Enrichment of Queries for Education Domain

Mohamed Rachdi, El Habib Benlahmar, EL Hassan Labriji, Aziz Chara, Kamal El Guemmat
Faculty of Computing Ben M’sik, Casablanca, Maroc

Abstract: The rising growth of information available on the Internet calls for the establishment of more efficient research tools to select relevant documents from a colossal set of documents available in the net. The selection of the most relevant documents necessarily requires going through several stages. Among these, there is the reformulation stage, a process where the user query is enriched so as to delimit the desired result and therefore guide the user to the most relevant document. Reformulation by reinjection is one of the several techniques for reformulation extensively tackled in the literature. The purpose of this paper is to propose a new approach for enriching queries, based on well-defined classes. The proposed enrichment will be applied in the querying system of a meta search engine dedicated to education. We will sketch in particular the case of algorithms courses. Our approach defines a set of classes that are classified in two categories: application classes representing the concepts of algorithms courses and utilization classes constituting of patterns related to application classes with each containing a collection of markers (metalinguistic, metalinguistic nominal, lexical, etc). These classes are employed to detect the need for the user at first, and also allowing the reformulation and enrichment of the user query. The different classes are complementary and their use can boost the relevance of the returned results.

Keywords: Reformulation of queries, meta-search engines, pertinence

1. Introduction

Because it has a direct impact on the returned result, query reformulation is nowadays one of the keys to the success of Information Search Systems. Thus, the quality of the result is always linked to the expressed need (the user query). A well reformulated user query will definitely return a relevant result with much greater precision and less noise. In the same concern, the existing reformulation techniques propose enrichment approaches based on ontologies of customized domain [1], the user profiles [2,3], .... The enrichment presented in this article belongs to the third type defined by [4] which aims at increasing accuracy and reducing noise. The proposed enrichment is applied to a well-defined domain: teaching, namely the algorithms module. In our approach, we base the enrichment of the user query is based on the application and utilization classes. Then, we move to the presentation of our approach. After that, we draw a comparison to evaluate our system. Finally, the paper ends with a conclusion and outlook. In this paper, we begin with a state of the art of the various techniques of query reformulation. After that, we spell out our contribution. The next part will be devoted to the evaluation of our system. Finally we draw the conclusion and point toward perspectives.

2. State Of The Art

The enormous amounts of information available on the Internet and the large number of users increase the likelihood of not selecting the most relevant documents. The best way to satisfy the customer (end user) in getting the intended results is query reformulation. In the same vein, several avenues were explored and the use of domain ontologies stands out as one of the finest methods to query reformulation. Several studies are already exploiting domain ontologies to reformulate queries [1, 5-12] using their own resources. Other studies on query reformulation make use of linguistic methods and lexical resources [11]. The contribution of the approaches these studies adopt is making a morphological analysis of the words used in the query before reformulation. Reinjection of the relevance appears in this stage the most appropriate solution in the sense that it integrates the user through the process of reformulation. Some studies that have been undertaken in the same framework to reformulate queries rely on documents deemed relevant by the user [7,8]. Further studies as in [9,10] use Reinjection of the relevance but without the intervention of the User. It is an automatic reinjection in which the process is applied on N results. The final returned result will be considered the best. In [6], another technique of reformulation has been proposed. Reformulation is based on definition sentences taken from the Internet. The disadvantage of this technique is its application on queries composed of one word only.

3. Our approach

The major concern of information retrieval systems nowadays is the selection of relevant documents from a huge mass of existing documents on the Internet. To accomplish this challenging task properly several reformulation techniques are already in use, some of
which were mentioned in the previous section. To our knowledge, until now there are no approaches of query reformulation dedicated to education, specifically for algorithms courses. The enrichment proposed in this paper stands at the heart of reformulation approaches. It allows reformulation of the user query based on the application and utilization classes and along with markers defining the utilization classes. In our approach, the Internet is directly exploited in the reformulation process.

Let us consider the following definitions:

**Application class:** represents an indispensable part of the algorithms course, as an example of application classes we can cite: variable, reading and writing, conditional structures, repetitive structures, arrays, functions and procedures, etc.

**Utilization class:** Refers to a pattern corresponding to an element of the course. It is defined by a set of markers that can identify it. As an example of the utilization classes we can mention: Definition, Objectives, declaration, assignment (use), syntax, templates, samples, etc.

