Running CLEF-IP experiments using a graphical query builder

W. Alink, R. Cornacchia, A. P. de Vries
Centrum Wiskunde en Informatica
{alink,roberto,arjen}@cwi.nl

Abstract

The CWI submission for the CLEF-IP track shows results for out-of-the-box querying using a graphical strategy design interface. This domain-independent search platform has been enriched with patent-specific information, which was then readily available to the query interface. The search strategies for the 4 runs submitted have been constructed by simple drag&drop operations in this graphical interface, subsequently compiled into probabilistic relational algebra (PRA) [3] and SQL, and then executed on a relational high-performance database system [2].

The four search strategies compare boolean search, ranked retrieval, and category-based re-ranking. The main lesson learned is that using selection on category only yields a high recall.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.3.7 Digital Libraries; H.2.3 [**Database Management**]: Languages—*Query Languages*

General Terms

Measurement, Performance, Experimentation

Keywords

Question answering, Questions beyond factoids

1 Introduction

The participation of the CWI in the CLEF-IP track has been triggered by the interesting results of a recent project, called LHM, in which a flexible approach towards querying intellectual property documents was developed.

The approach uses a probabilistic database language on top of a scalable database system, instead of using a dedicated IR system. In the user interface the query strategy can be graphically assembled, to provide non-computer scientists the tools to develop complex search strategies. The usage of a probabilistic database language abstracts the query design from directly having to deal with probabilities in queries. The usage of a high-performance database system makes it possible to execute the complex queries on real world data sizes.

The system is still under development, and the retrieval strategies have not been tuned yet. With the submitted runs (4 in total) it is shown that the system is capable of efficiently executing

different retrieval strategies without the need of re-programming or re-configuring it. More details on the runs are provided in Section 4.

2 Objectives

The main objectives of the submitted runs are to show flexibility in expressing different strategies for patent-document retrieval using a seamless combination of information retrieval and database technologies: possibility to mix freely probabilistic (IR) and exact (DB) match criteria, and a neat separation of the retrieval specifications (IR) and physical data management (DB).

Tasks performed by intellectual property specialists are often ad hoc, and continuously require new approaches to search a collection of documents. Our objective is therefore to focus on the ease of expressing new search strategies for IP search experts who not necessarily have high IR/database expertise. Intellectual property specialists also need a high degree of control over the searches that are performed. Our objective is therefore also to be able to combine exact match operators with ranking operators, and provide the intellectual property specialist with an intuitive overview of the search steps used in his strategy, so that results can be explained and verified.

By targeting the XL experiments, our aim also includes to build a scalable solution, although due to time restrictions, and the early stage of the system used, this issue has been not been thoroughly addressed.

3 Approach

The CWI submission for the CLEF-IP track was powered by the *LHM* project, a joint project with Apriorie [5] and a leading IP search provider company. The main aim of the project is to build an integrated system composed of:

Strategy Builder: a graphical user interface that enables patent experts to create complex search strategies in a drag&drop fashion.

HySpirit: a software framework for probabilistic reasoning on relational and object-relational data, developed by Apriorie.

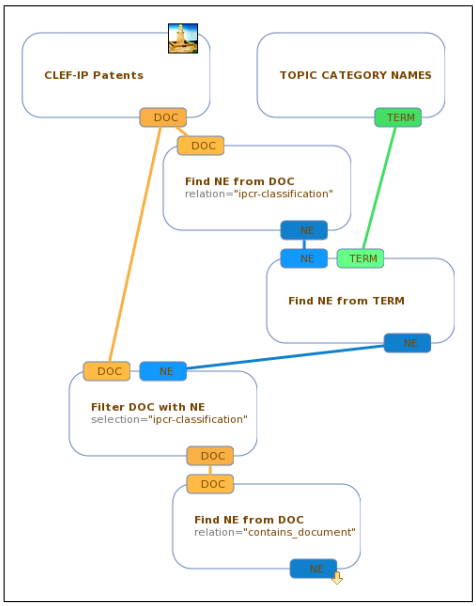
MonetDB: an open source high-performance database management system developed by CWI.

