FUZZY INTERVAL NUMBER (FIN) TECHNIQUES FOR CROSS LANGUAGE INFORMATION RETRIEVAL

Catherine Marinagi, Theodoros Alevizos, Vassilis G. Kaburlasos
Department of Industrial Informatics, T.E.I. of Kavala, Greece

Christos Skourlas
Department of Informatics, T.E.I. of Athens, Greece

Keywords: Fuzzy Interval Number (FIN), Information Retrieval (IR), Cross Language Information Retrieval (CLIR).

Abstract: A new method to handle problems of Information Retrieval (IR) and related applications is proposed. The method is based on Fuzzy Interval Numbers (FINs) introduced in fuzzy system applications. Definition, interpretation and a computation algorithm of FINs are presented. The frame of use FINs in IR is given. An experiment showing the anticipated importance of these techniques in Cross Language Information Retrieval (CLIR) is presented.

1 INTRODUCTION

Oard (Oard, 1997) classifies free (full) text Cross Language Information Retrieval (CLIR) approaches to corpus-based and knowledge-based approaches. Knowledge-based approaches encompass dictionary-based and ontology (thesaurus)-based approaches while corpus-based approaches encompass parallel, comparable and monolingual corpora. Dictionary-based systems translate query terms one by one using all the possible senses of the term. The main drawbacks of this procedure are:
a) the lack of fully updated Machine Readable Dictionaries (MRDs),
b) the ambiguity of the translations of terms results in a 50% loss of precision (Davis, 1996).
Since the machine translation of the query is less accurate than that of a full text, experiments have been conducted with collections having machine translations of all the collection texts to all languages of interest. Such systems are really multi-monolingual systems. Parallel and comparable corpora systems are different: the parallel (or comparable) corpora are used to "train" the system and after that no translations are used for retrieval. One such system, perhaps the most successful, is based on Latent Semantic Indexing (LSI) (Dumais, 1996), (Berry, 1995). The main problem with this approach is that it is not easy to find training parallel corpora related to any collection.
Fuzzy (set) techniques were proposed for Information Retrieval (IR) applications many years ago (Radecki, 1979), (Kraft, 1993), mainly for modeling.
Fuzzy Interval Numbers (FINs) were introduced by Kaburlasos (Kaburlazos, 2004), (Petridis, 2003) in fuzzy system applications.
A FIN may be interpreted as a conventional fuzzy set; additional interpretations for a FIN are possible including a statistical interpretation.
The special interest in these objects and associated techniques for IR stems from their anticipated capability to serve CLIR without the use of dictionaries and translations.
The basic features of the method presented here are:
1) Documents are represented as FINs; a FIN resembles a probability distribution.
2) The FIN representation of documents is based on the use of the collection term frequency as the term identifier.
3) The use of FIN distance instead of a similarity measure.
There are indications, part of which is presented in section 4 below, that a parallel corpora system can be build using FIN techniques.
The structure of the remainder of this paper is as follows: In section 2 a brief introduction to FINs and other relevant concepts is given. In section 3 the

Marinagi C., Alevizos T., G. Kaburlasos V. and Skourlas C. (2006).
FUZZY INTERVAL NUMBER (FIN) TECHNIQUES FOR CROSS LANGUAGE INFORMATION RETRIEVAL.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - AIDSS*, pages 249-256
DOI: 10.5220/0002467002490256

"conceptual" transition from document vectors to document FINs is presented. Section 4 presents the special interest of FINs in handling CLIR problems. Conclusions and current work on the subject are presented in section 5.

2 THEORETICAL BACKGROUND

A. Generalized Intervals

A generalized interval of height h ($\in$ (0,1]) is a mapping μ given by:

If $x_1 < x_2$ (positive generalized interval) then

$$\mu_{[x_1,x_2]^h}(x) = \begin{cases} h, & x_1 \le x \le x_2 \\ 0, & \text{otherwise} \end{cases}$$

elseif $x_1 > x_2$ (negative generalized interval) then

$$\mu_{[x_1,x_2]^h}(x) = \begin{cases} -h, & x_1 \ge x \ge x_2 \\ 0, & \text{otherwise} \end{cases}$$

elseif $x_1 = x_2$ (trivial generalized interval) then

$$\mu_{[x_1,x_2]^h}(x) = \begin{cases} \{-h,,h\} & x = x_1 \\ 0, & \text{otherwise} \end{cases}$$

In this paper we use the more compact notation $[x_1, x_2]^h$ instead of the μ notation.

