Semi-fuzzy quantifiers for information retrieval

David E. Losada[1], Félix Díaz-Hermida[2], and Alberto Bugarín[1]

[1] Grupo de Sistemas Inteligentes, Departamento de Electrónica y Computación. Universidad de Santiago de Compostela
{dlosada,alberto}@dec.usc.es
[2] Departamento de Informática, Universidad de Oviedo
diazfelix@uniovi.es

Recent research on fuzzy quantification for information retrieval has proposed the application of semi-fuzzy quantifiers for improving query languages. Fuzzy quantified sentences are useful as they allow additional restrictions to be imposed on the retrieval process unlike more popular retrieval approaches, which lack the facility to accurately express information needs. For instance, fuzzy quantification supplies a variety of methods for combining query terms whereas extended boolean models can only handle extended boolean-like operators to connect query terms. Although some experiments validating these advantages have been reported in recent works, a comparison against state-of-the-art techniques has not been addressed. In this work we provide empirical evidence on the adequacy of fuzzy quantifiers to enhance information retrieval systems. We show that our fuzzy approach is competitive with respect to models such as the vector-space model with pivoted document-length normalization, which is at the heart of some high-performance web search systems. These empirical results strengthen previous theoretical works that suggested fuzzy quantification as an appropriate technique for modeling information needs. In this respect, we demonstrate here the connection between the retrieval framework based on the concept of semi-fuzzy quantifier and the seminal proposals for modeling linguistic statements through Ordered Weighted Averaging operators (OWA).

1 Introduction

Classical retrieval approaches are mainly guided by efficiency rather than expressiveness. This yields to Information Retrieval (IR) systems which retrieve documents very efficiently but their internal representations of documents and queries is simplistic. This is especially true for web retrieval engines, which deal with huge amounts of data and their response time is critical. Nevertheless, it is well known that users have often a vague idea of what they are looking for and, hence, the query language should supply adequate means to express her/his information need.

Boolean query languages were traditionally used in most early commercial systems but there exists much evidence to show that ordinary users are unable to master

the complications of boolean expressions to construct consistently effective search statements [24]. This provoked that a number of researchers have explored ways to incorporate some elements of the natural language into the query language. To this aim, fuzzy set theory and fuzzy quantifiers have been found useful [2, 3]. In particular, fuzzy quantifiers permit to implement a diversity of methods for combining query terms whereas the classic extended boolean methods [24] for softening the basic Boolean connectives are rather inflexible [2]. This is especially valuable for web search as it is well known that users are reluctant to supply many search terms and, thus, it is interesting to support different combinations of the query terms. Indeed, fuzzy linguistic modelling has been identified as a promising research topic for improving the query language of search engines [14]. Nevertheless, the benefits from fuzzy quantification have been traditionally shown through motivating examples in IR whose actual retrieval performance remained unclear. The absence of a proper evaluation, using large-scale data collections and following the well-established experimental methodology for IR, is an important weakness for these proposals.

A first step to augment the availabilty of quantitative empirical data for fuzzy quantification in IR was done in [19], where a query language expanded with quantified expressions was defined and evaluated empirically. This work stands on the concept of semi-fuzzy quantifier (SFQ) and quantifier fuzzification mechanism (QFM). To evaluate a given quantified statement, an appropriate SFQ is defined and a QFM is subsequently applied, yielding the final evaluation score.

In this paper, we extend the research on SFQ for IR in two different ways. First, we show that the framework based on SFQ is general and it handles seminal proposals [30] for applying Ordered Weighted Averaging operators (OWA) as particular cases. Second, the experimentation has been expanded. In particular, we compare here the retrieval performance of the fuzzy model with state-of-the-art IR matching functions. We show that the model is competitive with respect to high-performance extensions of the vector space model based on document length corrections (pivoted document length normalization [28]), which have recurrently appeared among the top performance systems in TREC Web track competitions [27, 13, 29]. This is a promising result which advances the adequacy of fuzzy linguistic quantifiers for enhancing search engines.

The remainder of the paper is organized as follows. Section 2 describes some related work and section 3 explains the fuzzy model for IR defined in [19]. Section 4 shows that the framework based on SFQ handles the OWA-based quantification as a particular case. The main experimental findings are reported in section 5. The paper ends with some conclusions and future lines of research.

2 Related Work

Fuzzy set theory has been applied to model flexible IR systems which can represent and interpret the vagueness typical of human communication and reasoning. Many fuzzy proposals have been proposed facing one or more of the different aspects

around the retrieval activity. Exhaustive surveys on fuzzy techniques in different IR subareas can be found in [6, 3].

In seminal fuzzy approaches for IR, retrieval was naturally modeled in terms of fuzzy sets [23, 15, 16, 21]. The intrisic limitations of the Boolean Model motivated the development of a series of studies aiming at extending the Boolean Model by means of fuzzy set theory. The Boolean Model was naturally extended by implementing boolean connectives through operations between fuzzy sets. Given a boolean expression, each individual query term can be interpreted as 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, and the membership function has the form: $\mu_{t_i} : D \rightarrow [0, 1]$. The larger this degree is, the more important the term is for characterizing the document's content. For instance, these values can be easily computed from popular IR heuristics, such as tf/idf [25]. Given a Boolean query involving terms and Boolean connectors AND, OR, NOT (e.g. t_1 AND t_2 OR NOT t_3) a fuzzy set of documents representing the query as a whole can be obtained 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.

These seminal proposals are in one way or another on the basis of many subsequent fuzzy approaches for IR. In particular, those works focused on extending query expressiveness further on boolean expressions are especially related to our research. In [2] an extended query language containing linguistic quantifiers was designed. The boolean connectives AND and OR were replaced by soft operators for aggregating the selection criteria. The linguistic quantifiers used as aggregation operators were defined by Ordered Weighted Averaging (OWA) operators [31]. The requirements of an information need are more easily and intuitively formulated using linguistic quantifiers, such as all, $at\ least\ k$, $about\ k$ and $most\ of$. Moreover, the operator $and\ possibly$ was defined to allow for a hierarchical aggregation of the selection criteria in order to express their priorities. This original proposal is very valuable as it anticipated the adequacy of fuzzy linguistic quantifiers for enhancing IR query languages. Nevertheless, the practical advantages obtained from such quantified statements remained unclear because of the lack of reported experiments.

In [19], a fuzzy IR model was proposed to handle queries as boolean combinations of atomic search units. These basic units can be either search terms or quantified expressions. Linguistic quantified expressions were implemented by means of semi-fuzzy quantifiers. Some experiments were reported showing that the approach based on SFQ is operative under realistic circumstances.

In this paper we extend the work developed in [19] at both the theoretical and experimental level. On one hand, we show explicitly the connection between the pioneering proposals on fuzzy quantification for IR [2] and the framework based on SFQ. On the other hand, we compare here the retrieval performance of the SFQ fuzzy model with high performance IR matching functions. This will show whether or not the SFQ approach is comparable to state-of-the-art IR methods.

