SARIPOD: A POSSIBILISTIC SYSTEM FOR WEB INFORMATION RETRIEVAL

Bilel Elayeb[1], Fabrice Evrard[2], Montaceur Zaghdoud[1]
and Mohamed Ben Ahmed[1]

[1]Laboratoire RIADI-GDL, ENSI université de la Manouba, 2010 Tunisie.
{Bilel.Elayeb,Mohamed.Benahmed}@riadi.rnu.tn,
Montaceur.Zaghdoud@ensi.rnu.tn
[2]IRIT-ENSEEIHT, 2 rue Charles Camichel BP 7122 - 31071 Toulouse Cedex 7, France
Fabrice.Evrard@enseeiht.fr

ABSTRACT

This paper presents a web information retrieval system based on Hierarchical Small-Worlds (HSW) and Possibilistic Networks (PN). The first HSW consists in structuring the "Google" search results in dense zones of web pages which strongly depend on each other. We thus reveal dense clouds of pages which "speak" more or less about the same subject and which all strongly answer the user's query. The goal of the second HSW consists in considering the query as multiple in the sense that we don't seek only the keyword in the web pages but also its synonyms. The PN generates the mixing of these two HSW in order to organize the searched documents according to user's preferences. Indeed, SARIPOD is a new approach for Information Retrieval Model based on possibility and necessity measures. This model encodes relationship dependencies existing between query terms and web documents through naïve possibilistic networks and quantifies these relationships by two measures: possibility and necessity. The retrieved documents are those which are necessarily or possibly relevant given a user's query. The search process restores the plausibly or necessarily relevant documents for a user need.

KEYWORDS

Information Retrieval, Hierarchical Small-Worlds, Possibilistic Networks, Possibilistic relevance, user's preferences.

1. INTRODUCTION

The key issue of Information Retrieval (IR) is that documents must be retrieved from a large document collection in response to a user's need, often on the basis of poor information. Known models in the literature (Boolean, vector space, probabilistic, Bayesian) represent documents and queries only through weighted lists of terms and a measure of relevance is computed (vector space similarity, probabilistic relevance) based on those weighted lists. Devising a proper weighting scheme seems to be the fundamental element of actual IR models since the computation of relevance relies on it [14] [18]. Usually, the weighting scheme is the result of several combinations: term frequencies in document (tf), term frequencies in the whole collection (idf) and document length (dl) [15] [17]. Whatever the used model, the response to a user need is a list of documents ranked according to a unique relevance value. Many approaches consider term weights as a probability of relevance. In such models, the incompleteness of information is not considered when representing or evaluating documents given a query. Yet, the rough nature of document descriptions (a multiset of terms) and of the query description (a list of terms) are hardly compatible with the high precision of relevance values obtained by current methods.

The aim of this paper is to propose basic steps towards an IR mixed approach based on possibility and necessity measures. Instead of using a unique relevance value, we propose a

possibilistic approach for computing relevance. This model should be able to infer propositions like:

- It is plausible with a certain degree that the document is relevant for the user need.
- It is almost certain (in possibilistic sense) that the document is relevant to the query.
- The set D_1 of documents (possibly singleton) is better than the set D_2 of documents.

The first kind of proposition is meant to eliminate irrelevant documents (weak plausibility). The second answer focuses attention on what looks very relevant. The third proposition suggests that, since the raw information on documents is more qualitative than quantitative, ordinal approaches to the problem may be interesting as well. The use of probability theory in the definition of relevance given a query does not account for our limited knowledge of the relevance of a document, since it does not consider imprecision and vagueness intrinsic to relevance [7].

When we make a web search query on the Internet using key words (thanks to a search engine like Google), we obtain in general a considerable list of links of web pages answering this query. We can also, by using complementary keywords, seek something more relevant in the whole of the preceding results. And so on until obtaining required result. However we could differently proceed by structuring the whole of the web pages obtained in the first result. The structure would consist in classifying all these results by domains and sub-domains. A very promising technique emerges today and called upon the hierarchical small-worlds. Thus, each web page would be a node of a gigantic graph whose edges would be the hypertextual links of a page towards another. Certain calculations on this graph are capable of revealing regroupings sets of themes (web pages which "speak" almost about the same subject).

