

Integration of Semantic, Metadata and Image search engines with a text search engine for patent retrieval

Joan Codina¹, Emanuele Pianta², Stefanos Vrochidis³, Symeon Papadopoulos³

¹ Fundació Barcelona Media, Ocata 1, 08003 Barcelona Spain

² Fondazione Bruno Kessler, via Sommarive 18 38100 Trento, Italy

³ Aristotle University of Thessaloniki, Thessaloniki, Greece

Joan.codina@barcelonamedia.org, pianta@fbk.eu, {stefanos, [papadop](mailto:papadop@iti.gr)}@iti.gr

Abstract. The combination of different search techniques can improve the results given by each one. In the ongoing R&D project PATExpert¹, four different search techniques are combined to perform a patent search. These techniques are: metadata search, keyword-based search, semantic search and image search. In this paper we propose a general architecture based on web services where each tool works in its own domain and provides a set of basic functionalities to perform the retrieval. To be able to combine the results from the four search engines, these must be fuzzy (using a membership function or similarity grade). We focus on how the fuzzy results can be obtained from each technique, and how they can then be combined. This combination must take into account the query, the similarity of the patent to each part of the query, and the confidence on the technique

Keywords: Patent search, semantic search, image search, multimodal, user feedback, similarity search, fuzzy.

1 Introduction

In the field of information retrieval there is an increasing interest in patent retrieval. The legal style of patent documents, where text is obfuscated deliberately and very specific vocabulary is combined with very generic terms, makes patent retrieval a challenging task. Because of the legal implications of a patent invalidity search, it is crucial to get a high recall rate even at the expenses of losing precision. Expert users perform long Boolean queries (having from 5 to 30 statements) where each concept they are searching for is expressed by AND's and OR's of possible synonyms [1].

The use of semantic search allows searching for concepts, instead of words, and for relationships between them. However, semantic search has still to face a number of challenges in order to become the backbone of a search engine. First, it needs an ontology that copes with all the relevant terms. Although several ontologies exist, they do not cover most of the very specific terms found in patents, and the generic terms provide only little information. As illustration, consider the following sentence

¹ PATExpert is partially funded by the European Commission in its Sixth Framework Programme (FP6 028116).

from a patent claim of a compact disc reader: “An optical head device for use in combination with an optical source for generating an optical beam along an optical axis from a source point and comprising a lens system”. Here, words like “head”, “device” or “source” are combined with more specific ones like “axis” or “lens”. Additionally, many of these words are combined in multiwords such as “optical head device”, “optical axis” or “source point” which may not exist in the ontology.

Another problem arises when disambiguating terms since the most common choices may not apply to patent documents, hence broad-coverage parsers like Minipar [2] may take the wrong decisions. As an example, consider the word “means”, which can be either a verb or a noun. In natural language the most common choice would be to consider it a verb. However, this may not be true in patent documents of a given domain, where “means” is often “a noun denoting an instrumentality for accomplishing some end” as in “a transfer means mounted on the frame and operatively associated with the tool means for moving the tool means...”.

There is also a complexity problem. A patent can contain thousands of triples, each one composed by a noun, a verb and an object. Triples can be related between them when the same object appears in two triples. For example, the pair “we live in a house”, “the house is white” can be equivalent to “we live in a white house” if we know that the house of the first and second triples are the same. Ideally a patent could be represented by a single graph made of thousands of related triples.

In practice, however, all triples and relationships cannot always be determined and one gets a set of unconnected sub-graphs which may fall short to make use of the proper content representation.

Most patents are impossible to understand without the help of drawings. Images are a source of valuable information during search, but can also be a source of confusion since the same object can be drawn in so many different ways. Image search based on image content (and not captions or surrounding text) is still an open research problem, and though results are encouraging they are not reliable enough.

In short, semantic and image search techniques are promising but not yet mature enough to rely exclusively on them. On the other hand, expert patent users feel confident with traditional (but often too short-sighted) text search techniques. A multimodal patent search system may help to circumvent the weakness of the individual techniques. This multimodality characteristic is one of the prominent features in the PATExpert [3] retrieval module.

PATExpert is a European project devoted to the use of linguistic and image analysis tools for patent processing. This includes patent search, but also paraphrasing, summarization, classification, valuing and multilingual search. PATExpert advocates the specification of patent material in terms of techniques that operate on semantic representations rather than on textual ones.

