Indonesian Journal of Electrical Engineering and Computer Science
Vol. 2, No. 1, April 2016, pp. 205 ~ 214
DOI: 10.11591/ijeecs.v2.i1.pp205-214

Information Retrieval: Textual Indexing Using an Oriented Object Database

Mohammed Erritali
TIAD laboratory, Computer Sciences Department, Faculty of sciences and techniques, Sultan Moulay Slimane University, Beni-Mellal, BP: 523, Morocco
e-mail: m.erritali@usms.ma

Abstract

The growth in the volume of text data such as books and articles in libraries for centuries has imposed to establish effective mechanisms to locate them. Early techniques such as abstraction, indexing and the use of classification categories have marked the birth of a new field of research called "Information Retrieval". Information Retrieval (IR) can be defined as the task of defining models and systems whose purpose is to facilitate access to a set of documents in electronic form (corpus) to allow a user to find the relevant ones for him, that is to say, the contents which matches with the information needs of the user. Most of the models of information retrieval use a specific data structure to index a corpus which is called "inverted file" or "reverse index". This inverted file collects information on all terms over the corpus documents specifying the identifiers of documents that contain the term in question, the frequency of each term in the documents of the corpus, the positions of the occurrences of the word. In this paper we use an oriented object database (db4o) instead of the inverted file, that is to say, instead to search a term in the inverted file, we will search it in the db4o database. The purpose of this work is to make a comparative study to see if the oriented object databases may be competing for the inverse index in terms of access speed and resource consumption using a large volume of data.

Keywords*: Information Retrieval, indexation, oriented object database (db4o), inverted file*

1. Introduction

Due to the rapid growth in the volume of electronically stored information, the major problem which arises is to respond to a search query with relevant manner from a set of unstructured documents in a database called the corpus. This research problem is known as Information Retrieval (IR).

Information Retrieval can be defined as a set of techniques and tools dealing with access to information and its presentation, its organization and its storage [1], [2].

The term "information retrieval" is given by Calvin N. Mooers in 1948 for the first time in his thesis [3].

According to [18], an SRI is a set of computer programs that aims to select relevant information that meets users needs expressed in the form of queries. Lancaster cited in [19] notes that a SRI does not inform the user on the subject of his research .it simply reports the existence or non-existence of documents relating to his request.

From the above definitions we can deduce that a user translates its needs in a structured way as a query that it transmits to information retrieval system.

This one has as a main task to return to the user the maximum of relevant documents in relation to his need (minimum of irrelevant documents). For this, the information search system connects the available information (the corpus documents) and the requirements of the user (the user query).

In the literature we find several representations of the process of information retrieval [20]-[22] which show that the mapping information contained in a corpus on the one hand, and information needs of users on the other hand, is done through two mechanisms: indexing and search.

This operation is provided through a process known as the process in U as shown in Figure 1.

Received December 29, 2015; Revised March 12, 2016; Accepted March 27, 2016

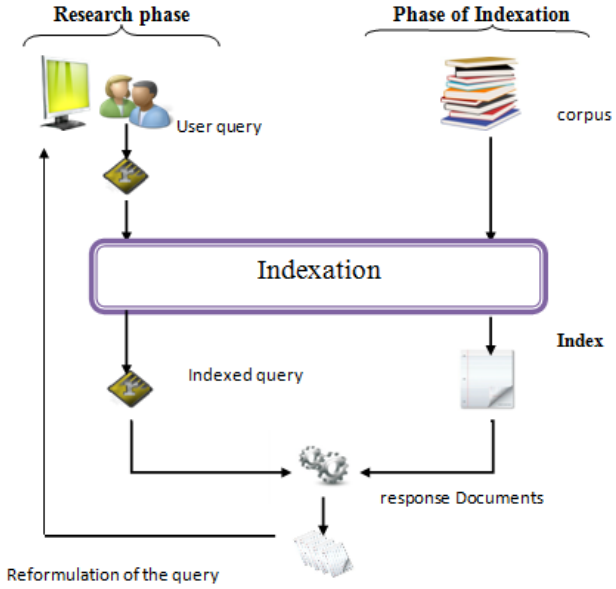

Figure 1. Process of Information Retrieval

2. Indexation for Information Retrieval

In information retrieval systems, the query and documents in the corpus are difficult to use in its raw state. To organize documents and the query as an intermediate representation to reflect as closely as possible their content, techniques and models are implemented. These techniques can describe the documents and the request by a set of descriptors. This process of representation is called the indexing process. Indexing consists in analyzing the documents and the query to extract a set of descriptors [2], [7], [5], [10], [11].

