a, b, c] $\underline{-}$ [a', b', c'] = [max(0, a-a'), max(0, b-b'), max(0, c-c')] (*limited difference in [0,1]*)

k*[a, b, c] = [ka, kb, kc] (*product with a real number k*)

Given the *fuzzy triangular numbers* x=[a, b, c] and y=[a', b', c'], one says that $x \preceq y$ if the following relations are true: $b \leq a'$, $c \leq b'$, $c \leq c'$. If the relation ≤ is true for each couple of elements of a set of fuzzy triangular numbers, this set is said *totally ordered*.

Let it be X a classical nonempty set and *F*[0, 1] the set of fuzzy sets defined on the interval [0, 1]. A type-2 fuzzy set on X is a function X $\rightarrow$ *F*[0, 1].

A *fuzzy partition* on A is a class of fuzzy set A_i on [0,1] so that for each x in A it is true that $\Sigma A_i(x) = 1$ and A_i belongs to the class, so that $A_j(x)>0$. In this paper a partition built with fuzzy triangular numbers on [0, 1] is used.

The set {[0, 0, 0.2], [0, 0.2, 0.4], [0.2, 0.4, 0.6], [0.4, 0.6, 0.8] [0.6, 0.8, 1], [0.8, 1, 1]} is an example of 6-dimensions fuzzy partition on [0, 1].

Fig. 1 shows graphically this fuzzy partition on [0, 1]. A fuzzy partition singles out a set of totally ordered fuzzy triangular numbers.

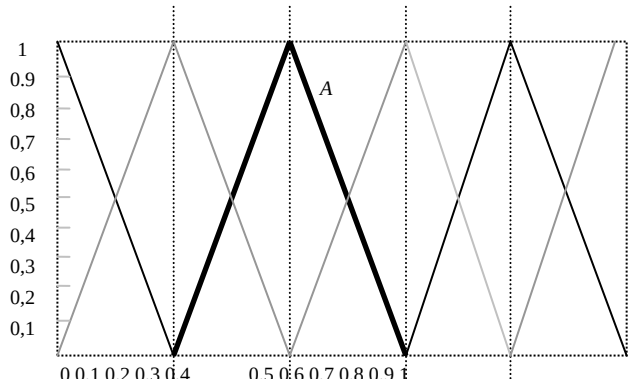

Figure 1: A triangular function in a triangular fuzzy partition on [0, 1].

Linguistic Variables

Intuitively each word in natural language is a *linguistic variable (lv)*. For example, Age is a lv. The values a variable can assume are called *linguistic terms (lt).* In this way, the word Age, for example, can assume the following values: young, old, not much young, old enough, not young but not too old. Formally, a lv is a quintuple (x, U, T(x), G, M) in

which *x* is the name of the variable, *U* is its domain of discourse, *T(x)* is the set of its linguistic terms, *G* is a grammar that establish the rules for the construction of correct terms of x, and *M* is the function that gives the meaning of each term: M: T(x) $\rightarrow$ *F*(U), where *F*(U) is the class of fuzzy sets on U. For the lv *Age*, the terms introduced before are the elements of T(*Age*), while U = [0, 120].

In the paper, the lv *Interest* is taken into account with its associated terms:

Table 1. Linguistic variable : *Interest*

Triangular Num.	*Linguistic Term*	*Triangular Num.*	*Linguistic Term*
[0.8, 1, 1]	Fully Interested (vi)	[0.2, 0.4, 0.6]	Sufficiently Interested (si)
[0.6, 0.8, 1]	Interested (i)	[0.0, 0.2, 0.4]	Little Interested (li)
[0.4, 0.6, 0.8]	Fairly Interested (fi)	[0.0, 0.0, 0.2]	Not Interested (un)

3. TOOLS FOR THE ANALYSIS OF DOCUMENTS

Some algorithms and functions are required to implement correctly the search: a linguistic approximation algorithm (used to translate triangular fuzzy numbers into linguistic terms); an operation of resemblance (between documents and a user profile) and a fuzzy measure for inequality (used in the selection algorithm).

3.1 *Linguistic approximation*

The algorithm for *linguistic approximation*, named $\boldsymbol{ApprLing_{k,d}}$ allows to map triangular fuzzy numbers into linguistic expressions referring to a specific linguistic variable.

