Abstract

The performance of an information retrieval system (IRS) may be degraded in terms of accuracy due to the difficulty for users to express their information needs precisely. Reformulation or query expansion is one of the answers to this problem within the IRS. In this paper, we propose a new method of reformulating conceptual queries seeking, from the initial user query and domain ontology, a set of concepts maximizing the performance of SRI. These are evaluated in an original way, using indicators for which a formalization is proposed. Then we calculate the matching score between the query and documents reformulated medical videoconferences to give the user the most relevant document.

Keywords: Query, reformulation query, matching document query, relevancy score, similarity calculation, calculation of correspondence, ontology, user profile, information retrieval, medical videoconferencing.

1. INTRODUCTION

The main task of SRI is to select from a collection of documents those which are able to respond to the user’s needs of information. To reduce the ‘silence’ (a proportion of relevant documents among those not found) and the ‘noise’ (proportion of irrelevant documents among those returned), three processes are generally implemented in SRI: first, an analysis process that aims at reprocessing the documents of the corpus, second, an indexing process that provides a compact and semantic representation of documents and queries using concepts and semantic relations between these concepts, and finally, a search process consisting of two main tasks. In the first step, this process supports the generation of a new application using external resources such as ontologies, user’s profile and the initial query. In the second step, it calculates the correspondence (matching) between the reformulated query and the documents of the corpus to satisfy the user’s needs. It is generally presented as in Figure 1.

2. QUERY REFORMULATION

The user sometimes faces a difficult situation. He is unable to find the exact words to express the needed information. So, if not the majority of the documents found may interest the user less than others. In addition, some queries are short and, therefore, not semantically rich, so that the IRS can return relevant documents. To overcome these problems, researchers in RI have turned to incorporating an additional step in the research process which is the reformulation or expansion of the query. It consists in changing the user’s initial query by adding significant terms and / or re-estimating their weight. Ralalason proposes a process of query’s reformulation in his thesis [7]. This reformulation can be automatic or manual depending on whether the system or the user carries it out. Thus, the query reformulation may particularly intervene at two levels:

- during the initial search, if no document is found, the system performs the reformulation;
- if the reader is not satisfied with the system’s response after an initial search, he may reformulate the query.

The reformulation of the query can be based on the user’s profile and/or external resources. In our work, we propose a reformulation of queries guided by an external resource. It is to rewrite the query of the user by taking into account synonymy and metonymy relations presented in the external resources. So, the query is enhanced by terms semantically close to the original ones. These semantically similar terms are issued from domain ontologies in order to search for documents in a medical corpus. Similarly, we rely on the user’s profile to reformulate queries. We use a progressive mechanism to categorize users by building a static contextual elements base used to reformulate the initial query to produce a new one which reflects better the user’s needs.
Figure 1 Research process
The structure of the reformulation step, its constituent tasks as well as the interactions between them are illustrated in Figure 2.

![Diagram of reformulation process]

Figure 2 Reformulation of the user’s query
The formulated query is matched with the documents’ indices. These indices having a sufficient “similarity” to the

Table 1 : Examples of query modification

<table>
<thead>
<tr>
<th>Type</th>
<th>Modified Query</th>
<th>Reformulation Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Q = migraine (initial query)</td>
<td>Changing the query relative to the concept of the corpus.</td>
</tr>
<tr>
<td>2</td>
<td>Q’ = migraine or headache (enlargement of the query to a wider domain)</td>
<td>Changing the query relative to the domain.</td>
</tr>
</tbody>
</table>

The reformulation of queries occurs in three phases: (i) capture of the static context, (ii) identification of the concepts that match the user’s needs, (iii) the composition of these terms in order to formulate the query. Generally speaking, queries as those formulated by users query are considered to be relevant. Below, we present an example of the query modification. According to the cases target: (i) concepts or terms related to the field of the query: migraine, “migraine aura” headaches; (ii) elements presented in various parts of a videoconference (media type).

2.1. phase of Capture of the static context
This phase is used to identify a user through a series of information for his categorization. The user defines the static context when first using the system. This context is composed of four categories:
- Connection parameter: login, password
- Personal characteristics: name, surname, country, ...
- Interest and preferences: domain, specialty, ...
- Competence: Occupation, education level, ...

