Experiments on using fuzzy quantified sentences in adhoc retrieval

David E. Losada, Félix Díaz-Hermida, Alberto Bugarín and Senén Barro
Intelligent Systems Group
Department of Electronics & Computer Science
University of Santiago de Compostela
Santiago de Compostela, Spain
{dlosada, felixdh, alberto, senen}@dec.usc.es

ABSTRACT

In this work we implement and evaluate a fuzzy approach to Information Retrieval whose query language incorporates fuzzy quantifiers. Fuzzy quantified sentences are suitable for imposing additional restrictions in the retrieval process which are not typical in classic information retrieval. Moreover, fuzzy quantifiers can be implemented in different relaxed ways leading to a wide range of methods for combining query terms. The large-scale evaluation conducted here shows clearly the practical benefits obtained in terms of retrieval performance. These empirical results strengthen previous theoretical works that already advanced the adequacy of fuzzy quantifiers for modeling information needs.

Categories and Subject Descriptors

H.3 [**Information Systems**]: Information Storage and Retrieval; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*

General Terms

Theory, Experimentation

Keywords

Information Retrieval, Fuzzy quantifiers

1. INTRODUCTION

The notion of vagueness shows up in many information retrieval (IR) tasks. Because fuzzy set theory supplies formal tools to handle approximate or vague notions, many researchers have devoted their efforts to the definition of fuzzy models for IR[7, 5].

Unfortunately, the application of fuzzy set theory for IR is not mature yet. Fuzzy techniques are not standard in IR applications and their actual role within the field is at least controversial. This is mainly because of the lack of extensive testing with state-of-the-art evaluation methodologies [1]. Experimental results and comparative data is the essence of modern research in the field (especially after the emergence of the TREC forum in early nineties [21]) and theoretical works should address the experimental confirmation of their formal foundations. This will help to motivate the whole IR community in favor of theoretically strong IR subareas such as fuzzy IR or logic-based IR.

This work is focused on conducting experiments on a fuzzy model for IR whose query language incorporates fuzzy quantifiers defined following the theory of semi-fuzzy quantifiers. It is well known that Boolean languages are rather strict and users find it difficult to formulate her/his information needs into Boolean form. Fuzzy quantified sentences provide us with mechanisms able to capture human subjectivity. On the other hand, fuzzy quantifiers permit to implement a diversity of methods for combining query terms whereas the classic procedures for softening the basic Boolean connectives [19] are rather inflexible. These promising characteristics were already advanced in [4], where an extended query language containing linguistic quantifiers and an special connector dealing with primary and optional criteria was proposed. Nevertheless, the practical advantages obtained from quantified statements are still unclear because of the lack of reported experiments.

From a theoretical perspective, our work starts from past fuzzy approaches for IR [17, 13], where retrieval was naturally modeled in terms of fuzzy sets. We think that this is a good beginning to face the inclusion of fuzzy quantified sentences. We propose an extension of the model designed in [17] to incorporate fuzzy quantifiers. We have implemented and evaluated relaxed versions of linguistic quantifiers and we provide experimental evidence about the benefits of our fuzzy approach. In particular, we will show that fuzzy quantified sentences are suitable tools for implementing additional restrictions in the retrieval process which lead to important benefits in retrieval performance.

The remainder of the paper is organized as follows. Section 2 describes our fuzzy model for IR and section 3 presents the main experimental findings. In section 4 we offer some remarks. The paper ends with some conclusions.

2. A FUZZY APPROACH TO INFORMATION RETRIEVAL

SAC'04, March 14-17, 2004, Nicosia, Cyprus

Our work starts from the fuzzy set model for IR developed by Ogawa, Morita and Kobayashi [17]. These researchers got inspiration from traditional fuzzy IR proposals [13] for designing an extended Boolean model (EBM) which implements Boolean connectives through operations between fuzzy sets. This model, which will be referred to as OMK model, is the basic foundation of a number of fuzzy approaches for IR [1]. Before proceeding, we briefly introduce some fundamental concepts of fuzzy set theory. Then, we sketch the basic foundations of the OMK model and the last part of this section is dedicated to explain the extension that we propose for the OMK model.

2.1 Fuzzy set theory

Fuzzy set theory allows us to define sets whose boundaries are not well defined. Given a universe of discourse U, a fuzzy set A can be characterized by a membership function with the form: $\mu_A : U \rightarrow [0,1]$. For every element $u \in U$, $\mu_A(u)$ represents its degree of membership to the fuzzy set A, with 0 corresponding to no membership in the fuzzy set and 1 corresponding to full membership. Operations on fuzzy sets can be implemented in several ways. For instance, the complement of a fuzzy set A and the intersection and union of two fuzzy sets A and B are typically defined by the following membership functions: $\mu_{\overline{A}}(u) = 1 - \mu_A(u)$, $\mu_{A \cup B}(u) = max(\mu_A(u), \mu_B(u))$ and $\mu_{A \cap B}(u) = min(\mu_A(u), \mu_B(u))$.

Some additional notation will be also of help in the rest of this paper. By $\wp(U)$ we refer to the crisp powerset of U and $\widetilde{\wp}(U)$ stands for the fuzzy powerset of U, i.e. the set containing all the fuzzy sets that can be defined over U. Given the universe of discourse $U = \{u_1, u_2, \ldots, u_n\}$, a fuzzy set A constructed over U is usually denoted as:

$A = \{\mu_A(u_1)/u_1, \mu_A(u_2)/u_2, \ldots, \mu_A(u_n)/u_n\}$

The α-cut operation on a fuzzy set produces a crisp set containing certain elements of the original fuzzy set. Formally,

DEFINITION 1 (α-CUT). *Given a fuzzy set $X \in \widetilde{\wp}(U)$ and $\alpha \in [0,1]$, the α-cut of level α of X is the crisp set $X_{\geq \alpha}$ defined as $X_{\geq \alpha} = \{u \in U : \mu_X(u) \geq \alpha\}$.*

EXAMPLE 1. *Let $X \in \widetilde{\wp}(U)$ be the fuzzy set $X = \{0.5/u_1, 0.7/u_2, 0.3/u_3, 0/u_4, 1/u_5\}$, then $X_{\geq 0.7} = \{u_2, u_5\}$.*

Fuzzy quantifiers are usually defined for relaxing the definition of crisp quantifiers. They have been widely applied for the evaluation of quantified statements, as in "approximately 80% of tall people are blonde" or "most modern cars have electric windows". Fuzzy quantifiers can be defined either directly or with the aid of semi-fuzzy quantifiers, which work on crisp sets [10, 11]. In this paper, we follow the latter approach and, therefore, we present first the definition of semi-fuzzy quantifier [11].