The hypothesis is that such an integrated system enables users of the Strategy Builder to formulate and execute expressive queries and retrieval strategies efficiently on the large-scale patent corpus of the CLEF-IP track. This goal is achieved by implementing several automatic translation steps. First, the graphical, user-oriented, strategy is composed of *building blocks* which are internally expressed in terms of the HySpirit Probabilistic Relational Algebra (PRA). This guarantees the search strategy to be grounded on a solid theoretical framework, that properly propagates relevance probabilities throughout the whole search process, while hiding explicit management of such probabilities. Second, the PRA specification is translated to a database query, and executed on the high-performance database engine MonetDB, using the standard SQL query language.

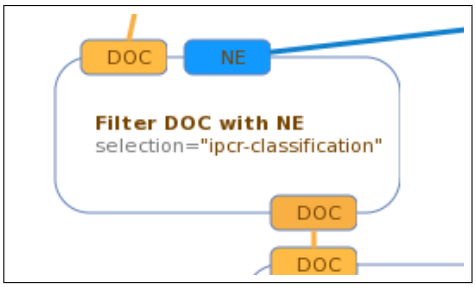
Fig. 1 shows the *category*-run strategy and excerpts of the intermediate compiled strategy representations. A graphical representation of the complete *category*-run strategy is shown in Fig. 1a. Fig. 1b zooms in on a single *building block* of that strategy. The corresponding probabilistic (PRA) that is attached to this building block is shown in Fig. 1c. The compilation of the PRA snippet yields the SQL code depicted in Fig. 1d which can be directly executed on a database engine.

4 Tasks performed

In total 4 runs have been submitted to the CLEF-IP track. A short explanation of each of the runs:



(a) Strategy Builder - a complete strategy



(b) Strategy Builder - a building block

```
BLOCK_SOURCE(docID) = INPUT1_result
BLOCK_SUBJECTS(neID) = INPUT2_result

BLOCK_prd(neID,docID)
 = PROJECT ALL [neID, docID] (
   SELECT[predicate="%SELECTION%"] (
    INPUT1_ne_doc ) );
BLOCK_nes(neID, docID)
 = PROJECT ALL [neID, docID] (
   JOIN INDEPENDENT [docID=docID] (
    BLOCK_SOURCE_result,
    BLOCK_prd ) );
BLOCK_result(docID)
 =  PROJECT DISTINCT[docID] (
   JOIN INDEPENDENT [neID=neID] (
    BLOCK_SUBJECTS_result,
    BLOCK_nes ) );
```

(c) PRA query for the building block in Fig. 1b

```
CREATE VIEW BLOCK_prd_1 AS
 SELECT neID AS a1, predicate AS a2,
        docID AS a3, prob
 FROM  INPUT1_ne_doc
 WHERE predicate='%SELECTION%';

CREATE VIEW  BLOCK_prd AS
 SELECT a1, a3 AS a2, prob
 FROM  BLOCK_prd_1;

CREATE VIEW  BLOCK_nes_1 AS
 SELECT INPUT1.a1 AS a1,
   BLOCK_prd.a1 AS a2,  BLOCK_prd.a2 AS a3,
   INPUT1.prob *  BLOCK_prd.prob AS prob
 FROM  INPUT1,  BLOCK_prd
 WHERE INPUT1.a1= BLOCK_prd.a2;

CREATE VIEW  BLOCK_nes AS
 SELECT a1, a3 AS a2, prob
 FROM  BLOCK_nes_1;

CREATE VIEW BLOCK_result_1 AS
 SELECT INPUT2.a1 AS a1,
   BLOCK_nes.a1 AS a2, BLOCK_nes.a2 AS a3,
   INPUT2.prob * BLOCK_nes.prob AS prob
 FROM  INPUT2, BLOCK_nes
 WHERE INPUT2.a1=BLOCK_nes.a1;

CREATE VIEW BLOCK_result AS
 SELECT a3 AS a1, 1-prod(1-prob) AS prob
 FROM BLOCK_result_1 GROUP BY a3;
```