The descriptor of a document or a query is a list of words or groups of significant terms for the corresponding textual unit, usually accompanied by a weight representing the degree of representation of the content that they describe [12].

A. Indexation Approaches

Indexing is traditionally performed manually that is to say a human operator, usually a librarian or a domain expert is responsible for characterizing, according to his knowledge, the content of a document. The analysis of Document is performed by a person, not a machine, which is very costly in time because on the first hand indexer must read and understand a document before it can be properly indexed. On the other hand this type of indexing is practically inapplicable to the large corpus of texts [10], [12].

This approach has another drawback that it is subjective, since the choice of indexing terms depends on the indexer and its domain knowledge. With the increase of the amount of documents to be indexed, indexing tends to be automated [10].

Automatic indexing is a completely automated process that is charged to extract words that characterize the document [14]. The advantage of this approach lies in its ability to process text faster than the previous approach, and therefore, it is particularly suitable for large corpus [12], [2], [5].

One of the problems of automatic indexation is located at the semantics of the words, because the process of automatic indexing counts the number of occurrences of a word without taking into account the meaning of each word. For example, the word “orange” in French means a color and a fruit. If the two meanings of the word are used in a single text the word will be counted twice when he was not the same.

Another disadvantage of the automatic indexing is in compound words. For example, consider the compound word "pomme de terre" in French. It will be indexed at "pomme" and "terre" but not at "pomme de terre" which is its original meaning.

There is an intermediary method of indexation is the semi-automatic indexation or controlled indexation. In this type of indexing a first automatic process is used to extract terms of

the document. However, the final choice rest to the specialist in the field to establish the relationship between words and select the significant terms. In this paper we are particularly interested in the automatic indexation approach.

B. Indexing Languages

The vocabulary of indexing language is formed from the set of indexation terms. This section presents the two main types of indexing language [10]-[12]: free language and controlled language.

- ✓ Controlled indexation language is constructed from a set of pre-defined terms and usually organized in a thesaurus. When a document is analyzed, we keep only words belonging to this thesaurus.
- ✓ The free language is a language close to our natural language (NL). In this language descriptor is automatically extracted from documents, or user request. This type of indexing is especially used by search engines making a fully automatic indexing as Google.

C. Automatic Indexing Process

Automatic indexing is a set of automated processes on a document which are: segmentation, removal of empty or stop words, stemming or radicalization of words, and weighting.

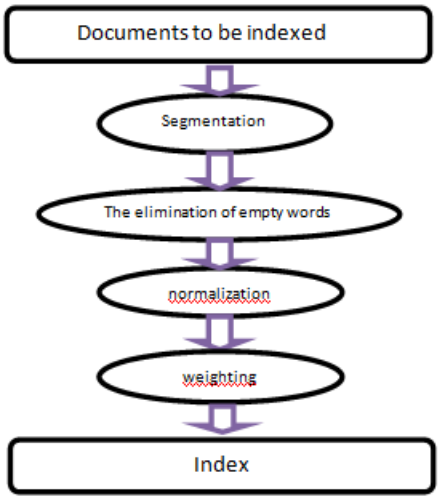

Figure 2. Phases of automatic indexation

1) The Tokenization (Segmentation)

The tokenization is also called segmentation. It consists to divide the text into elementary tokens. This is an operation which "locates" strings surrounded by separators (white space, punctuation), and identifies them as words.