We present, in this framework, an Internet information retrieval system, called SARIPOD. This system is based on Hierarchical Small-Worlds (HSW) and Possibilistic Networks (PN). The first HSW consists in structuring the "Google" search results in dense zones of web pages which strongly depend on each other. We thus reveal dense clouds of pages which "speak" more or less about the same subject and which all strongly answer the user's query. The goal of the second HSW consists in considering the query as multiple in the sense that we don't seek only the keyword in the web pages but also its synonyms. The PN generates the mixing of these two HSW in order to organize the searched documents according to user's preferences.

The hierarchical small-worlds graph is presented in section 2. We briefly recall some notions of possibility theory in section 3. We describe in section 4 the general architecture of SARIPOD system and we also present the functionality of each module with some examples. The experimentation of our system is in section 5 and it is evaluated in section 6. Section 7 suggests future works.

2. HIERARCHICAL SMALL-WORLDS

Recent work in graph theory has revealed a set of features shared by many graphs observed "in the field". These features define the class of "hierarchical small-worlds" networks [20]. The relevant features of a graph in this respect are the following [13]:

D : the density of the network. HSWs typically have a low D, i.e. they have rather few edges compared to their number of vertices.

L : the shortest path average between two nodes. It is also low in a HSW.

C : the clustering rate. This is a measure of how often neighbors of a vertex are also connected in the graph. In a HSW, this feature is typically high.

I : the distribution of incidence degrees (i.e. the number of neighbors) of vertices according to the frequency of nodes (how many nodes are there that have an incidence degree of 1, 2, ..., n). In a HSW network, this distribution follows a power law: the probability P(k) that a

given node has k neighbors decreases as $k^{-\lambda}$, with $\lambda > 0$. It means also that there are very few nodes with a lot of neighbors, and a lot more nodes with very few neighbors.

As a mean of comparison, table 1 [11] shows the differences between random graphs (nodes are given, edges are drawn randomly between nodes), regular graphs and HSWs. The graph of the web belongs to the class of HSW [8] [16].

Table 1. Comparing classes of graphs.

with equal D	L	C	I
Random graphs	small L (short paths)	Small C (few aggregates)	Poisson Law
HSW	**small L (short paths)**	**High C (aggregates)**	**power law**
Regular graphs	High L (long paths)	High C (aggregates)	constant

We show, in section 6 of evaluation of SARIPOD system, the benefit of the use of HSW in SARIPOD, and their role structuring (organisational) for the objective of the search of information. Indeed, we prove that SARIPOD doesn't seek only information, but also made coherent answers, structured by possible fields of answers.

3. POSSIBILISTIC LOGIC

Possibility theory introduced by Zadeh [21] and developed by Dubois-Prade [10], handles uncertainty in the interval [0,1] called possibility scale, in a qualitative or quantitative way.

3.1. Possibility distribution

Possibility theory is based on possibility distributions. The latter, denoted by π, are mappings from Ω (the universe of discourse) to the scale [0,1] encoding partial knowledge on the world. The possibility scale is interpreted in two ways. In the ordinal case, possibility values only reflect an ordering between possible states; in the numerical scale, possibility values often account for upper probability bounds [9].

3.2. Possibility and necessity measures

A possibility distribution π on Ω enables events to be qualified in terms of their plausibility and their certainty, in terms of possibility and necessity measures respectively.

The possibility $\Pi(A) = \max_{x \in A} \pi(x)$ of an event A relies on the most normal situation in which A is true.

The necessity $N(A) = \min_{x \notin A}(1 - \pi(x)) = 1 - \Pi(\neg A)$ of an event A reflects the most normal situation in which A is false.

The width of the gap between N(A) and Π(A) evaluates the amount of ignorance about A. Note that N(A) > 0 implies Π(A) = 1. When A is a fuzzy set this property no longer holds but the inequality N(A) $\leq \Pi$ (A) remains valid [10].

