Hybrid Search: Effectively Combining Keywords and Semantic Searches

Ravish Bhagdev[1], Sam Chapman[1], Fabio Ciravegna[1], Vitaveska Lanfranchi[1] and Daniela Petrelli[2]

[1] Department of Computer Science
[2] Department of Information Studies
University of Sheffield, Regent Court, 211 Portobello Street,
S1 4DP Sheffield, United Kingdom
{*N.Surname*}@shef.ac.uk

Abstract. This paper describes hybrid search, a search method supporting both document and knowledge retrieval via the flexible combination of ontology-based search and keyword-based matching. Hybrid search smoothly copes with lack of semantic coverage of document content, which is one of the main limitations of current semantic search methods. In this paper we define hybrid search formally, discuss its compatibility with the current semantic trends and present a reference implementation: K-Search. We then show how the method outperforms both keyword-based search and pure semantic search in terms of precision and recall in a set of experiments performed on a collection of about 18.000 technical documents. Experiments carried out with professional users show that users understand the paradigm and consider it very powerful and reliable. K-Search has been ported to two applications released at Rolls-Royce plc for searching technical documentation about jet engines.

Keywords: Semantic search, Semantic Web in use.

1 Introduction

The Semantic Web (SW) is a creative mix of metadata designed according to multiple ontologies and unstructured documents (e.g. classic Web documents). The assumption that the SW is not a Web of documents, but a Web of relations between resources denoting real world objects [4] is too restrictive of the true nature of the SW. There are a number of applications and situations where coexistence of documents and metadata is actually required. One example is the legal scenario, where access to documents is the main focus and the available metadata is the means to reach a specific set of documents [5]. However it may well happen that the available metadata does not cover parts of the document that are of interest to some users because: (i) the ontology used for annotation has a different focus and does not model that part of the content or (ii) annotations can be incomplete, whether user or system provided. A human annotator may miss some or provide spurious ones; in the same way automated means such as Information Extraction from texts (IE) may be unable to reliably extract the information required. This is because IE is a technology that

S. Bechhofer et al.(Eds.): ESWC 2008, LNCS 5021, pp. 554–568, 2008.

performs very well on simple tasks (such as named entity recognition), but poorly on more complex tasks such as event capture [8]. Therefore, some metadata modelled by an ontology may be impossible to capture with IE thus preventing any future operation (e.g. retrieval) via that metadata.

In this paper, we focus on searching the SW as a collection of both documents and metadata, with the aim of accommodating different user tasks: document retrieval and/or knowledge retrieval. A document retrieval task implies searching for documents using concepts or keywords of interest; a knowledge retrieval task concerns retrieving facts from a knowledge base (i.e. triples). Differently from previous literature [1, 2, 3, 4, 9], we consider the issue of working in a complex environment where metadata only partially covers the user information needs. We therefore propose to use a strategy (called Hybrid Search, (HS) where a mix of keyword-based and metadata-based strategies are used. We formally define the approach and describe how to organise a HS architecture. We then describe K-Search, a reference implementation of HS. In implementing an approach, a number of decisions are made: methodological (e.g. we selected a form-based approach [1]), and technical (e.g. on the expressivity of covered language and architecture design). We discuss how these choices impact the HS mechanism. Then we present two experiments performed using a K-Search application:

- *in vitro:* K-search was applied to a large corpus of legacy documents; an evaluation of the resulting application shows HS outperforming both keyword based searching and semantic searching;
- *in vivo:* the application was evaluated with real users; the results show that users appreciate the full power of the HS concept.

Finally we compare our work to the state of the art, we discuss how it is possible to extend the currently available semantic search paradigms to cope with HS, draw conclusions and highlight future work.

2 Hybrid Search

HS combines the flexibility of keyword-based retrieval (as in traditional search engines) with the ability to query and reason on metadata typical of semantic search systems. Metadata is information associated to a document describing both its context (e.g. author, title, etc.) and its content (as provided by RDF triples annotating portions of the documents with respect to an ontology, e.g. <*"installed_part" upon "engine_type"*>). In concrete terms, HS is defined as:

- the application of semantic (metadata-based) search for the parts of the user queries where metadata is available;
- the application of keyword-based search for the parts not covered by metadata.

