META-DOCUMENTS AND QUERY EXTENSION TO ENHANCE INFORMATION RETRIEVAL PROCESS

Mounira Chkiwa[1], Jedidi Anis[2], Faiez Gargouri[3]

[1]*University of Sfax, Multimedia Information Systems and Advanced Computing Laboratory, Tunisia*
m.chkiwa@gmail.com
[2]*University of Sfax, Multimedia Information Systems and Advanced Computing Laboratory, Tunisia*
anis.jedidi@isimsf.rnu.tn
[3]*University of Sfax, Multimedia Information Systems and Advanced Computing Laboratory, Tunisia*
faiez.gargouri@isimsf.rnu.tn

Abstract

In this paper, we present two facets indispensable for the efficiency of our information retrieval system: meta-documents and query extension. Meta-document represents a structure used to annotate our web documents collection and query extension represents an automatic process aimed to enhance user query expression by additional terms. The two facets are indirectly related since added terms to a given query are taken from an ontology based on meta-documents. The cooperation between meta-documents and query extension aims to have an enhanced information retrieval process. In this paper, we present our proposition particularity and its evaluation results which show its efficiency.

Keywords: *Information Retrieval, Meta-Document, Annotation, Query, Semantic Extension, OWL Ontology, Semantic Proximity.*

----------***----------

I. INTRODUCTION

In the information retrieval context, two structures must be consistent to lead to the success of the process: the set of keywords chosen by the user to express his information need and the set of keywords chosen by the system to annotate each document of its collection. The matching between the two structures is extremely important since it leads to get or not relevant results. The rest of paper is organized as follows: in the next section we present the web documents annotation by meta-document, section 3 presents our method of query extension, section 4 presents the contribution evaluation using standards measures, this evaluation represents a closing of the development of our information retrieval system presented in [1, 2 and 3]. Section 5 presents some related works and our particularities. Finally section 6 concludes the paper.

II. META-DOCUMENTS

In the web context, the document representation means how to highlight most meaningful web page parts. In our work, this operation is called annotation and it aims moreover to sort those meaningful found parts by importance in an independent structure called meta-document. The matching between a query representation and the set of meta-documents aims generally to find the most relevant documents to a query. This operation is generally called querying and it follows generally a matching algorithm which represents the particularity of an information retrieval system. We highlight in this section the meta-documents automatic creation and the exploitation of this structure in querying process.

A. Meta- Document creation

The annotation process allows in our system to automatically generate a meta-document descriptive of a document. In order to keep only significant terms that can describe a web document, we start by scanning the content of the target document and eliminating empty words. Empty words or stop words usually refer to the most common words in a language such as (the, is, or, by, in, with…). A stop word often has a high frequency of use for all documents to be annotated, but also a low semantic value. Since we are interested in French and English documents, our elimination process is based on a universal anti-dictionary called "stop-words-collection" [4] covering 671 English terms and 463 French terms to be eliminated. After running the stop words elimination process, we pass to the meta-document generation which aims to annotate the current web. A meta-document is composed by the source page content sorted in three lists of terms called "very important, "important" and "normal":

1. The very important list: this list contains some tags contents such as <title>, <meta name="keywords" ....> <h1> and <h2>. In addition, this list contains the 5% most frequent terms in the document.
2. The important list: this list covers the content of the following tags : <h3><h4><h5> <b> <meta name= "abstract" … > and <meta name= "description" …>. In addition, this list contains the top 10% of the rest most frequent terms.

3. The normal list covers the content of the rest of document tags except <script> tags, <style> tags and code commentaries.

In order to be updated with new web "trends", we consider equivalent new HTML tags such as the metadata defined by the Open Graph protocol [5]. Hence we can accept in the "very important" list the content of <meta property="og:title" … /> instead of <title> and the content of <meta property="og:description" …> instead of <meta name= "description" …> if they exist. The addition of 5% and 10% of most frequent terms to "very important" and "important" lists is made to make a balance between the lists size. Further, those added terms are indispensable to fulfil "very important" and "important" lists in the case where no or few of specified tags exist in the web page supposed to be annotated.

