

## University of Leeds – School of Computing

DB32 Knowledge Management, Coursework No 1: Building an Expert Finder

David J. Aumueller<sup>1</sup>

MSc in Communications Studies

March 11, 2003

10 pages

## Introduction

The goal of this coursework is to build an expert finder in form of an ‘intelligent’ search engine querying the data given on the website of the Department of Biology about their ‘experts’, i.e. the staff of this school present themselves with descriptions of their publications, research interests and current projects or activities. After populating the data from the website into a SQL Server database a method to measure hits is to be developed. With respect to this measure the actual query engine should be perfected. Finally, the own expert finder is compared with similar products on the web.

The URL of my system: <http://csiis.leeds.ac.uk/ics2dja/db32cw1/>

## The measure – description and results (1/2 page)

My measure ( $z$ ) takes in account two values: the number of returned hits and the position of the matching expert. To receive all hits the amount of rows to be returned is set to 48 (all rows with non-null value in `researchInterests`) or higher in the T-SQL experiment (`match.sql`). Firstly, to get a value between 0 and 1 for the position of the wanted row within the result-set, the following equation<sup>2</sup> is used:

$$x = (\text{hits} - \text{position} + 1) / \text{hits}$$

The 1 is added to get a perfect match (1) for this part of the measure when the expert is on position 1. e.g.  $(3 - 1 + 1) / 3 = 1$  or  $(3 - 2 + 1) / 3 = 0.6$ . The special case that the searched expert is not returned at all is handled separately ( $z = 0$ ).

Secondly, to get a value between 0 and 1 for the amount of returned hits to take into account a similar equation<sup>3</sup> is used:

---

<sup>1</sup> ics2dja@leeds.ac.uk

<sup>2</sup> the better (=lower in number) position, the higher the value (for a fixed number of hits)

<sup>3</sup> the fewer hits, the higher the value (for a fixed collection value)

$y = (\text{collection} - \text{hits} + 1) / \text{collection}$  , with collection being the number of returned distinct expert names. The added 1 is to get a 1 for perfect matches, i.e. when only the right expert is being returned. E.g.,  $(23 - 5 + 1) / 23 = 0.826\dots$  or  $(31 - 3 + 1) / 31 = 0.935\dots$

The two factors are weighed (each multiplied by a constant between 0 and 1; the two constants sum up to 1) and added to get the final measure. The factors are chosen to approximate the resulting order to the following hit order, which is (the beginning of) a list of hits and position values ordered according to my personal judgement/understanding of a good ranking scale for this special experiment where only one hit is relevant (rank gets lower from left to right):

Position:	1	1	1	1	2	2	1	2	1	2	3	3	2	3	3	4	4	4	5
Hits:	1	2	3	4	2	3	5	4	6	5	3	4	6	5	6	4	5	6	5

The weighing factors decided upon are 0.2 for the first value ( $x$ ) and 0.8 for the second ( $y$ ). Thus, the overall formula for the measure is:

$z = 0$  for position = 0; and  $z = 0.2x + 0.8y$  else, with  $x, y$  defined as above.

## Results of the experiment (1/2 page)

As example, two matching experts are presented. For “John Grahame” using column “publications” (collection=23) hits = 7 and position =4 and  $z = 0.705590062$ . For “John Turner” using column “projects” (collection=31) hits = 7 and position =5 and  $z = 0.730875576$ . Collection is defined as the number of returned distinct expert names (see above). Among all resulting experts the average of these z-values is calculated for each column:

Publications:  $z_{\text{pub}} = 0.714672788$  , with collection = 23.

Projects:  $z_{\text{pro}} = 0.684988851$  , with collection = 31.

Rounded values  $z_{\text{pub}} = 0.715$  and  $z_{\text{pro}} = 0.685$  are used further on.

## Design of query

These two values are used to create a weighted combination of the columns publications and projects using full outer joins as indicated in the lecture handout<sup>4</sup>. Using these joins in connection with the ISNULL-function makes sure that no sets are neglected even when some field might be empty. The column “research” is taken into account with an arbitrary weight of 0.1 to not completely loose these value bits of information in the query. (Executing the experiment on that column results in very high values of the measure; in fact all experts but “KJ McDowall” are matched in the first position of the returned hits. The artificiality

---

<sup>4</sup> see slide “Weighting fields independently” in DB32 “Combining inverted and relational data”

of the experiment does not justify the use of this value for weighting in T-SQL, though.) To get a standardised combined rank between 0 and 1000 the rank is divided by the sum of these three z-values, i.e. by  $1.5 = 0.715+0.685+0.1$ .