The interpretation of a generalized interval depends on an application; for instance if a feature is present it could be indicated by a positive generalized interval.

The set of all positive generalized intervals of height h is denoted by $\mathbf{M}_+^h$, the set of all negative generalized intervals by $\mathbf{M}_-^h$, the set of all trivial generalized intervals by $\mathbf{M}_0^h$ and the set of all generalized intervals by $\mathbf{M}^h = \mathbf{M}_-^h \cup \mathbf{M}_0^h \cup \mathbf{M}_+^h$.

Two functions, that are going to be used in the sequel, are defined:

Function support maps a generalized interval to the corresponding conventional interval; support $([x_1, x_2]^h) = [x_1, x_2]$ for positive, support$([x_1, x_2]^h) = [x_2, x_1]$ for negative and support$([x_1, x_1]^h) = \{x_1\}$ for trivial generalized intervals.

Function sign: $\mathbf{M}^h \rightarrow \{ -1, 0, +1 \}$ maps a positive generalized interval to +1, a negative generalized interval to –1 and a trivial generalized interval to 0.

Now, we try to define a metric distance and an inclusion measure function in the set (lattice) $\mathbf{M}^h$.

A relation $\le$ in a set S is called partial ordering relation if and only if it is:

1) $x \le x$ (reflexive)
2) $x \le y$ and $y \le x$ imply $x = y$ (antisymmetric)
3) $x \le y$ and $y \le z$ imply $x \le z$ (transitive)

Therefore a partial order relation $\le$ can be defined in the set $\mathbf{M}^h$, $h \in (0,1]$:

1) $[a, b]^h \le [c, d]^h \Leftrightarrow$ support$([a, b]^h) \subseteq$ support $([c, d]^h)$, for $[a, b]^h, [c, d]^h \in \mathbf{M}_+^h$
2) $[a, b]^h \le [c, d]^h \Leftrightarrow$ support$([c, d]^h) \subseteq$ support$([a, b]^h)$, for $[a, b]^h, [c, d]^h \in \mathbf{M}_-^h$
3) $[a, b]^h \le [c, d]^h \Leftrightarrow$ support$([c, d]^h) \cap$ support$([a, b]^h) \neq 0$, for $[a, b]^h \in \mathbf{M}_-^h$, $[c, d]^h \in \mathbf{M}_+^h$

A partial ordering relation does not hold for all pairs of generalized interval.

A lattice $(L, \le)$ is a partially ordered set and any two elements have a unique greatest lower bound or lattice meet $(x \wedge_L y)$ and a unique least upper bound or lattice join $(x \vee_L y)$.

A valuation v in a lattice L, defined as the area "under" a generalized interval, is a real function v:L $\rightarrow$ R which satisfies

$v(x)+v(y)= v(x \vee_L y) + v(x \wedge_L y)$, $x,y \in L$.

A valuation is called monotone if and only if $x \le y$ implies $v(x) \le v(y)$ and positive if and only if $x < y$ implies $v(x) < v(y)$ for $x,y \in L$.

A metric distance in a set S is a real function d:S×S$\rightarrow$ R which satisfies:

1) $d(x,y) \ge 0$, $x, y \in S$
2) $d(x,y) = 0 \Leftrightarrow x = y$, $x \in S$
3) $d(x,y) = d(y,x)$, $x, y \in S$ (symmetry)
4) $d(x,y) \le d(x,z) + d(z,y)$, $x, y, z \in S$ (triangle inequility)

Therefore a metric distance d:L×L$\rightarrow$ R can be defined in the lattice $\mathbf{M}^h$, $h \in (0,1]$ given by

$d(x, y) = v(x \vee_L y) - v(x \wedge_L y)$, $x,y \in L$.

A lattice is called complete when each of its subsets has a least upper bound and a greatest lower bound. In a complete lattice the positive valuation function v can be used to define an inclusion measure function k: L×L$\rightarrow$ [0, 1] given by

$$k(x, u) = \frac{v(u)}{v(x \vee_L u)}$$

The lattice $\mathbf{M}^h$ is not complete.