Meta-documents generated offers three levels of weighting to significant terms of an annotated web document. According to its belonging to a list, a term can be considered very important, important or normal. Our Partitioning choice is semantic and it considers the importance of some terms comparing with others: It is obvious for example that the contents of the <title> tag reflects more the sense of a web document than a <b> tag (bold), and a bold text is more important in a given document than a text without formatting. In addition, our annotation method is a compromise between the indexing in Boolean and vector models:

- In Boolean model, the indexing attribute a simple weighting to the terms of a document, either 1 or 0, respectively, reflecting the presence or absence of a given term in a document. While this weighting is simple to practice, but it penalizes important terms of the documents (titles, subtitles ...).
- In the Indexing vector model attribute weighting of terms requiring a complex calculation because it depends on: [6]
- The importance of the word in the document (local weight).
- The importance of the term in the collection (global weight).
- The importance of the document (normalized for the Document size).

In vector model, there is no big difference between a term has a weight equivalent to 0.98 and another has a weight equivalent to 0.95. In the other side, there is no big difference between a term has a weight equivalent to 0.01 and another has a weight equivalent to 0.04. Both of first terms are considered important and both of terms in the second example are considered trivial. Therefore, our annotation method could be considered, compared to vector model, as a clustering terms method in terms of importance. Subsequently, we avoid complex indexing weights calculation and we allow a simpler querying algorithm and faster in terms of execution.

It remains to mention that the annotation process is automatic; we simply specify a web page URL [1] and the system is charged to eliminate empty words and generate a meta-document corresponding to the current web page.

B. Querying

The querying process consists of the matching between the query representation and the meta-document in order to get the score relevance of the correspondent document to the query. This score reflect the relevance rate of a document to the user query. Before being used in the querying process, a user query (composed often by a set of keywords), passes through two main process:

- The elimination of empty words via the same anti-dictionary used in the annotation process.
- The query extension which adds some words having semantic relation with initial keywords. We have further explained this process section III.

The querying process matches the query representation and the set of meta-documents describing our web documents collection. We present below the querying algorithm abstract:

```
1. For each document Di
2. S←0
3. For each query term ti
4. if(ti in meta-document Di)
   1. if (ti is in normal list)
   2. S←S+n
   3. if (ti is in important list)
   4. S←S+n*2
   5. if (ti is in very-important list)
   6. S←S+n*3
5. end if
6. end for
7. return S
8. end for
```

As we have mentioned, the algorithm above represents an "abstract" version of the used querying process because we consider in addition of equivalence query-term/meta-document-term, the inclusion between them (e.g. real and reality or lead and leader...). The value *n* represents the number of occurrences of the current term and its variants in a given list. The output of a querying process is the score relevance *S* of the current document to the query. The set of scores of relevance is usable in the ranking process which aims to sort documents by relevance before returning them as result.

III. SEMANTIC QUERY EXTENSION

Since information retrieval's queries are often composed by sets of keywords formed by users, the semantic query extension can better formulate a given need by some additive terms. Additive terms can cover more lexical fields of the terms of an initial query without diverging from its

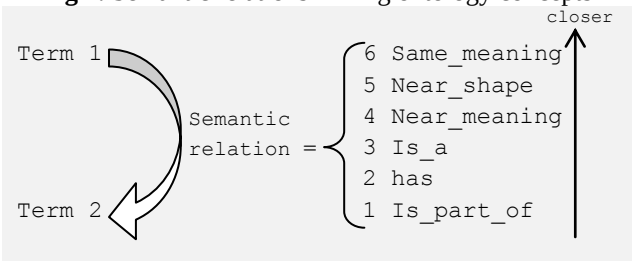
[1] URL: Uniform Resource Locator is a universal identifier of web pages.

sense. In order to find the most semantically related terms to a given query, our system use some semantic web technologies such as OWL [2] and SPARQL [3]. Our OWL ontology represents a structured vocabulary from which we extract the additional related terms while a query extension process. SPARQL is a querying language which we use while scanning OWL ontology in order to find terms having semantic proximity to the user query.