3.3. Possibilistic Networks (PN)

A directed possibilistic network on a variable set V is characterized by a graphical component and a numeric component. The first one is a directed acyclic graph. The graph structure encodes independence relation sets just like Bayesian nets [2] [4]. The second component quantifies distinct links of the graph and consists of the conditional possibility matrix of each node in the context of its parents. These possibility distributions should respect normalisation. For each variable V:

- If V is a root node and dom(V) the domain of V, the prior possibility of V should satisfy: $\max_{V \in dom(V)} \Pi(V) = 1$;
- If V is not a root node, the conditional distribution of V in the context of its parents context should satisfy: $\max_{V \in dom(V)} \Pi(V \mid Par_V) = 1$; $Par_V \in dom(Par_V)$
  where: $dom(V)$: domain of V; Par_V : value of parents of V; $dom(Par_V)$: domain of parent set of V.

4. GENERAL ARCHITECTURE OF SARIPOD SYSTEM

To support the re-use, we choose a modular approach for SARIPOD system. The co-operation of the various modules makes it possible to generate the general architecture of our system composed of six modules (see figure 1).

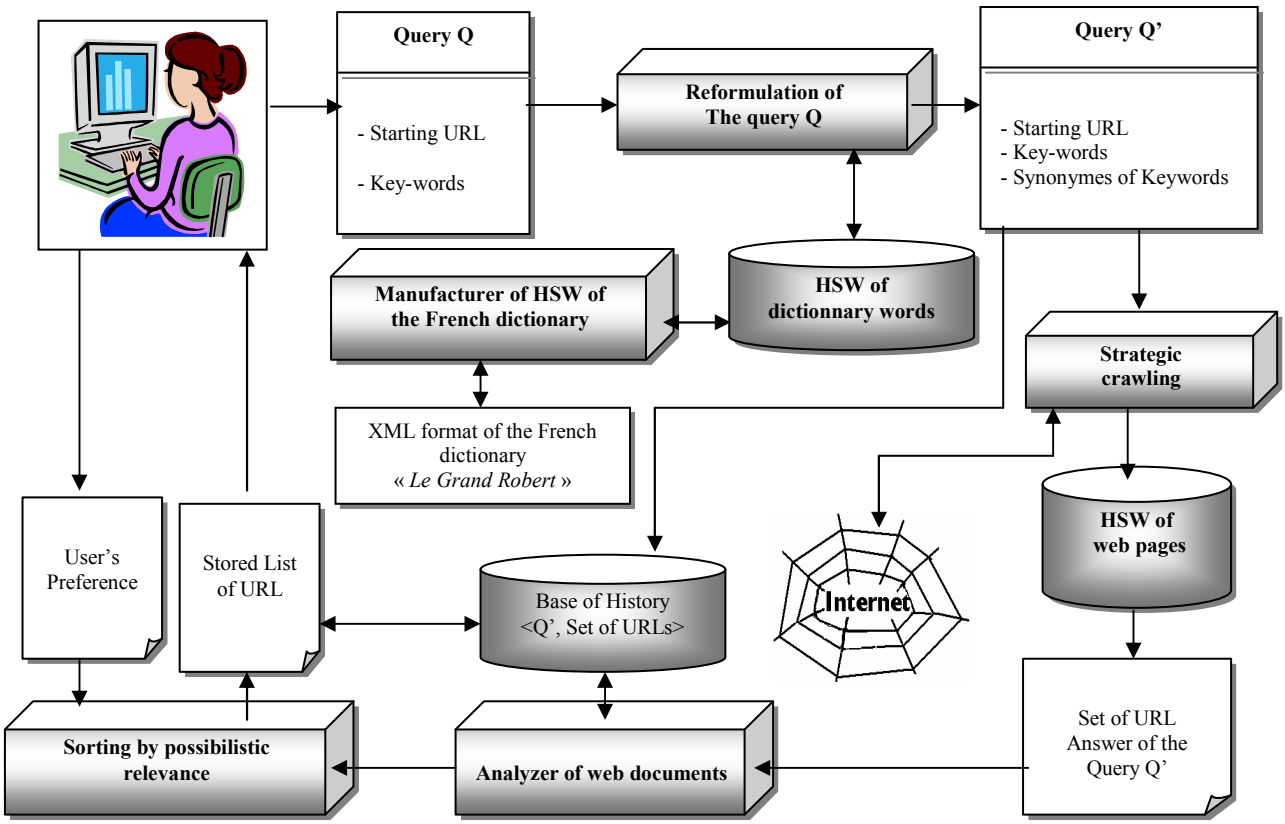

Figure 1. General architecture of SARIPOD system

4.1 HSW dictionary module

Our approach consists in representing the dictionary by a HSW graph: there is an arc of a word A towards a word B if and only if the entry B appears in the definition of entry A as a synonym. Consequently, the problem of synonymy in a dictionary is brought back to a study on the graphs seeking to exploit the networks thus established between the words. It acts, very often, to detect components having specific graph properties in terms of graph thus leading to the regrouping of synonymous words [1].

Our data base is the French dictionary "Le Grand Robert" in XML format in which each tagged element is associated with semantics. The enormous volume of this corresponding XML file constrained us to reduce the size of this dictionary by keeping just useful information to make a more effective treatment. This was done, obviously, without loss of relevant information. In

fact, we limit our vocabulary to nouns, verbs, adjectives and adverbs of this dictionary. It exempts us to use a stop-word list such as pronouns, definite and indefinite articles, prepositions, etc.

4.2 Reformulation of the user's query module

The reformulation of user's query is a primordial stage in SARIPOD system. Indeed, this module sends user's keywords to HSW dictionary module which determines their synonyms and expends them to the reformulation of the user's query module. With the reception of these synonyms, this module provides consequently, a new query Q' containing the starting URL chosen by the user of system and the set of keywords and their synonyms. In fact, the new query Q' is well enriched by news words.