Therefore we must define in a different way an inclusion measure to quantify the degree of inclusion of a lattice element into another one.

Definition

An inclusion measure σ in a non-complete lattice L is a map σ: L×L→ [0, 1] such that for u, w, x ∈ L:

1) σ(x, x) = 1

2) u < w ⇒ σ(w, u) < 1

3) u ≤ w ⇒ σ(x, u) ≤ σ(x, w) (consistency property)

We have interchangeable used the notations σ(x, u), σ(x ≤ u) because both the notations indicate a degree of inclusion of x in u.

Kaburlazos has proved the following proposition:

Let the underlying positive valuation function f: R→ R be a strictly increasing real function in R. Then the real function v: $\mathbf{M}^h \rightarrow R$ is given by

$v([a, b]^h) = \text{sign}([a, b]^h)\, c(h) \int_a^b [f(x) - f(a)]\, dx$

where v is a positive valuation function in $\mathbf{M}^h$,

c: (0,1] → R^+ is a positive real function for normalization. A metric distance in $\mathbf{M}^h$ is given by:

$d_h(x, y) = v(x \vee y) - v(x \wedge y)$

As an example consider f(x)=x, c(h)=h.

Then the distance is given by:

$d_h([a, b]^h, [c, d]^h) = h\,(|a-c|+|b-d|)$. As another example, for $f(x)= x^3$, h=1 and c(1)=0.5 we compute the distance (between the intervals $[-1, 1]^1$, $[2, 4]^1$)

$d_h([-1, 1]^1, [2, 4]^1) = f([-1, 1]^1 \vee [2, 4]^1) - f([-1, 1]^1 \wedge [2, 4]^1) = 32.5 + 3.5 = 36.$

The essential role of a positive valuation function v: L→ R is known to be a mapping from a lattice L of semantics to the mathematical field R of real numbers for carrying out computations.

B. Fuzzy Interval Numbers: Definition and Interpretation

A positive Fuzzy Interval Number (FIN) is a continuous function F: (0,1] → $\mathbf{M}_+^h$ such that

$h_1 \le h_2 \Rightarrow \text{support}(F(h_1)) \supseteq \text{support}(F(h_2))$,

where $0 < h_1 \le h_2 < 1$.

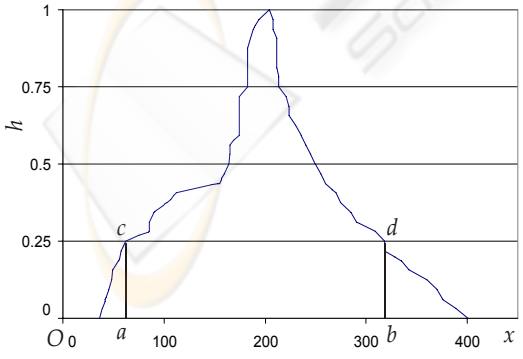

Figure 1: An 86 value FIN. In the support(F(0.25)) = [62,318] there are about 75% of the values.

The set of all positive FINs is denoted by $\mathbf{F}_+$. Similarly, trivial and negative FINs are defined.

Given a population (a vector) $x = [x_1, x_2, \ldots, x_N]$ of real numbers (measurements), sorted in ascending order, a FIN can be computed by applying the CALFIN algorithm given in the Appendix. In Fig.1 a FIN, calculated from a population of 86 values, is shown. Given a FIN, any "cut" at a given height h (∈ (0,1]) defines a generalized interval, denoted by F(h). In Fig.1, F(0.25) is the generalized interval $[a,b]^{0.25}$ represented by acdb.

A consequence of the CALFIN algorithm is the following: Let F(1) = $\{m_1\}$; approximately N/2 of the values of x are smaller than m_1 and N/2 are greater than m_1. Let F(0.5) = $[p_{1/2}, q_{1/2}]^{0.5}$; approximately N/4 of the values of x lie in $[p_{1/2}, m_1]$ and N/4 in $[m_1, q_{1/2}]$. In more general terms: for any h ∈ (0,1] approximately 100(1 - h)% of the N values of x are within support(F(h)).

C. FIN Metrics

Let m_h: $\mathbf{R} \rightarrow \mathbf{R}^+$ be a positive real function – a mass function – for h ∈ (0,1] (could be independent of h) and $f_h(x) = \int_0^x m_h(t)dt$.