The column activities can be neglected because it contains hardly information that is not available in the other columns and mostly the data are just links to other pages (which on the other hand a more sophisticated system easily could incorporate these as hyperlinks using e.g. regular expressions).

## The working system

### My Active Server Page, code listing with SQL

```
<!-- #include File="define_connection.asp" -->
<HTML>
<head>
    <title>Expert Finder for DB32cw1 by ics2dja</title>
    <link rel="stylesheet" type="text/css" href="style.css" />
</head>

<! getexperts.asp (was: sample1.asp)
    written by SA Roberts Feb 2002, Revised Feb 2003
    modified by DJ Aumueller Mar 2003
>
<%
    dim adoConn      ' ADO connection object
    dim varErrors    ' Keeps track of number of errors generated
    Set adoConn = Server.CreateObject("adodb.connection")
    make_connection adoConn, "CSMS11", "DB32_ics2dja",
        request.form("login"), request.form("password")
%>

<BODY>
<h1>Expert Finder for DB32cw1 by ics2dja</h1>
<p>Please enter a valid query to identify appropriate experts.</p>

<form action="getexperts.asp" method="post">
<table border="0" cellpadding="10">

<tr>
    <td><p>Query type:&nbsp;  </p></td>
    <td><p>
        <% ' get chosen querytype from radio-buttons
            and activate (check) chosen button again
            dim querytype
            querytype = request.form("querytype")
            dim checked
            %>
        <% if (querytype = "english") then
            checked = "checked" else checked = "" end if %>
        <input type="radio" name="querytype" value="english"
        <% Response.write(checked) %> />English language query<br />

        <% if (querytype = "boolean") then
            checked = "checked" else checked = "" end if %>
        <input type="radio" name="querytype" value="boolean"
        <% Response.write(checked) %> />Boolean query<br />
    </td>
</tr>
</table>
</form>
</BODY>
</HTML>
```

```

    <% if (querytype = "weighted") then
        checked = "checked" else checked = "" end if %>
    <input type="radio" name="querytype" value="weighted"
    <% Response.write(checked) %> />Weighted keyword query

</p></td>
</tr>
<tr>
<td><p>Query:&nbsp;</p></td>
<td><p>
    <input type="text" size="100" name="keyword"
        value="<% Response.write(request.form("keyword")) %>"
    </p></td>
</tr>
<tr>
<td><p>&nbsp;</p></td>
<td><p>
    <input type="submit" value="Find" />
    <input type="hidden" name="login" value="wwwuser" />
    <input type="hidden" name="password" value="wwwpass" />
</p></td>
</tr>
</table>
</form>

<p><a href="http://csiis.leeds.ac.uk/ics2dja/db32cw1/">back</a><p>
<p><hr /></p>
<h2>Result</h2>
<% ' get the Experts that match the keyword and list them
    dim sSqlSelectExpertList
    dim adoRsExperts
    set adoRsExperts = Server.CreateObject("adodb.recordset")

    ' querytype out of (english, boolean, weighted)
    select case querytype
        case ("boolean")
'sSqlSelectExpertList = "SELECT initials, rank, name, research FROM
expertise, expertlinks inner join CONTAINSTABLE (expertise, research,
'" & request.form("keyword") & "') AS ft ON ft.[KEY] =
expertise.[Staff_id] ORDER BY rank DESC, name ASC"

sSqlSelectExpertList = "select initials, name, research,
publications, projects, ROUND(( 0.685*ISNULL(k1.[rank],0) +
0.715*ISNULL(k2.[rank],0) + 0.1*ISNULL(k3.[rank],0) )/1.5,1) AS rank
FROM expertlinks, expertise FULL JOIN CONTAINSTABLE (expertise,
projects, '" & request.form("keyword") & "') k1 on
expertise.staff_id=k1.[key] FULL JOIN CONTAINSTABLE (expertise,
publications, '" & request.form("keyword") & "') k2 on
expertise.staff_id=k2.[KEY] FULL JOIN CONTAINSTABLE (expertise,
research, '" & request.form("keyword") & "') k3 on
expertise.staff_id=k3.[KEY] WHERE NOT (k1.rank IS NULL and k2.rank
IS NULL and k3.rank IS NULL)
AND staff_id=link_id ORDER BY rank DESC, name ASC"

        case ("weighted")
'sSqlSelectExpertList = "SELECT rank, name, research FROM expertise
inner join CONTAINSTABLE (expertise, research, 'ISABOUT('" &
request.form("keyword") & "') AS ft ON ft.[KEY] =
expertise.[Staff_id] ORDER BY rank DESC, name ASC"