4.3 Strategic crawl module

A web crawler is a program which automatically traverses the web by downloading documents and following links from page to page. It starts from a starting URL of a web page and it has a changeable depth of propagation [12]. This module accepts in entry the query Q' and generates as output the HSW of web pages and a set of their URLs.

We propose within this framework a systematic crawling technique via "Strat" algorithm, whose scenario is as follows :

1. While a page on N contains the word *W* it is necessary to continue to visit the outgoing pages of this page, for that would be nevertheless a shame not to go to visit these outgoing pages which have a very strong probability of containing the word *W*;
2. When N successive pages do not contain the word *W* (whatever the depth), we stop research in this branch, so we need then to backtrack.

In practice we took N=2 by carrying out several tests and by noting that N=2 is more flexible than N=1 (algorithm which would stop too quickly) and N>2 (algorithm which would keep too many pages).

4.4 Web document analysis module

This module allows the extraction of the logical structure of each web document. We mind to store such a document in an editable and exchangeable format that represents explicitly its structure and its content. The strategy of this module is based on a labelling method. It is composed of several analysis steps that leads to the transformation of the document in a logical structure where each text block has a level and a label that represents explicitly its logical role [5]. Searching the user's keywords in such logical entity of document and not in the whole document proves the qualitative character of the SARIPOD system.

4.5 Sorting documents by possibilistic relevance module

We allotted a coefficient of relevance to each logical entity (LE) according to its importance in the web document. These coefficients are calculated according to the following way:

$$\alpha_{ML} = ML + \max(\alpha_{Legends}, \alpha_{Paragraph}),\ \alpha_{Li} = ML - L_i + \max(\alpha_{Legends}, \alpha_{Paragraph})$$

Where *ML* is the maximal level, L_i is the level *i* of logical entity.

The quantitative relevance of each logical entity of a web document of the collection, with the query is $Q = (t_1, t_2, \ldots, t_T)$, is calculated in the following way:
The expression of $\Pi(LEd_j|Q)$ is then proportional to:

$$\Pi'(LEd_j \mid Q) = \Pi(t_1 \mid LEd_j) * \ldots * \Pi(t_T \mid LEd_j) = nft_{1j} * \ldots * nft_{Tj}$$

Where $nft_{ij} = tf_{ij} / \max(tf_{kj})$: the normalized frequency of the terms of the query in the logical entity.