This phase is presented in Figure 3.

Figure 3 Phase of the recovery of the user’s static context

2.2. Reformulation phase
The purpose of this phase is to produce a new query from the query initially formulated by the user by adding words from the context of his current research. The user gives his query using his own terms. The system performs the extraction of all the terms to be added to produce a new query. First, these terms are extracted from the base of the user’s contexts. Then, we retrieve the specific concepts from the ontology. In case the concept C1 is presented as a term in the original application, it will be expanded by the concept C2 which is retrieved from the ontology (UMLS, ONTOMENELAS ..). The extension of the concepts is based on the presence of a semantic relation between C1 and C2 (synonym or a more general concept).

This phase shall run in two steps acting on different aspects of the initial query. These steps are:

Extraction of concepts: we retrieve the query terms presented in the ontology. Then we search in (go over) the ontology using these terms as an entry point to extract concepts that are directly related to each term.

The concepts are retrieved from the ontology in the following way:
- If the concept (C1) is present as a term in the initial query, it will be expanded by the concept (C2) retrieved from the domain ontology.
- The choice of (C2) follows a navigation of the XML file describing the ontology.
- We seek a semantic link between (C1) and (C2). This link may be of type "Synonym", "more general concept" ...

In this case, the role of user is passive because it does not intervene in the selection of the concepts. The calculation
time is too high for a large ontology. In the context of a heuristic approach, it is reasonable to test only the concepts that are semantically close to the concepts indexing the documents. If $\mathcal{R}(C_x, C_y)$ is a measure of semantic proximity between CX and CY, two concepts of $\Theta$, we construct $C(\Theta)$ in the following way:

$$C(\Theta) = \{c_i \in \Theta | \exists c_j \in C(\Theta) \left( (c_i, c_j) > \epsilon \right) \}$$  \hspace{1cm} (1)

$C(\Theta)$ is constructed by identifying, for each concept those, the semantically-close concepts that should be considered [11].

Reformulation query: the integration in the phase of query’s reformulation consists of arguing the initial query by concepts issued from the stage of concepts’ extraction and terms issued from the user’s profile. The new query is consequently transmitted to the corresponding step (section II). This query will be presented in the form of a conceptual graph. The reformulation process is summarized in Figure 4 shown below:

![Figure 4 Architecture of the reformulation process.](image)

The algorithm of our reformulation approach is the following:

**Algorithm 1 : Reformulation**

```
Input :
Q : User’s query
O : Ontology
P : a user’s profile

Output : Q’ : reformulated query

Variables :
Lc : list of concepts
Lt : list of terms

Begin
Pretreatment of the query ; /* Extraction of terms*/
If (exist already (Lt) = true) then
Lt \leftarrow Lt ;
else
Lt \leftarrow Extraction (Lt) ;
End if
Q’ \leftarrow Q + Lt
End
```

3. MATCHING DOCUMENT-QUERY

The comparison between the document and the query enables to calculate a measure called relevance system supposed to represent the relevance of the document in comparison with the query. This value is calculated using a similarity function denoted $RSV(Q, D)$ (Retrieval Status Value), where Q is a query and D is a document. This measure takes into account the weight of terms in the documents. In general, matching the query-document and the indexing model allows characterizing and identifying a model of information retrieval. The order in which the returned documents expected to respond to the query is important. Indeed, the user, in general, simply considers the first returned documents (the first 10 or 20). If the required documents are not in this range, the user will consider SRI as bad comparing to his query. Many

**Algorithm 2 : Extraction**

```
Input:
Lt : List of terms
Lco : List of the ontology’s concepts
Iu : Interest of the user
Cu : competence of the user

Output :
Lc : List of concepts

Variables :
t : Term
c : Concept

Begin
For each term do
Identify the concepts related to t
End for
```