This paper focuses on the search and retrieval module of PATExpert where a multimodal search engine is built from four individual search engines: (1) a metadata information search engine, (2) a keyword-based retrieval engine, (3) a semantic search engine, and (4) an image search engine. The first two allow for keyword-based text search and for metadata search. They are mainly based on classical information retrieval techniques. The third one, namely the semantic search engine, allows for the search of patent documents according to content criteria (e.g., material of which an object is made, availability of a component with a specific functionality, purpose of a

component, etc.). Finally, the image search engine allows for the search of patent material with images similar to images or features provided by the user. The objective of the multimodal search is to improve the performance from the classical retrieval techniques with the inclusion of the results from the advanced search methodologies.

The remainder of the paper is organized as follows. Section 2 first presents the architecture of the multimodal search system and then describes how the individual search modules can be integrated. Section 3 discusses how the results are processed and combined with each other. Finally, conclusions and future directions are given in Section 4.

2 Multimodal Search Engine

As shown in Fig. 1, the multimodal search engine is built upon four independent search engines covering different user needs related to patent search: (1) metadata search, (2) keyword-based search, (3) semantic-criteria search, and (4) image-related search.

Apart from the search engines, the system facilitates a management tool for queries and retrieved patent objects (results); here referred to as *merger*. The merger splits the user query into sub-queries and distributes them to the different search engines. The search engines are independent and use very different approaches to find results and determine scores. Nonetheless, all of the search engines match and retrieve patent objects on the basis of the similarity of their query representation (i.e., similarity-based retrieval). These results are then properly combined by the merger and the final ranked results presented to the user. At this stage, the original query of the user is iteratively refined and adjusted, based on the feedback provided by the user.

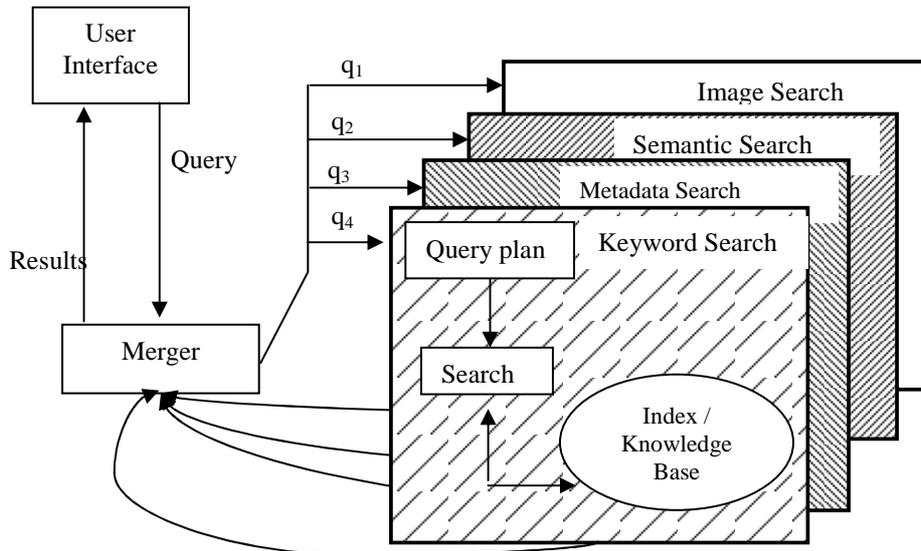


Fig. 1. Multimodal Search Engine

In the following paragraphs, we briefly describe the different search engines, and leave the discussion of the merger module for the next section.

2.1 Metadata Search

Queries posed to the metadata retrieval system relate to the attribute values of the patents (e.g. the name of the inventor, the publication date or the IPC² classification code). The metadata query is intended to offer the possibility to perform a query focused on the contents of a database field.

The standard approach for searching metadata is to perform an exact query based on a Boolean search constraint specified by the user (e.g. “pubdate > 01/01/2000 AND inventor = Y”). The returned results are the set of documents which completely fulfill the constraints. Thus, the result is crisp in the sense that a document either satisfies the query or it does not. This is quite a limitation since it does not allow for partial matching. Moreover, there is no fuzziness or ranking as known from classic information retrieval.

Fuzziness can be introduced in the constraints as well as in the Boolean operators. Fuzzy comparators like >~, <~, ~, and !~ are included. As an example consider the query “pubdate >~01/01/2000”. This fuzzy operator will return all records where “pubdate > 01/01/2000-FuzzyMargin”. The ones after 01/01/2000 will have a membership grade (ranking) of 1.0, while the documents within the *FuzzyMargin* range are assigned a decreasing membership. The size of the fuzzy margin is user defined.