3 Semi-fuzzy quantifiers for information retrieval

Before proceeding, we briefly review some basic concepts of fuzzy set theory. Next, the approach based on semi-fuzzy quantifiers proposed in [19] is reviewed.

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 discrete fuzzy set A constructed over U is usually denoted as: $A = \{\mu_A(u_1)/u_1, \mu_A(u_2)/u_2, \ldots, \mu_A(u_n)/u_n\}$

Fuzzy quantification is usually applied for relaxing the definition of crisp quantifiers. The evaluation of *unary* expressions such as "approximately 80% of people are tall" or "most cars are fast" is naturally handled through the concept of fuzzy quantifier[3]. Formally,

Definition 1 (fuzzy quantifier). *A unary fuzzy quantifier $\widetilde{Q}$ on a base set $U \neq \varnothing$ is a mapping $\widetilde{Q} : \widetilde{\wp}(U) \longrightarrow [0,1]$.*

For example, given the fuzzy set $X = \{0.2/u_1, 0.1/u_2, 0.3/u_3, 0.1/u_4\}$, modelling the degree of technical skill of four football players in a team, we can apply a quantifier of the kind most to determine whether or not most footballers are skillful. Of course, given the membership degrees of the elements in X, any coherent implementation of the most quantifier applied on X would lead to a low evaluation score.

The definition of fuzzy quantifiers for handling linguistic expressions has been widely dealt with in the literature [34, 31, 32, 5, 11, 7, 8]. Unfortunately, given a certain linguistic expression, it is often difficult to achieve consensus on a) the most appropriate mathematical definition for a given quantifier and b) the adequacy of a particular numerical value as the evaluation result for a fuzzy quantified sentence. This is especially problematic when linguistic expressions involve several fuzzy properties. To overcome this problem, some authors have proposed indirect definitions of fuzzy quantifiers through semi-fuzzy quantifiers [9, 11, 10]. A fuzzy quantifier can be defined from a semi-fuzzy quantifier through a so-called quantifier fuzzification mechanism (QFM). The motivation of this class of indirect definitions is that semi-fuzzy quantifiers (SFQ) are closer to the well-known crisp quantifiers and can be defined in a more natural and intuitive way. Formally,

[3] These expressions are called unary because each sentence involves a single vague property (tall in the first example and fast in the second one).

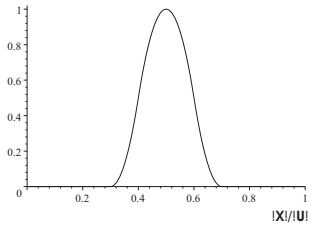
Definition 2 (semi-fuzzy quantifier). *A unary semi-fuzzy quantifier Q on a base set $U \neq \varnothing$ is a mapping $Q : \wp(U) \longrightarrow [0,1]$.*

In the next example we show a definition and graphical description of a *relative* semi-fuzzy quantifier $about\_half$ [4].

Example 1. $about\_half$ semi-fuzzy quantifier.
about_half : $\wp(U) \rightarrow [0,1]$

$$about\_half(X) = \begin{cases} 0 & \text{if } \frac{|X|}{|U|} < 0.3 \\ 2\left(\frac{\left(\frac{|X|}{|U|} - 0.3\right)}{0.2}\right)^2 & \text{if } \frac{|X|}{|U|} \geq 0.3 \wedge \frac{|X|}{|U|} < 0.4 \\ 1 - 2\left(\frac{\left(\frac{|X|}{|U|} - 0.5\right)}{0.2}\right)^2 & \text{if } \frac{|X|}{|U|} \geq 0.4 \wedge \frac{|X|}{|U|} < 0.6 \\ 2\left(\frac{\left(\frac{|X|}{|U|} - 0.7\right)}{0.2}\right)^2 & \text{if } \frac{|X|}{|U|} \geq 0.6 \wedge \frac{|X|}{|U|} < 0.7 \\ 0 & \text{otherwise} \end{cases}$$

Graphically,



Example of use:
Consider a universe of discourse composed of 10 individuals, $U = \{u_1, u_2, \ldots, u_{10}\}$. Imagine that X is a subset of U containing those individuals which are taller than 1.70m: $X = \{u_1, u_4, u_8, u_{10}\}$
The evaluation of the expression "about half of people are taller than 1.70m" produces the value: $about\_half(X) = 1 - 2((0.4 - 0.5)/0.2)^2 = 0.5$

Definition 3 (quantifier fuzzification mechanism). *A QFM is a mapping with domain in the universe of semi-fuzzy quantifiers and range in the universe of fuzzy quantifiers*[5]*:*

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

[4] This is a relative quantifier because it is defined as a proportion over the base set U
[5] Note that we use the unary version of the fuzzification mechanisms.

Different QFMs have been proposed in the literature [9, 10]. In the following we will focus on the QFM tested for IR in [19]. Further details on the properties of this QFM and a thorough analysis of its behaviour can be found in [8].

Since this QFM is based on the notion of α-cut, we first introduce the α-cut operation and, next, we depict the definition of the QFM.

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

Definition 4 (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 2. Let $X \in \widetilde{\wp}(U)$ be the fuzzy set $X = \{0.6/u_1, 0.2/u_2, 0.3/u_3, 0/u_4, 1/u_5\}$, then $X_{\geq 0.4} = \{u_1, u_5\}$.

In [19], the following quantifier fuzzification mechanism was applied for the basic IR retrieval task:

$$(F(Q))(X) = \int_0^1 Q\left((X)_{\geq\alpha}\right) d\alpha \tag{2}$$

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.

The crisp sets $(X)_{\geq\alpha}$ can be regarded as crisp representatives for the fuzzy set X. Roughly speaking, $\overline{2}$ averages out the values obtained after applying the semi-fuzzy quantifier to these crisp representatives of X. The original definition of this QFM can be found in [8].

If U is finite, expression (2) can be discretized as follows:

$$(F(Q))(X) = \sum_{i=0}^{m} Q\left((X)_{\geq\alpha_i}\right) \cdot (\alpha_i - \alpha_{i+1}) \tag{3}$$

where $\alpha_0 = 1, \alpha_{m+1} = 0$ and $\alpha_1 \geq \ldots \geq \alpha_m$ denote the membership values in descending order of the elements in U to the fuzzy set X.