(d) SQL query for the building block in Fig. 1b

Figure 1: Life-cycle of the *category*-run strategy used for the CLEF-IP track: from the graphical strategy builder to the executable SQL query.

boolean-run: In this run an attempt is made to mimic boolean retrieval. From the topic document 10 words with the highest tf-idf score are taken. Patent-documents that match at least half (5) of the words taken from the topic document are considered a match. From these patent-documents the patents have been found. This yields on average roughly 1000 matching patents per topic. In case more patents have been retrieved, only the first 1000 were submitted as result.

The reason we think this somewhat resembles boolean retrieval is that from various patent search experts we have heard that often the initial phase of a search is done by selecting key terms from the patent under inspection, and generating such a query with those words so that an amount of results is retrieved of which it is feasible to read all the abstracts.

bm25-run: A well-known and often applied strategy in information retrieval is the BM25 relevance model [7]. Our *bm25*-run uses 15 keywords taken from the topic patent, and searches the patent-document collection using the BM25 formula.

The keywords taken from the topic document are weighted based on tf-idf. The initial weight of the keywords is taken into account when ranking the documents.

category-run: In the *category*-run patent-documents are selected that matched one or more IPCR-categories of the topic-patent. The IPCR-categories are weighted based on idf. The patent-documents are ranked by the sum of matching category scores.

category-bm25-run: uses the category strategy and applies the bm25 strategy to the results of this strategy. This run combines the *bm25*-run and the *category*-run. First the patent-documents are selected that match one or more of the categories in the topic-patent, and afterwards this set of documents is searched using keywords extracted from the topic patent. Scores are propagated at each step, so BM25 gets as input a list of weighted documents. The same keywords have been used as were used in the *bm25*-run

For the *boolean*, *bm25*, and *category-bm25* runs, text search has been performed on all the 'textual' sections of a document (title, abstract, description, and claims), and no specific sections have been queried.

5 Experimental Setup

In our approach the process of creating indices for the data is separated from querying the data. The same indices are used for each of the submitted runs. The only differences between the 4 runs are changes in the strategy.

The schema used is comparable to RDF [9] triple schema; all entries are subject, predicate, object tuples. The most noticeable difference of our schema compared to RDF is that probabilities are attached to each tuple. The CLEF-IP corpus has been provided as a set of XML documents in a custom XML format. All data has been loaded as XML using MonetDB/XQuery [1]. Keyword indices have been created using the PF/TIJAH indexer [4]. Structural relations between patents are obtained by using a domain specific knowledge that was expressed as a set of XQuery [10] queries.

The strategies are expressed in the strategy builder's 'building blocks'. Each building block contains snippets of PRA code, and the combined code is then compiled into a full PRA expression, Subsequently the PRA is compiled into SQL statements using the PRA2SQL conversion of the HySpirit engine. This compiler is a result of the LHM project. The final script is then executed on a MonetDB/SQL database engine.

The strategy-builder had initially been build to run a single query with a given set of parameters. To allow the system to execute a full run of topics at a time, it was changed in such a way that it would compile the query-template once, and then would substitute the keywords and categories for each topic.

The main software components that are used:

- LHM Strategy Builder v0.2, configured with a specific workspace 'CLEF-IP'

run	MAP	nDCG
boolean	0.0217	0.3043
bm25	0.0774	0.4219
category	0.0453	0.3256
categorybm25	0.0739	0.3857
other1	0.2783	0.5816
other2	0.1206	0.4441

Figure 2: MAP and nDCG for our runs for bundle M, task Main. For comparison two runs from other participants have been added. (source: [6], Table 9)

- HySpirit PRA 2 SQL (HySpirit version 2.4.9.3)
- MonetDB Feb2009 SP2 release, patched so the MonetDB/TIJAH indexer results can be used in MonetDB/SQL.