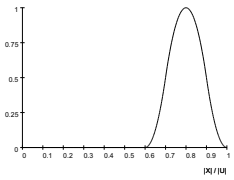
DEFINITION 2 (SEMI-FUZZY QUANTIFIER). *A unary semi-fuzzy quantifier Q on a base set $U \neq \varnothing$ is a mapping $Q : \wp(U) \longrightarrow [0,1]$ which maps each crisp set $X \in \wp(U)$ into a gradual result $Q(X) \in [0,1]$.*

A range of mathematical functions are usually applied in the definition of quantifiers. In the next example we show a definition and graphical description of a semi-fuzzy quantifier *approx_80%*. This is a relative quantifier because it is defined as a proportion over the base set U.

EXAMPLE 2. *approx_80% semi-fuzzy quantifier.* $approx\_80\% : \wp(U) \rightarrow [0,1]$

$$approx\_80\%(X) = \begin{cases} 0 & if\ \frac{|X|}{|U|} < 0.6 \\ 2\left(\frac{\left(\frac{|X|}{|U|} - 0.6\right)}{0.2}\right)^2 & if\ \frac{|X|}{|U|} \geq 0.6 \wedge \frac{|X|}{|U|} < 0.7 \\ 1 - 2\left(\frac{\left(\frac{|X|}{|U|} - 0.8\right)}{0.2}\right)^2 & if\ \frac{|X|}{|U|} \geq 0.7 \wedge \frac{|X|}{|U|} < 0.9 \\ 2\left(\frac{\left(\frac{|X|}{|U|} - 1\right)}{0.2}\right)^2 & if\ \frac{|X|}{|U|} \geq 0.9 \wedge \frac{|X|}{|U|} < 1 \\ 0 & otherwise \end{cases}$$

Graphically,



Example of use:
$U = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, u_9, u_{10}\}$
$X \subseteq \wp(U), X = \{u_1, u_2, u_3, u_4, u_5, u_8, u_{10}\}$
$approx\_80\%(X) = 1 - 2((0.7 - 0.8)/0.2)^2 = 0.5$

In order to apply the intuitions behind semi-fuzzy quantifiers for non-crisp sets, the concept of *fuzzy quantifier* is now introduced [11].

DEFINITION 3 (FUZZY QUANTIFIER). *A unary fuzzy quantifier $\widetilde{Q}$ on a base set $U \neq \varnothing$ is a mapping $\widetilde{Q} : \widetilde{\wp}(U) \longrightarrow [0,1]$ which maps each fuzzy set $X \in \widetilde{\wp}(U)$ into a gradual result $\widetilde{Q}(X) \in [0,1]$.*

The definition of fuzzy quantifiers for handling linguistic expressions has been widely dealt with in the literature [25, 23, 24, 6, 11, 8, 9]. Many proposals do not fulfill important properties, such as correct generalization and monotonicity, as reported in the reviews [10, 3], whilst the approaches suggested in [11, 9] exhibit a solid behaviour. In [11], fuzzy quantifiers are constructed from semi-fuzzy quantifiers through the so-called *quantifier fuzzification mechanisms*, which are mappings with domain in the universe of semi-fuzzy quantifiers and range in the universe of fuzzy quantifiers[1]:

$$F : (Q : \wp(U) \mapsto [0,1]) \mapsto \left(\widetilde{Q} : \widetilde{\wp}(U) \mapsto [0,1]\right)$$

In this way, the natural and clear semantics inherent to semi-fuzzy quantifiers can be applied when the sets considered are fuzzy instead of crisp. In [9] a quantifier fuzzification mechanism is defined from the notion of α-cut (definition 1). Formally, the Choquet integral [6] is applied as follows:

$$(F(Q))(X) = \int_0^1 Q\left((X)_{\geq \alpha}\right) d\alpha \qquad (1)$$

where $Q : \wp(U) \rightarrow [0,1]$ is a unary semi-fuzzy quantifier, $X \in \widetilde{\wp}(U)$ is a fuzzy set and $(X)_{\geq \alpha}$ is the α-cut of level α of X. Note that, as required by the semi-fuzzy quantifier

[1] Note that we use the unary version of the fuzzification mechanisms.

Q, $(X)_{\geq\alpha}$ is a crisp set. In this approach, $(X)_{\geq\alpha}$ are crisp representatives for the fuzzy set X and, roughly speaking, the integral over α-cuts in the interval $[0,1]$ averages out the values obtained after applying the semi-fuzzy quantifier to all the crisp representatives of X.

In this work, we will use the restriction to the unary case of this framework because this sort of quantifiers is expressive enough for the objectives pursued here[2]. Nevertheless, we plan to apply quantifiers of higher arity in the near future.

If the universe of discourse U is finite, the previous equation can be discretized as follows:

$$(F(Q))(X) = \sum_{i=0}^{m} Q\left((X)_{\geq\alpha_i}\right) \cdot (\alpha_i - \alpha_{i+1}) \qquad (2)$$

where $\alpha_0 = 1, \alpha_{m+1} = 0$ and $\alpha_1 > \ldots > \alpha_m$ denote the membership values in descending order of the elements in U to the fuzzy set X. The voting model interpretation of fuzzy sets [2] simply interprets the value $\alpha_i - \alpha_{i+1}$ as the probability that $(X)_{\geq\alpha_i}$ is selected as the crisp representative for the fuzzy set X. Therefore, the semi-fuzzy quantifier is applied for every crisp representative of X and those values are weighted by the probability of each crisp representative.

