Using Oracle® for Natural Language Document Retrieval An Automatic Query Reformulation Approach

Jens GRIVOLLA
Laboratoire Informatique d'Avignon (LIA)
jens.grivolla@univ-avignon.fr

ABSTRACT

In corporate applications, vast amounts of data are often stored in database systems such as Oracle. Apart from structured information this can include text documents which cannot easily be retrieved using traditional SQL queries.

Oracle includes means to deal with full text document retrieval (called *Oracle Text*) that offer special query operators for searches inside text fields. We have explored the effect of these different operators for queries derived from natural language queries. This article compares the retrieval performances achieved with different automatic reformulations from natural language to Oracle SQL queries.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

General Terms: Performance

Keywords: Oracle, natural language, query reformulation

1. INTRODUCTION

When dealing with great amounts of text data stored in a corporate database it can often be interesting to use a natural language query interface to search for information. However, it is usually not practical to have a natural language document retrieval system in addition to the existing structured database access.

Oracle addresses this issue by including full text retrieval methods in their database product. They have shown the potential of their methods through their participation at the TREC 8 evaluation campaign [3] where they obtained good retrieval performances [1]. Unfortunately, these results were obtained with manually reformulated queries, showing the best use of the available query operators by a team of expert users.

In an attempt to make some of that potential available to non-expert users of a database, we have investigated different means to automatically translate natural language (plain english) queries into SQL statements.

2. MOTIVATIONS AND APPROACH

For full text retrieval, Oracle allows us to use SQL statements of the following form:

SIGIR'05, August 15–19, 2005, Salvador, Brazil.
ACM 1-59593-034-5/05/0008.

```
select * from TABLE where contains(TEXTCOLUMN,
<query>, 1) > 0 order by score(1) desc;
```

This will return all documents (rows from table `TABLE`) for which the score obtained by matching the query to the contents of `TEXTCOLUMN` is positive.

A number of operators are available for building the query, some of which are:

`ABOUT` searches for themes, using the supplied knowledge base
`ACCUMulate (,)` ranks by cumulative score for all terms
`AND (&)` all terms need to be present
`OR (|)` any of the terms has to be present
`NEAR (;)` scores by proximity of terms
`stem ($)` uses stemming on term (find all terms with same stem)

The submission of the Oracle team at TREC 8 shows that it is possible to achieve very interesting performances for natural language document retrieval using the tools provided by *Oracle InterMedia* or *Oracle Text*.

At the same time, the examples of reformulated queries given in the article show that these results were obtained by combining several of the available operators, carefully selecting adequate search terms and phrases, and manually assigning weights to them.

Topic 414 (Cuba, sugar, exports) thus became:

```
ABOUT(cuba) * 3, ABOUT(sugar) * 5, ABOUT(Russia) *
2, (ABOUT(export) or ABOUT(import))
```

The query for Topic 450 (King Hussein, peace) was even more complex:

```
(King Hussein=Hussayn=Husayn=Husyan * 5,
ABOUT(peace talks) * 3, peace * 2, ABOUT(Israel))
MINUS (Iraq or ABOUT(Iraq))
```

We have used the same query and document corpus from TREC 8 and have generated several sets of queries, applying different transformations in order to obtain valid SQL statements.

2.1 Reformulations

We started out using the different operators in their basic form, then continued with combinations, and added term weighting based on term document frequencies (directly extracted from the database), similar to the classic $TF.IDF$ weighting in the vector space model [2].

A query such as Topic 401,

```
What language and cultural differences impede the
integration of foreign minorities in Germany?
```

would thus be converted to a variety of forms, e.g.:

```
$What * 2.48, $language * 2.96, $cultural * 4.82,
$differences * 4.79, $impede * 7.37, $integration
```

transformation type	mean avg. prec. medium	short
ACCUM	7%	16%
AND	1%	13%
NEAR	0%	14%
OR	1%	4%
stemmed	7%	17%
stemmed+weighting	12%	19%
about	16%	19%
about+weighting	13%	17%
about (separate terms)	10%	15%
about (expanded query)	7%	14%
about+stemmed	7%	20%
about+stemmed+weighting	13%	21%
ultrasearch	7%	17%

Table 1: mean average precision obtained for short (*title*) and medium (*title and description*) queries from TREC 8

`* 5.28, $foreign * 2.83, $minorities * 6.07, $Germany * 3.87` (with stemming and additional term weighting).