The data is physically distributed over 4 different databases, each holding the indices of 500k patent-documents. Creating the indices for the documents took little over 10 hours. For the experiments on the 2 million patent documents we have been allowed to use some of the IR Facility resources; the LDC, an Altix-4700 machine which has ample resources to use (80 cores and 360GB of shared memory). Both during indexing and querying (only) 4 cores in parallel are used. During querying 3 GB of memory per database are needed. During indexing much more memory is needed, but due to the fact that the LDC has ample memory available (360GB) no problems occurred.

6 CLEF-IP Results

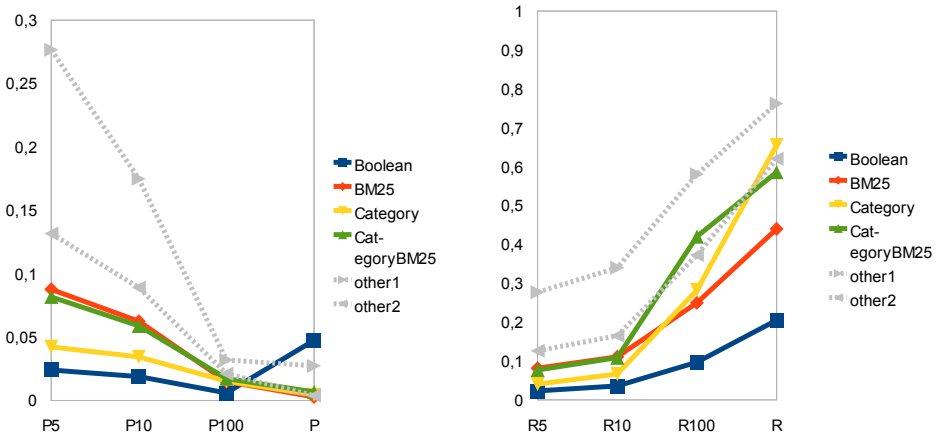
Results of the CLEF-IP runs have been made available in [6], and are summarised in [8]. To describe the overall result of the CWI submission for the CLEF-IP: in terms of scores none of the runs have real good results, at least when compared to other participants. The results over the S, M and XL bundles seem similar. The results for the M bundle (Table 9 in [6]) are used here for analysis, as this is the largest bundle for which all our 4 runs submitted results. The most interesting item using only category information of a patent can yield high recall In Fig. 3a and Fig. 3b the precision and recall scores are presented for the 4 submitted runs. For comparison, the results of other participants that obtained the highest scores are also shown.

There are a few observations to be made when looking at each of the runs individually. The *boolean*-run provides poor retrieval quality. The reason why it has relatively high precision is probably due to the fact that not always 1000 results are returned, and often much less. The *category*-run has high recall, but MAP2 is very low.

The *bm25*-run resembles the most classic method of ranking documents. Results are (slightly) lower than other participants methods. This is perhaps due to the fact that poor parameters have been used. The *category-bm25*-run does somewhat improve precision and recall over the *bm25*-run, but MAP is lower. Compared to the *category*-run, the *category-bm25*-run does somewhat improve precision and MAP, but recall is lower.

A part of the explanation for the results could be the aggregation of patent-documents to patents. For each of the runs the same aggregation method is used. The selection of the patents has in each of the runs been the final operation of the strategy: in the *category-bm25*-run the intermediate results between the category and bm25 part are patent-documents and not patents.

The execution time for a single topic in the *category*-, *bm25*- and *boolean*-runs is roughly 3 to 6 seconds. The execution time for a single topic in the *bm25-category*-run is roughly 30 seconds. Why the *bm25-category*-run is much slower than the other runs has not been analysed in detail, but it may possibly be due to query plan optimisation for this (more complex) query.