The next example illustrates the implementation of this process.

EXAMPLE 3. *Let $approx\_80\%$: $\wp(U) \rightarrow [0,1]$ be the semi-fuzzy quantifier depicted in example 2 and let X be the fuzzy set: $X = \{0.9/u_1, 0.8/u_2, 0.6/u_3, 0.5/u_4\}$. The next table presents the values produced by the semi-fuzzy quantifier $approx\_80\%$ at all α_i cut levels:*

	$(X)_{\geq\alpha_i}$	$approx\_80\%\left((X)_{\geq\alpha_i}\right)$
$\alpha_0 = 1$	$\emptyset$	$approx\_80\%(\emptyset) = 0$
$\alpha_1 = 0.9$	$\{u_1\}$	$approx\_80\%(\{u_1\}) = 0$
$\alpha_2 = 0.8$	$\{u_1, u_2\}$	$approx\_80\%(\{u_1, u_2\}) = 0$
$\alpha_3 = 0.6$	$\{u_1, u_2, u_3\}$	$approx\_80\%(\{u_1, u_2, u_3\}) = 0.875$
$\alpha_4 = 0.5$	$\{u_1, u_2, u_3, u_4\}$	$approx\_80\%(\{u_1, u_2, u_3, u_4\}) = 0$

Applying (2):

$(F(approx\_80\%))(X) = approx\_80\%((X)_{\geq 1}) \cdot (1 - 0.9) +$
$approx\_80\%((X)_{\geq 0.9}) \cdot (0.9 - 0.8) + approx\_80\%((X)_{\geq 0.8}) \cdot (0.8 - 0.6) +$
$approx\_80\%((X)_{\geq 0.6}) \cdot (0.6 - 0.5) + approx\_80\%((X)_{\geq 0.5}) \cdot (0.5 - 0) =$
$= 0 \cdot 0.1 + 0 \cdot 0.1 + 0 \cdot 0.2 + 0.875 \cdot 0.1 + 0 \cdot 0.5 = 0.0875$

This is a coherent result taking into account the definition of the fuzzy set X, where the degree of membership for the elements u_1 and u_2 is very high (0.9 and 0.8 respectively) whereas the degree of membership for u_3 and u_4 is medium (0.6 and 0.5 respectively). As a consequence, it is unlikely that 80% of the elements of the universe of discourse actually comply with the property modelled by X.

2.2 The OMK fuzzy model

Extended Boolean models (EBM) preserve the query structure inherent in the Boolean model while at the same time incorporate weighted terms into both queries and documents. The OMK model is an EBM whose basic foundations are as follows.

Each query term defines a fuzzy set in which each document has a degree of membership. Formally, each individual term, t_i, defines a fuzzy set whose universe of discourse is the set of all documents in the document base, D: $\mu_{t_i} : D \rightarrow [0,1]$. The larger this degree is, the more important the term is for characterizing the document's content. In [17], these values were estimated applying learning from co-occurrence data. Alternatively, these degrees can also be computed from classical IR weighting schemes, such as tf/idf [20].

Consider now a Boolean query involving terms and Boolean connectors AND, OR, NOT (e.g. t_1 AND t_2 OR NOT t_3). Since every t_i is a fuzzy set of documents, we can easily obtain a fuzzy set of documents which represents the whole query by operations between fuzzy sets. The Boolean connective AND is implemented by an intersection between fuzzy sets, the Boolean OR is implemented by a fuzzy union and so forth. Finally, a rank of documents can be straightforwardly obtained from the fuzzy set of documents representing the query.

A fundamental consideration around fuzzy approaches to IR stands on which fuzzy operators are suitable for implementing the Boolean operations. For instance, it is well known that the application of min and max operators for aggregating terms produces results which are counterintuitive. Lee et al. thoroughly studied a number of problems that arise from the use of such fuzzy operators and concluded that the so-called averaging operators are more appropriate for IR purposes [15, 14][3]. In the present work, our interest is not focused on revisiting the adequacy of different fuzzy operators for implementing Boolean connectives. On the contrary, we will show that, in practical situations, the introduction of fuzzy quantifiers makes that performance is not so dependent on the particular fuzzy operators used for implementing the basic Boolean connectives. In particular, following our experiments, fuzzy quantifiers appear as suitable tools for getting terms combined whereas basic Boolean operations behave well for aggregating different quantified statements.

2.3 Fuzzy quantifiers for IR

Fuzzy quantifiers provide a neat framework that allows the articulation of flexible expressions whose meaning is easily understood by users. Unfortunately, their application for IR is underexploited. A notable exception is the work developed by Bordogna and Pasi [4] in which linguistic quantifiers are employed to handle vague aggregation criteria in queries. Nevertheless, there is no clear evidence of the actual role of fuzzy linguistic quantifiers for enhancing retrieval systems because of the lack of experimentation.

Bordogna and Pasi apply ordered weighted averaging (OWA) operators [22] for implementing fuzzy quantifiers. We use the approach sketched in section 2.1 because: 1) it subsumes the fuzzy quantification model based on OWA (the OWA method is equivalent to the mechanism defined in equation (1) for increasing unary quantifiers [6, 8]) 2) it has been shown that OWA models [23, 24] do not comply with fundamental properties [10, 3] when dealing with n-ary quantifiers and 3) these problems are not present in the procedure summarized in equation (1). In section 4 we will enter into details about the limitations which have been described for OWA operators.

It is well known that users of retrieval systems find it difficult to make good use of Boolean connectives. The

[2]This is a general framework in which different models appear as particular cases [9]. In particular, for unary expressions, the instantiation produced by the general framework is equivalent to the approach based on Ordered Weighted Operators [22].

[3]In particular, the best-performing fuzzy averaging operators evaluated by Lee and other researchers work roughly the same than the well known P-Norm EBM measure [19].

distinction between AND and OR is frequently misunderstood, ORs are often used as exclusive ORs, etc. As a consequence, it is desirable to define flexible query languages able to include linguistic expressions whose meaning is easier to grasp. For instance, a query such as $at\_least\_2(car, engine, wheel, bumper)$ is much more compact and comprehensible than its Boolean counterpart.