A *lexical-syntactic pattern:* describes a regular expression formed of words belonging to grammatical or semantic categories, and symbols to identify text fragments corresponding to this format. In the particular case of looking for relationships, the pattern features a set of linguistic forms whose interpretation is relatively stable and corresponds to a semantic relationship between the terms [13]. Our method directs the dilemma of reformulation to seeking out a set of words having the same semantics of the words in the original query. This approach is intended to provide all the words that would help in reformulation. However, the real gain of the approach is the retrieval of the most relevant documents to the user.

Let a user need be expressed as a user query. How to detect this need? How to draw correspondence between this need and the defined classes and how to use these classes for enrichment? This paper is intended to answer these questions. In a general context, the application and utilization classes use predefined elements that help to initially detect the need of the user and for enrichment afterwards. Hence, our approach is to identify the general and specific needs to enrich the user query. The general need corresponds to the use of application classes that are exploited in detecting elements of algorithms courses sought by the user, such as variables, functions, etc. Once the general needs are detected, the set of utilization classes helps at this level in extraction of the specific needs (declaration, syntax, etc). In cases where specific needs are implicitly expressed, we detect them using the markers of utilization classes.

After detecting the general and specific needs (application and utilization classes), our approach enrich the query with the application classes and markers linked to the utilization classes.

The following diagram shows the general process of the approach:

![Diagram](image.png)

**Figure 1.** The approach process of enrichment by classes

The implementation of this process involves four steps:

### 3.1. Step 1: preprocessing the query

A user query is composed of one or more words represented separately or as a sentence, or question, etc. To understand this need we go through several steps, including a preprocessing step which is indispensable before starting the process. The preprocessing step enables the cleansing of the query from useless words and the maintenance of only the important ones that carry functional meaning. To select important terms representing the same semantics, we start with the removal of useless words, the first step in the process.

Let us consider Q0 the user query.

\[
Q_0 = \{t_1, \ldots, t_n\} \tag{1}
\]

Where \( t_n \) is the \( n \)-term of query preprocessing the query will result in a new query \( Q_1 \)

\[
Q_1 = \{M_1, \ldots, M_n\} \tag{2}
\]

where \( M_n \) is the \( n \)-word that carries the key meaning.

### 3.2. Step 2: General needs detection

The first step allows us (using preprocessing) to identify the most important terms in the query which semantically represent the user need. This second step is meant to detect the general needs expressed in the user query. We first start with a set of application classes (ECA). The purpose of this step is to make a
projection of the terms of the initial query Q1 on the ECA list. This projection allows us to know which application class (AC) does correspond the most to the query Q1 and especially identify elements of the algorithms course on searched by the user. Let us consider ECA the set of application classes

$$\text{ECA} = \{\text{CA}_1, \text{CA}_2, \ldots, \text{CA}_n\}$$

where CA_i is the application class i

The examination of the ECA list at this stage can spell out the general needs of the query. This step results in mining a set of application classes appropriate to the query Q1. The following step is to detect the specific needs of the user so as to refine the detection phase.

3.3. Step 3: Detection of specific needs

The user needs is not always well articulated and often expressed implicitly. To identify the exact needs of the user, we started out with the preprocessing of the initial user query. Then, we proceeded with the detection of the general needs of the user. In this step, we carry on the detection phase by identifying the user’s specific needs. In the is regard, two cases are presented:

- the need is explicitly expressed by the user so it can be directly detected.
- the need it is not explicitly stated and therefore it is detected by making reference to markers in the utilization classes.

We begin this step by examining all utilization classes (ECU) that have already been defined. Once examined, we look for the common elements between the ECU list and the set of terms in the query Q1.

Let us consider ECU the set of utilization classes

$$\text{ECU} = \{\text{CU}_1, \text{CU}_2, \text{CU}_3, \ldots, \text{CU}_n\}$$

where CU is a utilization class and where each CU is defined by a list of markers

$$\text{CU} = \{\text{MQ}_1, \ldots, \text{MQ}_N\}$$

MQ_i is the specified marker i

In the case where matching the ECU list with the terms in the query returns CU set of words, we deduce that this set represents the real specific needs of the query. In the case where the matching returns a null result, we move to a second level of matching between this time the words of Q1 and all markers defined in the utilization classes. The result in this case is a marker which is used the query for reformulation.

From 5, we considered that a CU class is defined by a set of markers. Applying this rule inversely, we deduce CU class that contains the marker returned. In the latter case, the CU class containing this marker is therefore considered specific needs of user query. At the end of this stage, we are in hold of the general needs (from step 2) and the specific needs (from this step). The next step exploits the identified needs, from the previous steps, in the enrichment of the query.