Example 3. Imagine a quantified expression such as "about half of people are tall". Let $about\_half : \wp(U) \rightarrow [0,1]$ be the semi-fuzzy quantifier depicted in example 1 and let X be the fuzzy set: $X = \{0.9/u_1, 0.8/u_2, 0.1/u_3, 0/u_4\}$. The next table shows the values produced by the semi-fuzzy quantifier $about\_half$ at all α_i cut levels:

	$(X)_{\geq\alpha_i}$	$about\_half\left((X)_{\geq\alpha_i}\right)$
$\alpha_0 = 1$	$\emptyset$	$about\_half(\emptyset) = 0$
$\alpha_1 = 0.9$	$\{u_1\}$	$about\_half(\{u_1\}) = 0$
$\alpha_2 = 0.8$	$\{u_1, u_2\}$	$about\_half(\{u_1, u_2\}) = 1$
$\alpha_3 = 0.1$	$\{u_1, u_2, u_3\}$	$about\_half(\{u_1, u_2, u_3\}) = 0$
$\alpha_4 = 0$	$\{u_1, u_2, u_3, u_4\}$	$about\_half(\{u_1, u_2, u_3, u_4\}) = 0$

Applying (3):

$$\begin{aligned}(F\,(about\_half))\,(X) = {} & about\_half((X)_{\geq 1}) \cdot (1 - 0.9) + \\ & about\_half((X)_{\geq 0.9}) \cdot (0.9 - 0.8) + \\ & about\_half((X)_{\geq 0.8}) \cdot (0.8 - 0.1) + \\ & about\_half((X)_{\geq 0.1}) \cdot (0.1 - 0) + \\ & about\_half((X)_{\geq 0}) \cdot (0 - 0) \\ = {} & 0.7\end{aligned}$$

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 very low (0.1 and 0 respectively). As a consequence, it is likely that about half ot the individuals are actually tall.

3.1 Query language

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.

Example 4. Given an alphabet of terms $\{a, b, c, d\}$ and the set of quantification symbols $\{most\}$ the expression b AND $most(a, c,$ NOT $c)$ is a syntactically valid query expression.

The range of linguistic quantifiers available determines how flexible the query language is.

3.2 Semantics

Given a query expression q, its associated fuzzy set of documents is denoted by $Sm(q)$. 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 [25]. Given the fuzzy set defined by every individual query term, the fuzzy set representing a Boolean query can be directly obtained applying operations between fuzzy sets.

Given 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, we have to articulate a method for combining the fuzzy sets $Sm(e_1), \ldots, Sm(e_r)$ into a single fuzzy set of documents, $Sm(Q(e_1, \ldots, e_r))$, representing the quantified sentence as a whole.

First, we associate a semi-fuzzy quantifier with every quantification symbol in the query language. For instance, we might include the quantification symbol $about\_half$ in the query language which associated to a semi-fuzzy quantifier similar to the one depicted in example 1[6]. Given a quantification syntactic symbol Q, by Q_s we refer to its associated semi-fuzzy quantifier. Given a QFM, F, $F(Q_s)$ denotes the fuzzy quantifier obtained from Q_s by fuzzification.

Let d_j be a document and $Sm(e_i)$ the fuzzy sets induced by the components of the quantified expression, we can define the fuzzy set C_{d_j}, which represents how much d_j satisfies the individual components of the quantified statement:

$$C_{d_j} = \{\mu_{Sm(e_1)}(d_j)/1, \mu_{Sm(e_2)}(d_j)/2, \ldots, \mu_{Sm(e_r)}(d_j)/r\} \quad (4)$$

From these individual degrees of fulfilment, the expression $Q(e_1, \ldots, e_r)$ can be evaluated by means of the fuzzy quantifier $F(Q_s)$:

$$\mu_{Sm(Q(e_1,\ldots,e_r))}(d_j) = (F(Q_s))(C_{d_j}) \quad (5)$$

For instance, if Q_s is a semi-fuzzy quantifier $about\_half$ then a document will be assigned a high evaluation score if it has a high degree of membership for about half of the quantifier components and low degrees of membership for the rest of the components.

3.3 Example

Consider the query expression $at\_least\_3(a, b, c, d, e)$ and a document d_j whose degrees of membership in the fuzzy sets defined by each indexing term are: $\mu_{Sm_a(d_j)} = 0$, $\mu_{Sm_b(d_j)} = 0.15$, $\mu_{Sm_c(d_j)} = 0.2$, $\mu_{Sm_d(d_j)} = 0.3$ and $\mu_{Sm_e(d_j)} = 0.4$.

The fuzzy set induced by d_j from the components of the query expression is: $C_{d_j} = \{0/1, 0.15/2, 0.2/3, 0.3/4, 0.4/5\}$.

Consider that we use the following crisp semi-fuzzy quantifier for implementing the quantification symbol at_least_3.

$$at\_least\_3 : \wp(U) \rightarrow [0, 1]$$

$$at\_least\_3(X) = \begin{cases} 0 \text{ if } |X| < 3 \\ 1 \text{ otherwise} \end{cases}$$

Now, several crisp representatives of C_{d_j} are obtained from subsequent α-cuts and the semi-fuzzy quantifier at_least_3 is applied on every crisp representative:

[6] Although many times the name of the quantification symbol is the same as the name of the semi-fuzzy quantifier used to handle the linguistic expression, both concepts should not be confused.

	$\left(C_{d_j}\right)_{\geq\alpha_i}$	$at\_least\_3\left(\left(C_{d_j}\right)_{\geq\alpha_i}\right)$
$\alpha_0 = 1$	$\emptyset$	$at\_least\_3\left(\emptyset\right) = 0$
$\alpha_1 = 0.4$	$\{e\}$	$at\_least\_3\left(\{e\}\right) = 0$
$\alpha_2 = 0.3$	$\{d, e\}$	$at\_least\_3\left(\{d, e\}\right) = 0$
$\alpha_3 = 0.2$	$\{c, d, e\}$	$at\_least\_3\left(\{c, d, e\}\right) = 1$
$\alpha_4 = 0.15$	$\{b, c, d, e\}$	$at\_least\_3\left(\{b, c, d, e\}\right) = 1$
$\alpha_5 = 0$	$\{a, b, c, d, e\}$	$at\_least\_3\left(\{a, b, c, d, e\}\right) = 1$

And it follows that

$$\begin{aligned}\left(F\left(at\_least\_3\right)\right)\left(C_{d_j}\right) &= 0 \cdot 0.6 + 0 \cdot 0.1 + 0 \cdot 0.1 + 1 \cdot 0.05 + \\ &\quad +1 \cdot 0.15 + 1 \cdot 0 = 0.2 \\ \mu_{Sm(at\_least\_3(a,b,c,d,e))}(d_j) &= 0.2\end{aligned}$$

Indeed, it is unlikely that at least three out of the five query terms are actually related to document d_j because all query terms have low degrees of membership in C_{d_j}.

4 Semi-fuzzy quantifiers and OWA quantification

In [19], the modeling of linguistic quantifiers was approached by semi-fuzzy quantifiers and quantifier fuzzification mechanisms (equation (3)) because: 1) this approach subsumes the fuzzy quantification model based on OWA (the OWA method is equivalent to the mechanism defined in equation (3) for increasing unary quantifiers [5, 7]) and 2) it has been shown that OWA models [31, 32] do not comply with fundamental properties [9, 1] when dealing with n-ary quantifiers. These problems are not present in the SFQ-based approach defined in [8].

In this section, we enter into details about these issues and, in particular, we show how the implementation of linguistic quantifiers through OWA operators is equivalent to a particular case of the SFQ-based framework. This is a good property of the SFQ approach because seminal models of fuzzy quantification for IR [2], which are based on OWA operators, can be implemented and tested under the SFQ framework. Note that we refer here to the OWA-based unary quantification approach [33]. Although alternative OWA formulations have been proposed in the literature, a thorough study of the role of these alternatives for quantification is out of the scope of this work.