Other advantage stands on the flexibility provided by quantifiers. In the framework of the EBM softened interpretations of AND and OR Boolean connectors were defined [19] but a choice of aggregation operators is not provided [4]. That is, only two basic methods can be used for combining operands (one for each Boolean connective). On the contrary, the application of fuzzy quantifiers leads to models which are flexible enough to handle a range of aggregation operators whose implementation can be relaxed in different ways.

2.4 Query language

In this section we present an extension of the Boolean query language that includes quantifiers.

Given a set of indexing terms $\{t_1, \ldots, t_m\}$ and a set of quantification symbols $\{Q_1, \ldots, Q_k\}$, query expressions are built as follows: a) any indexing term t_i belongs to the language, b) if e_1 belongs to the language then, NOT e_1 and (e_1) also belong to the language, c) if e_1 and e_2 belong to the language then, e_1 AND e_2 and e_1 OR e_2 also belong to the language and d) if $e_1, e_2, \ldots, e_n$ belong to the language then, $Q_i(e_1, e_2, \ldots, e_n)$ also belongs to the language, where Q_i is a quantification symbol.

The query language combines traditional Boolean expressions with quantified sentences. This means that users can articulate both Boolean queries (in which, as argued before, the range of aggregation methods is limited) and more evolved expressions involving quantifiers. The more quantifiers the language has, the more flexibility for the user when expressing her/his information needs.

EXAMPLE 4. *Given an alphabet of terms $\{a, b, c, d\}$ and the set of quantification symbols $\{at\_least\_2\}$ the expression d OR $at\_least\_2(a, b, NOT\ c)$ is a syntactically valid query expression.*

2.5 Semantics

Given a query complying with the rules expressed in the last section, we now address its semantics. Given a query expression q, we denote by $Sm(q)$ its associated fuzzy set of documents.

Every indexing term t_i is interpreted by a fuzzy set of documents, $Sm(t_i)$, whose membership function can be computed following classical IR weighting formulas, such as the popular tf/idf method [20]. Intuitively, t_i will be a good representative for documents with high degree of membership in $Sm(t_i)$ whereas t_i poorly represents the documents with low degree of membership in $Sm(t_i)$. Given a document d_j, we compute its degree of membership in the fuzzy set defined by a term t_i as:

$$\mu_{Sm(t_i)}(d_j) = \frac{f_{i,j}}{max_k f_{k,j}} * \frac{idf(t_i)}{max_l idf(t_l)} \qquad (3)$$

where $f_{i,j}$ is the raw frequency of term t_i in the document d_j and $max_k f_{k,j}$ is the maximum raw frequency computed over all terms which are mentioned by the document d_j. By $idf(t_i)$ we refer to a function computing an inverse document frequency factor. In this work we have used $idf(t_i) = log(max_l n_l / n_i)$, where n_i is the number of documents in which the term t_i appears and the maximum $max_l n_l$ is computed over all terms in the indexing vocabulary. The value $idf(t_i)$ is divided by $max_l idf(t_l)$, which is the maximum value of the function idf computed over all terms in the alphabet.

Note that $\mu_{Sm(t_i)}(d_j) \in [0, 1]$ because both the tf and the idf factors are divided by its maximum possible value. Of course, other membership functions might have been proposed (e.g. normalizing by document length) but to make extensive testing of the adequacy of different weighting schemes within the fuzzy model is out of the scope of this work.

As argued in section 2.1, given the fuzzy sets defined from individual query terms, the fuzzy set representing a Boolean query can be straightforwardly obtained applying classical fuzzy operators. At this point we will not make a decision about which operators (min, max, product, etc.) will provide an implementation for every Boolean connective. On the contrary, we will revisit this issue when presenting our experiments and, for each individual test, we will report on the specific operator(s) applied.

The most interesting case concerns quantifiers. The rest of this section is dedicated to explain how to define the membership function for a quantified sentence with the form $Q(e_1, \ldots, e_r)$, where Q is a quantification symbol and each e_i is an expression of the query language (whose associated fuzzy set is $Sm(e_i)$). That is, given the fuzzy sets $Sm(e_1), \ldots, Sm(e_r)$ (representing for each component of the quantifier its importance with regard to every document in the collection), we have to define a method for combining them into a single fuzzy set of documents, $Sm(Q(e_1, \ldots, e_r))$, representing the quantified sentence as a whole.

Recall that semi-fuzzy quantifiers work on crisp sets and, hence, its direct application for IR would entail a binary notion of importance for terms in documents. For instance, we could apply the semi-fuzzy quantifier defined in example 2 to determine whether or not 80% of query terms are important in a given document but the main limitation is that both query terms and document terms would have to be stored in crisp sets and, as a consequence, only a binary notion of importance can be handled. As a result, we propose to apply fuzzy quantifiers (which, as argued in section 2.1, work on fuzzy sets) defined from semi-fuzzy quantifiers by a fuzzification process.

First, we associate a semi-fuzzy quantifier with every quantification symbol in the query language. For instance, we might include the quantification symbol $approx\_80\%$ in the query language which is connected to a semi-fuzzy quantifier similar to the one depicted in example 2. Although many times the name of the quantification symbol is the same as the name of its semi-fuzzy quantifier, both concepts should not be confused. Given a quantification syntactic symbol Q, by Q_s we refer to its associated semi-fuzzy quantifier. The expression $F(Q_s)$ stands for the fuzzy quantifier which is obtained from Q_s by a fuzzification process such as the one defined in equation (1).