The linguistic approximation function LA is defined as follows: $\mathbf{LA_{L,k}[\alpha] = \lambda}$ where α is the fuzzy number that should receive a new linguistic label and λ is the label attached to the number. The algorithm requires two parameters, *k*, namely the number of new labels generated between each couple of linguistic terms and *L*, that is the level of precision, that allows to create complex and precise linguistic terms. With L=1, LA introduces **k** intermediate labels, obtained with linguistic modifiers starting from n basic terms, and it provides the generation of [(n – 1) * k + n] overall labels, then associated with the triangles. With L > 1, the algorithm introduces *k* linguistic terms between each couple of terms (Fig. 2):

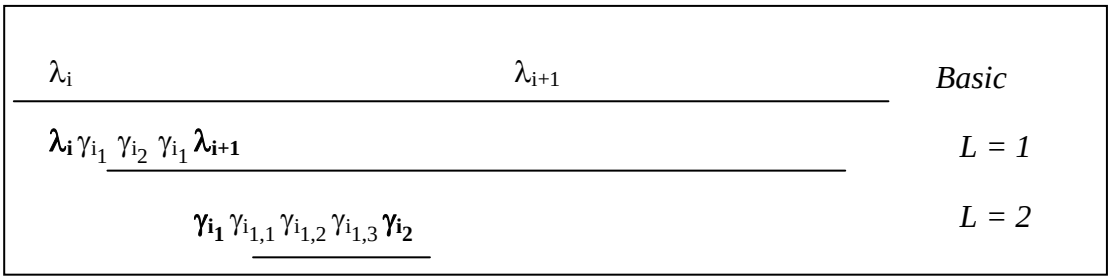

Figure 2: A graphical representation of $\boldsymbol{Apprling_{k,d}}$ for L = 1, 2

In the sequel only $LA_{1,3}$, (LA_3, for short) will be utilized as it allows to obtain a sufficient linguistic expressivity. As said, if n is the number of the basic linguistic labels, the number of linguistic labels generated by LA will be [(n – 1) * k] and so the overall number of terms is [(n – 1) * k]+ n.

From a formal point of view:

E = { $\lambda_n, \lambda_{n-1}, \ldots, \lambda_1$} is the set of *n* basic linguistic terms of a variable **V**;

F = { $\alpha_n, \alpha_{n-1}, \ldots, \alpha_1$} is the subset of **Tr** associated with the set **E**;

M is the semantic rule such that $\mathbf{M}(\lambda_i) = \alpha_i$.

Given two basic terms λ_i and λ_{i+1}, the LA algorithm splits this interval into a certain number (*k*) of sub-intervals. The greater is the number, the more precise is the approximation. To represent linguistically the new labels associated with the sub-intervals so generated, linguistic modifiers such as much, little, more or less are introduced. It is worth emphasizing that increasing the number of labels deeply affects the computational complexity of the procedure. On the

other hand, using few approximate labels the expressive power of the system decreases correspondingly. The value k = 3 allows to achieve a good compromise between the expressivity of the system and a reasonable complexity. For generating new labels between two existing labels, one has to single out if the linguistic value associated is positive (e.g., good, fair,…) or negative (poor, inadequate,…), in order to assign the correct linguistic modifiers. For each couple of the basic terms, let us consider three different cases, presented in Table 3.

Table 2. Possible situations when applying the Linguistic Approximation Algorithm.

Case	λ_i	λ_{i+1}
I	Positive	Positive
II	Negative	Positive
III	Negative	Negative

The next step leads to single out the modifiers to be applied in order to have intermediate labels:

I) both labels have positive meaning, thus one can use increasing modifiers applied to λ_i (more, very, a lot…) and decreasing modifiers applied to λ_{i+1} (almost, less…);

II) the lower label is negative while the other has positive meaning: one has to set decreasing modifiers for both λ_i and λ_{i+1} (almost, less…);

III) both labels are negative: one has to use decreasing modifiers for λ_i and increasing for λ_{i+1}.

Suppose that α is a triangular fuzzy number to be approximated; suppose that the central value of α (denoted by m) takes a value included between m_i and m_{i+1}, that are the central values of the numbers associated to the basic terms α_i and α_{i+1}, respectively, which have positive (negative, respectively) linguistic value. If $d = m_{i+1}-m_i$, Table 3 shows a possible implementation of LA_3 (it will be used in the following case study):

Table 3. Implementing LA_3.

λ_i	λ_{i+1}	Condition	Resulting Labels
Positive	Positive	if $m \in [m_i, m_i + d/10]$	λ_i
Negative	Positive		λ_i
Negative	Negative		λ_i
Positive	Positive	if $m \in]m_i + (d/10), m_i + (3/10)*d]$	More(Much)than λ_i
Negative	Positive		Little better than λ_i
Negative	Negative		Almost λ_i
Positive	Positive	if $m \in]m_i + (3/10)*d, m_i + (7/10)*d]$	Very (*or* A lot more) λ_i
Negative	Positive		Better than λ_i
Negative	Negative		Very λ_{i+1}
Positive	Positive	if $m \in]m_i + (7/10)*d, m_i + (9/10)*d]$	Almost λ_{i+1}
Negative	Positive		Less than λ_{i+1}
Negative	Negative		Little more than λ_{i+1}
Positive	Positive	if $m \in]m_i + (9/10)*d, m_{i+1}]$	λ_{i+1}
Negative	Positive		λ_{i+1}
Negative	Negative		λ_{i+1}

In this case the maximum number of obtainable labels is (4n-3), where *n* stands for the number of basic labels. Moreover, it is worth noting that the linguistic approximation is useful only to facilitate the interpretation of the results, while the operations are always carried out on not approximated type-2 fuzzy sets.