Obviously, f_h is strictly increasing. The real function v_h: $\mathbf{M}_+^h \rightarrow \mathbf{R}$, given by $v_h([a,b]^h) = f_h(b) - f_h(a)$ is a positive valuation function in the set of positive generalized intervals of height h.

$d_h([a,b]^h,[c,d]^h) =$

$[f_h(a\vee c) - f_h(a\wedge c)] + [f_h(b\vee d) - f_h(b\wedge d)]$,

where $a\wedge c = \min\{a,c\}$ and $a\vee c = \max\{a,c\}$, is a metric distance between the two generalized intervals $[a,b]^h$ and $[c,d]^h$. In Fig.2 an interpretation of a, b, c, d in the case of two FINs is shown.

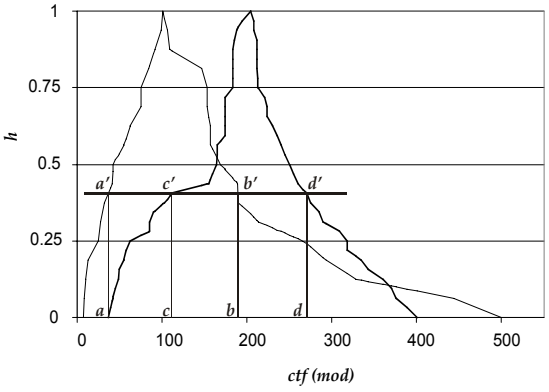

Figure 2: Two FINs F_1 and F_2 (representing two documents of the CACM test collection). The points a, b, c, d used to define $d_h(F_1(h),F_2(h)) = d_h([a,b]^h,[c,d]^h)$.

Given two positive FINs F_1 and F_2,

$$d(F_1,F_2)= c\int_0^1 d_h(F_1(h),F_2(h))dh$$

where c is an user-defined positive constant, is a metric distance (for a proof see (Kaburlazos, 2004))

3 USING FINS TO REPRESENT DOCUMENTS

In the Vector Space Model (Salton, 1983) for Information Retrieval, a text document is represented by a vector in a space of many dimensions, one for each different term in the collection. In the simplest case, the components of each vector are the frequencies of the corresponding terms in the document:

$$Doc_k = (f_{k1}, f_{k2}, \ldots f_{kn})$$

f_{kj} stands for the frequency of occurrence of term t_j in document Doc_k. In Fig.3 one such vector is shown as a histogram.

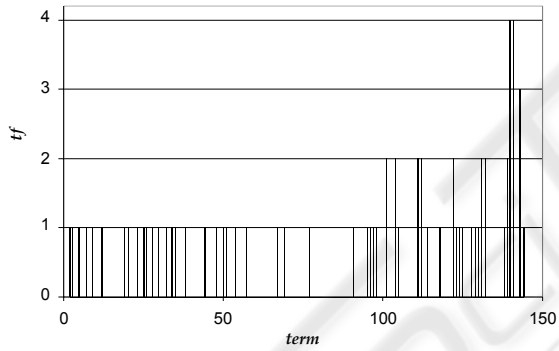

Figure 3: A document vector as a histogram. Each value on the term axis represents a term (stem), e.g. "51" stands for "industri" and "104" for "research".

Let ctf_j be the total frequency of occurrence of term t_j in the whole collection. Then ctf_j is equal to

$$\sum_k tf_{kj}$$

The collection term frequencies are going to be used as term identifiers. In order to ensure the uniqueness of the identifiers a multiple of a small ε is added to the ctfs when needed. In Fig.4 the new form of the document vector histogram is shown.

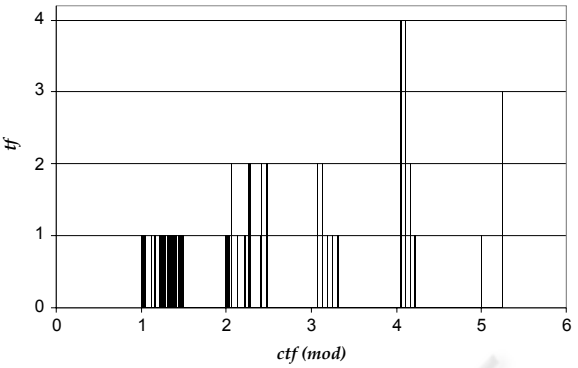

Figure 4: The document vector histogram in (modified) ctf abscissae. Each value on the ctf (mod) axis represents a term (stem), e.g. "1.24775" stands for "industri" and "2.09009" for "research".