Each individual document d_j has a degree of membership, $\mu_{Sm(e_i)}(d_j)$, in the fuzzy set $Sm(e_i)$ defined by each component of the quantifier. For instance, if each component e_i is simply an indexing term, the values $\mu_{Sm(e_1)}(d_j)$, $\ldots$, $\mu_{Sm(e_r)}(d_j)$ would be obtained from tf/idf computations, as

detailed in equation (3). We can easily build a fuzzy set, C_{d_j}, induced by each document in which each component of the quantifier is weighted in terms of its importance for the document d_j:

$C_{d_j} = \{\mu_{Sm(e_1)}(d_j)/1, \mu_{Sm(e_2)}(d_j)/2, \ldots, \mu_{Sm(e_r)}(d_j)/r\}$

Note that every quantifier defines a universe of discourse composed of all quantifier components and every document builds a fuzzy set C_{d_j} over that universe of discourse.

To measure how much does d_j satisfy the requirements expressed by the quantifier $Q(e_1, \ldots, e_r)$ we can apply now the fuzzy quantifier $F(Q_s)$ over the fuzzy set C_{d_j}. Formally,

$$\mu_{Sm(Q(e_1,\ldots,e_r))}(d_j) = (F(Q_s))(C_{d_j}) \quad (4)$$

For instance, given an $approx\_80\%$ quantifier, documents whose degree of membership is high for approximately 80% of the quantifier components will likely receive a high degree of membership in the fuzzy set $Sm(Q(e_1, \ldots, e_r))$, which represents the whole quantified sentence. As a consequence, a query with the form $Q(e_1, \ldots, e_r)$ will likely retrieve those documents.

2.5.1 Example

We will present now a complete example which helps to illustrate the basic intuitions behind that process. Consider the query expression $at\_least\_2(a, b, c, d)$ and a document d_j whose degrees of membership in the fuzzy sets defined by each indexing term are: $\mu_{Sm_a(d_j)} = 0$, $\mu_{Sm_b(d_j)} = 0.15$, $\mu_{Sm_c(d_j)} = 0.2$ and $\mu_{Sm_d(d_j)} = 0.3$.

The fuzzy set induced by d_j from the components of the query expression is: $C_{d_j} = \{0/1, 0.15/2, 0.2/3, 0.3/4\}$.

The approach depicted in section 2.1 for fuzzy quantification works now on C_{d_j} as follows. Consider that we use the following semi-fuzzy quantifier for implementing the quantification symbol at_least_2.

$$at\_least\_2 : \wp(U) \rightarrow [0, 1]$$

$$at\_least\_2(X) = \begin{cases} 0 & \text{if } |X| < 2 \\ 1 & \text{otherwise} \end{cases}$$

We can directly estimate the degree of satisfaction of the quantified sentence as:

	$\left(C_{d_j}\right)_{\geq \alpha_i}$	$at\_least\_2\left(\left(C_{d_j}\right)_{\geq \alpha_i}\right)$
$\alpha_0 = 1$	$\emptyset$	$at\_least\_2(\emptyset) = 0$
$\alpha_1 = 0.3$	$\{d\}$	$at\_least\_2(\{d\}) = 0$
$\alpha_2 = 0.2$	$\{c, d\}$	$at\_least\_2(\{c, d\}) = 1$
$\alpha_3 = 0.15$	$\{b, c, d\}$	$at\_least\_2(\{b, c, d\}) = 1$
$\alpha_4 = 0$	$\{a, b, c, d\}$	$at\_least\_2(\{a, b, c, d\}) = 1$

And it follows that

$(F\ (at\_least\_2))\left(C_{d_j}\right) = 0 \cdot 0.7 + 0 \cdot 0.1 + 1 \cdot 0.05 + 1 \cdot 0.15 + 1 \cdot 0 = 0.2$

$\mu_{Sm(at\_least\_2(a,b,c,d))}(d_j) = 0.2$

That is, it is unlikely that at least two out of the four query terms are actually related to the document d_j.

The flexibility of fuzzy quantifiers permits to capture some intuitions which are typically left aside by IR models. For instance, given two documents d_1 and d_2 and a query q, imagine that d_1 is not related to any query term whereas d_2 is related to a single query term. The quantified statement depicted in the previous example would give the same score to both documents because the semi-quantifier at_least_2 produces the value 0 for all sets whose cardinality is less than 2. In this way, a document with a single matching term is considered as bad as a document with no matching terms. Think that this matching term might occur just by chance and, especially for a low number of matches, this approach is reasonable. As it will be shown in the next sections, this kind of expressions is also beneficial in terms of retrieval performance.

3. EXPERIMENTS

We evaluated the ability of this fuzzy approach to improve search effectiveness for the adhoc retrieval task. Our experiments were conducted on the Wall Street journal (WSJ) corpora from the TREC collection. This dataset contains about 173000 news articles spread over six years (total size: 524 Mb). A total of 50 topics were selected from TREC experiments (topics #151-#200 from TREC-3 adhoc retrieval task [12]). We used a stoplist of 571 words and Porter's stemmer [18]. The inverted file was built with the aid of GNU mifluz [16], which supplies a C++ library to build and query a full text inverted index.

For a baseline, we implemented a linear fuzzy quantified sentence Q_{lin}, whose associated semi-fuzzy quantifier is:

$$Q_{lin} : \wp(U) \rightarrow [0, 1], \quad Q_{lin}(X) = \frac{|X|}{|U|}$$

Note that this semi-fuzzy quantifier produces a result that grows linearly with the size of the input set X.

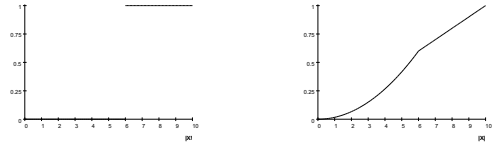
Terms are collected from the TREC topic and, after stopword and stemming, a fuzzy query with the form $Q_{lin}(t_1, \ldots, t_n)$ is built.

It can be easily proved that the ranking produced from such a query is equivalent to the one generated from the inner product matching function in the vector-space model [20]. Specifically, if document vectors are weighted as in equation 3 and query vectors are binary, both approaches result in the same ranking of documents. This is a good property of our fuzzy approach because it can handle popular IR retrieval methods as particular cases.

