Exploiting the Search Culture modulated by the Documentation Retrieval applications

Nikitas N. Karanikolas[†], Christos Skourlas[†]

[†] *Technological Educational Institute of Athens. Department of Informatics, Aigaleo 12210, Athens, Greece.*
{nnk, cskourlas} @teiath.gr

Abstract: Th*e documentation retrieval applications have been influenced by each other and have contributed in the creation of some search culture which is shared between the hidden design assumptions (existing behind) of search interfaces and the user skills. The purpose of this work is to exploit the hidden design assumptions in order to define scenarios that combine concrete features for specific retrieval. One such interesting case is the scenario for finding the citations that a researcher has got, excluding self citations.*

Keywords: *abstracting and indexing; citation; documentation retrieval; search engines; search culture.*

I. INTRODUCTION

It is generally acceptable that query languages should be simple and easy to learn for the end-user, and they should provide facilities for the experienced user / information scientist (Gebhardt and Stellmacher 1978). Search behaviour is an area that is often oversimplified, and it is important to highlight different kinds of search activity (Ruthven, 2008). In recent years, the focus has been on search on the Web. Search culture took hold in the late 1990s with Yahoo and Google and a key development occurred with the movement of services onto the Web (Brown and Dumouchel, 2007).

An effective query language is simple enough for novices/end-users but also offers possibilities of complex query languages, to cover specific user needs. An alternative solution is to use a combination of simple form-based searches of a tool, and also templates of ready to use sophisticated and precise queries for specific applications.

Previous studies (Karanikolas, 2011) have focused on the modulated search culture shared between the various applications of documentation retrieval, their interfaces and their users. In this paper, we summarize a portion of web and GUI based search culture to exploit it for evaluating the research work of researchers, on the base of citation counts (NUS Libraries, 2011). In order to achieve our goal, we will concentrate to two concrete documentation retrieval systems/sources (Google Scholar and Scopus) and to a number of customized query templates. It is worth mentioning that there are also other suggestions for evaluating academic reputation of authors (more refined than the simple citation count) (Hou, Li and Niu, 2011; Burns, 2011).

The motivation for using concrete custom query templates, instead of the standard research evaluation methods provided by the specific applications and tools of documentation retrieval systems/sources, is to improve the unsatisfactory results returned by the standard methods. These custom query templates are based on both the complete bibliographical records and also to the full text of articles. Various methods for the evaluation of research activities, provided by documentation retrieval systems/sources, usually underestimate the individual academic's actual impact (underestimate the actual citations count) (Harzing, 2008). There are a lot of reasons for such an underestimation (Chapman, 1989).

The paper is organized as follows: In section 2 we discuss the design assumptions and the related search culture we came across while in section 3 we describe and discuss some experiments. Finally, in section 4 we draw conclusions and point to future work.

II. SEARCH CULTURE AND METHODS

From the diversity of documentation retrieval applications' interfaces and their behavior, previous studies (Karanikolas, 2011) have concluded a set of (hidden) assumptions, which constitute a search culture modulated through time. The set of assumptions includes:

- Possibility for selection of search fields (from the pool of the available ones) by the user for the construction of simple (basic) criteria. An alternative possibility is the apposition of a fixed set of search fields, that can be used into the formation of a query (if the user enters a required value) or remain inactive.
- Possibility for selection of Boolean operators by the user, for building composite queries from simple criteria. An alternative possibility is the suppression of Boolean operators, with the hidden assumption that logical conjunction (AND) is always used.
- Possibility of using (the implied) left to right nesting of simple criteria. There is not any alternative solution based on form-type interfaces. Only the search command languages permit the definition of nested criteria, by using parentheses.
- Semantics of equality is dependent from context. The meaning of equality can vary from "exact match" to "contains" and this is implicitly determined by the used search field in the criterion. Command language based interfaces do not (usually) provide handlers (expressions) for altering the context dependent semantics of equations. Form based interfaces can provide handlers for defining (explicitly by user) the semantics of equality. The user defined semantics of equality can be global or specifically defined for any simple criterion.
- Possibility to use meta-fields in order to search simultaneously, in a set of fields, for the value of a given criterion. Alternatively, for the same purpose,

the system can automatically perform query transcription for creating expanded queries that search the specific (search) value in a set of fields.