Oracle gives some examples of natural language to *Oracle Text* query transformations for their application *Ultrasearch*, which combine in decreasing order of importance exact phrase matching, `NEAR-` , and finally `ACCUM`-queries.

3. EXPERIMENTAL RESULTS

We have tested a variety of query reformulations on the different sets of natural language queries provided for the TREC campaign, in particular very short *title only* queries consisting of just a few keywords, as well as mid-length full sentence queries, composed of a *title* and a *description* part.

An overview of the average precision achieved with these different query sets is shown in table 1.

The Oracle team had achieved around 47% average precision with their TREC participation, which is far better than the performances of even the best automatic systems, and a good performance for a "manual" system. Our results should rather be compared to other automatic systems, which typically obtain an average precision of around 25 to 30%.

We notice that basic `ABOUT` queries yield quite acceptable results, while still clearly below those of systems specifically designed for TREC type queries. Simple accumulation of query terms (with or without stemming) gives significantly weaker performances, especially for longer queries.

Adding additional term weighting (Oracle already uses some term document frequency based weighting internally) improves performances considerably for `ACCUM` queries. However, this is not easily reproduced for `ABOUT` queries. This seems to be due to a massive degradation of performances when applying the `ABOUT` operator to individual query terms rather than the entire query, which is necessary in order to add individual term weights.

We have sought to obtain more insight into the exact treatment of theme expansion in `ABOUT` queries, which can be obtained via the `ctx_query.explain` procedure. However, the expanded queries returned by that procedure do not reach the performances of the initial query, possibly due to a loss of information about relative weights of the individual terms and phrases in the expanded query.

`AND`, `NEAR` and `OR` expressions have proven to be entirely inadequate for longer queries, the first two being far too restrictive for the TREC setting and returning hardly any hits at all while the evaluation is based on having 1000 documents retrieved for every query. The `OR` operator on the other hand considers all documents containing at least one of the terms as equally good and therefore obtains very low precision. Use of these operators in combination with other means of formulating the query has either no effect, or decreases precision, except when applied manually for very specific queries. The query transformation as described in the *Ultrasearch* application thus gives the exact same result as a plain `ACCUM` query.

For short keyword queries, `AND` and `NEAR` do return adequate results. However, they do so with far less documents returned (which may be good depending on the specific application setting). For some queries, they do not return any documents. The mean average precision in the table is calculated only over those queries that do have results, and should therefore actually be lower than indicated when averaging over all queries.

4. DISCUSSION AND PERSPECTIVES

We found that it is possible to obtain good retrieval results for short queries, especially when combining theme (`ABOUT`) and term (`stem`) querying with additional weighting.

Based on the gain obtained by adding term weights to `stem` queries, we had hoped to be able to similarly enhance longer queries using the `ABOUT` operator. Unfortunately, too much of the internal processing of those queries appears to be hidden from the user, and it is very difficult (or even impossible) to tweak the way the documents are ranked.

While performances simply using the `ABOUT` operator with no further work on the query can be acceptable, particularly for short queries consisting essentially of selected keywords, other systems that are specifically designed for natural language document retrieval do perform considerably better. It would therefore still be of great interest to be able to optimize the retrieval process for the type of queries in a particular application setting.

We are continuing to work on some other query transformation and try to extract further information about the internal expansion and term weighting within the Oracle system. We are also currently validating our results on other collections, such as the queries from TREC 6 and 7, as well as possibly in the future entirely different document and query sets.

We also noticed a possible problem resulting from the fact that *Oracle Text* only assigns integer scores from 0 to 100 to the documents. This results in a great number of ties, where the ordering of the documents in the results list is not determined. We are investigating the effect this can have on retrieval performance, especially with regard to average precision, and eventually ways to avoid this problem.

5. ACKNOWLEDGEMENTS

The work presented in this article was conducted with the help of *Digitech S.A.* and *Région PACA*.

6. REFERENCES

[1] K. Mahesh, J. Kud, and P. Dixon. Oracle at TREC 8: A lexical approach. In *TREC-8 Proceedings*, 1999.

[2] G. Salton. Developments in automatic text retrieval. In *Science*. 1991.

[3] E. M. Voorhees and D. Harman. Overview of the eighth text retrieval conference. In *TREC-8 Proceedings*, 1999.