The certainty to restore a logical entity of a relevant document d_j for a query, noted $N(LEd_j|Q)$, is given by: $N(LEd_j \mid Q) = 1 - \Pi(\neg LEd_j \mid Q)$

Where: $\Pi(\neg LEd_j \mid Q) = (\Pi(Q \mid \neg LEd_j) * \Pi(\neg LEd_j)) / \Pi(Q)$

In the same way, $\Pi(\neg LEd_j| Q)$ is then proportional to:

$\Pi'(\neg LEd_j \mid Q) = \Pi(t_1 \mid \neg LEd_j) * ... * \Pi(t_T \mid \neg LEd_j)$

This numerator can be expressed by: $\Pi'(\neg LEd_j \mid Q) = (1 - \phi LE_{1j}) * ... * (1 - \phi LE_{Tj})$

Where: $\phi LE_{ij} = Log_{10}(nCLE / nLEd_i) * (nft_{ij})$

And :

nCLE = The number of logical entity of the documents of the collection,

$nLEd_i$ = The number of logical entity of the documents of the collection, containing the term t_i,

Let us note the degree of relevance mixed possibilistic of logical entity of the document d_i by:

$DRMPLE_j(d_i) = \Pi(LEd_i \mid Q) + N(LEd_i \mid Q)$

We note finally the degree of relevance mixed possibilistic of the document d_i by:

$DRMP(d_i) = \sum_j (\alpha_j * DRMPLE_j(d_i))$

User's preferences of SARIPOD system are defined as the quality of the document which he seeks; i.e. his preferences for certain stylistics attributes in the searched documents : information located either in the principal title of the document, or in the sub-titles, or in the paragraphs... and also his preferences for certain types of informations : information in figures, tables or multimedia sequences.

The preferred documents are those which have a high value of $DRMP(d_i)$. Let us note that the α_j are parameterized in our system and can be modified according to the user's preferences. Indeed, if we seek, for example, the documents containing the word "M" in figures, it is enough to give the greatest importance to the coefficient of relevance corresponding to the figure legend (α_{FL}). Consequently, $DRMP(d_i)$ of these documents will be most significant and will be posted in the heading of the list result of sorted documents.

Example. Assume a three documents collection containing the four terms t_1, t_2, t_3 and t_4:

$d_1 = \{t_1, t_1, t_1, t_2, t_2, t_3\}$, $d_2 = \{t_1, t_1, t_2, t_2, t_2, t_2\}$ and $d_3 = \{t_1, t_3, t_3, t_3, t_3, t_4, t_4\}$

These terms are distributed in the logical entities of these three documents as table 2 indicates.

Table 2. Distribution of terms in the logical structures of the three documents

Logical structure of document	**d_1**	**d_2**	**d_3**
Maximal Level (ML)	t_1	t_1, t_2	t_4
ML-1	t_2		t_1, t_3
ML-2			t_3
ML-3		t_2	
ML-4	t_3		
Figure Legend (FL)			t_3
Table Legend (TL)	t_2		
Multimedia Sequence Legend (MSL)		t_2	
Paragraph (P)	t_1, t_1	t_1, t_2	t_3, t_4
DRMP(d_i)	**11.28**	**11.76**	**19.07**

The evaluation of the documents d_1, d_2 and d_3 for the query $Q = (t_1, t_2, t_3, t_4)$ gives (for the three documents only representatitivity is given):

$\Pi(LE_j d_i|Q) = 0, \forall LE_j \in \{ML, (ML-1), (ML-2), (ML-3), (ML-4), FL, TL, MSL, P\}, \forall i=1, 2, 3$
$N(ML_{d1}|Q) = N(P_{d1}|Q) = 0.18,$
$N((ML-1)_{d1}|Q) = N((ML-4)_{d1}|Q) = N(TL_{d1}|Q) = N((ML-3)_{d2}|Q) = N(MSL_{d2}|Q) = N((ML-2)_{d3}|Q) = N(FL_{d3}|Q) = N(ML_{d3}|Q) = 0.48, N(ML_{d2}|Q) = N(P_{d2}|Q) = 0.58,$
$N((ML-1)_{d3}|Q) = N(P_{d3}|Q) = 0.73.$

The query Q interpreted as a conjunction of terms is too restrictive since no document contains all query terms simultaneously. Thus, necessity and possibility degrees of the documents equal 0. To avoid such case, we retrieve documents contained at least two terms and, if not productive, at least one term. If the query does not involve enough terms, the possibility of documents is equal 1 and their necessity 0. We then seek the documents which treat sets $\{t_1, t_2\}$ or $\{t_1, t_4\}$, or $\{t_2, t_4\}$. We see through this example, the need for allowing the user to express preferences between the query terms.