Example 1: Let us consider the lv *Interest* and the correspondences triangular fuzzy numbers/linguistic terms of Table 1, suppose that the value k = 3 is selected and the approximation algorithm is applied to the number: [0.74, 0.90, 1]. Its central value (0.90) is included between m_i and m_{vi} which are the central values of the triangular fuzzy numbers corresponding to "Interested" and "Fully interested", respectively.

Then $d = m_{vi} - m_i = 0.2$; but $0.90 \in [m_i + (7/10)*0.2, m_i + (9/10)*0.2]$ and thus the algorithm gives the linguistic modification associated in this case: [0.74, 0.90, 0.1] ≈ "Almost Fully interested".

Example 2: Let us consider the lv *Compatibility* and the correspondences triangular fuzzy numbers/linguistic terms of Table 2, suppose that k = 3 again and the approximation algorithm is applied to the number: [0.33, 0.42, 0.61]. In this

case the central value (0.42) is included between m_i and m_{vi} which are the central values of the triangular fuzzy numbers corresponding to "Medium" and "Good", respectively.
Then d= m_{vi} - m_i= - 0.25; but 0.42 ∈ [m_i + (3/10)*(-0.25), m_i + (7/10)*(-0.25)] and thus the algorithm gives the linguistic modification associated with this case: [0.33, 0.42, 0.61] ≈ "A lot more than Medium (but less than good)".

3.2. *A Type-2 Resemblance Index*

Let us consider now a similarity index between two type-2 fuzzy sets A and B:

$$\delta(A,B) = 1 - \frac{1}{(m-1)*n} \frac{\sum_{i=1}^{n} | P(\lambda_{iA}) - P(\lambda_{iB}) |}{n_c},$$

where:
- *m* is the number of linguistic terms (both basic and generated by $ApprLing_k$)
- *n* is the number of basic terms
- λ^h_{iA} and λ^h_{iB} are the linguistic labels associated with the element p^i_h ,respectively, in A and B
- n_c = min(*a'*, *b'*) where *a'* and *b'* are the numbers of crisp non-empty partitions in A and B.
- P: $All\_Terms \rightarrow N$, $P(\lambda^h_i) = i$, $\forall i \in \{1, \ldots, n+k*(n-1)\}$
  - *k* is the number of intermediate labels of $ApprLing_k$
  - *All_Terms* is the set of involved linguistic terms, both basic terms and generated ones;it associates with a term λ^h_i its position in the ordered set of the terms generated by $ApprLing_k$

It can be easily shown that δ(A, B) ∈ [0,1], δ(A, A) = 1 and δ(A, B)= δ(B, A).

The maximum value of $\frac{\sum_{i=1}^{n} | P(\lambda_{iA}) - P(\lambda_{iB}) |}{n_c}$ is just (m-1)*n.

Example 3: Given two type-2 fuzzy sets:
A = vi/{c, d} + si/{a, e}+li/ b and B = vi/{a, c} +fi /d + (ai)/ b + li/e,
with n = 6 (see Table 1), k = 3, m = (n-1)*k + n = 21, an so (m - 1) * n = 120, n_c = 3, one has:

	a	***b***	***c***	***d***	***e***
A	si	li	vi	vi	si
B	vi	ai	vi	fi	li
Σ	***12***	***11***	***0***	***8***	***4***

where *12* corresponds to the number of terms between "*Sufficiently Interested*" and "*Very Interested*", associated with the element ***a*** in A and B, respectively; *11* corresponds to the number of terms between "*Low Interested*" and "*Almost Interested"* associated with the element *b* in A and B respectively, and the other values of are calculated in the same way. From these values, *δ(A, B) = 1 – (1/120)* * (12 + 11 + 0 + 8 + 4) */3 = 0.903.*

Consider another example: given the following type-2 fuzzy sets: C = vi/{a, b, c, d, e}, D = un/{a, b, c, d, e} with n=6 (see Tab. 1), k = 3, m = (n - 1) * k + n = 21, and so (m - 1) * n = =120, nc = 1, one has:

	a	***b***	***c***	***d***	***e***
C	vi	vi	vi	vi	vi
D	un	un	un	un	un
Σ	22	22	22	22	22

Thus Σ = 22*5 = 120, and *δ(A, B) = 0.*

3.3. A fuzzy measure for inequality

Let it be **Tr** the class of totally ordered triangular fuzzy numbers on [0,1]. The following operation **¤: Tr^2 → Tr**, is used in order to obtain an assessment of the inequality between two triangular numbers.