The next step is to break the "high bars" to multiple pieces of height 1, placed side by side, separated by some $\varepsilon' < \varepsilon$; this is shown in Fig.5. Now, the original histogram has been transformed to a "density graph".

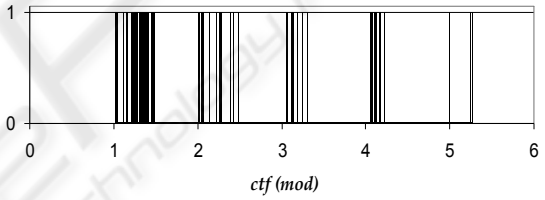

Figure 5: The "density graph" representation of the document of Fig. 1.

The abscissae vector is exactly the "number population" from which the document FIN (Fig.6) is computed from by the CALFIN algorithm described in the Appendix.

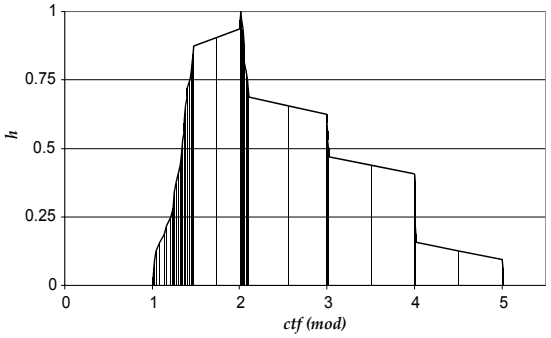

Figure 6: The document FIN along with the median bars. The numbers of terms on the left and right sides of the bar with height 1 are equal.

The FIN distance is used instead of the similarity measure between documents: the smaller the distance the more similar the documents. This

a) means that for each query a FIN must be calculated and
b) imposes a serious requirement on the queries: they must have many terms. As can be seen in the CALFIN algorithm (appendix), at least two terms are needed to calculate a FIN. Moreover, experience shows that many term documents (queries) give "better" fins.

4 FINS FOR CLIR

A. The Hypothesis

Consider a document set $\{doc(l)_1, doc(l)_2, \ldots, doc(l)_n\}$, all in language l. Suppose that for each document $doc(l)_k$ there exist translations $doc(j)_k$ to the languages j = 1, …, m. So we have a multilingual document collection:

$\{ 1 \le j \le m, 1 \le k \le n , doc(j)_k \}$

Assume that:

1. The stopword lists of all languages are translations of each other (partially unrealistic).
2. All the translations are done "1 word to 1 word", i.e. we have no phrasal translations of words (highly unrealistic (Ballesteros, 1997)).
3. There is no different polysemy between any two languages (highly unrealistic (Ballesteros, 1998)).

Under these assumptions: Consider a term in language l, $t(l)_j$. Let $tf(l)_{jk}$ be the term frequency in $doc(l)_k$ and $ctf(l)_j$ the total frequency of the term in the collection. The following equalities hold:

$tf(1)_{jk} = tf(2)_{jk} = \ldots = tf(m)_{jk}$
$ctf(1)_j = ctf(2)_j = \ldots = ctf(m)_j$

If $docF(l)_k$ is the FIN representing $doc(l)_k$ then the distances of the translated document FINs will be approximately 0:

A_1: $d(docF(l_1)_k, docF(l_2)_k) \approx 0$

The distances can be nullified exactly with the use of a dictionary.

Moreover, let qryF(l) be the FIN of a query submitted to a FIN-based IR System that manages the collection. Then:

A_2: $d(qryF(l), docF(l)_k) \approx d(qryF(l), docF(j)_k)$, $j \in 1..m$

This means that Cross Language Information Retrieval is achievable without the use of dictionaries.

B. The Experiment

The experiment aims to test the aforementioned statements A_1 and A_2 in the "real world".