4.1 Linguistic quantification using OWA operators

OWA operators [30] are mean fuzzy operators whose results lie between those produced by a fuzzy MIN operator and those yielded by a fuzzy MAX operator.

An ordered weighted averaging (OWA) operator of dimension n is a non linear aggregation operator: OWA: $[0,1]^n \rightarrow [0,1]$ with a weighting vector $W = [w_1, w_2, \ldots, w_n]$ such that:
$\sum_{i=1}^{n} w_i = 1, w_i \in [0,1]$
and
OWA$(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} w_i \cdot Max_i(x_1, x_2, \ldots, x_n)$
where $Max_i(x_1, x_2, \ldots, x_n)$ is the i-th largest element across all the x_k, e.g. $Max_2(0.9, 0.6, 0.8)$ is 0.8.

The selection of particular weighting vectors W allows the modeling of different linguistic quantifiers (e.g. $at\ least$, $most\ of$, etc.).

Given a quantified expression $Q(e_1, \ldots, e_r)$ and a document d_j, we can apply OWA quantification for aggregating the importance weights for the selection conditions e_i. Without loss of generality, these weights will be denoted here as $\mu_{Sm(e_i)}(d_j)$. Formally, the evaluation score produced would be:

$$\text{OWA}_{op}(\mu_{Sm(e_1)}(d_j), \ldots, \mu_{Sm(e_r)}(d_j)) = \sum_{i=1}^{r} w_i \cdot Max_i(\mu_{Sm(e_1)}(d_j), \ldots, \mu_{Sm(e_r)}(d_j)) \tag{6}$$

where OWA_{op} is an OWA operator associated with the quantification symbol Q.

Following the modelling of linguistic quantifiers via OWA operators [2, 4], the vector weights w_i associated to the $\text{OWA}_{operator}$ operator are defined from a monotone non-decreasing relative fuzzy number $FN : [0,1] \rightarrow [0,1]$ as follows:

$$w_i = FN(i/r) - FN((i-1)/r), i : 1, \ldots, r \tag{7}$$

The fuzzy numbers used in the context of OWA quantification are *coherent*. This means that it is guaranteed that $FN(0) = 0$ and $FN(1) = 1$.

Without loss of generality, we can denote $Max_1(\mu_{Sm(e_1)}(d_j), \mu_{Sm(e_2)}(d_j), \ldots, \mu_{Sm(e_r)}(d_j))$ as α_1, $Max_2(\mu_{Sm(e_1)}(d_j), \mu_{Sm(e_2)}(d_j), \ldots, \mu_{Sm(e_r)}(d_j))$ as α_2, etc. and the evaluation value equals:

$$\sum_{i=1}^{r} w_i \cdot \alpha_i = \sum_{i=1}^{r} (FN(i/r) - FN((i-1)/r)) \cdot \alpha_i \tag{8}$$

This equation depicts the evaluation score produced by an OWA operator. In the next section we show that an equivalent result can be obtained within the SFQ framewok if particular semi-fuzzy quantifiers are selected.

4.2 Linguistic quantification using SFQ

Recall that, given a quantified expression $Q(e_1, \ldots, e_r)$ and a document d_j, the evaluation scored computed following the SFQ approach is:

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

A key component of this approach is the quantifier fuzzification mechanism F, whose discrete definition (equation 3) is repeated here for the sake of clarity:

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

where $\alpha_0 = 1, \alpha_{m+1} = 0$ and $\alpha_1 \geq \ldots \geq \alpha_m$ denote the membership values in descending order of the elements in the fuzzy set X.

Putting all together:

$$\mu_{Sm(Q(e_1,\ldots,e_r))}(d_j) = \sum_{i=0}^{r} Q_s\left(\left(C_{d_j}\right)_{\geq \alpha_i}\right) \cdot (\alpha_i - \alpha_{i+1}) \tag{9}$$

Without loss of generality, we will assume that the e_i terms are ordered in decreasing order of membership degrees in C_{d_j}, i.e. $\mu_{C_{d_j}}(e_1) = \alpha_1 \geq \mu_{C_{d_j}}(e_2) = \alpha_2 \ldots \geq \mu_{C_{d_j}}(e_r) = \alpha_r$. Note also that equation 9 stands on a sequence of successive α-cuts on the fuzzy set C_{d_j}. The first cut (α_0) is done at the membership level 1 and the last cut (α_r) is performed at the level 0. This means that the equation can be rewritten as:

$$\mu_{Sm(Q(e_1,\ldots,e_r))}(d_j) = \sum_{i=0}^{r} Q_s(CS_i) \cdot (\alpha_i - \alpha_{i+1}) \tag{10}$$

where $CS_0 = \emptyset$ and $CS_i = \{e_1, \ldots, e_i\}$, $i = 1, \ldots, r$.

The equation can be developed as:

$$\begin{aligned}
\mu_{Sm(Q(e_1,\ldots,e_r))}(d_j) &= \sum_{i=0}^{r} Q_s(CS_i) \cdot (\alpha_i - \alpha_{i+1}) \qquad (11)\\
&= Q_s(\emptyset) \cdot (1 - \alpha_1) + \\
&\quad Q_s(\{e_1\}) \cdot (\alpha_1 - \alpha_2) + \ldots + \\
&\quad Q_s(\{e_1, e_2, \ldots, e_r\}) \cdot \alpha_r \\
&= Q_s(\emptyset) + \alpha_1 \cdot (Q_s(\{e_1\}) - Q_s(\emptyset)) + \\
&\quad \alpha_2 \cdot (Q_s(\{e_1, e_2\}) - Q_s(\{e_1\})) + \ldots + \\
&\quad \alpha_r \cdot (Q_s(\{e_1, e_2, \ldots, e_r\}) - Q_s(\{e_1, e_2, \ldots, e_{r-1}\}))
\end{aligned}$$

The unary semi-fuzzy quantifier Q_s can be implemented by means of a fuzzy number as follows: $Q_s(CS_i) = FN(|CS_i|/r)$. Hence, the previous equation can be rewritten in the following way:

$$\mu_{Sm(Q(e_1,\ldots,e_r))}(d_j) = \sum_{i=0}^{r} Q_s\left(CS_i\right) \cdot (\alpha_i - \alpha_{i+1}) \tag{12}$$
$$= FN\left(0\right) + \alpha_1 \cdot (FN\left(1/r\right) - FN\left(0\right)) +$$
$$\alpha_2 \cdot (FN\left(2/r\right) - FN\left(1/r\right)) + \ldots +$$
$$\alpha_r \cdot (FN\left(1\right) - FN\left((r-1)/r\right))$$

It is straightforward that we can replicate the OWA-based evaluation (equation 8) if we select a SFQ whose associated fuzzy number is the same as the one used in the OWA equation. Note that, $FN(0) = 0$ provided that the fuzzy number is coherent.