2) Elimination of Empty Words (Stop Words)

Stop words (empty words) are prepositions and conjunctions. Elimination of empty words reduces the index, then we gain in storage space, but also the no treatment of empty words reduces the execution time of a System of information retrieval [8]. Seen that reducing the number of terms increases the performance, some systems consider, too, such as empty words some verbs, adjectives and adverbs. There are two techniques to filter out empty words:

- ✓ The use of a predefined list of stop words (also called anti-dictionary / stop-list).
- ✓ Counting the number of occurrences of a word in a document collection. Followed by striking with a frequency that exceeds a certain threshold and become empty words.

In this work we have chosen to use the first technique which is the anti-dictionary, in this phase the treatment is simple: if a term of the corpus appears in the anti-dictionary, it is not considered as an index term.

3) Normalization of Index Terms

Normalization is a process that allows grouping the morphological variants of words as a single base. Its goal is to keep in the indexing language, the forms of representative words, which offers considerable gain of storage memory and an effective research. The normalization is based on one of two procedures: Stemming or lemmatization [13].

a) Lemmatization

Lemmatization is used to group the words of the same grammatical category and transform them to their canonical form called lemma (e.g. different forms of a verb are transformed to infinitive) [7], [10]. This technique is based on the use of software and resources on lemmatization namely: TreeTagger, WinBrill and LEFFF.

Some lemmatizers can treat multiple languages (e.g. TreeTagger treats the English and German languages).

b) Stemming

Stemming transforms a word to its root. A stemmer seeks the root of a word based on its shape and the desired language. For example in French: "écologie, écologiste, écologique" are stemming by one word: "écologie" [8] [13].

In the literature there are several algorithms that are used in stemming as the algorithm of Lovins [23], Paice / Husk [24] algorithm and Porter [25] algorithm.

Snowball [26] is another Stemming tool which was invented by Martin Porter (the creator of the Porter algorithm). There are Snowball stemmers for various languages (French, English, Spanish …)

Experiments have shown that the Stemming and lemmatization significantly increases the search performance for morphologically rich languages such as French and Italian [13].

4) Weighting of Terms

To measure the importance of a word in a document indexing uses the concept of weight. The weighting is to assign a weight to terms of indexing and search. This weight is used to specify the relative importance of words represented in the documentation with respect to those identified in the request. The weighting consists to answer the question if all terms have the same importance and how to assign a weight to the extracted terms?

In general, the weighting formulas used are based on the combination of a local weighting factor quantifying the local representation of the word in the document [4], and a global weighting factor quantifying the overall representation of the term with respect to the collection of documents [2] [10].

✓ Local Weighting

Local weighting is used to measure the local representation of a term. It takes into account the local information of the term in relation to a given document. It indicates the importance of the term in this document. This weighting is generally measured by the frequency of the term t_j (term frequency, denoted tf_{ij} in the document $d_i$considered.

✓ Global Weighting

The global weighting is based on the idea that a term does not distinguish the documents from each other during the search, if it is distributed uniformly in all documents in the collection. Thus, this term does not have any discriminatory power. Therefore, the terms that appear in few documents are discriminating and weights are assigned to them. This weighting is expressed by the inverse document frequency idf_j of a term t_j in the collection. It is generally defined by the following formula:

$$idf_j = \log\left(\frac{N}{n_j}\right)$$

Where N is the number of documents in the collection; n_j is the number of documents indexed by the term t_j .

Salton [6] has defined a weighting formula tf * idf by:

$$w_{ij} = tf_{ij} * idf_j = tf_{ij} * \log\left(\frac{N}{n_j}\right)$$

The measure tf * idf is a good approximation of the importance of a term in the document collections composed of document with homogeneous sizes. However, for collections containing documents of varying sizes, words in longer documents appear frequently with very high weight than those in shorter documents. So the longer documents are more likely to be selected [2], [13], [15].

3. Models of Information Retrieval

An information retrieval system is based on a theoretical model. This model allows us to interpret the notion of relevance of a document with respect to a query in a formal setting. It therefore provides a theoretical cadre for modeling relevance of this measure [7].