- Whenever the data collection contains full-text documents, the search mechanism can provide some kind of word normalization (stemming, lemmatization, etc). Documentation retrieval applications' interfaces can provide handlers for activation of this feature by the user, or ignore it.
- Use of Relevance Ranking. It is mainly applied to free text collections. The documents are ranked (using a similarity function) according to their similarity to the given question and are presented (usually) in descending order of similarity.

III. EXPERIMENTS & PRELIMINARY DISCUSSION

An interesting and useful exploitation of the above mentioned search culture can be applied for evaluating the research work of academicians and researchers. Some interesting types of queries that can be used for evaluating research work are the following:

1. Citations of an author,
2. Citations of an author, without self-citations,
3. Citations of an author, without self-citations from the author or specific collaborator,
4. Citations that have received two (or more) researchers in their common publications,
5. Citations that have received two (or more) researchers in their common publications, without self-citations,
6. Citations of a specific article title, without self-citations from the first author,
7. Citations of a specific article title, without self-citations from any of the contributing authors.

In the following we give templates and examples of queries that can be applied to the selected systems/sources (Google Scholar and Scopus). Query types 1 and 6 are not examined. Using the term Scopus we assume that "Scopus Advanced search" is used and whenever we mention Google Scholar we actually use "Google Scholar search box" (the simple – default – search of Google Scholar). In the following, numbers at end of place holder names (as is number 1 in the following "author name 1" and number 2 in the following "author name 2") are used for permitting the user to provide different values for the same entity (e.g. name variations of the same researcher).

Query type 2 – Google Scholar – template:

```
"author name 1" -author:"author name 2"
```

This template searches for the phrase "author name 1" inside any field in the bibliographic record and in the full text of articles. It excludes from the results all the articles that contain the phrase "author name 2" in the authors' field of the bibliographic record. In Google Scholar this is the only way for finding citations of an author, without self-citations, because the alternative interface (Advanced Scholar search) provides only the possibility for defining that the phrase exist either "anywhere in the article" or "in the title of the article".

Query type 2 – Google Scholar – examples:

```
"n. karanikolas" -author:karanikolas

"n.n. karanikolas" -author:karanikolas

"nikitas karanikolas" -author:karanikolas
```

Since the author names are not always written in the same way, the query type 2 should be executed in Google Scholar with all variations used for filling the template's place holder "author name 1". However, using only the authors surname in the template's "author name 1" will retrieve too many irrelevant articles. On the contrary, the value used for filling the place holder "author name 2" (the value that accompanies the exclusion operator "-author:") should be as simple as possible (only the surname of author). These suggestions have been used in the previous examples.

Query type 2 – Scopus – template:

```
( REF("author name 1")
  AND NOT
  AUTHOR-NAME("author name 2")
)
```

Query type 2 – Scopus – example:

```
( REF(Karanikolas)
  AND NOT
  AUTHOR-NAME(Karanikolas)
)
```

On the contrary to Google Scholar, queries of type 2 can be executed in Scopus by providing only the author's surname as the required value for the REF field. This is a consequence of the structured nature of Scopus and, more precisely, it occurs because the provided surname should be contained in a specific structured field (REF). However, it is a matter of experimentation whether the query will work properly by providing only the author's surname for the REF field or something more precise (e.g. surname with initials) will be needed.

Query type 3 – Google Scholar – template:

```
"author name 1" -author:"author name 2"
-author:"collaborator name"
```

Query type 3 – Google Scholar – example:

```
"n. karanikolas" -author:karanikolas
-author:skourlas
```

Query type 3 – Scopus – template:

```
( REF("author name 1")
  AND NOT
  ( AUTHOR-NAME("author name 2")
    OR
    AUTHOR-NAME("collaborator name")
  )
)
```

Query type 3 – Scopus – example:

```
( REF(Karanikolas)
  AND NOT
  ( AUTHOR-NAME(Karanikolas)
    OR
    AUTHOR-NAME(Skourlas)
  )
)
```

Unfortunately, query type 4 is not supported by Google Scholar, since Google Scholar does not provide fields for structured queries.