Three types of queries are possible with HS: (i) pure semantic search via unique identification of concepts/relations/instances (e.g. via *URIs* or unique identifiers); (ii) keyword-based search on the whole document and (iii) keyword-in-context search. Keyword-in-context searches the keywords only within the portion of the document annotated with a specific concept or relation; for example in the aerospace domain, it

enables searching for the string "fuel" but only in the context of all the text portions annotated with the concept *affected-engine-part*. The keyword-in-context mechanism was the core of the mechanism proposed in [14].

It is important to stress that differently from other approaches (e.g. [9]), in HS conditions on metadata and keywords coexist. For example consider an application in the aerospace diagnostic domain where metadata is associated to documents for events described (e.g. discoloration) and the affected component (e.g. a high pressure blade) but not for the part of the component affected (e.g. the trailing edge). An example of hybrid query could be:

```
∀z /(discoloration y) & (component x) & (located-on y x) &
     (provenance-text-contains x "blade") & (document z) &&
     (provenance y z) & (contains z "trailing edge")
```

This can be read as: retrieve all documents *(document z)* that contain the string "trailing edge" *(contains z "trailing edge")* with associated metadata *(provenance y z)* involving:

- an instance of discoloration – *(discoloration y)*
- an instance of component where the provenance text contains the word "blade" *(component x) & (provenance-text-contains x "blade")*
- the component is affected by the discoloration *(located-on y x)*

To our knowledge, no other approaches allows such flexibility in qiuerying, as most of them just allow queries based on metadata [1, 2, 3, 4, 9].

2.1 Hybrid Search for Document Retrieval

The most commonly used method for document retrieval is keyword-based search (KS). KS effectiveness is often affected by two main issues, ambiguity and synonymity. Ambiguity arises in traditional keyword search systems because keywords can be polysemous, i.e. they can have multiple meanings. A search containing ambiguous terms will return spurious documents (low precision). Synonymity is found when an object can be identified by multiple equivalent terms. When searching documents using just one of the terms, the documents containing other synonym are not retrieved (low recall). Semantic search as metadata-based search defined according to an ontology, enables overcoming both issues because annotations are unambiguous and do not suffer from synonymity.

Nonetheless when pure Semantic Search is applied to a document retrieval task, it can fail to encompass the user information needs (either because of limitations in the ontology or because the metadata is unavailable for a specific document), as it would restrict the types of queries users can perform (low recall).

HS combines the disambiguation capabilities of semantic search (when metadata is available) with the generality and extensibility of keyword-based search (for the other cases). The expected result is that:

- precision and recall are increased with respect to the standard keyword-based search because ambiguity and synonymity are dealt with by semantic search when available;

- the use of keywords where metadata is missing enables to answer otherwise impossible queries (increased recall with respect to semantic search). As keywords are combined with metadata in the same query, the context given by the available metadata helps in disambiguating keywords as well (higher precision than keyword-based search).

2.2 Hybrid Search for Knowledge Retrieval

In addition to document retrieval, HS can provide highly effective knowledge retrieval by using keywords as "context" of the metadata, hence enabling to further focus the results in a way that is impossible with semantic search.
In the aerospace example, it will be possible to retrieve

```
∀y,x /(discoloration y) & (component x) & (contains-
     keyword x "blade") & (located-on y x) & (document
     z) && (provenance y z) & (contain-keyword z
     "trailing edge")
```

Searching on metadata only allows a query like

```
∀y,x /(discoloration y) & (component x) & (located-on y x)
```

which would return a large amount of spurious results (low precision). The results of the hybrid query would still be sub-optimal (i.e. not equivalent to a semantic search where all the metadata is available) because the keyword search part will still suffer problems of synonymity and polysemy, but it would be far more high quality than the pure semantic search which will miss essential conditions. Also, it is expected that the matching of metadata will help reducing the issue of polysemy because it will work as context for the keywords.

3 Architecture for Hybrid Search

This section discusses a generic architecture for HS, while the next one presents an actual implementation.