4.3 Remarks

Given a query and a document d_j, the application of SFQ for IR proposed in [19] involves a single fuzzy set, C_{d_j}. In these cases, as shown in the last section, equivalent evaluation results can be obtained by an alternative OWA formulation [7]. This means that the advantages shown empirically for the SFQ framework can be directly extrapolated to OWA-based approaches such as the one designed in [2]. This is good because the evaluation results apply not only for a particular scenario but for other well-known proposals whose practical behaviour for large document collections was unclear.

Nevertheless, some counterintuitive problems have been described for OWA operators when handling expressions involving several fuzzy sets. 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 [9, 1].

One of the major drawbacks of OWA's method is its nonmonotonic behaviour for propositions involving two properties [1]. This means that, given two quantifiers Q_1, Q_2 such that Q_1 is more specific than Q_2[8], 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 produce an evaluation score which is less than or equal than the score produced by $at\_least\_60\%$. 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.

[7] That is, the SFQ formulation is equivalent to the OWA formulation for monotonic unary expressions.

[8] 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.

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.

These problems are not present for the fuzzification mechanisms defined in [8], which stand on the basis of the SFQ-based framework. This fact and the intrinsic generality of the SFQ-based approach are convenient for the purpose of IR.

5 Experiments

The behaviour of the extended fuzzy query language has been evaluated empirically. This experimental research task is fundamental in determining the actual benefits that retrieval engines might obtain from linguistic quantifiers. The empirical evaluation presented in this section expands the experimentation carried out in [19]. In particular, only a basic tf/idf weighting scheme was tested in [19]. We report here performance results for evolved weighting approaches. Our hypothesis is that these weighting methods, which have traditionally performed very well in the context of popular IR models, might increase the absolute performance attainable by the fuzzy approach. The results of the experimentation conducted in [19] are also shown here because we want to check whether or not the same trends hold when different weighting schemes are applied.

The experimental benchmark involved the Wall Street journal (WSJ) corpora from the TREC collection, which contains about 173000 news articles spread over six years (total size: 524 Mb), and 50 topics from TREC-3 [12] (topics #151-#200). Common words were removed from documents and topics[9] and Porter's stemmer [22] was applied to reduce words to their syntactical roots. The inverted file was built with the aid of GNU mifluz [20], which supplies a C++ library to build and query a full text inverted index.

As argued in section 3.2, every indexing term t_i is interpreted as a fuzzy set of documents, $Sm(t_i)$, whose membership function can be computed following classical IR weighting formulas. In [19], a normalized version of the popular tf/idf weighting scheme was applied as follows. Given a document d_j, its degree of membership in the fuzzy set defined by a term t_i is defined as:

$$\mu_{Sm(t_i)}(d_j) = \frac{f_{i,j}}{max_k f_{k,j}} * \frac{idf(t_i)}{max_l idf(t_l)} \tag{13}$$

[9] The stoplist was composed of 571 common words.

In the equation, $f_{i,j}$ 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[10]. 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.

Although the basic *tf/idf* weighting was very effective on early IR collections, it is now accepted that this classic weighiting method is non-optimal [26]. The characteristics of present datasets required the development of methods to factor document length into term weights. In this line, pivoted normalization weighting [28] is a high-performance method which has demonstrated its merits in exhaustive TREC experimentations [26]. It is also especially remarkable that pivot-based approaches are also competitive for web retrieval purposes [27, 13, 29]. As a consequence, it is important to check how this effective weighting scheme works in the context of the SFQ-based method. Furthermore, a comparison between the fuzzy model powered by pivoted weigths and a high performance pivot-based IR retrieval method will also help to shed light on the adequacy of the fuzzy approach to enhance retrieval engines. More specifically, we will compare the fuzzy model against the inner product matching function of the vector-space model with document term weights computed using pivoted normalization.

The fuzzy set of documents induced by every individual query term can be defined using pivoted document length as follows:

$$\mu_{Sm(t_i)}(d_j) = \frac{\frac{1+ln(1+ln(f_{i,j}))}{(1-s)+s\frac{dl_j}{avg_{dl}}}}{norm\_1} * \frac{qtf_i}{max_l qtf_l} * \frac{ln(\frac{N+1}{n_i})}{norm\_2} \quad (14)$$

where $f_{i,j}$ is the raw frequency of term t_i in the document d_j, s is a constant (the pivot) in the interval $[0,1]$, dl_j is the length of document d_j, avg_{dl} is the average document length, qtf_i is the frequency of term t_i in the query and $max_l qtf_l$ is the maximum term frequency in the query. The value N is the total number of documents in the collection and n_i is the number of documents which contain the term t_i. The normalizing factors $norm\_1$ and $norm\_2$ are included to maintain $\mu_{Sm(t_i)}(d_j)$ between 0 and 1. In the experiments reported here $norm\_1$ is equal to $\frac{1+ln(1+ln(maxdl)))}{(1-s)}$ ($maxdl$ is the size of the largest document) and $norm\_2$ is equal to $ln(N+1)$. This formula arises straightforwardly from the pivot-based expression detailed in [26].

The rationale behind both equations, 13 and 14, is that 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 $Sm(t_i)$. Note that, it is not guaranteed that there exists a document d_j such that $\mu_{Sm(t_i)}(d_j) = 1$. Indeed,

[10] The function used in [19] was $idf(t_i) = log(max_l n_l / n_i)$, 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 same function has been used in the new experiments reported here.

the distribution of the values $\mu_{Sm(t_i)}(d_j)$ depends largely on the characteristics of the document collection[11]. Anyway, for medium/large collections, such as WSJ, most $\mu_{Sm(t_i)}(d_j)$ values tend to be small. We feel that the large success of tf/idf weighting schemes and their evolved variations in the context of IR is a solid warranty for its application in the context of the SFQ framework. Other mathematical shapes could have been taken into account for defining a membership function. Nevertheless, these membership definitions are convenient because, as sketched in the next paragraphs, the SFQ framework can thus handle popular IR methods as particular cases.

For both weighing methods (equations 13 and 14) we implemented a baseline experiment by means of a linear fuzzy quantified sentence Q_{lin}, whose associated semi-fuzzy quantifier is:

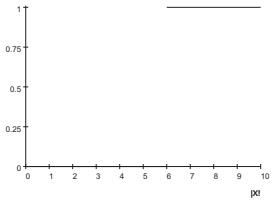
$$Q_{lin} : \wp(U) \rightarrow [0,1]$$
$$Q_{lin}(X) = \frac{|X|}{|U|}$$

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 [25]. The details can be found in appendix A. This is a good property of the fuzzy approach because it can handle popular IR retrieval methods as particular cases.