C. OWL Ontology

Our information retrieval system is based on a collection of web documents which are annotated by a set of meat-documents covering the essence of each of them. During an information retrieval process, the system scans the meta-documents set in order to find relevant documents and return them as results. In order to enhance results, we propose to extend user query by some additional terms which have semantic relation with query terms meaning. The structured vocabulary from which the system picks additional terms must cover the same lexical field covered by our collection of web documents in order to make the extension process more targeted. In other side, it's not useful to propose additional words that are semantically linked to the user query but not found in the set of meta-documents scanned while an information retrieval attempt. For this reason, we built an "on measure" ontology based on the meta-documents set. Our ontology covers a consistent set of concepts, each two concepts or terms are linked by one among our six fundamental semantic relations. In fig. 1. We present the set of semantic relations used to link between ontology concepts, in this figure, relation are ordered by proximity strength.

Fig 1. Semantic relations linking ontology concepts



The number attributed to a semantic relation type shows the proximity strength between two terms, bigger number means a closer semantic relation between two terms. Hence, two terms linked by the relation "same_meaning" are closer than two other terms linked by the relation "has". For example the semantic relation between "trip" and "travel" is stronger than the relation between "trip" and "agency" since the first couple of terms are linked by "same_meaning" relation. Our OWL ontology based on meta-documents is validated by the "OWL validator" available online by Manchester University [7].

[2] OWL Web Ontology Language 1.0 Abstract Syntax W3C Working Draft 29 July 2002 Peter F. Patel-Schneider Ian Horrocks Frank van Harmelen

3 SPARQL Query Language for RDF W3C Recommendation 15 January 2008

D. Query extension

After an information retrieval attempt, the user could simply "refresh" the results page if he is not convinced and the query extension process takes place. This extension subsequently aims to better rerank relevant documents found during the first search attempt and eventually find other relevant documents. Then, the display of new results is accompanied by the initial user query concatenated to additional found terms. The semantic query extension follows the abstract algorithm as follows (Q is a set of keywords query):

```
1. A=""
2. T=""
3. N=size(Q)/3
4. Remove empty words of query Q
5. For each term ti of Q
       Find the set T of terms related to
   ti
       Concatenate T to A
       T=""
    End for
6. Order A terms by semantic proximity
7. Remove any repetitions in A
8. Concatenate the first N terms of A to Q
```

In order to get a useful query extension, two conditions have to be respected:

- The number of added terms is proportional to the query size; for this reason, we add only N of the found related terms to the initial query (see the 8th point of the query extension algorithm). N represents one third of the initial query size. The added set represents the most semantically related terms to the query since they are ordered by semantic proximity (see the 6th point of the query extension algorithm).
- Only significant terms are concerned by the query extension, hence, we remove first empty words from the user query (see the 4th point of the query extension algorithm). To do it, we use the same anti-dictionary used in the annotation process.

In the 6th point of the query extension algorithm, the set of found terms of A are ordered by semantic proximity according to semantic relation strength detailed in fig. 1. Hence, it is more probable to find firstly in the set A, terms linked by the "same_meaning" relation to query user terms. This addition allows enriching the initial query sense without diverging it.

IV. THE SYSTEM EVALUATION

To evaluate the performance of our system, we have relied on three classic but proven measures in the field of information retrieval: *Recall*, *Precision* and *F-measure*. While an information retrieval process:

- Recall (R): is the ratio of the number of relevant documents retrieved to the total number of relevant annotated documents.