(a) Precision at 5, 10, 100 and overall precision for bundle M, task Main.In gray are highest scores by other participants.

(b) Recall at 5, 10, 100 and overall recall for bundle M, task Main.In gray are highest scores by other participants.

Figure 3: Precision and recall curves for our 4 submissions, for comparison two runs from other participants have been added. (source: [6], Table 9)

7 Conclusion

Participating in CLEF-IP 2009 has been an interesting experience. The first of our objectives, flexibility and ease of use, is reached: constructing strategies in a graphical interface using high-level abstract concepts worked well and proved to be flexible enough to express the retrieval strategies for the CLEF-IP 2009 submission, without any collection-specific additional coding. In particular, combining exact and ranked matches required no effort: this distinction and the proper propagation of scores (probabilities) are totally transparent in the graphical user interface. The integration with a general purpose database engine as a back-end worked smoothly as well, with all the physical details abstracted away from the query interface. It would be interesting to see whether other retrieval strategies used in CLEF-IP 2009 could be easily formulated in our Strategy Builder.

The second of our objectives, scalability, is only partly reached: we were able to handle the 2M patents of the CLEF-IP 2009 corpus, but this is still an order of magnitude less than all patent-documents digitally available.

Quality of retrieval results is not excellent, and should be improved. The main interesting results: the *category*-run exhibits good recall, but poor MAP. This could mean that the IPCR classification is good for selecting relevant patent-documents, but seems a poor criteria for ranking in the way it is used in our strategy. More investigation is needed to determine whether category information can be effectively used for ranking patents.

The main points to be improved, or at least to be further investigated:

- The parameters used in the BM25 ranking formula were not chosen carefully, also the 'patent-document to patent' aggregation may be an interesting point of research, and could be improved upon.
- For intermediate steps inside a composed strategy, it might be more useful to regard all patent-documents of a retrieved patent again, instead of the individual retrieved patent-documents. This has not been analysed.
- The strategies used are oblivious of the language in which the patent has been written. Better retrieval should be possible if language is taken into account

- IDF of top topic-terms is computed over the XL bundle of topic documents, rather than over the patent corpus. It is currently unknown whether this has a high effect on the results.

Finally, we would like to thank the IRF for providing access to the LDC, which made the experiments much easier to perform.

References

[1] Peter Boncz, Torsten Grust, Maurice van Keulen, Stefan Manegold, Jan Rittinger, and Jens Teubner. Monetdb/xquery: a fast xquery processor powered by a relational engine. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 479–490, New York, NY, USA, 2006. ACM.

[2] CWI. MonetDB website. `http://www.monetdb.nl/`.

[3] Norbert Fuhr and Thomas Rölleke. A probabilistic relational algebra for the integration of information retrieval and database systems. *ACM Trans. Inf. Syst.*, 15(1):32–66, 1997.

[4] Djoerd Hiemstra, Henning Rode, Roel van Os, and Jan Flokstra. PF/Tijah: text search in an XML database system. In *Second International Workshop on Open Source Information Retrieval*, Seattle, USA, 2006.

[5] Apriorie LTD. Apriorie website. `http://www.apriorie.co.uk/`.

[6] Florina Piroi, Giovanna Roda, and Veronika Zenz. CLEF-IP 2009 Evaluation Summary, 2009.

[7] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In *Third Text REtrieval Conference (TREC 1994)*, 1994.

[8] Giovanna Roda, John Tait, Florina Piroi, and Veronika Zenz. CLEF-IP 2009: retrieval experiments in the Intellectual Property domain. In *CLEF working notes 2009*, Corfu, Greece, 2009.

[9] W3C. Resource description framework. `http://www.w3.org/RDF/`.

[10] W3C. XQuery 1.0: An XML Query Language. `http://www.w3.org/TR/xquery/`.