Fuzziness has also been introduced in the Boolean operators. This means that the user may choose to perform an ORF or ANDF instead of a regular OR/AND. The difference is that the fuzzy operators will give a greater membership if both conditions are true than if only one is true. The Boolean operators for OR/AND over fuzzy values will become the maximum/minimum of the membership grades. The fuzzy operators are the product T-norm (AND) and probabilistic sum for the S-norm (OR).

The drawback of having fuzzy operators is that the FAND becomes an OR when translated to the corresponding SQL query, and then it needs to compute the membership grade for each result.

In the next sample, we show how a fuzzy query is transformed to get a list of patents with the membership:

The Original Query:

```
(appcountry in ('ES', 'FR')) ORF pubdate >~1/2/2002
```

will generate an sql statement in two steps; in the first step the similarity for each condition present in the query is computed, while in the second, the global similarity applying the fuzzy formulas is computed.

```
SELECT id, sim1+sim2-sim1*sim2
FROM
```

² IPC (International Patent Classification) is a hierarchical classification system providing a common classification for patents.

```
(
  SELECT DISTINCT Patent_id ,
  CASE
    WHEN patents.pubdate>'1/1/2005' THEN 1.0
    WHEN patents.pubdate<'1/1/2004' THEN 0.0
    ELSE (patents.pubdate-'1/1/2004')/365.0
  END as sim1 ,
  CASE
    WHEN appcountry in ('ES', 'FR') THEN 1.0
    ELSE 0.0
  END as sim2
  FROM patents
  WHERE (patents.pubdate>'1/1/2004')
    OR appcountry IN ('ES', 'FR')
)
```

2.2 Keyword-based Search

The majority of the search engines available for patent retrieval are keyword-based. Some include a query pre-processing procedure allowing for the use of wildcards, weighting of query terms, query expansion by using thesaurus relations, proximity search, etc. The vector model is one of the most widely used search techniques as it gives very good results with a rather simple model.

In PATExpert we use Lucene [4], with some adaptations to deal with certain idiosyncratic aspects of patents (such as recognition of patent numbers or IPC codes).

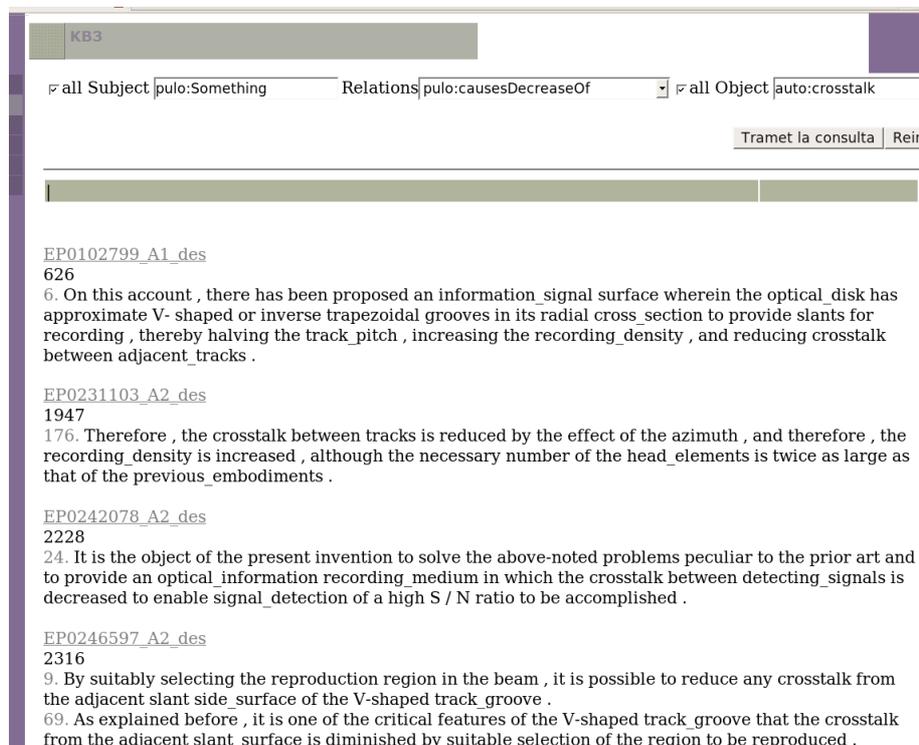
2.3 Semantic Search

State of the art patent processing makes use of the semantic-web formalism based on text labels to extract semantic information. In PATExpert patent documents are first processed with general purpose language processing tools, such as TextPro [5], and MiniPar [2], which carry out PoS tagging, multiword recognition, lemmatization, and dependency parsing. Linguistic annotation are then exploited to recognize frame instances (see FrameNet [6]), and finally concepts and triples.