Given $\alpha=[a_1,b_1,c_1]$, $\beta=[a_2,b_2,c_2]$, the number $[a',b',c'] = [a_1,b_1,c_1]$ ¤ $[a_2,b_2,c_2]$ is so obtained:

b' = $| b_1 - b_2 |$;
a' = max $\{ 0 ; b' - (|a_1 - a_2|)/2 \}$
c' = min $\{ 1 ; b' + (|c_1 - c_2|)/2\}$

This operation, given two triangular numbers, computes a sort of *triangular distance*, that is a triangular number that expresses how they are faraway each other. It is easy to show that this operation is *Reflexive*: α ¤ α = [0, 0, 0] and *Symmetric*: α ¤ $\beta = \beta$ ¤ α .

Example 4: By using the lv *Interest* and the linguistic terms/triangular fuzzy numbers of *Example 1*, it is possible to see that the operation on two following numbers gives always the same result: [0.2 , 0.4 , 0.6] ¤ [0.4 , 0.6 , 0.8] = [0 , 0.2 , 0.4] while [0.2 , 0.4 , 0.6] ¤ [0.6 , 0.8 , 1] = [0.2 , 0.4 , 0.6] (the distance increases); other examples are: [0.8 , 1 , 1] ¤ [0 , 0 , 0.2] = [0.9 , 1 , 1] and [0.4 , 0.6 , 0.8] ¤ [0.4 , 0.6 , 0.8] = [0, 0, 0].

4. LINGUISTIC ATTRIBUTES AND USER PROFILES REPRESENTATION

As said before, the approach is based on the search of a matching between the metadata assigned to each page and that used to describe the user profile.

So one needs to single out a linguistic variable, a set of linguistic attributes and terms on it and a set of corresponding triangular fuzzy numbers. With this elements the author of the page can express by words all the features of a page and, on the other side, the final user can fill in easily his/her linguistic profile.

Through a suitable choice of preferences applied to the linguistic variable *Interest (or others of one's choice)*, it is possible both to assign a type-2 fuzzy set describing adequately the contents of a web page and to fully represent the user profile. So the author of a web page, can use a type-2 fuzzy set (in a simple linguistic way) to make explicit the contents, the level of widening involved and the degree of satisfaction of the page with respect to some user preferences. In the following a web page is defined as a couple (w, s_2): it represents the document contents and its features.

In a similar way, the final user of the information can express his/her interests to the search engine by answering with linguistic terms to a short questionnaire, obtaining a type-2 fuzzy set A(w) that represents his/her profile.

4.1. *Formal definition of a document*

A *Web Document* is defined as a couple (w, s_2), where *w* represents the contents, and s_2 is a type-2 fuzzy set: in such way one captures the imprecision of the linguistic description of the features of the document given by the author. In fact a metadata attribution of this kind (that uses of linguistic variables) allows to give more expressivity and to add more information to the contents.

Let us consider the following elements:

P = {p_1, …, p_n}, a finite crisp set of *features*, as for example, *contents area, addressing people, length, form, difficulty* and so on.

Π(P) = {s_{p1}, …, s_{pk}} a set of classical partitions on P;

W: a set of documents on the web;

Tr = {α_1, …, α_m} a set of *totally ordered* triangular fuzzy numbers on [0,1];

V_I: the linguistic variable "*Interest*";

$T(V_I)$ = {λ_1, …, λ_m} a set of linguistic terms on V_I;

V_C: the linguistic variable "*Compatibility*";

$T(V_C)$ = {γ_1, …, γ_n} a set of linguistic terms on V_C;

M: a semantic rule that assigns to each linguistic term $t \in T(V_I) \cup T(V_C)$ its meaning, i.e., M: $T(V_I) \cup T(V_C) \rightarrow$ Tr.

Using all these elements, one can give the following

Definition: Given (w, s_2), its metadata is the following:

$s_2(w) = \Sigma\ \alpha_i / s_{pi}$.

The fuzzy linguistic description of $s_2(w)$ is given by:

$FLD[s_2(w)] = \Sigma\ \lambda_i / s_{pi}$, where $M(\lambda_i) = \alpha_i$.

Using the last definition, a web document can be expressed as the couple (w, FLD[s_2(w)]) or for the sake of simplicity as (w, FLD[w]) that is a general definition of linguistic metadata, which does not set any limitation or restriction on the performable features as happens using XML; however, in the case study of Section 6, the set P is organized in classes that form a subset of the IEEE LOM basic metadata structure [10]: *General / Technical / Educational / Annotations / Classification*.