The first pool of experiments considered only terms from the title of the topic. In order to check whether *non-linear* quantifiers are good in terms of retrieval performance, we implemented relaxed versions of *atleast* quantifiers. We now show a usual crisp implementation of an *atleast 6* quantifier (left-hand side) and our proposed relaxation (right-hand side).

EXAMPLE 5. $at\_least\_6 : \wp(U) \rightarrow [0, 1].\ at\_least\_6(X) =$

$$\begin{cases} 0 & if\ |X| < 6 \\ 1 & otherwise \end{cases} \qquad \begin{cases} (10/6) * (|X|/10)^2 & if\ |X| < 6 \\ |X|/10 & otherwise \end{cases}$$



Clearly, the typical atleast implementation is too rigid to be applied in IR. For instance, it is not fair to consider that a document matching 10 query terms is as good as one matching only 6 terms[4]. Moreover, it is too rigid to consider a document that matches 0 query terms is as bad as

[4]Indeed, analogous considerations for Boolean conjunctions and disjunctions provoked that Lee and other researchers were rather critic with the application of fuzzy min and max operators [15, 14].

Recall	Q_{lin}	$atleast\_2$	$atleast\_3$	$atleast\_8$
0.00	0.5979	0.6165	0.6329	0.6597
0.10	0.4600	0.4776	0.4905	0.5037
0.20	0.3777	0.3997	0.4203	0.4253
0.30	0.3092	0.3336	0.3454	0.3483
0.40	0.2430	0.2680	0.2751	0.2784
0.50	0.1689	0.2121	0.2191	0.2226
0.60	0.1302	0.1592	0.1704	0.1770
0.70	0.0853	0.1100	0.1215	0.1273
0.80	0.0520	0.0734	0.0855	0.0892
0.90	0.0248	0.0428	0.0467	0.0495
1.00	0.0034	0.0070	0.0107	0.0105
Avg.prec.	0.2035	0.2241	0.2362	0.2411
% change		+10.12%	+16.07%	+18.5%

Table 1: Effect of simple atleast queries on retrieval performance

one matching 5 query terms. We believe that the intuitions behind atleast quantifiers can be good for retrieval purposes if implemented in a relaxed form. In particular, we propose an intermediate implementation which is between a classical atleast and a linear implementation (which is typical in popular IR matching functions, as shown above). It is not strange that unrelevant documents match a few query terms simply by chance. To minimize this problem our relaxed formulation makes that documents matching few terms (less than 6 for the example depicted above) receive a lower score compared to an alternative linear implementation. On the other hand, unlike the rigid atleast implementation, documents matching many terms (more than 6 for the example) receive a score that grows linearly with the number of those terms. Of course, a number of relaxed formulations may be proposed for every fuzzy quantifier. Nevertheless, we just want to advance the practical benefits that IR systems might get from flexible query languages and to make extensive testing on different relaxations is out of the scope of this work.

The first results are shown in table 1. We ran experiments for the baseline ($Q_{lin}(t_1, t_2, \ldots, t_n)$) and for several atleast formulations. All atleast quantifiers were implemented in the relaxed way shown in example 5 (right-hand side). Although topic titles consist typically of a very small number of terms, the outcome of these experiments clearly shows that flexible query formulations can lead to significant improvements in retrieval performance. There is a steady increment of performance over all recall levels and for $atleast\_x$ with $x \geq 8$ the performance values become stabilized.

Subsequent experiments made use of all topic subfields (Title, Description & Narrative). Several strategies were tested in order to produce fuzzy queries from topics. In all cases, every subfield is used for generating a single fuzzy quantifier and the fuzzy query is the conjunction of these quantifiers. Figure 1 depicts an example. We use the symbol $\wedge$ to refer to the Boolean AND connective. This simple method allows to obtain fuzzy representations from TREC topics in an automatic way. This advances that fuzzy query languages might be adequate not only to assist users when formulating their information needs but also to transform textual queries into fuzzy expressions.

We tried out varied combinations of atleast and linear quantifiers. For implementing the conjunction connective we applied both the fuzzy min operator and the product operator. The product operator outperforms clearly the min operator. Performance results (product operator) are summarized in table 2. The combination of linear quantifiers is clearly inferior to the combination of at_least_x quantifiers. There is a progressive improvement in retrieval perfor-

TREC topic:
<title> Topic: Vitamins - The Cure for or Cause of Human Ailments
<desc> Description:
Document will identify vitamins that have contributed to the cure for human diseases or ailments or documents will identify vitamins that have caused health problems in humans.
<narr> Narrative:
A relevant document will provide information indicating that vitamins may help to prevent or cure human ailments. Information indicating that vitamins may cause health problems in humans is also relevant. A document that makes a general reference to vitamins such as "good for your health" or "having nutritional value" is not relevant. Information about research being conducted without results would not be relevant. References to derivatives of vitamins are to be treated as the vitamin.
Fuzzy query:
atleast_3(vitamin,cure,caus,human,ailment) $\wedge$ atleast_3(document,identifi, vitamin,contribut,cure,human,diseas,ailment,caus,health,problem) $\wedge$ atleast_3(relevant,document,provid,inform,indic,vitamin,prevent,cure, human,ailment,caus,health,problem,make,gener,refer,good, nutrit,research,conduct,result,deriv,treat)

Figure 1: Fuzzy query from a TREC topic

	Q_{lin}(title terms) $\wedge$ Q_{lin}(desc terms) $\wedge$ Q_{lin}(narr terms)	at_least_x(title terms) $\wedge$ at_least_x(desc terms) $\wedge$ at least x(narr terms)		
Recall		$x = 2$	$x = 3$	$x = 8$
0.00	0.7277	0.7473	0.7311	0.7664
0.10	0.5513	0.5524	0.5576	0.5991
0.20	0.4610	0.4665	0.4711	0.4769
0.30	0.3608	0.3802	0.3869	0.3983
0.40	0.2915	0.3142	0.3133	0.3172
0.50	0.2428	0.2684	0.2638	0.2660
0.60	0.1857	0.2069	0.2111	0.2136
0.70	0.1160	0.1407	0.1496	0.1561
0.80	0.0720	0.0882	0.0932	0.1014
0.90	0.0431	0.0522	0.0559	0.0634
1.00	0.0067	0.0089	0.0126	0.0157
Avg.prec.	0.2572	0.2722	0.2760	0.2849
% change		+5.8%	+7.3%	+10.8%

Table 2: Conjunctions between quantifiers

mance as the value of x grows from 2 to 8. The performance ratios become stabilized for values of x around 8. It is important to emphasize that a combination of linear quantifiers is not a common characteristic of popular IR approaches, where a single linear operation is usually applied over all topic terms. As a consequence, the comparison presented in table 2 aims at checking the effect of atleast quantifiers vs linear quantifiers within our fuzzy approach and, later on, we will compare our best results with a classic approach in which a linear quantifier is applied over all topic terms.