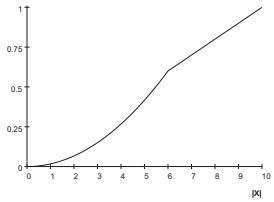
5.1 Experiments: tf/idf

The first pool of experiments considered only terms from the topic title. In order to check whether *non-linear* quantifiers are good in terms of retrieval performance, relaxed versions of *at least* quantifiers were implemented. For example, a usual crisp implementation of an *at least 6* quantifier (left-hand side) and its proposed relaxation (right-hand side) can be defined as:

$$at\_least\_6 : \wp(U) \rightarrow [0,1] \qquad at\_least\_6 : \wp(U) \rightarrow [0,1]$$
$$at\_least\_6(X) = \begin{cases} 0 & \text{if } |X| < 6 \\ 1 & \text{otherwise} \end{cases} \qquad at\_least\_6(X) = \begin{cases} (10/6) * (|X|/10)^2 & \text{if } |X| < 6 \\ |X|/10 & \text{otherwise} \end{cases}$$



a) crisp definition



b) relaxed definition

[11] For instance, in eq. 13 this will only happen if the term(s) that appear(s) the largest number of times within the document is/are also the most infrequent one(s) across the whole collection.

		at_least_k						
Recall	Q_{lin}	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$
0.00	0.5979	0.6165	0.6329	0.6436	0.6568	0.6580	0.6582	0.6597
0.10	0.4600	0.4776	0.4905	0.4968	0.5019	0.5036	0.5035	0.5037
0.20	0.3777	0.3997	0.4203	0.4208	0.4243	0.4251	0.4253	0.4253
0.30	0.3092	0.3336	0.3454	0.3479	0.3486	0.3483	0.3483	0.3483
0.40	0.2430	0.2680	0.2751	0.2805	0.2792	0.2786	0.2784	0.2784
0.50	0.1689	0.2121	0.2191	0.2234	0.2228	0.2226	0.2226	0.2226
0.60	0.1302	0.1592	0.1704	0.1774	0.1772	0.1768	0.1770	0.1770
0.70	0.0853	0.1100	0.1215	0.1261	0.1267	0.1269	0.1273	0.1273
0.80	0.0520	0.0734	0.0855	0.0888	0.0892	0.0892	0.0892	0.0892
0.90	0.0248	0.0428	0.0467	0.0497	0.0496	0.0495	0.0495	0.0495
1.00	0.0034	0.0070	0.0107	0.0106	0.0105	0.0105	0.0105	0.0105
Avg.prec. (non-interpolated)	0.2035	0.2241	0.2362	0.2403	0.2409	0.2410	0.2410	0.2411
% change		+10.12%	+16.07%	+18.08%	+18.4%	+18.4%	+18.4%	+18.5%

Table 1. Effect of simple *at least* queries on retrieval performance

The crisp *at least* implementation is too rigid to be applied in IR. It is not fair to consider that a document matching 9 query terms is as good as one matching only 6 terms. On the other hand, it is too rigid to consider that a document matching 0 query terms is as bad as one matching 5 query terms. The intuitions behind *at least* quantifiers can be good for retrieval purposes if implemented in a relaxed form. In particular, intermediate implementations, between a classical *at least* and a linear implementation (which is typical in popular IR matching functions, as shown above), were proposed and tested in [19]. Non-relevant documents might match a few query terms simply by chance. To minimize this problem the 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 *at least* implementation, documents matching many terms (more than 6 for the example) receive a score that grows linearly with the number of those terms.

The first set of results, involving the baseline experiment ($Q_{lin}(t_1, t_2, \ldots, t_n)$) and several *at least* formulations, are shown in table 1. The *at least* quantifiers were relaxed in the form shown in the example above. Although topic titles consist typically of very few 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 across all recall levels and for at_least_x with $x \geq 8$ the performance values became stabilized.

The next pool of experiments used all topic subfields (Title, Description & Narrative). Different strategies were tested in order to produce fuzzy queries from topics. For all experiments, every subfield is used for generating a single fuzzy quantifier and the fuzzy query is the conjunction of these quantifiers. Figure 1 exemplifies the articulation of fuzzy queries from a TREC topic[12]. 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 tested several combinations of *at least* and linear quantifiers. For implementing the conjunction connective both the fuzzy MIN operator and the product operator were applied. Performance results are summarized in tables 2 (MIN operator) and 3

[12] We use the symbol $\wedge$ to refer to the Boolean AND connective.

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:
at_least_4(vitamin,cure,caus,human,ailment) $\wedge$ at_least_4(document,identifi,
vitamin,contribut,cure,human,diseas,ailment,caus,health,problem) $\wedge$
at_least_3(relevant,document,provid,inform,indic,vitamin,prevent,cure,
human,ailment,caus,health,problem,make,gener,refer,good,
nutrit,research,conduct,result,deriv,treat)

Fig. 1. Fuzzy query from a TREC topic

(product operator). In terms of average precision, the product operator is clearly better than the MIN operator to implement the boolean AND connective. Indeed, all the columns in table 3 depict better performance ratios than their respective columns in table 2.

On the other hand, the combination of linear quantifiers is clearly inferior to the combination of at_least_x quantifiers. There is a progressive improvement in retrieval performance as the value of x grows from 2 to 8. This happens independently of the operator applied for implementing the conjunction. Performance becomes 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 tables 2 and 3 aims at checking the effect of *at least* quantifiers vs linear quantifiers within the SFQ fuzzy approach and, later on, we will compare the best SFQ results with a classic approach in which a linear quantifier is applied over all topic terms.

Experiments using averaging-like operators (such as the ones tested by Lee and others in [18, 17]) for implementing the boolean conjunction were also run but further improvements in performance were not obtained. This might indicate that, although T-norm operators (e.g. MIN and product) worked bad to combine terms within conjunctive boolean representations [18, 17], they could play an important role to combine more expressive query components, such as quantifiers.

	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		$x = 2$	$x = 3$	$x = 4$	$x = 8$
0.00	0.6822	0.6465	0.6642	0.6917	0.7577
0.10	0.4713	0.4787	0.4739	0.4804	0.5290
0.20	0.3839	0.3803	0.4011	0.4080	0.4408
0.30	0.3071	0.3132	0.3236	0.3283	0.3371
0.40	0.2550	0.2621	0.2671	0.2720	0.2722
0.50	0.2053	0.2127	0.2190	0.2221	0.2256
0.60	0.1557	0.1457	0.1578	0.1613	0.1709
0.70	0.1053	0.1117	0.1146	0.1192	0.1311
0.80	0.0641	0.0685	0.0744	0.0788	0.0849
0.90	0.0397	0.0440	0.0436	0.0403	0.0430
1.00	0.0060	0.0097	0.0140	0.0142	0.0171
Avg.prec. (non-interpolated)	0.2225	0.2232	0.2282	0.2321	0.2481
% change		+0.3%	+2.6%	+4.3%	+11.5%

Table 2. Conjunctions between quantifiers - MIN operator

	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		$x = 2$	$x = 3$	$x = 4$	$x = 8$
0.00	0.7277	0.7473	0.7311	0.7375	0.7664
0.10	0.5513	0.5524	0.5576	0.5542	0.5991
0.20	0.4610	0.4665	0.4711	0.4671	0.4769
0.30	0.3608	0.3802	0.3869	0.3830	0.3983
0.40	0.2915	0.3142	0.3133	0.3154	0.3172
0.50	0.2428	0.2684	0.2638	0.2643	0.2660
0.60	0.1857	0.2069	0.2111	0.2077	0.2136
0.70	0.1160	0.1407	0.1496	0.1531	0.1561
0.80	0.0720	0.0882	0.0932	0.0972	0.1014
0.90	0.0431	0.0522	0.0559	0.0609	0.0634
1.00	0.0067	0.0089	0.0126	0.0136	0.0157
Avg.prec. (non interpolated)	0.2572	0.2722	0.2760	0.2750	0.2849
% change		+5.8%	+7.3%	+6.9%	+10.8%