An ontological framework is needed to work with concepts. In PATExpert³, the Core Upper Level Ontology (SUMO) with mappings to Wordnet has been employed and several ontologies have been developed: a Patent Upper Level Ontology (PULO), and domain ontologies with concepts of the specific technical fields. As patents are documents where new concepts are forged, PATExpert has the ability to automatically expand existing ontologies with new concepts (marked as *auto*) [7].

In triple-based semantic search the user specifies a target triple by selecting a relation and two concepts filling the subject and object role of the relation. The relation is chosen from a limited list (few tens) of significant relations recognized by the system (e.g. sumo:hasPart, pulo:moves, pulo:hasSubstance). Subject and object are selected from a much larger list (over 30.000) of domain specific concepts. A wizard helps the user to select the KB-concepts matching the concepts he/she has in mind.

In its basic functionality, the search engine will select all sentences in the corpus



³ A detail description of content extraction and developed ontologies in PATExpert can be found in [3].

containing the target triple, whatever the linguistic form in which the triple is expressed (e.g. "An optical head has a prism" or "the prism included in the optical head" or the "prism of the optical head"). However, the user can also choose to expand the search by instructing the system to consider also concepts related to the object or subject of the target relation. For instance instead of searching only for triples having as object the "prism" concept, the user can search also for all kind of more specific "prisms" known by the system according to the domain ontology (hyponyms), e.g. "trapezoidal_prism", "anamorphic_prism", etc. Alternatively, the user can search for concepts generalizing the concept of "prism", like "optical_component" (hypernyms).

If the user chooses one expanded query, the retrieved sentences can be ordered according to their similarity to the base (non-expanded) target triple. The semantic distance between the target and the retrieved triples is measured according to the distance of the retrieved concepts (hypernyms and/or hypopnyms) from the target concepts according to the domain ontology (e.g. "trapezoidal_prism" is closer to "optical_component" than to "engineering_component"). Assuming that the similarity of two equal triples is 1, we multiply this value by a factor $a < 1$ for each step down the hyponyms hierarchy, and by a factor $b < a < 1$ for each step up in the hypernyms chain. In this way we obtain a set of patents having a given concept or triple with a similarity value b .

The result of a sample search using semantics is shown in Fig. 2.

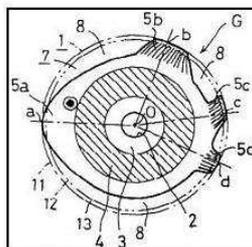
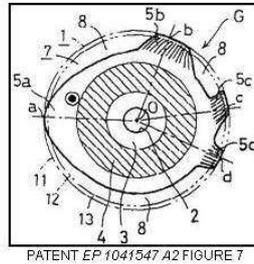
2.4 Image Search

Apart from the textual information, patent documents usually include a number of figures which are descriptive of the overall content. The most important characteristic of these figures is that they are black and white binary images as they mainly represent technical drawings, charts and mathematical relationships. Under these circumstances, image search could be rather challenging as color information, which is one of the main characteristics that content based retrieval relies on, cannot be considered.

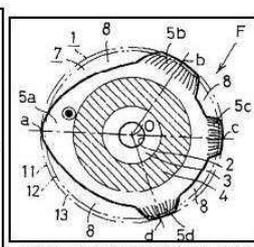
Taking into account the information above, the image similarity search module for patents was implemented in order to allow efficient image retrieval based on visual similarity. This module requires an off line pre-processing step in order to run on-line and provide the desirable results. The whole procedure is described below.

The first step in the off line processing is to detect and extract the pages of the patent that include figures as raster images. Secondly, orientation detection takes place. Connected components regions are extracted from the page (by use of the 8-neighborhood property) and the direction is identified along which the higher number of regions that lie on the same line [8]. Subsequently, individual figures need to be separated as normally such a page may contain more than one figure. The figure separation can be done automatically with an acceptable error⁴ while it can be also

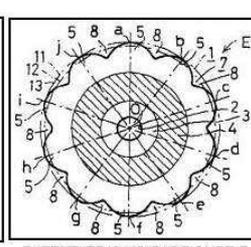
⁴ The reason for accepting error at this stage has to do with the fact that the figures are placed randomly in the page, some times really close to each other and the labels can be handwritten. In such cases the borders between different figures are very hard to specify.