For the example of this query Q, the document d_3 is more preferred than documents d_2 and d_1. We notice that the most relevant document is the one whose query's terms exist in its logical entities having the significant coefficients of relevance α_j, such as the maximum level (e.g., the principal title of the document), Maxlevel -1 (e.g., title 1), Maxlevel -2 (e.g., title 2), etc. Thanks to our mixed possibilistic approach, we also noticed that: even if the selected terms tend to select this document, these terms are not most frequent of the document (t_4 isn't the most frequent term of d_3).

4.6 Optimization system module

The optimization system module of SARIPOD allows its users a significant profit in term of response time of our system. Indeed, this module makes it possible to build a base of history of the querys and their answers, already passed by the system. In the reception of a new query, the system consults this base of history, seeks the nearest query in this base, using Case Base Reasoning (CBR) technique [3] and finally, it updates the answer by eliminating URLs that are not available on the web and by adding nonexistent new URLs in this base of history. This requires the expend of the new query to the strategic crawl module. This new obtained answer will be useful, in the same way, like history for later queries. Let us note that the user will be able to use the result of the history directly or to change his preferences while launching a new sorting of the documents according to his new parameters.

5. EXPERIMENTATION

We made the tests of SARIPOD system on a local site containing 10 000 pages and for keywords having a variable number of synonyms. Table 3 gives our preliminary results.

The determination of keywords' synonyms is very dependent on the degree of cleaning of French dictionary "Le Grand Robert", used as a data base of the words' HSW. According to table 3, we notice that any increase of the synonyms' number gives more chance to collect more URLs, and increases consequently the duration of of the query treatment. Thanks to successive tests, we deduced the number three, as an optimal value of synonyms, for each user's keyword, by taking account of the quality of the collected documents as well as the duration of the query treatment.

We also notice that the difference between the degree of possibilistic relevance of the most relevant page (DRMP (d_1)) and of the least relevant page of the collection (DRMP (d_N)) decreases when the number of selected URL increases. It proves that the first goal which motivated us for the use of the HSW is checked here: the change of answers to research query given by *Google* web search engine, by structuring it in a HSW so that, if a page among the returned answers seems relevant then all neighbours in this HSW will be too. So, we increase

the number of Google's returned documents and we consequently change the Google's PageRank [19].

Table 3. Experimentations results

Number of synonyms	Required keywords	Numbers of URLs obtained	Duration of research query (in second)	Degree of relevance of document 1 DRMP (d_1)	Degree of relevance of document N DRMP (d_N)
0	vérifier	61	15	39,22	7,33
1	vérifier examiner	93	17	29,74	5,77
2	vérifier examiner voir	207	20	16,23	3,69
3	vérifier examiner voir éprouver	363	21	9,53	2,87
4	vérifier examiner voir éprouver reconnaître	412	24	7,19	1,88
5	vérifier examiner voir éprouver reconnaître essayer	517	29	5,37	1,45
6	vérifier examiner voir éprouver reconnaître essayer contrôler	761	35	3,86	0,83
7	vérifier examiner voir éprouver reconnaître essayer contrôler expérimenter	833	42	1,34	0,67
8	vérifier examiner voir éprouver reconnaître essayer contrôler expérimenter constater	904	55	0,53	0,18

6. EVALUATION

As evaluation of the SARIPOD system we present on one hand a justification of the use of the HSW graph in information retrieval, and on the other hand its evaluation compared to the traditional web search engine systems such as Google [19]. Indeed, we distinguish two very significant uses of these two HSW and their mixing in SARIPOD system:

The first use consists in structuring the "Google" search results in dense zones of web pages which strongly depend on each other. We thus reveal dense clouds of pages which "speak" more or less about the same subject and which all strongly answer the user's query. For another cloud of web pages strongly related to each other, it is similar: all of them answer this same query. The essential difference is that each cloud of web pages strongly answers the query in a particular way.