Table 3. Conjunctions between quantifiers - product operator

In order to conduct a proper comparison against popular IR methods, an additional baseline experiment was carried out. In this test, all terms from all topic subfields were collected into a single linear quantifier. Recall that this is equivalent to the popular vector-space model with the inner product matching function (appendix A). The results obtained are compared to the previous best results in table 4. 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, *at least* quantifiers appear as powerful tools to establish additional requirements for a document to be retrieved. Although the combination of linear quantifiers (e.g. table 3, col. 2) outperforms significantly the single linear quantifier approach (table 4, col. 2), it is still clear that a Boolean query language with linear quantifiers is not enough because further benefits are obtained when *at least* quantifiers are applied (e.g. table 3, cols. 3-5).

5.2 Experiments: pivoted document length normalization

It is well known that the classic tf/idf weighting approach is nowadays overcome by weighting schemes based on document length corrections [26]. Thus, the actual

Recall	Q_{lin} (title, desc & narr terms)	at_least_8(title terms) $\wedge$ at_least_8(desc terms) $\wedge$ at_least_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 4. Linear quantifier query vs more evolved query

impact of the SFQ fuzzy approach can only be clarified after a proper comparison against state-of-the-art matching functions. Moreover, there is practical evidence on the adequacy of pivoted weights for web retrieval purposes [27, 13, 29] and, hence, the comparison presented in this section will help to shed light on the role of fuzzy quantifiers to enhance web retrieval engines.

We have run additional experiments for evaluating the SFQ approach with pivot-based weighting methods (equation 14). For the sake of brevity, we will not report here every individual experiment but we will summarize the main experimental findings. Our discussion will be focused on tests using all topic subfields because the inner product matching function (baseline experiment with linear quantifier) yields its top performance when applied to all topic subfields. Indeed, as expected, the performance of the baseline experiment is substantially better than the tf/idf baseline (table 5, column 2 vs table 4, column 2).

The following enumeration sketches the main conclusions from the new pool of tests:

1. Again, the product operator is better than the MIN operator to implement the boolean AND connective.
2. The fuzzy approach with relaxed *at least* statements was not able to produce better performance results than the inner product matching function (baseline).
3. The fuzzy approach with a linear quantifier applied on every individual topic subfield (whose results are combined with the product operator) is able to produce modest improvements with respect to the baseline.

The pivot constant s was fixed to the value of 0.2 [13]. The main performance results are shown in table 5.

Further research is needed to determine the actual role of *at least* statements in the context of a high performance weighting technique such as pivoted document length normalization. At this point, the aplication of relaxed *at least* expressions produced performance results which are worse than those obtained for the baseline.

[13] Some tests with varying values of s were run for the fuzzy model (with both linear & *at least* statements) but no improvements were found. The baseline performance is also optimal for the value of 0.2. Indeed, the ideal value of the pivot s has also been considered very stable in previous experimentations on pivoted document length normalization schemes [26].

	Q_{lin} **(title, desc & narr terms)** (baseline)	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			$x = 2$	$x = 3$	$x = 4$	$x = 8$
0.00	0.8741	0.8637	0.8702	0.8470	0.8065	0.8030
0.10	0.7211	0.7276	0.7114	0.6984	0.6733	0.6563
0.20	0.6159	0.6467	0.6326	0.6160	0.5787	0.5544
0.30	0.5074	0.5405	0.5253	0.5114	0.4869	0.4487
0.40	0.4380	0.4584	0.4265	0.4128	0.3965	0.3697
0.50	0.3673	0.3722	0.3509	0.3430	0.3292	0.3024
0.60	0.3186	0.3200	0.2910	0.2778	0.2652	0.2434
0.70	0.2461	0.2511	0.2276	0.2138	0.2052	0.1864
0.80	0.1761	0.1876	0.1737	0.1567	0.1502	0.1340
0.90	0.1122	0.1239	0.1082	0.1027	0.0982	0.0854
1.00	0.0374	0.0350	0.0380	0.0375	0.0377	0.0365
Avg.prec. (non-interpolated)	0.3858	0.3977	0.3799	0.3666	0.3488	0.3278
% change		+3.1%	-1.5%	-5.0%	-9.6%	-15%

Table 5. Experimental results - Pivoted document length normalization

As depicted in table 5, the overall performance gets worse as the *at least* statement becomes stricter. An at_least_2 statement is slightly worse than the baseline (1.5% worse) but the at_least_8 formulation yields significantly worse performance ratios (average precision decreases by 15%). In the near future we plan to make extensive testing on different relaxations of *at least* formulations in order to shed light on this issue.

On the contrary, the fuzzy model with linear quantifiers was able to overcome the baseline. Although the baseline experiment follows a high performance state-of-the-art IR retrieval technique (inner product matching function of the vector-space model with pivoted document length normalized weights), the fuzzy approach was still able to construct slightly better rankings. This is an important circumstance as it anticipates that fuzzy methods can say a word in future retrieval engines.

6 Conclusions and Further Work

Classical IR approaches tend to oversimplify the content of user information needs whereas flexible query languages allow to articulate more evolved queries. For instance, the inclusion of quantified statements in the query language permits to express additional constraints for the retrieved documents. IR matching functions can be relaxed in different ways by means of quantified statements whose implementation is handled efficiently by semi-fuzzy quantifiers and quantified fuzzification mechanisms.

In this work we showed that our proposal based on the concept of semi-fuzzy quantifier handles pioneering fuzzy quantification proposals for IR as particular cases. On the other hand, we conducted large-scale experiments showing that this fuzzy approach is competitive with state-of-the-art IR techniques. These popular IR methods have recurrently appeared among the best retrieval methods for both ad-hoc and web retrieval tasks and, hence, it is very remarkable that our SFQ approach performs at the same level.

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

seminal proposals of fuzzy quantification for IR. This is guaranteed by the subsumption proved in this work.

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. On the other hand, future research efforts will also be dedicated to analyze the practical behaviour of alternative models of fuzzy quantification. In this respect, besides *at least* expressions, we plan to extend the evaluation to other kind of quantifiers. For the basic retrieval task we have only found benefits in retrieval performance when this sort of quantifiers were applied. Nevertheless, we will study the adequacy of other sort of linguistic quantifiers in the context of other IR tasks.

Acknowledgements

Authors wish to acknowledge support from the Spanish Ministry of Education and Culture (project ref. TIC2003-09400-C04-03) and Xunta de Galicia (project ref. PGIDIT04SIN206003PR). 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.