At indexing time, documents are indexed using a standard keyword–based engine such as SolR[1]. Annotations (e.g. generated by an IE system) are stored in a Knowledge Base (e.g. a triple store like Sesame[2]) in the form of RDF triples. Provenance of facts must be recorded, for example in the form of triples connecting the facts' URIs and those of the document of origin, as well as the original strings used in the documents.

At retrieval time, HS performs the following steps:

- the query is parsed and the different components (keywords, keywords-in-context and metadata-based) identified;
- keyword matches are sent to the traditional information retrieval system;

[1] http://lucene.apache.org/solr/
[2] http://www.openrdf.org/

- metadata searches are translated into a query language like SPARQL[3] and sent to a triple store;
- keywords-in-context queries are matched with the provenance of annotations in documents (again using SPARQL and a triple store);
- finally, the results of the different queries are merged, ranked and displayed.

Merging of results. A direct matching between keyword and semantic results is not straightforward as their results are incompatible. Keyword matching returns a set of *URIs* of documents (`KSDocUriSet`) of size *n*.

$$KSDocUriSet \subset URIs, \text{ where } KSDocUriSet = \begin{vmatrix} uri1, \\ uri2, \\ \dots \\ uri_n \end{vmatrix}$$

while a semantic search performed on a knowledge base returns an unordered set *rSet* (size *m*) of individual assertions < *subj, rel, obj*>[4]

$$OSTripleSet = \text{all triples} \in R \text{ that satisfy the Ontology-based Query}$$

Using the provenance information associated to each triple, it is possible to compute the set of documents that contain the information retrieved from the RDF store.

$$OSDocUriSet = \text{Union of Provenance}(triple^i) \text{ for all } i \text{ where } triple^i \in OSTripleSet$$

In order to provide the answer for users interested in **document retrieval**, the list of *URIs* of documents generated using provenance information is now directly compatible with the output of keyword matching. The result of the query is given by the intersection of the two sets of document *URIs.*

$$HybridSearchUriSet = KSDocUriSet \cap OSDocUriSet$$

In order to provide answers to users interested in **knowledge retrieval**, a list of triples must be returned. In this case, the list of triples is filtered so to remove those whose provenance does not point to any of the documents returned by the keyword-based search engine. Formally:

$$HSTripleSet = \begin{vmatrix} \text{All triples} \in OSTripleSet \\ \text{Where Provenance}(triple^i) \in KSDocUriSet \end{vmatrix}$$

Ranking. Effective ranking (i.e. the ability to return relevant documents first) is extremely important for a positive user experience. The results returned by the different modalities provide material for orthogonal ranking methods:

[3] http://www.w3.org/TR/rdf-sparql-query/
[4] Both ontology-based and keyword in context queries are covered here.

- **keyword-based systems** like Lucene enable ranking of documents according to (1) their ability to match the keyword-based query; (2) the keywords used in anchor links (i.e. the text associated to hyperlinks pointing to a specific document) and (3) the document popularity measured as function of the weight of the links referring to the document itself;
- **semantic search** ranks according to the presence and quality of metadata.

Different ranking solutions can be adopted accordingly to the use case. The most natural one is to adopt the ranking provided by the keyword based search, as it is based on solidly proven methods, especially the use of anchor texts and hyperlinking However more sophisticated strategies can be designed, especially for organisational repositories where such interlinking is generally not present [14].

Presentation of results. Depending on the task (i.e. document retrieval Vs knowledge retrieval), results can be presented in different ways: as a list of ranked documents, as aggregated metadata (e.g. via graphs or charts) with associated provenance, etc.

4 K-Search: Putting Hybrid Search into Practice

K-Search is an implementation of the HS paradigm. In realising HS in a real world system, a number of choices need to be made in order to:

- create an interface that communicates to the user the optimal strategy to mix metadata and keywords for the task at hand, so to maximize effectiveness and efficiency of searches;
- decide what strategies to adopt for ranking, visualisation, annotation, etc.