Query type 4 – Scopus – template:

```
( REF("author name")
  AND
  REF("other author name")
)
```

Query type 4 – Scopus – example:

```
( REF(Karanikolas)
  AND
  REF(Skourlas)
)
```

Query type 5 – Google Scholar – template:

```
"author name 1" +"other author name 1"
-author:"author name 2"
–author:"other author name 2"
```

Query type 5 – Google Scholar – examples:

```
"n. karanikolas" +"c. skourlas"
-author:karanikolas -author:skourlas
```

```
karanikolas +skourlas
-author:karanikolas -author:skourlas
```

The previous two examples have the only difference that the first one uses surnames with initials in the "author name 1" and in the "other author name 1", while the second one uses only surnames. During the time of writing the present article, these queries returned 3 and 13 results, respectively. Consequently, we should be very careful when using name variations. Using only surnames in place of the "author name 1" and the "other author name 1" will result too many irrelevant articles. But using only one variation of surname with initials in "author name 1" and in the "other author name 1" will drive in loss of relevant articles (loss of citations). The best result will be achieved by repeating the query with different variations of surnames with initials. It is almost the same technique used in the query type 2 examples in Google Scholar. This applies also for query type 3.

Query type 5 – Scopus – template:

```
( ( REF("author name 1")
    AND
    REF("other author name 1")
  )
  AND NOT
  ( AUTHOR-NAME("author name 2")
    OR
    AUTHOR-NAME("other author name 2")
  )
)
```

Query type 5 – Scopus – example:

```
( ( REF(Karanikolas)
    AND
    REF(Skourlas)
  )
  AND NOT
  ( AUTHOR-NAME(Karanikolas)
    OR
    AUTHOR-NAME(Skourlas)
  )
)
```

Query type 7 – Google Scholar – template:

```
"article title" -author:"author1 name"
-author: "author2 name"
-author: "author3 name" …
```

Query type 7 – Google Scholar – examples:

```
"Computer Assisted Information Resources
Navigation" -author:karanikolas
-author:skourlas
```

```
"CUDL language semantics: updating FDB data"
-author:karanikolas -author:skourlas
-author:nitsiou -author:yannakoudakis
```

Query type 7 – Scopus – template:

```
( REF("article title")
  AND NOT
  ( AUTHOR-NAME("author1 name")
    OR
    AUTHOR-NAME("author2 name")
    OR
    AUTHOR-NAME("author3 name")
    …
  )
)
```

Query type 7 – Scopus – example:

```
( REF("Computer Assisted Information
  Resources Navigation")
  AND NOT
  ( AUTHOR-NAME(Skourlas)
    OR
    AUTHOR-NAME(Karanikolas)
  )
)
```

In both cases of using these systems/sources, Google Scholar or Scopus, the last type (7) of queries decreases (and maybe nullifies), in comparison to the queries of type 5, the percentage of irrelevant returned articles. For example, submitting the following query (of type 5) to Scopus has the consequence of returning two articles that do not contain reference to any common (co-authored) article of the mentioned authors (Karanikolas and Vassilakopoulos):

```
( ( REF(Vassilakopoulos) AND
    REF(Karanikolas)
  )
  AND NOT
  ( AUTHOR-NAME(Vassilakopoulos) OR
    AUTHOR-NAME(Karanikolas)
  )
)
```

Obviously, the same problem (of finding false citations) can also happen for queries of type 7. However, in our opinion it extremely rarely occurs in this case. The reference list of each returned paper should be examined in order to verify that the returned document contains valid citations for the provided authors (in case of queries of type 5) or for the provided article title (in case of queries of type 7). The superiority of type 7 queries, that decrease (or even nullify) the false citations, coexists with the possibility of reduced recall of citations, in some cases. The main reason for the reduced recall of citations is usually caused by slight modifications of articles' titles (e.g. substitutions of punctuation marks). Such modifications often occur when authors write their article's reference list. To overcome this drawback of type 7 queries, we should try, instead of the whole title of the article, subsets of consecutive words of the title. For example, the next two (type 7) queries can be used as alternative ways for finding citations for the same publication, through Scopus:

```
( REF("Computer Assisted Assessment (CAA) of
  Free-Text: Literature Review and the
  specification of an alternative CAA system")
  AND NOT
  AUTHOR-NAME(Karanikolas)
)

( REF("Literature Review and the specification
  of an alternative CAA system")
  AND NOT
  AUTHOR-NAME(Karanikolas)
)
```

One considerable conjecture that should be investigated is the following:

The citations that are recognized automatically by the Scopus system/source (the citations included in the index card of a researcher) are always less than the citations that can be found by using either queries of type 5 (for each group of researchers that participates the researcher of interest) or queries of type 7 (for each one of the titles of author's articles). A similar conjecture is also investigated in the case of Google Scholar.