References

1. 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.
2. 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.
3. 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.
4. G. Bordogna and G. Pasi. Modeling vagueness in information retrieval. In M. Agosti, F. Crestani, and G. Pasi, editors, *ESSIR 2000, LNCS 1980*, pages 207–241. Springer-Verlag Berlin Heidelberg, 2000.
5. 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.
6. F. Crestani and G. Pasi (eds). *Soft Computing in Information Retrieval: techniques and applications*. Studies in fuzziness and soft computing. Springer-Verlag, 2000.
7. 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.
8. 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*, 146:97–120, 2004.
9. I. Glöckner. A framework for evaluating approaches to fuzzy quantification. Technical Report TR99-03, Universität Bielefeld, May 1999.

10. I. Glöckner. *Fuzzy Quantifiers in Natural Language: Semantics and Computational Models*. PhD thesis, Universität Bielefeld, 2003.
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. Hawking, E. Voorhees, N. Craswell, and P. Bailey. Overview of the trec-8 web track. In *Proc. TREC-8, the 8th Text Retrieval Conference*, pages 131–150, Gaithersburg, United States, November 1999.
14. E. Herrera-Viedma and G. Pasi. Fuzzy approaches to access information on the web: recent developments and research trends. In *Proc. International Conference on Fuzzy Logic and Technology (EUSFLAT 2003)*, pages 25–31, Zittau (Germany), 2003.
15. D.H. Kraft and D.A. Buell. A model for a weighted retrieval system. *Journal of the american society for information science*, 32(3):211–216, 1981.
16. D.H. Kraft and D.A. Buell. Fuzzy sets and generalized boolean retrieval systems. *International journal of man-machine studies*, 19:45–56, 1983.
17. 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.
18. 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.
19. D. E. Losada, F. Díaz-Hermida, A. Bugarín, and S. Barro. Experiments on using fuzzy quantified sentences in adhoc retrieval. In *Proc. SAC-04, the 19th ACM Symposium on Applied Computing - Special Track on Information Access and Retrieval*, Nicosia, Cyprus, March 2004.
20. GNU mifluz. http://www.gnu.org/software/mifluz. 2001.
21. 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.
22. 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.
23. T. Radecki. Outline of a fuzzy logic approach to information retrieval. *International Journal of Man-Machine studies*, 14:169–178, 1981.
24. G. Salton, E. A. Fox, and H. Wu. Extended boolean information retrieval. *Communications of the ACM*, 26(12):1022–1036, 1983.
25. G. Salton and M.J. McGill. *Introduction to modern information retrieval*. McGraw-Hill, New York, 1983.
26. A. Singhal. Modern information retrieval: a brief overview. *IEEE Data Engineering Bulletin*, 24(4):35–43, 2001.
27. A. Singhal, S. Abney, M. Bacchiani, M. Collins, D. Hindle, and F. Pereira. At&t at trec-8. In *Proc. TREC-8, the 8th Text Retrieval Conference*, pages 317–330, Gaithersburg, United States, November 1999.
28. A. Singhal, C. Buckley, and M Mitra. Pivoted document length normalization. In *Proc. SIGIR-96, the 19th ACM Conference on Research and Development in Information Retrieval*, pages 21–29, Zurich, Switzerland, July 1996.
29. A Singhal and M. Kaszkiel. At&t at trec-9. In *Proc. TREC-9, the 9th Text Retrieval Conference*, pages 103–116, Gaithersburg, United States, November 2000.

30. R.R. Yager. On ordered weighted averaging aggregation operators in multi criteria decision making. *IEEE Transactions on Systems, Man and Cybernetics*, 18(1):183–191, 1988.
31. R.R. Yager. Connectives and quantifiers in fuzzy sets. *Fuzzy Sets and Systems*, 40:39–75, 1991.
32. R.R. Yager. A general approach to rule aggregation in fuzzy logic control. *Applied Intelligence*, 2:333–351, 1992.
33. R.R. Yager. Families of owa operators. *Fuzzy Sets and Systems*, 59(2):125–244, 1993.
34. L.A. Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Comp. and Machs. with Appls.*, 8:149–184, 1983.

Appendix A

Given a query expression such as $Q_{lin}(t_1, \ldots, t_n)$, where each t_i is an atomic term, and a document d_j, the fuzzy set induced by the document can be expressed as: $C_{d_j} = \{w_{1,j}/1, \ldots, w_{n,j}/n\}$.

Without any loss of generality, we will assume that query terms are sorted in descending order of its membership degree in C_{d_j}.

The linear semi-fuzzy quantifier Q_{lin} operates on C_{d_j} as follows:

$$\begin{aligned}
(F(Q_{lin}))(C_{d_j}) &= (1 - w_{1,j}) * Q_{lin}((C_{d_j})_{\geq 1}) + (w_{1,j} - w_{2,j}) * Q_{lin}((C_{d_j})_{\geq w_{1,j}}) + \\
&\quad (w_{2,j} - w_{3,j}) * Q_{lin}((C_{d_j})_{\geq w_{2,j}}) + \ldots + \\
&\quad (w_{n-1,j} - w_{n,j}) * Q_{lin}((C_{d_j})_{\geq w_{n-1,j}}) + w_{nj} * Q_{lin}((C_{d_j})_{\geq w_{n,j}}) = \\
&= (1 - w_{1,j}) * 0 + (w_{1,j} - w_{2,j}) * (1/n) + \\
&\quad (w_{2,j} - w_{3,j}) * (2/n) + \ldots + (w_{n-1,j} - w_{n,j}) * ((n-1)/n) + w_{nj} * 1 = \\
&= (1/n) * ((w_{1,j} - w_{2,j}) + (w_{2,j} - w_{3,j}) * 2 + \\
&\quad + \ldots + (w_{n-1,j} - w_{n,j}) * (n-1) + w_{nj} * n) = \\
&= (1/n) * \sum_{t_i \in q} w_{ij}
\end{aligned}$$

leading to:

$$\mu_{Sm(Q_{lin}(t_1,\ldots,t_n))}(d_j) = (1/n) * \sum_{t_i \in q} w_{ij}$$

Let us now analyze the two weighting schemes (equations 13 and 14) independently:

- **tf/idf weights** (equation 13). Consider now a vector-space approach in which document vectors are weighted as in equation 13 and query vectors are binary. The inner product equation, $\sum w_{i,j} * q_i$, where $w_{i,j}$ (q_i) is the weight for term t_i in document d_j (query), can be reduced to $\sum_{t_i \in q} w_{i,j}$ when query weights are binary. It follows that both approaches result in the same ranking of documents because the value $1/n$ does not affect the ranking of every query.

- **pivoted weights** (equation 14). Consider now a vector-space approach in which document vectors are weighted as:

$$\frac{\frac{1+ln(1+ln(f_{i,j}))}{(1-s)+s\frac{dl_j}{avg_{dl}}}}{norm\_1} * \frac{ln(\frac{N+1}{n_i})}{norm\_2}$$

and query vector weights are:

$$\frac{qtf_i}{max_l qtf_l}$$

Again, it follows that the inner product matching yields the same ranking that the one constructed from the fuzzy model.