- Precision (P): is the ratio of the number of relevant documents retrieved to the total number of retrieved documents (irrelevant and relevant).
- F-measure: it can be interpreted as a weighted average of the precision and recall: F-measure= 2.R.P / (R + P)

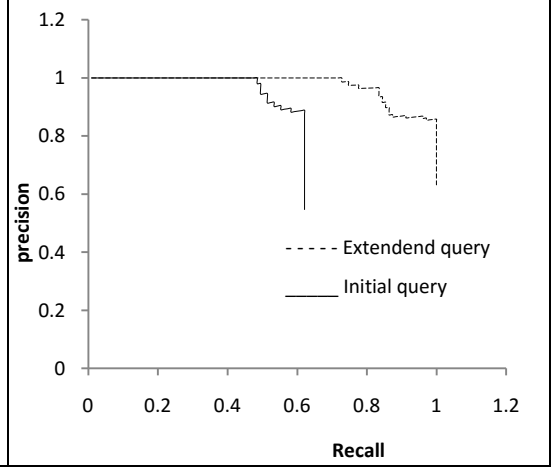
To study the contribution of our proposition in terms of the query extension, we use queries written in French and English languages since our collection covers French and English web documents. Hence we use in the evaluation tests three typical queries (α, β, and γ) in both languages where:

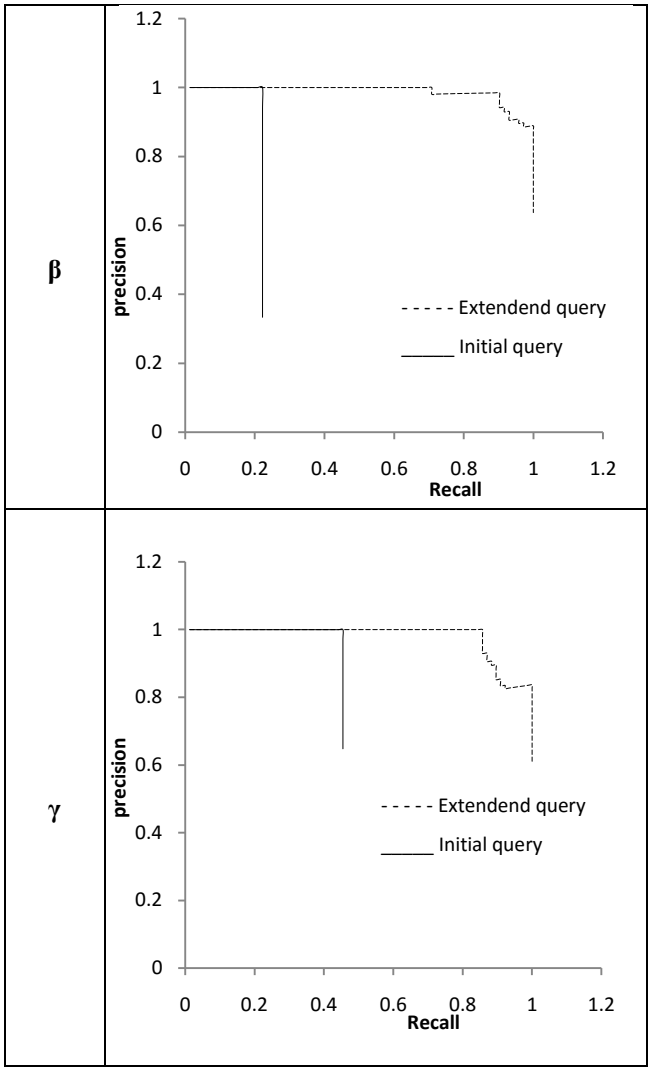
- α: apprendre des contes pour les enfants (learn tales for children).
- β: rules and basics of football.
- γ: jeux mathématiques pour les primaires (math games for primary).

We use such keywords in queries α, β, and γ because our system is dedicated for kids and all annotated web documents of our collection have childish interests [1, 2 and 3].

In Table I, we present the variation of curves recall / precision according to the query submitted nature (with and without extension). The dotted curves represent the evolution recall / precision while using queries α, β, and γ after semantic extension and continuous curves represent the evolution recall / precision by using queries α, β, and γ without extension.

Table I. The Differentiation Of Curves Recall / Precision According To The Query Submitted Nature (With And Without Extension)

query	recall / precision curves with and without semantic extension
α	



By studying visually table I, it is clear that querying based on extended queries is more efficient than that based on initial queries without extension. To consolidate this point of view by numerical results, we present in table II the considerable difference of F-measure values computed based on last values taken from recall / Precision curves presented in table I using queries α, β, and γ .