In order to evaluate our conjecture, we suggest a methodology based on type 5 or type 7 queries (templates) for finding citations (excluding self-citations) of an author/researcher. Two alternative versions/variations of this methodology (one for each type of used queries) are synoptically presented in the following.

Methodology variation for type 5 queries:

- Create query of type 5 for finding citations (excluding self citations) that has received one group of (two or more) co-researchers (co-authors) in their common publications and in which participates the researcher under investigation,
- Repeat such kind of queries for anyone of the groups in which the researcher under investigation participates,
- Unify the results of the previous queries, for finding the citation count for the researcher under investigation.

Methodology variation for type 7 queries:

- Create query of type 7 for finding citations (excluding self citations) that has received one concrete publication (on the basis of the publication title),
- Repeat such kind of queries for every article co-authored by the researcher under investigation,
- Unify the results of the previous queries, for finding the citation count for the researcher under investigation.

In the following, we focus on the methodology variation with queries of type 7 and evaluate it for finding citations of an author. We prefer methodology variation for type 7 queries, because it permits us a simpler comparison of systems/sources. It also easily combines results retrieved by equivalent queries submitted in two or more systems/sources. For our experimentation, we can choose a small set of publications (journal articles and conference papers) and create queries of type 7 for each publication and for each system/source: Google Scholar and Scopus. The chosen set of publications contains eight (8) journal articles and ten (10) conference papers. Before providing measurements of efficiency for each system/source, we will concentrate in few of the thirty-six (36, 18 for Google Scholar and 18 for Scopus) queries. These queries have a behavior that should be focused. The following four (4) queries submitted to Google Scholar return only irrelevant results:

```
"Automatic Diagnosis Classification of
patient discharge letters"
-author:karanikolas -author:skourlas
```

0/3 (relevant retrieved / retrieved, precision)

```
"Conceptual Universal Database Language: Moving
Up the Database Design levels"
-author:karanikolas -author:vassilakopoulos
```

0/1

```
"Bootstrapping the Albanian Information
Retrieval" -author:karanikolas
```

0/1

```
"Computer Assisted Assessment (CAA) of Free-
Text: Literature Review and the specification
of an alternative CAA system"
-author:karanikolas
```

0/1

The following two (2) queries submitted to Google

Scholar return only relevant results, while the next adjunct four (4) queries submitted to Scopus (2 queries equivalent with the following 2 queries submitted to Google Scholar and two variations) return nothing:

```
"Interconnection of Laboratory Information
System and Hospital Information System. The
case of ARETEION University Hospital"
-author:karanikolas -author:skourlas
```

1/1

```
"Strengthening the Security of E-Banking
Transactions. The case of NBG"
-author:karanikolas -author:marinakis
```

2/2

```
( REF("Interconnection of Laboratory
  Information System and Hospital
  Information System. The case of ARETEION
  University Hospital")
  AND NOT
  ( AUTHOR-NAME(Skourlas)
    OR
    AUTHOR-NAME(Karanikolas)
  )
)
```

0/0

Variation with part of the title

```
( REF("Interconnection of Laboratory
  Information System and Hospital
  Information System")
  AND NOT
  ( AUTHOR-NAME(Skourlas)
    OR
    AUTHOR-NAME(Karanikolas)
  )
)
```

0/0

```
( REF("Strengthening the Security of E-Banking
  Transactions. The case of NBG")
  AND NOT
  ( AUTHOR-NAME(Marinakis)
    OR
    AUTHOR-NAME(Karanikolas)
  )
)
```

0/0

Variation with part of the title

```
( REF("Strengthening the Security of E-Banking
  Transactions")
  AND NOT
  ( AUTHOR-NAME(Marinakis)
    OR
    AUTHOR-NAME(Karanikolas)
  )
)
```

0/0

From the above groups of queries (with 4, 2 and 4 queries), we have an indication of the increased recall of Google Scholar versus Scopus, and the decreased precision of Google Scholar versus Scopus.