A small document collection comprising 3 short documents in english and their greek translations (*the documents originate from EU databases)* was used. The documents were slightly modified in order to improve their compliance with the hypotheses of part A. The term content of the documents is shown in Table 1. Apparently it was not easy to avoid phrasal translations and terms with different polysemy.

Table 1: The term content of the documents.

	Total Number of Terms		Number of Distinct Terms	
	english	greek	English	greek
doc1	69	69	67	58
doc2	83	83	50	51
doc3	79	86	65	71
Total	Nt_{en} =231	Nt_{gr} =238	151	154

The FINs of the documents were calculated taking all terms without exceptions. The FINs of the english documents are shown in Fig.7. The steep ascent of the left side of the curves is due to the inclusion of all the terms appearing only once in the collection.

In Fig.8 the FINs of the greek and english versions of document 3 are shown. They are almost identical except for the right "tail" of the greek FIN. This is a result of different polysemy.

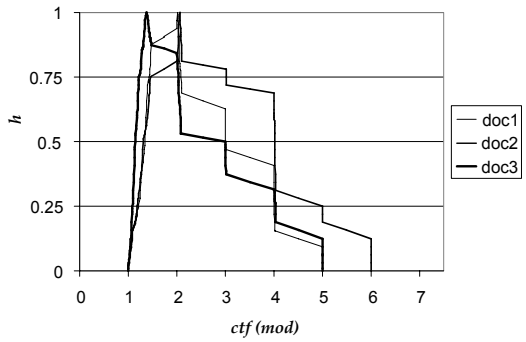

Figure 7: The FINs of the english documents.

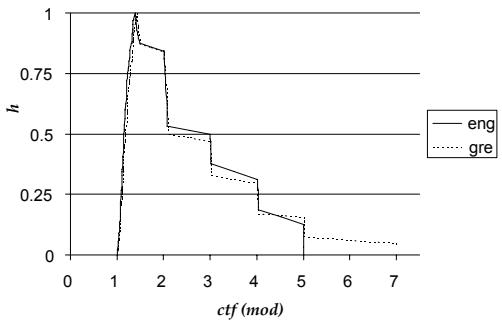

Figure 8: The FINs of the english and greek versions of document 3.

For the distance calculations a bell-shaped mass function is selected:

$$m_h(t) = \frac{\alpha + \beta h}{A^2 + \left(t - \frac{\max(ctf)}{2}\right)^2}$$

The positive real numbers A, α, β, are parameters; max(ctf) is the maximum value of all the collection term frequencies.

That kind of mass function degrades the contribution of to the FIN distance of the terms with ctf = 1. The same degradation applies to the high ctf terms. So, the contribution to the distance of the "tail" to the right of the greek FIN in Fig.8 is degraded but the same applies in general to all high frequency terms even those that do not have high document frequency.

The verification of A1 and A2 is translated as follows:

A_1: $d(docF(e)_1, docF(g)_1)$, $d(docF(e)_2, docF(g)_2)$ and $d(docF(e)_3, docF(g)_3)$ are significantly smaller that other distances between FINs.

A2: Instead of query documents the FINs of the documents of the collection are used. So instead of $d(qryF(l), docF(l)_k) \approx d(qryF(l), docF(j)_k)$, $j \in 1..m$ the following is examined:

$d(docF(l)_m, docF(l)_n) \approx d(docF(l_1)_m, docF(l_2)_n)$, m, n = 1, 2, 3, $m \neq n$, $l, l_1, l_2 = e, g$

In Table 2 the FIN distances of all pairs of documents in the collection are shown. As expected the smallest are the distances between the greek and english versions of the same documents.

If $d_{1e1g} = d(doc(e)1F, doc(g)1F) = 0.03515$ then
$d(doc(e)2F, doc(g)2F) \approx 1.2\ d_{1e1g}$ and
$d(doc(e)2F, doc(g)2F) \approx 2.06\ d_{1e1g}$

Apart from these, the smallest distance is $d(doc(e)1F, doc(e)3F) = 0.27195 \approx 7.7\ d_{1e1g}$. That is: the largest distance between two versions of the same document is about 3.8 times smaller than the smallest distance between versions of different documents. Moreover:

$d(docF(l_1)_m, docF(l_2)_n) \leq 1.15\ d(docF(l_3)_m, docF(l_4)_n)$, m, n = 1, 2, 3, $m \neq n$, $l_1, l_2, l_3, l_4 = e, g$

That is: the distance between any two versions of the same document is at most 1.15 times larger than the distance of any two other versions of the same document.