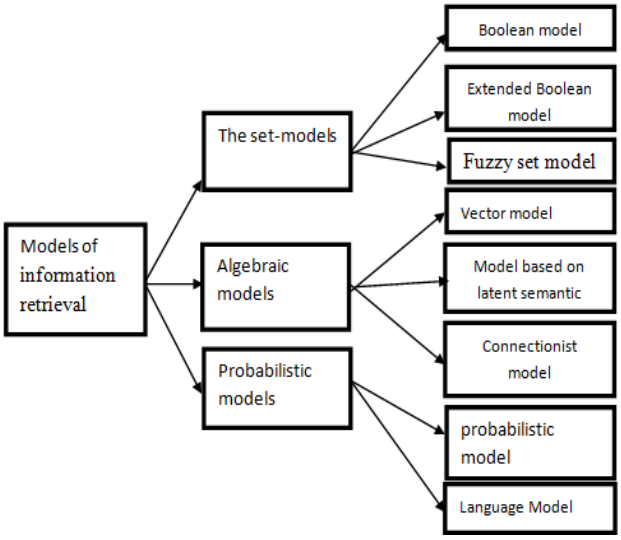

Figure 3. The various models of information retrieval

In the literature, many models of information retrieval have been proposed (Figure 3). They are divided into three major categories which are the set-models, vector models and probabilistic models.

The Boolean model was the first model to be used in information retrieval, because of its simplicity. However, the lack of weight in this model limits its uses. Thus, extended versions of this model have been proposed, they include weighting, such as the use of fuzzy set theory [7], [10]. The vector model is probably the most widely used in information retrieval. Its popularity is due to its ability to order found documents and its good performance. Probabilistic models are based on probability theory; the performance of these models appears in the 1990 years [10].

We present in the following the principle of the three models: Boolean model, vector model and probabilistic model.

a) Boolean Model

The Boolean model was introduced in 1983 by Salton and McGill [6]. This model is the oldest model in the field of information retrieval. It was emerged due to the simplicity and speed of its implementation. The query interface of most search engines (Google, Alta Vista) is based on the principles of this model. The Boolean model is based on set theory. The query is represented as a logical expression. In this expression, the descriptors are combined together using the Boolean operators "¬ NOT", "Λ AND" and "v OR ". Documents satisfying the logical expression representing the query are considered relevant [2] [7].

$$RSV(d, t_i) = 1 \ if \ t_i \in d \ , \ 0 \ otherwise$$
$$RSV(d, t_i \ ET \ t_j) = 1 \ if \ (t_i \in d) \wedge (t_j \ d) \ , 0 \ otherwise$$

$$RSV(d, t_i\ OU\ t_j) = 1\ \ if\ \ (t_i \in d) \vee (t_j\ d)\ , 0\ otherwise$$
$$RSV(d,\ NON\ t_i) = 1\ \ \ if\ \ t_i \notin d\ , 0\ \ \ \ otherwise$$

Although this model is simple to implement, it has major drawbacks [10]:

- ✓ matching is strict and does not allow documents to be classified in two categories, the relevant documents and non-relevant documents, whose terms are not orderable, and all terms of a document or a query are equal in importance (weighted at 0 or 1), which is not the case in reality,
- ✓ Boolean expressions are not accessible to a wide audience and confusion exist because of the difference in "meaning" of the logical AND and OR and their connotations in natural language operators.

To overcome these drawbacks, the extended Boolean model [2] [7] [10] has been proposed. It takes into account the importance of terms in the document representation and the query by assigning weights to each word of the document and the query.

b) The Vector Model

The vector model was proposed by G. Salton [6]. It is based on mathematical bases of vector spaces. In this model documents and the query are represented by vectors in indexing space i.e. the coordinates of a document represent the weight of their words. Formally, a document d_i is represented by a vector of a dimension N which is the number of indexing terms of the collection of documents [2] [10].

$$d_i = (w_{i1}, w_{i2}, w_{i3}, \ldots\ldots, w_{in}) \quad i = 1,2,3, \ldots, m$$

Where w_{ij} is the weight of term t_j in the document; d_i, m is the number of documents in the collection, and n is the number of index terms.

A query Q is represented by a vector of keywords defined in the same space vector as the document.

$$Q = (w_{Q1}, w_{Q2}, w_{Q3}, \ldots\ldots, w_{Qn})$$

Where w_{Qj} is the weight of term t_j in the Q query.