Example 1: Given the document w_f = *English_Medieval_Poetry.pdf*, by using the lv of Table 1 and the following classification items: Language, Poetry, Researcher, Medium length, English, History, University Student, Theoretical presentation, Deepening presentation, Political, Short, Scholar, Long, Scientific, a possible classification of w is the following:

FLD(w_3) = vi / {Language, Poetry, Researcher, Medium length } + i / { English , History, University Student, Theoretical presentation } + fi / { Deepening presentation, Political, Short} + li / { Scholar, Long} + ni /{Scientific}.

4.2. *The user profile*

An instance of search of a document (on the web, but also in a library, on newspapers or in others information files), is featured by a well-defined idea, on what he/she really needs in that moment, in terms of contents, form, complexity: this idea can be seen as a model of the "perfect document" the user has in mind.

In the development of this profile, each user owns not only his/her wanted features, but also a particular degree of preference about the characteristics of a document, and he/she knows perfectly how to linguistically express them. In this model the user profile can be represented as a set of *features* he/she is looking for in a document whose contents are explicitly given. The result is that one can use a user representation that is similar to the metadata associated with the document, defined on F$\subseteq$ P = {f_1, f_2, ..., f_k}, a finite crisp set of *features*.

Definition: The *User Profile* is a couple *Up* = (U_i, *Rc*), where:

$U_i = \Sigma\ \alpha_i / s_{fi}$, or linguistically as $LU_i = \Sigma\ \lambda_i / s_{fi}$, where $M(\lambda_i) = \alpha_i$, and *Rc* is a linguistic term of lv *Compatibility* that represents the *degree of suitability* of the selected documents according to the user.

Rc represents a sort of *tolerance limit of suitability* on the proposed results. The meanings of the values of *Rc* are fixed in Table 4.

Table 4. Linguistic terms and triangular fuzzy numbers of the lv *Compatibility*

Linguistic Variable: *Compatibility*				
Triangular Fuzzy Number	*Linguistic Term*		*Triangular Fuzzy Number*	*Linguistic Term*
[0.0, 0.0, 0.25]	High (h)		[0.5, 0.75, 1]	Sufficient (s)
[0.0, 0.25, 0.5]	Good (g)		[0.75, 1, 1]	Low (l)
[0.25, 0.5, 0.75]	Medium (m)			

This set of triangular fuzzy numbers is a fuzzy partition on [0, 1].

4.3. Using the attributes

The problem, now, is that the web author and the final user have to agree on the universe of *features*, in order to make comparable the two representations.

Let P be the set of features in the representation of a document and F that of the user profile, five cases can occur:

1. F = P : there is no problem in comparing the user profile and the document representation, because they contain information that can be qualitatively different, but given about the same features;

2. F $\subset$ P : in this case one can think that the user is not interested in some attributes, and so the compatibility can be compared on the basis of the features present in F;

3. P $\subset$ F: this means that the author of the document has left out some features; in this case the type-2 fuzzy set of document representation can be completed with all features present in the set F-P, by assigning them the special linguistic term NI (that stays for No Information and corresponds to the fuzzy number [0,0,0]. So the comparison will be made again on the features searched by the user and expressed in his/her profile.

4. P $\neq$ F and P$\cap$ F $\neq\varnothing$: this case (the most common in real situations) can be managed by considering only the features present in F in both user profile and document representation; this type-2 fuzzy set will be completed as said in case 3.

5. P $\cap$ F =$\varnothing$: No comparison is made.

5. THE SELECTION ALGORITHM

As said before, the aim of this approach is to add a second refinement on the result given by a Boolean search (*R*), by linguistically comparing the metadata of the documents and the active user profile. In this way, a subset ***SFS*** (Search Filtered Results) of *R* is defined.

Suppose that *Up* is the user profile, *Rc* a term of the lv *Interest* and $W = \{(w_j, FLD(w_j))\}$ the set of documents found through specific input parameters. The matching algorithm ***LCluster(w_j, FLD(w_j), Up, Rc)*** clusters and orders the documents in function of the compatibility of their attributes with what requested by the user in his/her profile. It uses the function ***FMatch(w_j, FLD(w_j), Up)***. The algorithm calculates $\mathbf{LDif_j}$ that is a term of the lv *Compatibility*. If $\mathbf{LDif_j} < \mathbf{Rc}$ then w_j in discarded, else it is put in the cluster containing all documents for which ***FMatch(w_j, FLD(w_j), Up, Rc)*** = $\mathbf{LDif_j}$. Each term so calculated defines a cluster. So the documents w_j for which ***FMatch(w_j, FLD(w_j), Up, Rc)*** = $\mathbf{LDif_j}$ will belong to that cluster.