In conclusion:

1) The distances of different versions (languages) of the same document are considerably smaller than others.
2) The distances of two different documents are almost the same irrespective of the language of the documents.

In the second phase of the experiment one more english language document (doc(e)4) is inserted in the collection but not its greek version. The number of english terms is increased by $\Delta Nt_{en} = 58$. Now the distances are changed:

$d(docF(l_1)_m, docF(l_2)_n) \leq 1.32\ d(docF(l_3)_m, docF(l_4)_n)$, m, n = 1, 2, 3, $m \neq n$, $l_1, l_2, l_3, l_4 = e, g$

To rebalance the collection characteristics we renormalize the greek FIN absissae multiplying χ_j by:

$$\left(1 + k_r \frac{x_j}{\max(ctf)} \frac{\Delta NT_{en}}{NT_{en}}\right)$$

Table 2: FIN distances of the documents of the collection.

	Doc(*e*)1	*doc*(*e*)2	*doc*(*e*)3	*doc*(*g*)1	*doc*(*g*)2	*doc*(*g*)3
doc(*e*)1	0.00000	0.34044	0.27195	**0.03515**	0.32658	0.29084
doc(*e*)2	0.34044	0.00000	0.60588	0.31314	**0.07230**	0.61259
doc(*e*)3	0.27195	0.60588	0.00000	0.29938	0.59363	**0.04258**
doc(*g*)1	**0.03515**	0.31314	0.29938	0.00000	0.30065	0.31147
doc(*g*)2	0.32658	**0.07230**	0.59363	0.30065	0.00000	0.60035
doc(*g*)3	0.29084	0.61259	**0.04258**	0.31147	0.60035	0.00000

After that:

$d(\text{docF}(l_1)_m, \text{docF}(l_2)_n) \leq d(\text{docF}(l_3)_m, \text{docF}(l_4)_n)$,

$m, n = 1, 2, 3, m \neq n, l_1, l_2, l_3, l_4 = e, g$

5 CONCLUSIONS AND FUTURE WORK

FIN techniques seem promising for IR and related applications; the prospect of CLIR without dictionaries is very intriguing. Nevertheless there are quite enough topics to be considered carefully. These techniques, for monolingual and cross language IR, work with long documents and queries that can give "good" FINs. Unfortunately, this is not the case with the queries submitted to Internet search engines; these queries very often have just a couple of terms (Kobayashi, 2000). The FIN techniques can be more successful in problems of document classification where documents with many terms – and "better" FINs– must be handled.

The quality of a FIN does not depend on number of terms only; it must be considered with the mass function for the distance calculation. In the previous paragraph a "soft" degradation of the contribution to the distance of terms with ctf = 1, has been attempted through the mass function. A better idea would be probably to ignore completely these terms in the FIN computation. In FIN construction, term document frequency (df) must be taken into account as well.

The bell-shaped mass function seems to be a reasonable one but other ideas should be considered in conjunction with FIN computation.

Last but not least the renormalization scheme: in any multilingual collection the numbers of terms in different languages are random and a solid and flexible re-balancing scheme is needed, which is not independent of the FIN construction method and the distance calculation (mass function).

The optimal determination of the parameters A, α, β and k_r is part of the system training process using parallel corpora.

At present, experiments are been conducted along these lines mainly with two of the standard monolingual collections, namely CACM and WSJ. These collections are of interest because they have relatively longer queries.

On the other hand, a trilingual – greek / english / french – test collection is been built for CLIR experimentation. The parallel corpora are created by translation by hand (although there is some mechanical help). Experiments are conducted and records of performance are kept during various stages of the parallel corpora creation.

ACKNOWLEDGMENTS

This work was co-funded by 75% from the E.U. and 25% from the Greek Government under the framework of the Education and Initial Vocational Training Program – Archimedes.

REFERENCES

Ballesteros, L. and W. B. Croft, "Phrasal Translation and Query Expansion Techniques for Cross-Language Information Retrieval" in the Proceedings of the 20th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-97), pp. 84-91, 1997.

Ballesteros, L. and W. B. Croft, "Resolving Ambiguity for Crosslanguage Retrieval" in the Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-98), pp. 64-71, 1998.