Table II. Difference of F-measures values according the query nature (extended or not)

	query					
	α		β		γ	
With extension	**yes**	**no**	**yes**	**no**	**yes**	**no**
recall	1	0,6214	0,2222	1	1	0,4545
precision	0,6319	0,547	0,3333	0,6372	0,6111	0,6481
F-measure	0,7744	0,5818	0,2666	0,7784	0,7586	0,5343
difference	0,1926		0,51176		0,2243	

The structure of a meta-document based on three lists enables to exhibit the document contents by differentiating

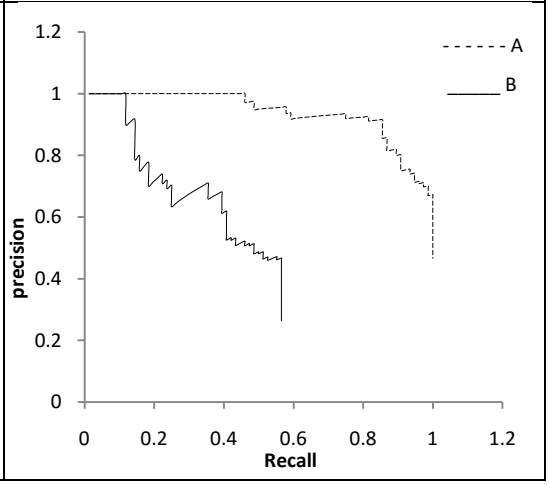
the importance of certain terms versus others. To study the contribution of this structure in the information retrieval process, we compare the querying of two meta-document collections (A and B) annotating the same documents set. A and B are distinguished as follows:

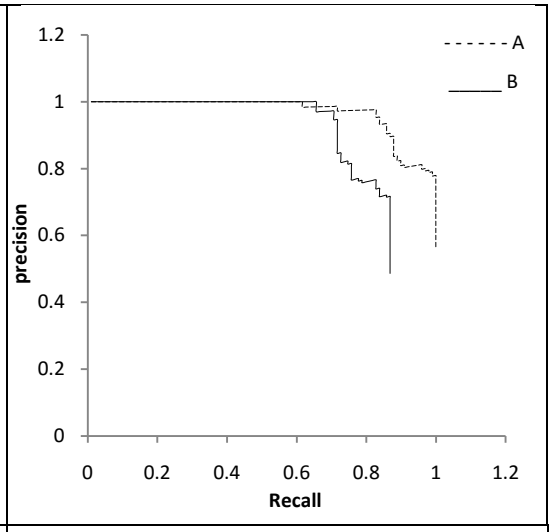
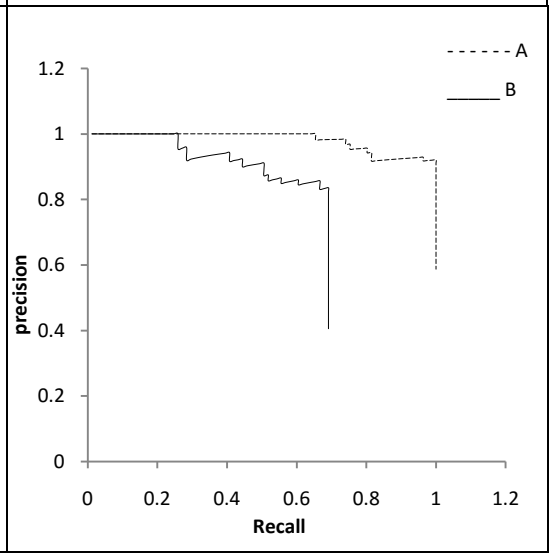
- Each meta-document of the collection A is composed by three lists (very important, important and normal)
- Each meta-document of the collection B is composed by one list covering all terms of the three lists without importance distinction.

In order to have a fair comparison between the two meta-documents collections, we eliminate empty words using the same anti-dictionary for the meta-documents generated in the collection A and B. To study the contribution of our proposition in terms of the structure of meta-documents generated during the annotation process, we use same queries (α, β, and γ) used when studying the contribution of semantic query extension.