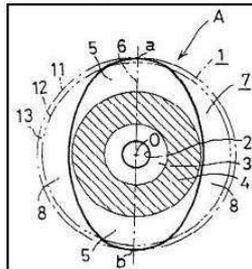
PATENT EP1041547A2 FIGURE 7
[Full Image](#) [Show similar](#)



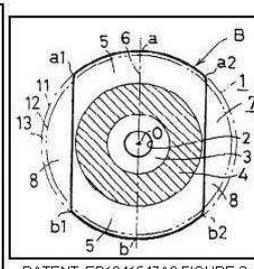
PATENT EP1041547A2 FIGURE 6
[Full Image](#) [Show similar](#)



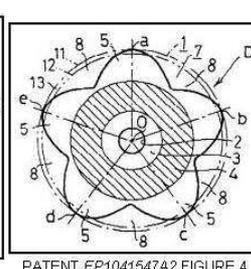
PATENT EP1041547A2 FIGURE 5
[Full Image](#) [Show similar](#)



PATENT EP1041547A2 FIGURE 1
[Full Image](#) [Show similar](#)



PATENT EP1041547A2 FIGURE 2
[Full Image](#) [Show similar](#)



PATENT EP1041547A2 FIGURE 4
[Full Image](#) [Show similar](#)

Fig. 2. Examples of retrieved results by image search

manually supported to improve the results. Finally, the extracted images are stored and indexed in a database.

At this stage, the feature extraction takes place. The employed feature extraction method relies on the computation of the Adaptive Hierarchical Geometric Centroids proposed in [9]. The reason for selecting these features was the fact that the majority of the figures are binary so the only useful information could be extracted from the geometry and the shape of the depicted objects.

Assuming that the origin of a 2-d space lies in the first level geometric centroid, we split the image plane into four disjoint quadrants, compute for each one its geometric centroid, and divide it into 4 sub-quadrants in an analogous way. This is recursively performed up to some number of levels n . Note that after n levels, there are 4^n disjoint

unequal rectangle areas, i.e., 4^n possible partitions that can be classified in pattern groups. As the feature vector of a binary patent image we use the n histograms -one for each level- of the partitions. Consequently, the resulting vector dimension is low in comparison to most standard techniques whose feature vector dimension may reach tens of thousands. Based on this method the feature vectors are extracted and stored in a database in order to be online accessible.

During the online search, we compute the L1 distances of the feature vector of the given image query against every other feature vector from the database. The smaller the distance is between the feature vectors of two images, the more common visual characteristics they share. One specific distance threshold is set in order to distinguish relevant from irrelevant images. High threshold values could result in many results at low precision levels, while lower ones could result in very small or even empty sets of images. For this reason, the threshold was empirically tuned in order to optimize the performance.

A use case for the image retrieval module is presented in Fig. 3. In this, the user selected a figure with cyclic characteristics. The results depicted in Fig. 3 provide an indication of the high relevance achieved by the module.

In order to evaluate the derived results, recall and precision metrics were calculated. The experiments were conducted in a database of 1400 images extracted by European Patents and a 100 of them were arbitrarily chosen to perform the image similarity for each one. By tuning the distance threshold that compromises between these complementary metrics leads to 77% Recall for 49% Precision.

3 Merger

PATExpert does not contain a single language for doing all four kinds of queries, when the user specifies a query, she uses different syntax depending on the search engine that she is writing the query for. Then the user can combine with Boolean or Fuzzy-Boolean operators some queries written for the different query engines, to build a search.

The merger is responsible for the distribution of sub-queries and combines the results back together to produce a single list of results. The query dispatching and collection of results does not have any special feature; the challenge remains on how to combine them. This is done within a fuzzy framework.

PATExpert also provides a similarity search. The similarity search could be the common interface for querying: The user introduces a text she is looking for, and the system returns a list of patents that are similar. This simple approach is not as simple as it seems, first because the task is intrinsically difficult, and from the point of view of the user (and even more: the expert users) there is no control on what the system is doing, or how to control the search process to be sure that the patents retrieved are the correct set. The expert user needs to be able to monitor the process to be sure that the list of patents contains all the patents that could lead to an infringement or invalidation of the patent. For this reason PATExpert provides this functionality in two steps: During the first step the system receives a text of a patent (or portions of it) and produces as output a query, that when executed would provide patents similar to