Berry, M. and P. Young, "Using latent semantic indexing for multi-language information retrieval" Computers and the Humanities, vol. 29, no 6, pp. 413-429, 1995.

Davis, M., "New experiments in cross-language text retrieval at NMSU's Computing Research Lab" in D. K. Harman, ed., The Fifth Text Retrieval Conference (TREC-5), NIST, 1996.

Dumais, S. T., T.K. Landauer, M.L. Littman, "Automatic cross-linguistic information retrieval using latent semantic indexing" in G. Grefenstette, ed., Working Notes of the Workshop on Cross-Linguistic Information Retrieval. ACM SIGIR.

Kaburlasos, V.G., "Fuzzy Interval Numbers (FINs): Lattice Theoretic Tools for Improving Prediction of Sugar Production from Populations of Measurements," IEEE Trans. on Man, Machine and Cybernetics – Part B, vol. 34, no 2, pp. 1017-1030, 2004.

Kobayashi, M. and K. Takeda, "Information Retrieval on the Web," ACM Computing Surveys, 32, 2 (2000), pp 144-173.

Kraft, D.H. and D.A. Buell, "Fuzzy Set and Generalized Boolean Retrieval Systems" in Readings in Fuzzy Sets for Intelligent Systems, D. Dubius, H.Prade, R.R. Yager (eds) 1993.

Oard, D.W., "Alternative Approaches for Cross-Language Text Retrieval" in Cross-Language Text and Speech Retrieval, AAAI Technical Report SS-97-05. Available at http://www.clis.umd.edu/dlrg/filter/sss/papers/

Petridis, V. and V.G. Kaburlasos, "FINkNN: A Fuzzy Interval Number k-Near-est Neighbor Classifier for prediction of sugar production from populations of samples," Journal of Machine Learning Research, vol. 4 (Apr), pp. 17-37, 2003 (can be downloaded from http://www.jmlr.org).

Radecki,T, "Fuzzy Set Theoretical Approach to Document Retrieval" in Information Processing and Management, v.15, Pergammon-Press 1979.

Salton, G. and M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, 1983.

APPENDIX – FIN COMPUTATION

Consider a vector of real numbers $x = [x_1,x_2,\ldots,x_N]$ such that $x_1 \leq x_2 \leq \ldots \leq x_N$. A FIN can be constructed according to the following algorithm CALFIN where dim(x) denotes the dimension of vector x, e.g. dim([2,-1])= 2, dim([-3,4,0,-1,7])= 5, etc.

The median(x) of a vector $x = [x_1,x_2,\ldots,x_N]$ is defined to be a number such that half of the N numbers $x_1,x_2,\ldots,x_N$ are smaller than median(x) and the other half are larger than median(x); for instance, the median($[x_1,x_2,x_3]$) with $x_1 < x_2 < x_3$ equals x_2, whereas the median($[x_1,x_2,x_3,x_4]$) with $x_1 < x_2 < x_3 < x_4$ was computed here as median($[x_1,x_2,x_3,x_4]$)= $(x_2 + x_3)/2$.

Algorithm CALFIN

1. Let x be a vector of real numbers.
2. Order incrementally the numbers in vector x.
3. Initially vector pts is empty.
4. function calfin(x) {
5. while (dim(x) ≠ 1)
6. medi:= median(x)
7. insert medi in vector pts
8. x_left:= elements in vector x less-than number median(x)
9. x_right:= elements in vector x larger-than number median(x)
10. calfin(x_left)
11. calfin(x_right)
12. endwhile
13. } //function calfin(x)
14. Sort vector pts incrementally.
15. Store in vector val, dim(pts)/2 numbers from 0 up to 1 in steps of 2/dim(pts) followed by another dim(pts)/2 numbers from 1 down to 0 in steps of 2/dim(pts).

The above procedure is repeated recursively $\log_2 N$ times, until "half vectors" are computed including a single number; the latter numbers are, by definition, median numbers. The computed median values are stored (sorted) in vector pts whose entries constitute the abscissae of a positive FIN's membership function; the corresponding ordinate values are computed in vector val. Note that algorithm CALFIN produces a positive FIN with a membership function $\mu(x)$ such that $\mu(x)=1$ for exactly one number x.