Relevance of the document d_i for a query Q is measured as the degree of correlation of the corresponding vectors. This correlation can be expressed by one of the following measures [2], [5], [10], [16]:

The scalar product [10]:

$$RSV(d_i, q) = \sum_{j=1}^{n} w_{Qj} * w_{ij} \tag{1}$$

The cosine measure [10], [14]:

$$RSV(d_i, q) = \frac{\sum_{j=1}^{n} w_{Qj} * w_{ij}}{\sqrt{\sum_{j=1}^{n} w_{Qj}} * \sqrt{\sum_{j=1}^{n} w_{ij}}} \tag{2}$$

The measure of Dice [10]:

$$RSV(d_i, q) = \frac{2 * \sum_{j=1}^{n} w_{Qj} * w_{ij}}{\sum_{j=1}^{n} {w_{Qj}}^2 * \sum_{j=1}^{n} {w_{ij}}^2} \tag{3}$$

The measure of Jacard [10]:

$$RSV(d_i, q) = \frac{\sum_{j=1}^{n} w_{Qj} * w_{ij}}{\sum_{j=1}^{n} {w_{Qj}}^2 * \sum_{j=1}^{n} {w_{ij}}^2 - \sum_{j=1}^{n} w_{Qj} * w_{ij}} \tag{4}$$

Superposition coefficient [10]:

$$RSV(d_i, q) = \frac{\sum_{j=1}^{n} w_{Qj} * w_{ij}}{\min\left(\sum_{j=1}^{n} {w_{Qj}}^2, \sum_{j=1}^{n} {w_{ij}}^2\right)} \quad (5)$$

c) The Probabilistic Model

The probabilistic model is based on decision theory. The aim is to compute the probability of relevance of a document D with respect to a query Q. The first probabilistic model has been proposed by Maron and Kuhns [17].

In the probabilistic model, the documents and the query are represented by vectors in indexing space as in the vector model. In these vectors the weights of the index are binary. For a query q all documents available are divided into two subsets: the set R of relevant documents and NR irrelevant documents. For each document two probabilities are associated [5] [2]:

- P (R / d): the probability that the document is relevant to the query q.
- P (NR / d): the probability that the document is not relevant to the query q.

The similarity between the document and the query q is calculated as a function of these two probabilities as follows:

$$RSV(d_j, q) = \frac{\mathrm{P(R/d)}}{\mathrm{P(NR/d)}} \quad (6)$$

4. Description of the Proposed Solution and Experimental Results

After discussing the theoretical aspects of information retrieval, this section is devoted to the description of our system and the comparison of two techniques for recording text index. Our information retrieval system is based on the vector model, and provides the following two features:

✓ Indexation:

In the indexing phase (shown in Figure. 4) the first operation is the removal of separators (common punctuation character) according to a file that contains a set of delimiters. After the phase of segmentation, the corpus passes to the second phase which is the removal of empty words. The treatment is simple: if a term appears in the anti-dictionary, it is not considered as an index term, finally we pass to the important step that is linguistic normalization, in which we use a dictionary of roots to replace a term with its lemma, if the word appears in the dictionary of roots; after this phase the result is the index that will be recorded in an inverted file or in a object database.

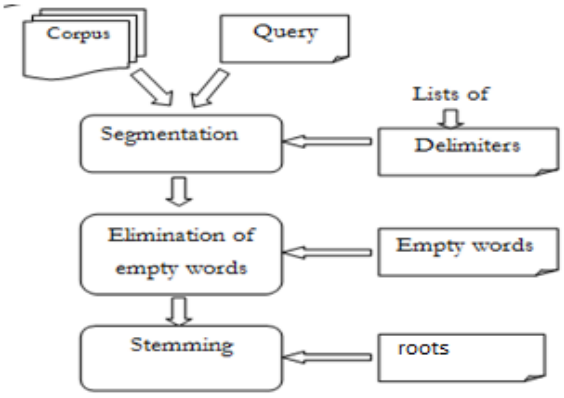

Figure. 4 Phases of indexing of proposed system

In the case of use of an object database DB4O its structure will be as shown in Table 1.