```

```
sSqlSelectExpertList = "select initials, name, research,
publications, projects, ROUND(( 0.685*ISNULL(k1.[rank],0) +
0.715*ISNULL(k2.[rank],0) + 0.1*ISNULL(k3.[rank],0) )/1.5,1) AS rank
FROM expertlinks, expertise FULL JOIN CONTAINSTABLE (expertise,
projects, 'ISABOUT(" & request.form("keyword") & "') k1 on
expertise.staff_id=k1.[key] FULL JOIN CONTAINSTABLE (expertise,
publications, 'ISABOUT(" & request.form("keyword") & "') k2 on
expertise.staff_id=k2.[KEY] FULL JOIN CONTAINSTABLE (expertise,
research, 'ISABOUT(" & request.form("keyword") & "') k3 on
expertise.staff_id=k3.[KEY] WHERE NOT (k1.rank IS NULL and k2.rank
IS NULL and k3.rank IS NULL)
AND staff_id=link_id ORDER BY rank DESC, name ASC"
```

```
case else english
'sSqlSelectExpertList = "SELECT RANK, name, research FROM expertise
inner join FREETEXTTABLE (expertise, research, '' &
request.form("keyword") & '') AS ft ON ft.[KEY] =
expertise.[Staff_id] WHERE name is not null ORDER BY rank DESC, name
ASC"
```

```
sSqlSelectExpertList = "select initials, name, research,
publications, projects, ROUND(( 0.685*ISNULL(k1.[rank],0) +
0.715*ISNULL(k2.[rank],0) + 0.1*ISNULL(k3.[rank],0) )/1.5,1) AS rank
FROM expertlinks, expertise FULL JOIN FREETEXTTABLE (expertise,
projects, '' & request.form("keyword") & '') k1 on
expertise.staff_id=k1.[key] FULL JOIN FREETEXTTABLE (expertise,
publications, '' & request.form("keyword") & '') k2 on
expertise.staff_id=k2.[KEY] FULL JOIN FREETEXTTABLE (expertise,
research, '' & request.form("keyword") & '') k3 on
expertise.staff_id=k3.[KEY] WHERE NOT (k1.rank IS NULL and k2.rank
IS NULL and k3.rank IS NULL)
AND staff_id=link_id ORDER BY rank DESC, name ASC"
```

```
end select
```

```
' adoRsExperts is an adodb.Recordset
response.write "<p>Your <em>" & querytype & " query</em> in
T-SQL:</p><p class=query>" & sSqlSelectExpertList & "</p>"
adoRsExperts.open sSqlSelectExpertList, adoConn , 3
```

```
expertsNo = adoRsExperts.RecordCount
```

```
if (expertsNo = 1) then ' singular/plural case
response.write "<h3>There is one expert matching your "
& querytype & " query '" & Request.Form("keyword") & "'.</h3>"
else
response.write "<h3>There are " & expertsNo & " experts matching
your " & querytype & " query '" & Request.Form("keyword") &
"'.</h3>"
end if
%>
```

```
<TABLE CELLSPACING = 3 CELLPADDING = 12>
<tr>
<th>Rank</th><th>Name</th><th>Research</th><th>Projects</th>
</tr>
<tr><td colspan=4><hr /></td></tr>
<% ' list each Expert with ascending counter
dim rowCounter
Do While Not adoRsExperts.EOF
```

```

    rowCounter = rowCounter + 1
%>
<tr><td valign="top">
<% ' show ascending counter and value of T-SQL rank
    response.write "<p class=counter>" & rowCounter & "</p>"
    <p class=digit>" & adoRsExperts("rank") & "</p>"
%>
</td><td valign="top">
<% ' show expert's name with link to the corresponding
    webpage on the website of the Biology Department
    response.write "<p class=small>"
    <a href=http://www.fbs.leeds.ac.uk/staff/staff.htm?uid="
        & adoRsExperts("initials") & ">"
    response.write adoRsExperts("name") & "</a></p>"
    response.write "<p class=tiny>"
    (<a href=http://www.fbs.leeds.ac.uk/?staff=" &
        adoRsExperts("initials") & " target=_blank>new window</a></p>"
%>
</td><td valign="top">
<% ' show research column as used for query
    response.write "<p>" & adoRsExperts("research") & "&nbsp;</p>"
%>
</td><td valign="top">
<% ' show project column as used for query
    response.write "<p>" & adoRsExperts("projects") &
        "&nbsp;</p></td></tr>"
%>
<tr><td colspan=2 valign=top align=right>
    <p><b>Pub's:</b></p></td><td colspan=2>
<% ' show publication column as used for query
    (in separate row below research and project field)
    response.write "<p>" & adoRsExperts("publications") &
        "&nbsp;</p><hr /></td></tr>"
    adoRsExperts.MoveNext
    loop
%>
</TABLE>

<%
    adoRsExperts.Close ' close recordset
    adoConn.Close      ' close connection
%>
</BODY>
</HTML>