Case 1 : $I_u$ is « not medicine » :
If t belongs to Lco then
Add (c(t)) to Lc   /* c is the identified concept */
End if
```

Case 2 : $I_u$ is « medicine » :
Case Cu do
Case1 : Cu is « expert » or « doctor » :
If t belongs to Lco then
Add(c) to Lc   /* c is the identified concept*/
End if
Case2 : Cu is « student » :
If t belongs to Lco then
Add (c(t)) to Lc   /* c is the identified concept */
End if
End case
End case
End for
End
matching models have been proposed in the literature: The
Boolean model [8], the weighted Boolean model [12]. The
vector space model, the extended vector model [4], the
probabilistic model [3], the inferential model [2], the
differential Bayesian model [13] and language models [6].
In order to take into account the semantics when
evaluating, and therefore when generating the new
classification and its comparison with the default
classifications of the SRI, Bouramoul associates to each
term in the query all words that are semantically related to
it. The idea is to project the query terms in the ontology’s
ccepts using both semantic relations "synonymy" and
"hypernymy" to extract the different meanings of the
query. Thereafter, all retrieved concepts for each term are
used in conjunction with the term itself when weighting
using the calculation module. The objective is to promote a
document that contains words semantically closed to what
the user is looking for, even if those words do not exist in
the query [1]. The concepts found after the reformulation
phase of the request will be presented in the form of GC.
Similarly, in the indexing module of our SRI [14], the
index of each video conferencing is also treated in the form
of a GC. To satisfy the user’s need for research, there are
two steps to be followed:
- Search of correspondence: this step can be done using
two methods. First, we perform a projection of the
query’s conceptual graph on the documents’
conceptual graph. In case this task does not give a
result, we use, secondly, a similarity function between
ccepts.
- Calculation of relevance of the documents found.

3.1. Search of correspondence

3.1.1. Projection correspondence

The formalism of conceptual graph enables to ask and seek
for sub-graphs in a graph using the projection operator
[10]. The latter takes into account the concepts and
relations. The projection of a query graph GRi on a
conceptual graph GAi noted \( \Pi_{i}^{G_{A}}(G_{Ri}) \), concludes that
there is a sub-graph GAi that is specific to the graph GRi.
Informally, a conceptual graph GD is a projection of a
graph GR if each concept GR has a specific concept in
GD. If such a projection exists, then, it has been proven
that the document responds the query (as shown in Figure
5). In [5], it was shown that a search algorithm
corresponding to the projection operator on the conceptual
graph can be implemented very efficiently in terms of
complexity. We propose to use this algorithm. We quantify
the correspondence between a query graph GR and an
indexing graph GD by combining a matching on the
concepts and a matching on the arcs.

![Figure 5 Operation of projection of GCs](image)

We call an arch a triplet (concept, relation, concept)
linking three nodes - two concepts and a relation- in a
conceptual graph. From where :

\[
F(G_{R}, G_{D}) = \sum [f(c) \cdot f(d) \cdot c \in Concepts de G_{R}] + \sum [f(a) \cdot f(d) \cdot a \in Arches de G_{D}]
\]

In the formula (2), the frequencies of terms ft and the
inverse frequencies of documents fid are calculated as follows:
- The frequency of the term ft associated with a concept
C in a conceptual graph G is defined as the number of
the concepts G which are specific to the concept C.
- The inverse frequency of a document fid associated
with a concept C is based on documents that are
described by C or by a specific concept C.

We use a formula inspired from [9].

\[
fid(C) = \log(1 + D|d(C)|)
\]

Where
- D: the corpus of documents
- D: the documents corresponding to C

For an arch, the principle is similar to concepts. Given that
indexation can be a conceptual graph, or a set of unrelated
cceptual graph, we define the matching between a query
graph GR and a set S of an indexing graph GD as the
maximum matching between GR and each graph GD of S:

\[
M(G_{R}, S) = \max_{G_{D}} \{ M(G_{R}, G_{D}) \}
\]

3.1.2. Similarity Calculation

In the case where the projection of the query graph on the
indexing graphs does not give a result, we use a measure of
similarity between the conceptual graphs associated to
the query and those associated with the different
documents in the corpus. In our system, we take into
account the various terms that constitute a concept. The
similarity between two concepts is assimilated to the