In the following table, we summarize the efficiency of three approaches (the methodology variation with type 7 queries applied to Scopus, using the first mentioned set of 18 queries; the same methodology variation applied to Google Scholar, using the second mentioned set of 18 queries; the standard method automatically provided by Scopus for an author's citation count). The well known measures of precision and recall are used for estimating the performance of the proposed search methodology. Precision is defined as the fraction which is equal to the number of the relevant documents retrieved divided by the number of all the retrieved documents. Recall is defined as the fraction which is equal to the relevant documents retrieved divided by the number of all the documents that are relevant. Next table depicts the results of the proposed methodology and shows the calculation of precision and recall for 18 publications of a specific researcher (Karanikolas) without taking into account the self-citation.

Approach	**precision**	**recall**
Automatically measured by Scopus	5/12	5/18
Using queries of type 7 in Scopus	8/8	8/18
Using queries of type 7 in Google Scholar	10/16	10/18

It is obvious from the previous table that 8 relevant documents are retrieved by queries of type 7 in Scopus and 10 relevant documents are retrieved by queries of type 7 in Google Scholar. Taking into account that 4 of the relevant retrieved documents are common, we can calculate the performance of methodology for the combination of results: precision=14/20, recall=14/18.

It is worth of mentioning that when we use queries of type 5 in Scopus for the same researcher (Karanikolas) and for each of the groups in which he contributes, the following values for precision are calculated: 6/6, 1/1, 0/2 και 0/0. Therefore, the total performance in terms of precision is equal to 7/9 and in terms of recall is equal to 7/18. Comparing the results of using queries of type 5 with the results of using queries of type 7, both in Scopus, we conclude that the calculated precision and recall is better in the later variation of the proposed methodology (the variation with queries of type 7). This is another reason (except the already mentioned simplification of overall combination of results by two or more systems/sources) for suggesting the variation with queries of type 7.

IV. CONCLUSIONS

Our experimentation gave us evidence that our conjecture that "the proposed methodology (either variation with type 5 queries either variation with type 7 queries) improves the estimation of a researcher's citation count versus the automatically measured by Scopus" is valid. It also seems that there is a slight advantage of using templates of queries of type 7 instead of using queries of type 5. Apart from this, the methodology of using queries of type 5 applies fewer and simpler queries and it is therefore an effective and fast solution for estimating the citation count. The combination of results from the use of the methodology variation with type 7 queries in both Scopus and Google Scholar seems to approach very close to the actual number of a researcher's citations. Future work involves performing further experimentation using a corpus of publications organized per research groups, to uncover research policies. Another research direction is oriented towards the examination of other systems/sources and methods.

REFERENCES

David Brown and Bernard Dumouchel, "Understanding user behaviour and its metrics," Information Services & Use, **27**, 3–34 (2007), IOS Press.

C. Sean Burns, "Collecting bibliographic references: A bibliometric analysis of CiteULike's collection as grounds for in-depth interviews," Research round table at the Canadian Association for Information Science, Fredericton, N.B. Canada, June 2011.

Chapman, A.J., "Assessing research: citation-count shortcomings," The Psychologist: Bulletin of the British Psychological Society, **8**, 336-344 (1989).

Friedrich Gebhardt, Imant Stellmacher, "Opinion paper: Design criteria for documentation retrieval languages," Journal of the American Society for Information Science, **29**, Issue 4, 191–199 (1978).

Anne-Wil Harzing, "Google Scholar - a new data source for citation analysis," 2008.

Wen-Ru Hou, Ming Li, Deng-Ke Niu, "Counting citations in texts rather than reference lists to improve the accuracy of assessing scientific contribution," BioEssays, **33**, 724–727 (2011).

Nikitas N. Karanikolas, "Search Culture," Proc. 15th Panhellenic Conference on Informatics, Kastoria, Greece, 229-234 (2011).

NUS Libraries, "Citation Count Workshop for NUS Staff," August 2011.

Ian Ruthven, Interactive information retrieval, Annual Review of Information Science and Technology, Volume 42, Issue 1, pages 43–91, 2008.