We also ran some experiments using averaging-like operators (such as the ones tested by Lee and others in [15, 14]) for implementing the Boolean conjunction but their results were worse. This might indicate that, although T-norm operators (e.g. min and product) worked bad for combining terms within conjunctive Boolean representations [15, 14], they could play an important role in combining more expressive query components, such as quantifiers.

To understand the actual benefits obtained from such expressive queries, we conducted an additional baseline experiment in which all terms from all topic subfields are collected into a single linear quantifier (recall that this approach is equivalent to the popular vector-space model with the inner product matching function). The results obtained are compared to the previous best results in table 3. The application of relaxed non-linear quantifiers leads to very significant improvements in retrieval performance. Clearly, a linear strategy involving all topic terms is not the most appropriate way to retrieve documents. On the other hand, expressive query languages provide us with tools to capture topic's contents in a better way. In particular, our evaluation shows clearly that non-linear fuzzy quantifiers are appropriate for enhancing search effectiveness. For example, atleast quantifiers appear as powerful tools to establish additional

Recall	Q_{lin} (title, desc & narr terms)	atleast_8(title terms) ∧ atleast_8(desc terms) ∧ atleast_8(narr terms)
0.00	0.6354	0.7664
0.10	0.4059	0.5991
0.20	0.3188	0.4769
0.30	0.2382	0.3983
0.40	0.1907	0.3172
0.50	0.1383	0.2660
0.60	0.0885	0.2136
0.70	0.0530	0.1561
0.80	0.0320	0.1014
0.90	0.0158	0.0634
1.00	0.0019	0.0157
Avg.prec.	0.1697	0.2849
% change		+67.9%

Table 3: Linear quantifier query vs more evolved query

requirements for a document to be retrieved. Although the combination of linear quantifiers (e.g. table 2, col. 2) outperforms significantly the single linear quantifier approach (table 3, col. 2), it is still clear that a Boolean query language with linear quantifiers is not enough because further benefits are obtained when atleast quantifiers are applied (e.g. table 2, cols. 3-5).

Of course, many combinations of different quantifiers might have been tested. For instance, we only experimented on atleast quantifiers while many other fuzzy quantifiers exist on the literature. We found interesting properties of atleast quantifiers for adhoc retrieval but other classes of quantified sentences might be useful for this or other IR tasks. To produce a detailed report on different quantifiers and combinations between them was not an objective in this paper. Our main motivation was simply to check whether fuzzy quantifiers can work in realistic scenarios and to show some examples within an experimental framework. We believe that this objective was achieved because the evaluation presented concludes not only that the fuzzy approach is operative but also that it can produce important benefits in retrieval performance.

4. REMARKS

In previous sections we have already advanced that the formulation proposed here is equivalent to the OWA case for monotonic unary expressions. This means that the advantages shown empirically can be directly extrapolated to OWA-based approaches such as the one designed in [4]. We believe that this is a good property of our work because the evaluation results apply not only for our particular scenario but for other well-known proposals whose practical behaviour for large document collections was unclear.

We decided to skip OWA operators because, otherwise, the approach is less general and it should be restricted to unary quantification expressions. We offer now additional details about these problems and we sketch their implications in the context of IR. A thorough comparative between different fuzzy operators can be found in [10, 3].

One of the major drawbacks of OWA's method is its non-monotonic behaviour for propositions involving two properties [3]. This means that, given two quantifiers Q_1, Q_2 such that Q_1 is more specific than Q_2[5], it is not assured that the application of the quantifiers for handling a quantified proposition maintains specificity. This is due to the assumption that any quantifier is a specific case of OWA interpolation between two extreme cases: the existential quantifier and the universal quantifier. Let us illustrate this with an example. Consider two quantifiers $at\_least\_60\%$ and $at\_least\_80\%$ and two fuzzy sets of individuals representing the properties of being blonde and tall, respectively. Obviously, $at\_least\_80\%$ should be more specific than $at\_least\_60\%$. Unfortunately, the evaluation of a expression such as $at\_least\_80\%$ *blondes are tall* does not necessarily produce a value which is less or equal than the value obtained from $at\_least\_60\%$ *blondes are tall*. This means that, given two fuzzy sets *blondes* and *tall*, it is possible that these sets are better at satisfying the expression $at\_least\_80\%$ *blondes are tall* than satisfying the expression $at\_least\_60\%$ *blondes are tall*. This is clearly unacceptable.

This is also problematic for the application in IR. Imagine two quantifiers such that Q_1 is more specific than Q_2. This means that Q_1 is more restrictive than Q_2 (e.g. a crisp at_least_5 vs a crisp at_least_3). The application of these quantifiers for handling expressions with the form Q_i *A's are B's* cannot be faced using OWA operators. This is an important limitation because it prevents the extension of the fuzzy approach in a number of ways. For instance, expressions such as *most* t_i *are* t_k, where t_i and t_k are terms, can be used to determine whether or not most documents dealing with t_i are also related to t_k. In general, statements with this form involving several fuzzy sets are promising for enhancing the expressiveness of IR systems in different tasks.