number of common terms between them. This function is
called Sim. The more common terms two concepts have,
the more they are closer to each other. Firstly, we calculate
the proximity of a given concept. This proximity is measured by the ratio between the common terms (between the two concepts) and all the terms of the concept.

Figure 6 : calcul of similarity

We propose:
- CR: a found concept in the query;
- Term (CR): the set of existing words in CR;
- Term (CR) = \{A, B, D, H, M\}
- CD: a found concept in video-conference;
- Term (CD): the set of existing words in CD;
- Term (CD) = \{A, B, D, G, L\}

We call the common terms between the two concepts by Ancestor.

\[ \text{Ancestor (CR, CD)} = \text{Term (CR)} \cap \text{Term (CD)} = \{A, B, D\} \]

The Proximity of concept CR against Ancestor (CR, CD) is expressed by:

\[ \text{Proximity(CR)} = \frac{\text{Cardinal(Ancestor(CR,CD))}}{\text{Cardinal(Terme(CR))}} \] (5)

Similarly, we propose the formula of semantic similarity between concepts:

\[ \text{Sim(CR, CD)} = \text{Proximity(CR) * Proximity(CD)} \] (7)

Take the example cited above:
- Term(CR) = \{A, B, D, H, M\}
- Ancestor (CR, CD) = \{A, B, D, H, M\} \cap \{A, B, D, G, L\} = \{A, B, D\}

\[ \text{Proximity(CR)} = \frac{\text{Cardinal(Ancestor(A,B,D))}}{\text{Cardinal(A,B,D,H,M)}} = \frac{3}{5} = 0.6 \]

\[ \text{Proximity(CD)} = \frac{\text{Cardinal(Ancestor(A,B,D))}}{\text{Cardinal(A,B,D,G,L)}} = \frac{3}{5} = 0.6 \]

3.2. calculating relevance

In the previous section, we calculated the similarity between the queries’ conceptual graphs and the videoconferences’ conceptual graph and we assigned to each videoconference a similarity score. We can get many documents that correspond to the query. It is necessary to order them. Only documents that have a non-zero similarity score will be sorted by decreasing order. The search module will return either the most relevant concepts per document, or the set of all relevant concepts sorted by a decreasing order and grouped by document or the documents will be sorted in turn by their relevance to the set of the query’s concepts. The relevance score is calculated by the following formula:

\[ \text{Score(R, D)} = \sum \text{Pondération} \times \text{Sim(CR, CD)} \] (8)

where:
- R is a query;
- D is videoconference;
- CD Weighting:
- Weighting concept in videoconference.

This score helps to promote video-conferences having many concepts of a query.

Algorithm 4 : Pertinence Document-query

Input :
R : query
V : Corpus of video-conferences

Output :
S : Score of pertinence of a document D / query
3.3. presentation for the user

The results of SRI are generally presented in the form of a list of links accompanied by a title and a summary describing the content of each page. Before being presented to the user, these results should be ordered according to the relevance score assigned by the algorithms of each SRI. In our approach, and to respect this principle typically used to display the results of a search, the task of presentation supports the display part once the results are processed. More precisely, this task presents an account of a search session as follows:

- All results in response to a query, where each result is represented by a 2-uplet (title, summary).
- The semantic relevance score associated with each result.
- All concepts that are associated to each term of the query.

![Figure 7](image)

**Figure 7**: illustrates the task of presentation

### 4. CONCLUSION

The research module consists, mainly, of two steps. The first is the reformulation of the initial query by exploiting the semantic relations between concepts using domain ontology. It is an evolutionary and interactive step. It uses the initial query to initiate the search in order to re-weight the terms of the initial query, or to add (or to remove) other terms to it. The newly-obtained query can correct the direction of the search for the meaning of the relevant documents. This reformulation is guided by an external resource in order to take into account synonymy and metonymy relations that can be presented without forgetting the addition of terms semantically similar to the query’s original terms, while taking into account the user profile. The second step is the matching query-document. It primarily uses the principle of projection of conceptual graphs. In case this principle does not make any results, the matching step is based on a similarity function that allows to compare the representation of the query to that of each document. The selected documents will be classified using the correspondence function and presented to the user by the 2-uplet (title, summary).

### REFERENCES


AUTHOR