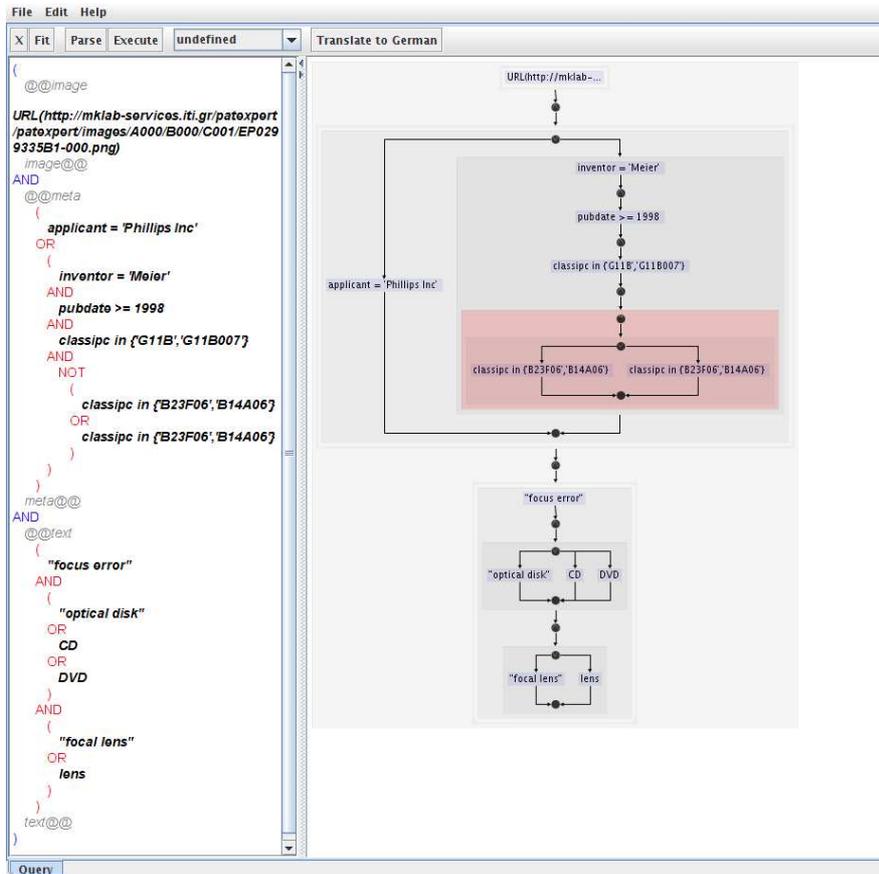


Fig. 3. Sample query as introduced in the user interface: The user specifies a combination of different searches to be performed by the different search engines.

the one provided by the user. The way that the query is generated from the text is out of the scope of this paper.

In the IR literature, the paradigm of a *broker* exists that distributes a query to different search engines, sends the same query to one or all of them, and then merges the result. Usually a broker is associated to distributed systems and the task of the broker is to send the query to the appropriate node that may have the data to answer the question. In PATExpert the role of the merger is different: First it does not send the same query to all the search engines, as each portion of a query is only solved by one search module and secondly when the merger gets the results back, it has to merge them, taking into account the Fuzzy-Boolean operators that combined the original sub-queries. For this reason the “merger” is not called “broker” in PATExpert.

3.1 An overview

We illustrate the merger function through an example. Let us consider the user query as written in the interface, shown in Fig. 3

This query is composed of several sub-queries: the first one is an image query the second one a metadata query and the third one a keyword-based. Each sub-query is sent to the corresponding search engine that will return an ordered list of (patent_id, similarity). The similarity of patent p in query q is denoted as:

$$\text{Sim}(q, p) = s_{q,p} \in [0,1], \quad (1)$$

and represents the extent to which the patent p fulfills query q . The larger the grade is, the better the match. In particular, a grade of 1 represents a perfect match while, on the other hand, a grade of 0 implies no match at all. The merger must take these lists and merge them to produce a single list, where for each patent_id the overall similarity grade is computed from the sub-queries similarities. The strategy used to merge the list results takes into account the operator, the similarity and also the degree of confidence on each search technique.

3.2 Combining the results

The combination of results is done within a fuzzy framework in order to overcome the shortcomings of the strict Boolean model (see Section 2.1). This fuzzy framework encompasses most common fuzzy approaches such as fuzzy set models [10] or the extended Boolean models with weights associated to the index terms [11].