3.4. Step 4: Enrichment of the query

In the preceding steps, we could detect the general and specific needs of the user. These needs are exploited in this step to enrich the user query.

In rule (5), we defined a class with a list of CU markers that are associated with it. From the specific needs (utilization class), we extract all markers associated with this class. These markers are used to enrich the final query. The final query will therefore consist of the general needs, detected in step 2, and all markers extracted from the specific needs, detected in step 3.

The final query will be

$$\text{Req} = \{\text{CA}_1 \text{CA}_2 \text{MQ}_1, \ldots, \text{CA}_n \text{MQ}_N\}$$

**Example:** if the user types in "how to declare a variable"

1. Initially, query preprocessing is used to remove the words that are not useful in terms of semantics, namely "how" and "a". The two terms that remain in the query after preprocessing are "variable" and "declare".

2. In the detection phase of the general needs that corresponds to application class of, we conclude that the user wants to be informed about variables.

3. The detection of specific needs that corresponds to the utilization class allows us to determine the exact needs of the user, the word "declare" in the our example.

4. The query will be enriched with the application class, the word "variable", and markers of the Utilization class, the word "declare", to become as follows.

   $$\text{req} = \{\text{Declare Variable}, \text{declare variable syntax}, \text{variable use}, \ldots\}$$

**Algorithm of the approach**

The following algorithm demonstrates the steps of the enrichment process.

Input : $$\text{Q0} = \{t_1...t_n\}$$

Output : $$\text{Q}'$$

Where Req is the user query

CU is the utilization classes table

Mq is the markers table

We begin by preprocessing with the preprocessing function. Then, we move to perform the first level of concordance between the words of the query and the list of application classes (ECA) in order to identify the general needs. After that, we proceed with we running in the second level of concordance between the words of the query and the list of utilization classes of use (ECU) so as to spot the specific needs. Once the user needs are detected, the query is enriched with the application classes and the markers of the utilization classes that are found.

$$\text{Req} = \text{preprocessing} (\text{Q0})$$

For $$i = 0$$ to $$|\text{req}|$$

If $$t(i)$$ \ ACE then

\begin{align*}
&\text{bgeneral.Add (t (i))}
\end{align*}

else

If $$t(i)$$ \ ECU then


for m = 0a|t (i).
bspecifique. Add (t (i) .mq (m))
end for
else
for k = 0 to.ECU |
for j = 0 to.CU (k) |
If t (i)  mq (j)
for m = 0a|t (i).
bspecifique. Add (t (i) mq (m))
end if
end for
end for
end if
end if
End for
Q' = {bgeneral + bspecifique}

4. Results
To test our approach, the Google search engine has been used as a platform for the implementation of our system. The example cited earlier of a user looking for how to declare a variable in algorithms was closely studied. The system has sent as query "variable declaration, to declare a variable, variable declaration syntax ...". As a result we obtained a list of documents containing the answer to the query "how to declare a variable.". The result returned with our approach seems interesting and has a high accuracy rate compared to the sending of the query without reformulation. The test we have undertaken in this example was run on the 20 first pages returned by Google and showed that without reformulation 11 non-relevant documents feature in the results. Implementing our approach, however, 7 non-relevant documents among the 20 first pages returned. It is noteworthy that our approach represents a significant contribution in the reformulation step of query processing.

Example of pertinent links returned by our approach:
http://mescal . imag . fr / members / Arnaud . Legrand / teaching / 2003/ Algo - DEUG -03/ Course / node 3 . html
http://www . mathmaurer . com / visitors /2 NDE / Algo _01_ bases / www . mathmaurer . com -2 NDE _ Algo _ course _01_ Bases _ in _ algorithmic . PDF
http://www . FSR . AC . my / course / computer / elmarraki / Algo _ ch 1_3 . PDF

5. Conclusion
This study falls within the scope of developing a system for the enrichment of user queries in information search systems in the area of education, namely an algorithms course. To that end, we have used the concept of application and utilization classes, and markers for reformulation. The ultimate aim of this process is to increase accuracy and reduce noise in the returned result. The feasibility of the method has been checked using the Google search engine.

We have shown that our approach represents a contribution in querying while using information search systems. Although the results yielded in our approach are significant, improvement has to be made so that the general computer science course could be taken into consideration and not only the algorithms course. The use of these classes for the construction of domain ontology is the prospect after this study.

References
[9] El younoussi Yacine, Doukkali Sdigu, Ben Lahmer El habib (2010). "Promoting the relevance of the research information systems via a query reformulation