Table 1. Example of the Structure of the Database

Terms	Document 1			Document 2		
	Id	frq	tabPos	Id	frq	tabPos
Recherche	1	22	3, 10,…	2	22	4,5,…
Information	1	5	0,2,19,	2	9	3,1,…
Base	1	1	4,5,…	2	0	NULL
structure	1	0	NULL	2	11	0,9,…

Figure 5 shows the difference between indexing time using the inverted file and the object database db4o. We note that indexing using a database is faster compared to using an inverted file for indexation (FI).

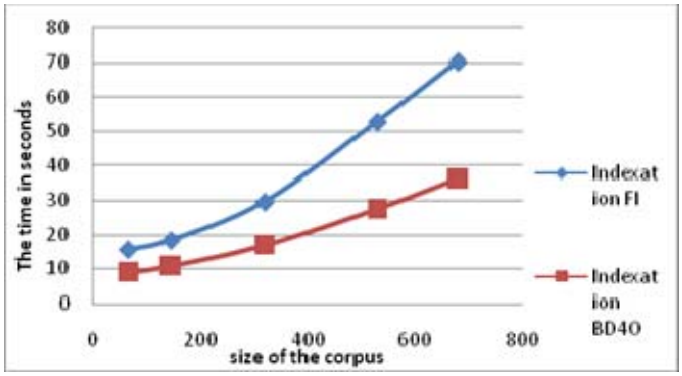

Figure. 5 Comparison of indexing time in db4o and the inverted file

✓ Research

Research is the second feature of our system; that maps the representation of the collection of documents (the index) which is in the database (db4o) and the representation of user needs (the indexed request); to return a set of relevant documents to the query.

Treatment associated to research passes through two stages:

- Indexing of the request in the same way as documents in the corpus;
- Put in match the representation of documents with that of the query using the similarity measure (tf * idf)

The search process using a database (db4o) is illustrated in Figure 6.

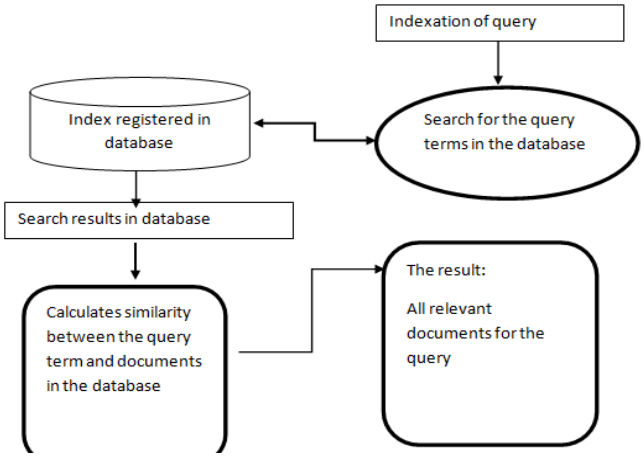

Figure 6. The process of search in a database (db4o)

Figure 7 provides a view of the difference between the search time in the database and the inverted file:

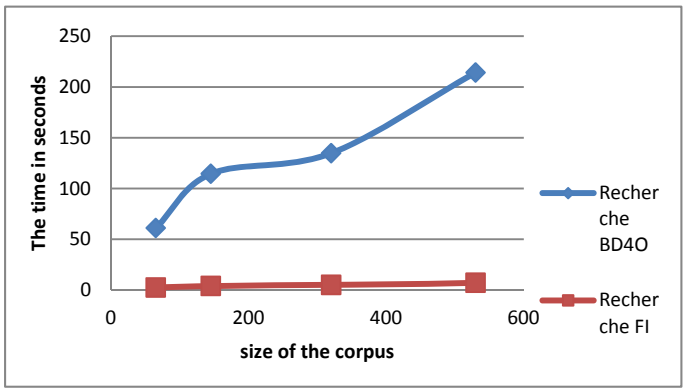

Figure 7. Comparison of search time in the BD4O and FI size of the corpus

Figure 7 shows that the time of the search in an object database BD4O is very slow compared to research in an inverted file.