Within the classical fuzzy set model, we could define the overall similarity of a patent p under the disjunctive and conjunctive queries as

$$\text{Sim}(q_{\text{or}}, p) = \text{Sim}(q_1 \vee q_2 \vee \dots \vee q_m, p) = \max(s_{q_1,p}, s_{q_2,p}, \dots, s_{q_m,p}) \quad (2)$$

$$\text{Sim}(q_{\text{and}}, p) = \text{Sim}(q_1 \wedge q_2 \wedge \dots \wedge q_m, p) = \min(s_{q_1,p}, s_{q_2,p}, \dots, s_{q_m,p}) \quad (3)$$

However, the use of maximum (or minimum) does not reflect any change in the similarity when values different from the maximum (or minimum) change without becoming bigger (or smaller) than the maximum (or minimum). As an illustration, consider the similarities in Table 1.

Table1: sample similarities to illustrate the differences between operators

Similarity $s_{q,p}$	q_1	q_2
p_1	0.8	0.7
p_2	0.81	0.3

Using the OR boolean operator we get

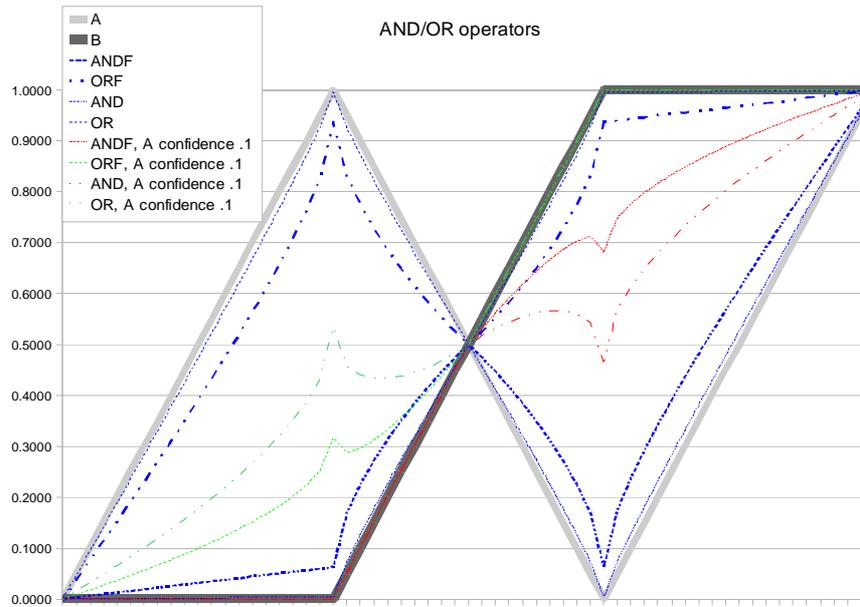


Fig. 4. Graphical representation of the weighted power-mean averaging operators operating as Boolean or Fuzzy and with different confidence levels of the fuzzy variable A. The graph shows for each operator the fuzzy result of the operators depending on the Fuzzy weight of A and B. The operators are the Boolean (AND OR) and fuzzy (ANDF, ORF) . When A has lower confidence level, the result is closer to B.

$$q_{or} = q_1 \vee q_2; \tag{4}$$

$$S_{q_{or},1} = \max(0.8,0.7)= 0.8; \quad S_{q_{or},2} = \max(0.81,0.3)= 0.81$$

which favors the second one even if one its queries has a very low score, and also despite the fact that both queries scores in the first one are high. To prevent this kind of behaviors, one may use alternatively T-norms (triangular norms) to compute the disjunctive with the corresponding T-conorms for the conjunctive operator [12], [13] but these operators include a non intuitive behavior: $AND(x,x) < x$ for $x < 1$. So the use of T-norms was discarded.

In addition to combining the partial similarities, the merger needs also to deal with a belief or confidence factor associated to each result set. For this purpose, a confidence factor (obtained during the training phase) is assigned to each of the search techniques. This process, however, is out of the scope of the paper.

To deal with Boolean/fuzzy operators in a similar way together with the confidence factor, and after testing the result of different fuzzy paradigms, we adopted the weighted power-mean averaging operators [14]. These operators include some nice properties from the user point of view that can be seen in Fig. 4: When both variables have the same value the result coincides with them and also $AND(0, x) \geq 0$ for $x \in [0,1]$, which means that a change on x , influences the result. The formula for the *AND* operator is

$$\begin{aligned}
 q_{and} &= q_1 \wedge q_2 \wedge \dots \wedge q_m; \\
 q_1 &< q_2 < \dots < q_m \\
 S_{q_{and}} &= \left[\frac{1}{m^2} \sum_{k=1}^m (u_k \times e_k^r) \right]^{\frac{1}{r}} \\
 u_k &= (m-k)^2 - (m-k-1)^2
 \end{aligned} \tag{5}$$

Where the m terms to compute the *AND* operator, are sorted in decreasing order, the weights are then assigned accordingly to that order before being averaged. The different values in the power r will make the operator Boolean (value of 0.0001) or fuzzy (when 0.5).