5. CONCLUSIONS

The main contribution of this paper is of empirical nature. We have confirmed previous intuitions about the benefits of quantified sentences on retrieval performance for the case of large document collections. Fuzzy set theory appears as a powerful framework in which matching functions can be relaxed in different ways by means of quantified statements. Classical IR approaches tend to oversimplify the content of user information needs whereas flexible query languages allow to articulate more evolved queries. A novel contribution of this work stands on a report of the practical behaviour of different functions when combining terms in quantified statements. We obtained significant experimental evidence about the important benefits that retrieval applications might obtain from linguistic quantifiers. The inclusion of quantified statements in the query language permits to express additional constraints for the retrieval process leading to solid improvements in performance. Furthermore, we still do not know the actual performance limits of our approach because well-behaved normalization methods, such as those based on document length, were not tested.

It is important to observe that the benefits shown here empirically are not restricted to our particular fuzzy apparatus but also hold in the framework of some past fuzzy IR proposals.

Note also that we applied very simple methods for building automatically fuzzy queries from TREC topics. In the near future we plan to study other means for obtaining fuzzy statements from user queries. It is particularly interesting to design methods for building n-ary statements involving several fuzzy sets. Moreover, the extended query language

[5]Roughly speaking, if Q_1 is more specific than Q_2 then for all the elements of the domain of the quantifier the value produced by Q_1 is less or equal than the value produced by Q_2.

presented here is adequate to drive the construction of appropriate user interfaces to help users in the articulation of fuzzy queries.

6. ACKNOWLEDGMENTS

Authors wish to acknowledge support from the Spanish Ministry of Education and Culture through grants TIC2000-0873 and TIC2003-09400-C04-03. D. E. Losada is supported by the "Ramón y Cajal" R&D program, which is funded in part by "Ministerio de Ciencia y Tecnología" and in part by FEDER funds.

7. REFERENCES

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, ACM press, 1999.

[2] J.F. Baldwin, J. Lawry, and T.P. Martin. Mass assignment theory of the probability of fuzzy events. *Fuzzy Sets and Systems*, 83:353–367, 1996.

[3] S. Barro, A. Bugarín, P. Cariñena, and F. Díaz-Hermida. A framework for fuzzy quantification models analysis. *IEEE Transactions on Fuzzy Systems*, 11:89–99, 2003.

[4] G. Bordogna and G. Pasi. Linguistic aggregation operators of selection criteria in fuzzy information retrieval. *International Journal of Intelligent Systems*, 10(2):233–248, 1995.

[5] G. Bordogna and G. Pasi. Modeling vagueness in information retrieval. In F. Crestani M. Agosti and G. Pasi, editors, *Lectures on Information Retrieval (LNCS 1980)*. Springer Verlag, 2000.

[6] P. Bosc, L. Lietard, and O. Pivert. Quantified statements and database fuzzy querying. In P.Bosc and J. Kacprzyk, editors, *Fuzziness in Database Management Systems*, volume 5 of *Studies in Fuzziness*, pages 275–308. Physica-Verlag, 1995.

[7] F. Crestani and G. Pasi (eds). *Soft Computing in Information Retrieval: techniques and applications*. Studies in fuzziness and soft computing. Springer-Verlag, 2000.

[8] M. Delgado, D. Sánchez, and M. A. Vila. Fuzzy cardinality based evaluation of quantified sentences. *International Journal of Approximate Reasoning*, 23(1):23–66, 2000.

[9] F. Díaz-Hermida, A. Bugarín, P. Cariñena, and S. Barro. Voting model based evaluation of fuzzy quantified sentences: a general framework. *Fuzzy Sets and Systems*, 2003. Accepted.

[10] I. Glöckner. A framework for evaluating approaches to fuzzy quantification. Technical Report TR99-03, Universität Bielefeld, May 1999.

[11] I. Glöckner and A. Knoll. A formal theory of fuzzy natural language quantification and its role in granular computing. In W. Pedrycz, editor, *Granular computing: An emerging paradigm*, volume 70 of *Studies in Fuzziness and Soft Computing*, pages 215–256. Physica-Verlag, 2001.

[12] D. Harman. Overview of the third text retrieval conference. In *Proc. TREC-3, the 3rd text retrieval conference*, 1994.

[13] D.H. Kraft and Buell D.A. Fuzzy sets and generalized boolean retrieval systems. *International journal of man-machine studies*, 19:45–56, 1983.

[14] J. H. Lee. Properties of extended boolean models in information retrieval. In *Proc. of SIGIR-94, the 17th ACM Conference on Research and Development in Information Retrieval*, Dublin, Ireland, July 1994.

[15] J. H. Lee, W. Y. Kim, and Y. J. Lee. On the evaluation of boolean operators in the extended boolean framework. In *Proc. of SIGIR-93, the 16th ACM Conference on Research and Development in Information Retrieval*, Pittsburgh, USA, 1993.

[16] GNU mifluz. http://www.gnu.org/software/mifluz.

[17] Y. Ogawa, T. Morita, and K. Kobayashi. A fuzzy document retrieval system using the keyword connection matrix and a learning method. *Fuzzy sets and systems*, 39:163–179, 1991.

[18] M.F. Porter. An algorithm for suffix stripping. In K.Sparck Jones and P.Willet, editors, *Readings in Information Retrieval*, pages 313–316. Morgan Kaufmann Publishers, 1997.

[19] G. Salton, E. A. Fox, and H. Wu. Extended boolean information retrieval. *Communications of the ACM*, 26(12):1022–1036, 1983.

[20] G. Salton and M.J. McGill. *Introduction to modern information retrieval*. McGraw-Hill, New York, 1983.

[21] TREC: The text retrieval conference. http://trec.nist.gov.

[22] R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man and Cybernetics*, 18(1):183–191, 1988.

[23] R.R. Yager. Connectives and quantifiers in fuzzy sets. *Fuzzy Sets and Systems*, 40:39–75, 1991.

[24] R.R. Yager. A general approach to rule aggregation in fuzzy logic control. *Applied Intelligence*, 2:333–351, 1992.

[25] L.A. Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Comp. and Machs. with Appls.*, 8:149–184, 1983.