```

## Example queries

### First stage, only using column "research" for query

11 results for english query "plants and leaves": (showing first two only)

```

311 Dr David J. Pilbeam
102 Dr Hillel Fromm

```

1 result for boolean query "plants and leaves":

```

48 Dr David J. Pilbeam

```

5 results for weighted query "plants weight (0.8), leaves weight (0.7)":

```

84 Dr David J. Pilbeam
47 Prof. Peter Meyer

```

3 results for english query "the sound of bats in flight":

301 Prof. Jeremy M.V. Rayner  
274 Dr Dean A. Waters  
130 Prof. John D. Altringham

2 results for boolean query "(sound or flight) and bat":

26 Dr Dean A. Waters  
26 Prof. Jeremy M.V. Rayner

2 results for weighted query "sound weight(.4), flight weight (.1), bat weight(.5)":

65 Dr Dean A. Waters  
62 Prof. Jeremy M.V. Rayner

### Improved/final version with queries weighted over three columns

There are 14 experts matching your english query 'plants with leaves'.

1	231.4	Prof. Peter Meyer
2	181	Prof. Howard J. Atkinson
3	175.9	Dr Jurgen Denecke
4	121.1	Dr Stephen G.A. Compton
5	104.5	Dr Hillel Fromm
6	100.1	Prof. David J. Cove
7	96.8	Dr Henry M.R. Greathead
8	81.4	Prof. J. Michael Forbes
9	55.1	Dr Brendan H. Davies
10	20.7	Dr David J. Pilbeam
11	6	Dr Celia D. Knight
12	3.3	Dr Geoffrey M. Whiteley
13	3.3	Dr Lynton D. Incoll
14	3.3	Dr William E. Kunin

There is one expert matching your boolean query 'plants and leaves'.

1	3.2	Dr David J. Pilbeam
---	-----	---------------------

There are 6 experts matching your weighted query 'plants weight (0.8), leaves weight (0.7)'.

1	27.9	Prof. Peter Meyer
2	24.8	Prof. Howard J. Atkinson
3	19.7	Prof. J. Michael Forbes
4	5.6	Dr David J. Pilbeam
5	1	Dr Celia D. Knight
6	1	Dr Lynton D. Incoll

There are 7 experts matching your english query 'the sound of bats in flight'.

1	165.1	Dr Graham N. Askew
2	118.2	Dr Jens Krause
3	118.2	Prof. Christian D. Thomas
4	101.8	Dr Dean A. Waters
5	91.3	Prof. Jeremy M.V. Rayner
6	79.9	Prof. John D. Altringham
7	41.6	Prof. Roger K. Butlin

There are 2 experts matching your boolean query '(sound or flight) and bat'.

1	1.7	Dr Dean A. Waters
2	1.7	Prof. Jeremy M.V. Rayner

There are 6 experts matching your weighted query 'sound weight (.4), flight weight (.1), bat weight (.5)'.

1	542.5	Dr Graham N. Askew
2	244.3	Prof. Jeremy M.V. Rayner
3	240.2	Prof. John D. Altringham
4	113.4	Dr Jens Krause
5	113.4	Prof. Christian D. Thomas
6	4.3	Dr Dean A. Waters

## Suggested and implemented improvements

The query based interface allows to choose between English language, Boolean and weighted keyword query via (X)HTML radio buttons. It is assumed that the query-string itself is entered syntactically correct according to the needs of the selected query type. The resulting hits are listed numbered including following fields: rank, name (with hyperlink<sup>5</sup> to the expert's webpage on the Department's website), research, projects and publications. The constructed T-SQL query code is shown. For ease of use to state another query, the search form is available at the top of the result page.