For example, the query "jouer", in the HSW of french synonyms, gives four clouds of verbs close to "jouer": the first cloud concerns A = {parier, risquer, miser, hasarder,...}, the second B = {tromper, mystifier, abuser, berner,...}, etc. for the two others. For the web, it's the same thing; a query (expressed with some keywords) returns a set of web pages (Google answers) which it's necessary to organize in HSW to reveal some large clouds of web pages among all these answers. Each cloud gathers a batch of pages which answer the query in relevant ways: as A pertinently answers the query "jouer" if A is interested in the "pari", as B which also answers pertinently the same query "jouer" if B is interested in the "abus ", etc. For the web each cloud of web pages will be relevant and, thanks to additional keywords, it will be possible to select a particular cloud.

Quality lies in the fact that when we look at the web pages of the same cloud, all the pages are relevant, but if this degree is not yet sufficient, we can only make queries in this only cloud (contrary to Google which never organizes its 300.000 answers in clouds) to obtain a subset of web pages which we can again (thus recursively) organize in under-HSW. With the deepest of this structure we find web pages alone. The set of answers was thus organized in HSW and

under-HSW to constitute a kind of decision tree (or structure of classification) on web pages according to the used keywords. Google can't do the same thing, but it can only search again in the set of preceding answers. In fact, Google is able to return web pages which our system would have put them in different clouds since the first query.

The second very significant use of the HSW consists in not taking the keywords just as they are but regarding a query as multiple in the sense that we don't search only the keyword in the web pages but also its synonyms. In fact, beyond strict synonymy, we will search for this keyword but also words close to it. The proximity of two words relies on circuits in the dictionary HSW (see section 4.1). The words considered as nearly relations thus include the synonyms of this word but don't narrow down to them. There will be potentially (in practice that will be limited by a terminal) all the words more or less near query's keyword. This number of words is skeletal (1, 5, 100...). A query is thus now very flexible since it tolerates that a web page is a good answer even if it doesn't contain the searched keyword.

However to be able to have this flexibility we need obviously a dictionary and especially to have structured this dictionary (all its entries) in HSW to precisely know which word is near to which other. However there are many ways of emerging a structure of HSW starting from a dictionary (that of Gaume and al. [11] for example consists in using words' definitions: The word W_1 is connected to the word W_2 if and only if W_2 belongs to the definition of W_1, using this relation he deduces a "semantic proximity" from any word to any other). Our system SARIPOD takes again this definition and calculates the proximity between the words in ordre to make the query more flexible. We can quantify from there the web pages obtained following a query using certain keywords. Each answer page will be characterized by a degree of relevance which will result from the combination of the degrees of proximity between the query's keywords and the words effectively present in this page.

7. CONCLUSION

This paper presents a web information retrieval system based on Hierarchical Small-Worlds (HSW) and Possibilistic Networks (PN). The first HSW consists in structuring the "Google" search results in dense zones of web pages which strongly depend on each other. We thus reveal dense clouds of pages which "speak " more or less about the same subject and which all strongly answer the user's query. The goal of the second HSW consists in considering the query as multiple in the sense that we don't seek only the keyword in the web pages but also its synonyms. The PN generates the mixing of these two HSW in order to organize the searched documents according to user's preferences.

Indeed, SARIPOD system is a new approach for Information Retrieval Model using possibilistic calculus. The approach proposed by Brini and al. [6] is only based on the quantitative setting of possibility calculus, our mixed approach extends it by also focusing on the ordinal setting. In fact, we add a qualitative approach which introduces preferences between retrieved documents. A user of our system has the possibility then to have his retrieved documents in any wished order via an interface which allows him to choose his own preference parameters. This approach had been implemented within Java language and test results were very encouraging.

Our tests are only limited to the verbal occurrences in the dictionary HSW of verbs definitions, but we plan to extend the tests to other grammatical categories, like refining our approach for the substantives by considering for example also the verbal occurrences in names definitions, etc. We also plan to carry out finer measurements of the performances of SARIPOD system by extending the tests with other types of web documents and by giving preferences between query terms. We will also hope to show how this model based on possibility calculus could be a counterpart to Bayesian probabilistic models.