The algorithm in pseudo-code is the following:

Step 1) Given a user profile Up_t, where F is the set of features chosen by the user, for each document doc_j:

If F = P then step 2 **else**:

i. Complete the type-2 fuzzy set of document representation by using the special term NI for each missing feature;

ii. All the features of the document representation that are missing in the fuzzy set of the user profile become part of the set *Additional Info*$_{(j)}$. This set contains the information that is present in document metadata but that is not requested/specified by the user. This information will be shown in linguistic form to the user at the end of the algorithm, in order to give him/her "linguistic additional information" to understand if a document is what he/she is looking for.

Step 2)

FMatch *(w_j, FLD(w_j), Up)*

{ For each document w_j **in** R,

For each $p_i \in$ F **in the type-2 fuzzy set** *FLD(w_j)*

If the label $(\lambda_i)_{FLD(wj)} \neq (\lambda_i)_{(Up)}$

then $Dif_j = Dif_j + |(\alpha_i)_{\alpha i = M(\lambda i) \in FLD(wj)} \text{¤} (\alpha_i)_{\alpha i = M(\lambda i) \in Up}|$

$T\_Dif_i = Dif_j / |F|$ */*extends the average of triangular numbers */*

$LDif_i$ = *ApprLingk, d* (T_Dif_i , *CompatibilityLinguisticTerms*)

}

The algorithm uses a set of temporary variables Dif_j (initialized to zero), and the variables T_Dif_j and $LDif_j$, that express (linguistically and numerically) an *average compatibility* between the preferences of the user (declared in the User Profile), and the features of the document $\boldsymbol{w_j}$. So if the value of $\mathbf{T\_Dif_j}$ is *numerically small* this means that the document features and contents are near to what the user is looking for, and it will be approximated with a *high compatibility linguistic term*.

Now, using these results, the documents retrivied by the search can be filtered, presenting those satisfing the relation data of Step 3. Note that the result documents can be simply presented on the basis of the calculated linguistic compatibility $\mathbf{LDif_j}$.

Step 3)

If $LDif_j > Rc$ **then** the document w_j is introduced in *$LDif_i$-cluster.*

In this step the algorithm filters the documents on the basis of the their compatibility calculated before; so, for each compatibility level higher than the lv *Rc* chosen by the user, it creates a cluster just labeled with the compatibility level.

Step 4)

For each *$LDif_i$-cluster* **so that** *|$LDif_i$-cluster| > 1,*

For each document w_t **in** *$LDif_i$-cluster,*

$\delta_{(w_t)} = \delta(M(Up), s_2(w_t))$

Sort **each** document w_t **in** *$LDif_i$-cluster* **in function of** $\delta_{(w_t)}$

A sorting algorithm that orders the documents of the same cluster on the basis of the numerical value $\delta \in [0, 1]$ is utilized. If a cluster is populated only by one document, δ is not calculated.

So a refinement is done on the clustering, by computing the Similarity Index on the documents that belong to the same cluster: this index is calculated between a document and the user profile in order to associate a Similarity numerical value (between 0 and 1) with which it is possible to organize the documents linguistically grouped in SFS.

Step 5)

The user chooses if he/she wants to have *additional information* about the returned documents.

If no the resulting documents are returned to the user clustered and ordered

If yes the resulting documents are shown together with Additional Information

6. A CASE STUDY

This case study shows how the algorithm works when the set of preferences of the user profile perfectly matches that of documents' representation. Let us define: p_3: scientific contents, p_4: IA concepts, p_5: logic concepts, p_7: formality, p_8: technical language, p_9: student. As said, $\mathbf{F = P = \{ p_3, p_4, p_5, p_7, p_8, p_9\}}$. Then let us consider the triangular fuzzy numbers and linguistic terms of Table 1 and Table 2 for the lv *Interest* and lv *Compatibility*, respectively. As said, there is no need to complete the representation type-2 fuzzy sets of the documents, because **P=F** (case 1 in Section 4.3). In this case, the special term NI will not be used, so the total number of considered terms (m) is 21. Suppose that the user selects the following profile: $\mathbf{U_i}$ = vi/{p_3, p_4} + i/{p_8} + si/{p_7, p_9} + li/{p_5}, while ***Rc*** = *Sufficient*. Let us consider the documents of the following Table 5:

Table 5. An example of documents and their semantic information.