5. Conclusion

We presented in this paper, the main steps of the process of information retrieval, as well as basic models of information retrieval. This paper focused mainly on the study and evaluation of performance of two indexing approaches which are the use of the object database BD4O and the inverted file. We performed some comparisons that show that the use of BD4O is quick for indexing, but that the search time in inverted file is better than research in databases. In our future work, we will introduce the notion of semantic based on ontology to our system to render it a semantic search engine.

References

[1] Ricardo BY, Berthier RN. Modern information retrieval, ACM (Association for Computing Machinery).
[2] Baziz M. Indexation conceptuelle guidée par ontologie pour la recherche d'information (Doctoral dissertation, Toulouse 3). 2005.
[3] Mooers CN. Application of random codes to the gathering of statistical information (Doctoral dissertation, Massachusetts Institute of Technology). 1948.
[4] Karbasi S. Pondération des termes en Recherche d'Information (Doctoral dissertation, Toulouse 3).
[5] Harrathi F. Extraction de concepts et de relations entre concepts à partir des documents multilingues: approche statistique et ontologique. 2009.
[6] Salton G. A comparison between manual and automatic indexing methods. American Documentation. 1969; 20(1): 61-71.
[7] Mallak I. De nouveaux facteurs pour l'exploitation de la sémantique d'un texte en Recherche d'Information (Doctoral dissertation, Université Paul Sabatier-Toulouse III). 2011.
[8] Aouicha MB. Une approche algébrique pour la recherche d'information structurée (Doctoral dissertation). 2009.
[9] Barry CL. User-defined relevance criteria: an exploratory study.JASIS. 1994; 45(3): 149-159.
[10] Boubekeur-Amirouche F. Contribution à la définition de modèles de recherche d'information flexibles basés sur les CP-Nets (Doctoral dissertation, Université de Toulouse, Université Toulouse III-Paul Sabatier). 2008.
[11] Roussey C. Une méthode d'indexation sémantique adaptée aux corpus multilingues. Institut National des Sciences Appliquées de Lyon Lyon, Ecole Doctorale Informatique et Information pour la Société. 2001.
[12] Azzoug W. Contribution à la définition d'une approche d'indexation sémantique de documents textuels. 2014.
[13] Porter MF. An algorithm for suffix stripping. Program: electronic library and information systems. 1980; 14(3): 130-137.

[14] Buckley C, Singhal A, Mitra M & Salton G. *New retrieval approaches using SMART: TREC 4*. In Proceedings of the Fourth Text REtrieval Conference (TREC-4). 1995: 25-48.
[15] Brini AH. Un modèle de recherche d'information basé sur les réseaux possibilistes (Doctoral dissertation, Toulouse 3). 2005.
[16] Maron ME & Kuhns JL. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM (JACM)*. 1960; 7(3): 216-244.
[17] Agrawal R, Imieliński T & Swami A. Mining association rules between sets of items in large databases. In *ACM SIGMOD Record*. *ACM*. 1993; 22(2): 207-216.
[18] Tebri H. Formalisation et spécification d'un système de filtrage incrémental d'information. Thèse de doctorat de l'université Paul Sabatier, Toulouse. 2004.
[19] V Rijsbergen CJ. Information Retrieval. Department of Computing Science University of Glasgow.
[20] Iadh O. Un modèle d'indexation relationnel pour les graphes conceptuels fondé sur une interprétation logique, Thèse pour obtenir le grade de Docteur de l'Université Joseph Fourier. 1992.
[21] Piwowarski B, Denoyer L, Gallinari P. Un modèle pour la recherche d'information sur des documents structurés. 6es Journées internationales d'Analyse statistique des Données Textuelles. LIP6, PARIS – France. 2002.
[22] Denos N. Modélisation de la pertinence en recherche d'information: modèle conceptuel, formalisation et application. Thèse pour obtenir le grade de Docteur de l'Université Joseph Fourier-Grenoble I. 1997.
[23] http://www.comp.lancs.ac.uk/computing/research/stemming/Links/lovins.htm
[24] http://www.comp.lancs.ac.uk/computing/research/stemming/Links/paice.htm
[25] http://tartarus.org/martin/PorterStemmer/
[26] http://snowball.tartarus.org/