REFERENCES

[1] Awada, A. & Chebaro, B. (2004). "Etude de la synonymie par l'extraction de composantes N-connexes dans les graphes de dictionnaires", *JEL2004*, Nantes, France.

[2] Benferhat, S., Dubois, D., Garcia, L. & Prade, H. (2002). "On the transformation between possibilistic logic bases and possibilistic causal networks", *Int. Journal of Approximate Reasoning*, 29, n°2, 135-173.

[3] Berry, M. J. A. & Linof, G. (1997). *Data Mining : Techniques appliquées au marketing, à la vente, et aux services clients*, InterEditions, Paris.

[4] Borgelt, C., Gebhardt, J. & Kruse, R. (2000). "Possibilistic Graphical Models, Computational Intelligence in Data Mining", *CISM Courses and Lectures*, 408, 51- 68.

[5] Bounhas, I., Habacha, A. & Ben Ahmed, M. (2007). "A Labelling Based Approach for Logical Structuration of Scientific Web papers", In *The ICSOFT'* 2007, Barcelona, Spain.

[6] Brini, A. H., Boughanem, M. & Dubois, D. (2004). "Towards a possibilistic approach for information retrieval", *Data and Knowledge Engineering Proceedings EUROFUSE 2004*, Varsovie, Pologne, p. 92-102.

[7] Brini, A. H. & Boughanem, M. (2003). "Relevance feedback : introduction of partial assessments for query expansion", *Proc. 2d. EUSFLAT Conf*, Zittau, Allemagne, 67-72.

[8] Douglas, W. & Houseman M. (2002). "The navigability of strong ties: small worlds, ties strength and network topology", *eScholarship Repository*, University of California.

[9] Dubois, D. & Prade, H. (1998). "Possibility Theory : Qualitative and Quantitative Aspects", In : *Handbook on Defeasible Reasoning and Uncertainty Management Systems*, vol. 1, Kluwer , 21-42.

[10] Dubois, D. & Prade, H. (1987). *Théorie des Possibilités : Application à la Représentation des Connaissances en Informatique*, Edition MASSON. Paris.

[11] Gaume, B., Hathout, N. & Muller, P. (2004). "Désambiguïsation par proximité structurelle", *TALN 2004*, Fès, Maroc.

[12] Miller, R. C. & Bharat, K. (1998). "SPHINX : A framework for creating personal, site-specific web crawlers", In *the 7th International World Wide Web Conference (WWW7)*, Computer Network and ISDN System v.30, pp. 119-130, Brisbane, Australia.

[13] Newman, M. E. J. (2003). "The structure and function of complex networks", *SIAM Review*, volume 45, 167–256.

[14] Ribeiro-Neto, B., Silva, I. & Muntz, R. (1996). "A Belief Network Model for IR", *Proc. Of the 19th ACM SIGIR Conf. on Research and Development in Information Retrieval*, 253-260.

[15] Salton, G., Allan, J., Buckley, C. & Singhal, A. (1994). "Automatic Analysis, Theme Generation and Summarization of Machine Readable Texts", *Science*, 264, 3, 1421-1426.

[16] Sergi, V. & Ricard, V. S. (2004). "Hierarchical Small Worlds in Software Architecture", *IEEE transaction on Software Engineering*, USA.

[17] Singhal, A., Salton, G., Mitra, M. & Buckley, C. (1996). "Document Length Normalization", *Information Processing and Management*, 32(5) : 619-633.

[18] Sparck, J. K. (1998). "A Look Back and a Look Forward", *Proceedings of the 11th annual international ACM SIGIR Conf. on Research and Development in Information Retrieval*, 13-29.

[19] Vise, D. & Malseed, M. (2006). *Google story*, Edition Dunod, Paris.

[20] Watts, D. & Strogatz, S. (1998). "Collective dynamics of 'small-world' networks", *Nature*, (393), 440–442.

[21] Zadeh, L. A. (1978). "Fuzzy Sets as a basis for a theory of Possibility", *Fuzzy Sets and Systems*, 1 :3-28.