Document w_i	**FLD(w_i)**
w_1	vi/{p_4, p_9} + i/{p_5, p_7} + li/{p_3, p_8}
w_2	vi/{p_3, p_8} + fi/{p_4} + si/{p_5, p_7, p_9}
w_3	vi/{p_5, p_7, p_9} + ni/{p_3, p_4, p_8}
w_4	vi/{p_3, p_4, p_8} + fi/{p_9} + si/{p_5, p_7}
w_5	vi/{p_4} + i/{p_3, p_8} + fi/{p_7} + li/{p_5, p_9}

On these documents, the classification algorithm is applied as follows:

T_Dif_1 = ([0.6, 0.8, 0.9] + [0.0, 0.2, 0.3] + [0.6, 0.8, 0.9] + [0.2, 0.4, 0.6] + [0.4, 0.6, 0.8] + [0.4, 0.6, 0.8])/6 = [0.366, 0.566, 0.7] and so, using the algorithm $\boldsymbol{ApprLing_{k=3}}$ (see par. 2.3), applied on the linguistic terms defined in table 2, one has $LDif_1$ = "*Medium*". In the same way: $LDif_2$ = "*Very Good*", $LDif_3$ = "*Almost Sufficient*", $LDif_4$ = "*Very Good*", $LDif_5$ = "*Very Good*". So the set of "compatible" documents **RFS = {"Very Good"/{w_2, w_4, w_5}, "Medium"/w_1 },** whereas the document w_3 is excluded because its compatibility level is *Almost Sufficient*, less than the chosen Rc . Now the similarity indices are calculated as follows: $\delta(U_i, w_2)$ = 1-((0+8+4+0+4+0)/3)/(20*6) = 0,94667. In the same way: $\delta(U_i, w_4)$ = 0,96; $\delta(Ui, w_5)$ = 0,97, $\delta(U_i, w_1)$ = 0,8.

Finally the documents are organized in the cluster labeled as *Included Between Interested-Fully interested* as follows: $\boldsymbol{w_5, w_4, w_2}$; hence w_5 is the document nearest to user needs. Finally the ***ordered* SFS** presented in Table 6 is obtained.

Table 6. The final result of the method: ***Ordered RFS***.

Document-User Compatibility	**Documents**	***Document -User Similarity***
Very Good		
	w_5	0.97
	w_4	0.96
	w_2	0.94667
Medium		
	w_1	-

Thanks to the simplicity of the example, it is possible to compare the user profile with the documents, and it is simple to see that w_5 is effectively the document nearest to user preferences. In case $P \neq F$ the algorithm allows to take into account only the features present in F in both user profile and document representation.

6.1. Results obtained by modifying the user profile

Now, to verify the soundness of the method, let us modify the user profile step by step in order to draw a user nearer and nearer to the document w_1 (see user profile modification in Table 7), in order to consider how the results depend on these modifications.

Table 7. The modified user profiles.

UserProfile	**Representation**	**Modified features**
$\mathbf{U_i}$	vi/{p_3, p_4} + i/{p_8} + si/{p_7, p_9} + li/{p_5}	Original profile
$\boldsymbol{U_{i1}}$	vi/{p_4} + i/{p_8} + si/{p_7, p_9}+ li/{p_3, p_5}	*p3*
$\boldsymbol{U_{i2}}$	vi/{p_4} + i/{p_8, p_5} + si/{p_7, p_9}+ li/{p_3 }	*p5*
$\boldsymbol{U_{i3}}$	vi/{p_4} + i/{p_8, p_5, p_7} + si/{p_9}+ li/{p_3 }	*p7*
$\boldsymbol{U_{i4}}$	vi/{p_4} + i/{ p_5, p_7} + si/{p_9}+ li/{p_3, p_8}	*p8*
$\boldsymbol{U_{i5}}$	vi/{p_4, p_9} + i/{p_5, p_7} + li/{p_3, p_8}	*p9*

The results are presented in Table 8:

Table 8. Final results with user profile closer and closer to w_1.

User Profile	*Document-User Compatibility*	**Documents**	*Document -User Similarity*
$\boldsymbol{U_{i1}}$	*More than Good*	w_5	-
	Good	w_4	-
	Almost Good	w_2	-
	More than Medium	$\mathbf{w_1}$	-
	More than Sufficient	w_3	-
$\boldsymbol{U_{i2}}$	***Almost Good***	w_5 , w_4, $\mathbf{w_1}$	0.93 , 0.9112 , 0.9112
	Very Medium	w_2	-
	A lot more than Sufficient	w_3	-
$\boldsymbol{U_{i3}}$	***More than Good***	$\mathbf{w_1}$	-
	Almost Good	w_5	-
	A lot more than Medium	w_4	-
	More than Medium	w_2	-
	Medium	w_3	-
$\boldsymbol{U_{i4}}$	***Very Good***	$\mathbf{w_1}$	-
	A lot more than Medium	w_5	-
	More than Medium	w_3, w_4 , w_2	0.8667 , 0.8556 , 0.8445
$\boldsymbol{U_{i5}}$	***High***	$\mathbf{w_1}$	(1)
	A lot more than Medium	w_3	-
	More than Medium	w_5 , w_4	0.88334 , 0.8445
	Almost Medium	w_2	-

It is worth stressing that, by changing the term of one feature in the user profile, all results change. In particular, as expected, the importance of the document w_1 starts to grow up and it gradually reaches the maximum value when the user profile becomes exactly equal to w_1 representation. Moreover, by considering the results with respect to the other documents, it's clear how a document became more or less important in function of the modifications carried out on the profile.