The query may be seen as a tree of *AND*'s/*OR*'s which can be evaluated bottom-up. Each Boolean operator must return a list of documents with the similarity for each document and a belief factor for the full list. This belief is computed as a weighted sum of the beliefs, where the weight is the own belief, giving the formula:

$$w_{and/or} = \frac{\sum w^2}{\sum w} \tag{6}$$

The weighted operator is computed using the following formula:

$$\begin{aligned}
 q_{and} &= w_1 q_1 \wedge w_2 q_2 \wedge \dots \wedge w_m q_m; \\
 \text{where } q_1 &< q_2 < \dots < q_m \\
 \text{and } \sum_{k=1}^m w_k &= 1 \\
 S_{q_{and}} &= \left[\frac{1}{m^2} \sum_{k=1}^m (u_k \times e_k^r) \right]^{\frac{1}{r}} \\
 u_k &= \left(m \sum_{l=k}^m w_l \right)^2 - \left(m \sum_{l=k+1}^m w_l \right)^2
 \end{aligned} \tag{7}$$

The main disadvantage of this approach is that the membership grades (q_i) need to be sorted before performing the computations, and then the weights need to be computed accordingly.

5 Conclusions and Future Work

Combining the different search methods improves over the single-modal search. Moreover, by allowing different search modalities, the users are no longer confined by the text-only interface. They could freely pick a modality to best represent the query element or mix and match several modalities to construct a complex query. The crucial part is the combination of the results obtained by the different search modalities. This combination must take into account the query, the similarity of the patent to each part of the query, and the confidence on the technique.

PATExpert is reaching a stage where expert users can experiment with it. We need them to use the system and give us their feeling about the quality of the results to tune each of the search engines. After this step we plan the use of machine learning tools to automatically adjust the weights based on user feedback.

There is a lack of information share between the different search engines. In order to optimize the search process; the search engines should be connected to a global optimizer that could help provide them information to reduce the search space.

The extension of PATExpert to other domains is highly dependent on the ontologies, and makes it difficult to be used as is with standard repositories like TREC or the ones containing patents, as we need ontologies for the specific technical domains, this means that there are no reference queries, and the training has to be done by our users.

References

- [1] Homan H.S.: Making the Case for Patent Searchers. Searcher, vol. 12, March 2004
- [2] Lin D.: Principle-based parsing without overgeneration. Proceedings of the 31st conference on Association for Computational Linguistics, pp. 112-120, 1993.
- [3] PATExpert home page, <http://www.patexpert.org>
- [4] LUCENE, <http://lucene.apache.org/java/docs>
- [5] Pianta, E., Girardi, C. and Zanoli, R.. "The TextPro tool suite", Proc. of LREC 2008, Marrakech, Morocco, May 2008.
- [6] Potrich, A, and Pianta, E., "Learning Domain Specific Isa-Relations from the Web", Proc. of LREC 2008, Marrakech, Morocco, May 2008.
- [7] FrameNet site: <http://framenet.icsi.berkeley.edu/>
- [8] Hones F., Lichter J.: Layout extraction of mixed mode documents. Machine Vision and Applications, Springer-Verlag 1994
- [9] Yang M., Qiu G., Huang Y., Elliman, D.: Near-Duplicate Image Recognition and Content-based Image Retrieval using Adaptive Hierarchical Geometric Centroids. Proceedings of the 18th International Conference on Pattern Recognition, 2006
- [10] Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, 1999

- [11] Salton, G., Fox, E., Wu, H.: Extended Boolean Information Retrieval, CACM 26(11): pp. 1022--1036, 1983
- [12] Hájek P.: Mathematics of fuzzy logic. Kluwer, 1998.
- [13] Lee, J. H. et al.: On the evaluation of Boolean operators in the extended Boolean retrieval framework. Proceedings of the 16th annual international ACM SIG
- [14] W.S. Hong et al., "A new approach for fuzzy information retrieval based on weighted power-mean averaging operators," *Computers and Mathematics with Applications*, vol. 53, 2007, pp. 1800-1819.