In Table III, we present the differentiation of curves recall / precision according to the queried collection of meta-documents. The dotted curves represent the evolution recall / precision by querying the collection A (our collection of structured meta-documents) and continuous curves represent the evolution recall / precision by querying the collection B. Table III shows obviously that that querying based on the collection A is more efficient than that based on the collection B.

Table III. Difference of Recall / Precision curves according the structure of queried meta-documents

query	Recall / Precision corresponding curves distinguished by collection
α	

β	
γ	

V. RELATED WORKS

In the semantic query extension field, many works are proposed, in this context, our proposition is simpler and it combines ideas from several works:

- [8] and [9] use WordNet ontology to distinguish similitude of sense of terms in order to find a target concept added to query terms.
- [8] applies a synthesis of collection of queried documents in order to find candidate terms added to a query.

Our proposition of query extension is based on ontology such as [8 and [9] but our ontology is more "targeted" since it is based on terms taken from meta-documents which represent the essence of queried web documents collection. Support on queried documents collection idea is not far from [8] proposition. But our approach differs on how to select additional terms. Indeed [10] use statistical approache to find the target terms. This approache seeks several calculations which is demanding in terms of computation complexity and execution time. However, our query extension proposal is based on a simple selection of words semantically related to the terms of a query from the ontology and their order by semantic proximity. Furthermore, our proposed semantic query extension is more flexible than that developed [11].

Indeed, the extension query in [11] is based only on the inclusion relation between concepts and the analysis of its values to evaluate the semantic proximity between terms. However, we have handled six different types of relations between terms in our contribution (Same_meaning, Near_shape, Near_meaning, is_a, and Has Is_part_of).

VI. CONCLUSIONS

In this paper we presented our contribution to enhance results of information retrieval process. The annotation process leads to the creation of meta-documents describing our web documents collection. The extension process is based on our "on measure" ontology created using meta-documents. In this paper we present evaluation results which show the approach efficiency, in addition we enumerate some related works, Our method creates a compromise between them by giving a simpler and more flexible version. Our perspective in the short term is to enlarge moreover the set of semantic relations between ontology concepts to enrich furthermore user queries.

REFERENCES

[1] Mounira Chkiwa, Anis Jedidi, Faïez Gargouri . Simplest Information Retrieval For Kids. 2nd International Conference on Information Technology in Education. ICITE - 2014 , Zurich, Switzerland, June 14 ~ 15, 2014.

[2] Mounira Chkiwa, Anis Jedidi, Faïez Gargouri. Fuzzy Score Relevance Valorization. 2nd International Conference of Artificial Intelligence & Fuzzy Logic AIFL 2014. Dubai,UAE April 4~5, 2014. 5.

[3] Mounira Chkiwa, Anis Jedidi, Faïez Gargouri. Simple and Efficient Information Retrieval Process. Third International Conference on Artificial Intelligence, Soft Computing (AISC-2015), Sydney Austria; 02/2015.

[4] Stop words collection. https://code.google.com/p/stop-words/

[5] The Open Graph protocol http://ogp.me/

[6] G. Salton , A. Wong , C. S. Yang, A vector space model for automatic indexing , Communications of the ACM, v.18 n.11, p.613-620, Nov. 1975

[7] Owl validator http://mowl-power.cs.man.ac.uk:8080/validator/

[8] Min S, Il-Yeol S, Xiaohua H, et Robert Al. Ontologies-driven Semantic Query Expansion 2006.

[9] M. Baziz, N Aussenac-Gilles, M. Boughanem. Exploitation des Liens Sémantiques pour l'Expansion de Requêtes dans un Système de Recherche d'Information.

[10] Lam-Adesina A.M., and Jones, G.J.F. Applying Summarization Techniques for Term Selection in Relevance Feedback, Proceedings of the 24th annual international ACM SIGIR Conference on Research and Development in Information Retrieval: 1-9, 2001.

[11] G.Akrivas, M. Wallace, G. Andreou, G. Stamou, S Kollias context – sensitive semantic query expansion. http://www.image.ece.ntua.gr/papers/202.pdf