7. CONCLUDING REMARKS

In this paper a fuzzy-based methodology for searching web documents has been illustrated. It involves the introduction, through type 2 fuzzy sets, of linguistic terms to enrich the documents metadata and to represent the user profile. Then an algorithm for comparing user profile/documents metadata and for clustering and ordering the results in function of user

needs is presented. Both metadata representation and algorithm introduced in this paper present several aspects deserving further investigation:

- A possible extension of the methodology concerns the introduction of a weighting function. In such way the final user could associate higher weights to the features he/she considers more important;
- The introduction of more Linguistic Variables to give more expressivity to the documents representations and to deal with the complexity of the user profile;
- The introduction of special labels that represent no information or not compatible to complete the matching, in order to tackle the problem of coherence between the attributes used for documents metadata and those for the user profile;;
- In some situations, it could be useful to present the rejected results of the search; the user, in fact, could be also interested in something different or even opposite to his profile to take general information on a context;
- Another possible extension concerns the introduction of grouped clustering, in which the selection is made not on the single attributes, but on several ones (e.g., contents, form and so on), or generic ones such as technical, educational, etc.
- An ambitious goal might be the automatic construction of the type-2 fuzzy set that describes the document, through an extension of the algorithm ***LClustering***. This could be accomplished through statistical analysis of the *key terms* present in a document.

8. ACKNOWLEDGEMENTS

The authors thank the referees for giving valuable suggestions to improve the quality of the paper.

9. REFERENCES

[1] Demartini G.: "From People to Entities: New Semantic Search Paradigms for the Web", in Studies on the Semantic Web, IOS Press, pp. 1-162, 2014
[2] Ding W., Liu Y., Zhang J.: "Chinese-keyword Fuzzy Search and Extraction over Encrypted Patent Documents", Proc. Knowledge Discovery and Information Retrieval, pp.168-176, 2015
[3] IEEE 1484.12.1-2002, Draft Standard for Learning Object Metadata, http://www.ieee.org,, 2002.
[4] IMS Learning Resource Metadata Information Model, Version 1.2.1 Final Specification, http://www.imsglobal.org/metadata, 2001.
[5] Khattak A.M., Mustafa J., Ahmed N., Latif K., Khan S.: "Intelligent Search in Digital Documents", Web Intelligence, pp.558-561, 2008
[6] Klusch M., Kapahnke P., Schulte S., Lécué F., Bernstein A.: "Semantic Web Service Search: A Brief Survey", Kunstliche Intelligenz 30(2), pp.139-147, 2016
[7] Patro S., Malhotra V.M., Johnson D.: "An Algorithm to Use Feedback on Viewed Documents to Improve Web Query - Enabling Naïve Searchers to Search the Web Smartly", Proc. Web Information Systems and Technologies (1), pp.287-294, 2006
[8] Qumsiyeh R., Ng Y. "Searching web documents using a summarization approach", Int.J. Web Information Systems 12(1), pp.83-101, 2016
[9] Saraçoglu R., Tütüncü K., Allahverdi N.: "A new approach on search for similar documents with multiple categories using fuzzy clustering", Expert Syst. Appl. 34(4), pp.2545-2554, 2008
[10] Sartori E., Velegrakis Y.,Guerra F.: "Entity-Based Keyword Search in Web Documents", Trans. Computational Collective Intelligence 21, pp.21-49, 2016
[11] Su F.,Xiao C.,, Gao C.,Gao Y.: "Adaptive method to support effective searching over large-scale web Documents", Proc. Fuzzy Systems and Knowledge Discovery, pp.2428-2432, 2010
[12] Ullah I., Khusro S.: "In Search of a Semantic Book Search Engine on the Web: Are We There Yet ?", Proc. Computer Science On-line Conference (1), pp.347-357, 2016
[13] Yao Z., Wang B.: "Using section-semantic relation structures to enhance the performance of Web search", in Proc. Database and Expert Systems Applications, London, pp. 512-516, 2000
[14] Weiland L., Scherp A.: "A Novel Approach for Semantics-Enabled Search of Multimedia Documents on the Web", in Multimedia Modeling, pp.450-61, 2014
[15] World Wide Web Consortium (W3C), Semantic Web, http://www.w3c.org,, 2001.
[16] Zadeh L. A., "The Concept of a Linguistic Variable and its Application to Approximate Reasoning- I, II, III", Information Sciences I 8 – II 8 – III 9, pp. 199-249; pp.301-357; pp. 43-80, 1970