On the result page the matched keywords could be highlighted to guide the user to the searched information. Since the search engine uses stemming this is a more difficult task than incorporating exact matches only where some regular expressions suffice.

The result page could offer to ask feedback from the user by finding and presenting special keywords from controlled index terms to allow narrowing using (subordinate) hyponyms, or broadening, using hypernyms (superordinate), the query by clicking on these terms.

Also nice to see would be the experts' contact details such as email-address.

Based on the experiments which resulted here in very similar outcomes of the measure it is important to use more than one column for the query so that the valuable pieces of information of columns are not lost (some experts might do research not based on their previous publications, for example). Incorporating more columns, on the other hand, makes it likely to retrieve much more hits (too many to find the most relevant documents?), and this is what happens as can be seen in the example queries above.

---

<sup>5</sup> for this, staff\_id is matched with another table "expertlinks" which holds the initials of the experts. These initials are used to build the URL to link to the external website.

## Evaluation and comparison

The indexing language is uncontrolled and post-coordinate. Depending on the query type matching is exact (for Boolean and weighted) or partial (for English). No query broadening is evident. The reported information on number of hits is useful, the value of the rank between 0 and 1000 as offered from SQL Server is hardly to derive from the stated query and returned results. A more intuitive measure would be an improvement.

Using Boolean queries with AND-connector usually results in the least amount of hits, due to the obvious fact that both (or all) keywords have to be in the result but also because the words even have to be present in this exact spelling. The English language query results in much more hits due to partial matches and stemming. Also it is more 'relaxed' since it may ignore some keywords.

### Other systems

**Bath:** Using keywords "plant/s", "(zebra)fish", or "toxin" resulted in zero hits although there are relevant experts in the Bath Department of Biology and Biochemistry<sup>6</sup>. It seems that this expert finder does not use a very up-to-date data-pool, in fact, the engine uses pages specially designed for finding experts and thus does not seem to reflect the current status in the various departments. Moreover, the search-engine used for these pages, "htdig", an open-source project, is not configured very sophisticatedly, since e.g. a search for "plant" and "plants" result in two completely different sets, i.e. "htfuzzy" could have been utilized<sup>7</sup>. Overall identified system properties: both browse- and query-based, uncontrolled index language, Boolean query available, exact match, no broadening, number of hits given. Fast.

**California:** Database seems to be well-kept and up-to-date. Instead of entering keywords the interface only allows to (browse through and) select multiple pre-defined keywords to build the query using either all or any of the selected keywords. Overall: Query-based, controlled indexing language, post-coordinate, Boolean logic, no broadening, order of hits undefined. Slow.

**Queensland:** Query-based, uncontrolled indexing language, Boolean logic, exact matching, no broadening, number of hits reported. Fast. E.g., exact Boolean match "sound and recording" returns hits, but "sound and recordings" not.

---

<sup>6</sup> see <http://www.bath.ac.uk/bio-sci/bsstaff.htm>

<sup>7</sup> see <http://www.htdig.org/>

**Cardiff:** Query based, uncontrolled indexing language, Boolean logic, no stemming but partial matching, no broadening, number of hits reported. Fast.

**Dublin:** only browsing through subject areas (alphabetically) without connection to experts possible.

## **Conclusion**

The presented system works much better than most of the ones looked at on the web. The 'California' expert finder uses a distinctive approach in allowing only predefined keywords. This seems not too intuitive, although a combination of free keyword search and refinement/feedback using predefined keywords could be an ideal system. Web users now are used to do free-text and even Boolean queries, but entering weights is not very common. Again, more sophisticated interfaces with feedback are needed to ease the use in this case. Also, most of them are used to get exact matches which is used in most search engines, but partial matches (with stemming) offer much better results since that way the user need not be aware of the various forms of words. The user can concentrate on the meaning of her/his search terms.

High influence has to be seen on the underlying data. Data that is well edited and supplied with metadata (using RDF for example) 'searching' could become 'finding'. Improvements from both sides (provided data and search engine) are needed to accomplish this high goal.