We have chosen to model our search interface on a form data entry paradigm. The interface (Fig 1) works in a standard browser and enables the definition of complex hybrid queries in an intuitive way. Keywords can be inserted into a default

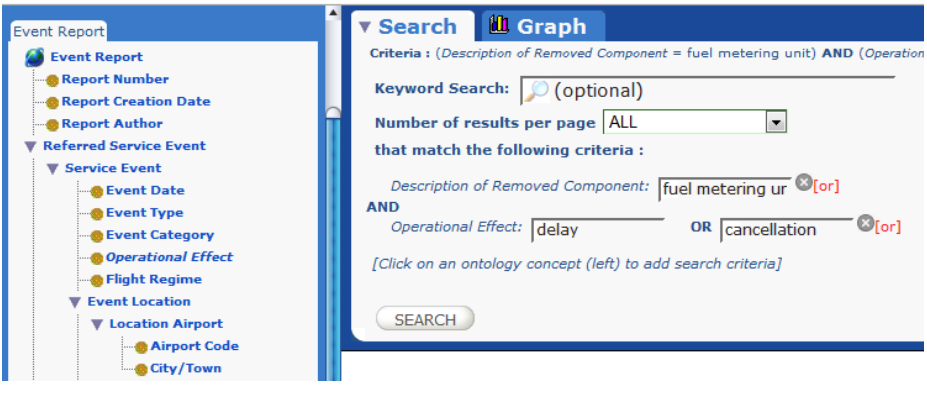

Fig. 1. Interface detail: the query form. Clicking a concept on the ontology creates a form item enabling inserting restrictions on metadata. Disjunctions are easily introduced by clicking [or].

form field in a way similar to that required by search engines; Boolean operators *OR* and *AND* can be used in their combination. Conditions on metadata can be added to the query by clicking on the ontology tree (left side of interface in Figure 2). This creates a form item to insert conditions on the specific concept. As multiple constraints can be added to the query, the **logical language** is restricted to provide a simple and intuitive interface: only common Boolean combinations are supported. This decision was supported by the observation that in carrying out their tasks, users adopted strategies that do not require the full logical language; furthermore research done in human-computer interaction shows that graphical representation of the whole Boolean logic is not understood by most users [10].

AND constructs are allowed among conditions checking different concepts in the ontology. So for example, *contains*(*removed-component, "fuel") AND contains*(*jet-engine-name, "engineA"*) is acceptable, but *contains*(*removed-component, "fuel") AND contains(removed-component, "meter")* is not. The latter is acceptable if formulated as *contains* (*removed-component, "fuel meter").* Conditions in AND are displayed on different lines in the interface (Figure. 3 shows an example of a combination of *removed-component AND operational-effect*). The expressivity restrictions are motivated by the results of our user studies, which showed which types of queries the users wanted to make.

OR constructs are acceptable only if between conditions on the same concept. So *contains*(*removed-component, "fuel") OR contains(removed-component, "meter")* is accepted, but *contains*(*removed-component, "fuel") OR contains*(*jet-engine-name, "engineA"*) is not. The latter must be split into two different queries. Again, these restrictions are motivated by results of our user studies.

Figure 1 shows how the query *retrieve all events where removal of a fuel meter unit caused delay or cancellation"* - logically translated in *(contains(removed-component "fuel meter unit"))* AND equal(*operational-effect (delay OR cancellation)*) - appears at the interface level: two concepts (*removed-component* and *operational-effect*) have been selected; removed-component has been specified with a single option (fuel meter unit) while *operational-effect* covers two alternatives (delay or cancellation).

4.1 Ranking and Presentation of Results

In K-Search the ranking of results is performed by relying on the keyword ranking, in this case based upon TF/IDF, because - as the matching on the metadata part of the query is strict (i.e. only the documents that match *all* the conditions are returned) - all the documents tend to be equivalent in semantic content. However, the visualisation interface enables the user to change the ranking by focusing on specific metadata values. For example, given the query in Figure 1, documents can be sorted according to e.g. the value of the removed part (this is done by clicking on the appropriate column header of the interface shown in Figure 2).

K-Search supports the tasks of document retrieval and knowledge retrieval also at the presentation level, by providing different views on the search results. The default use of K-Search is for document retrieval. Therefore when a query is fired, a set of ranked documents displayed as a list are presented (see mid-right panel of the interface in Figure 2). Each item in the list is identified by the title (or file name) of

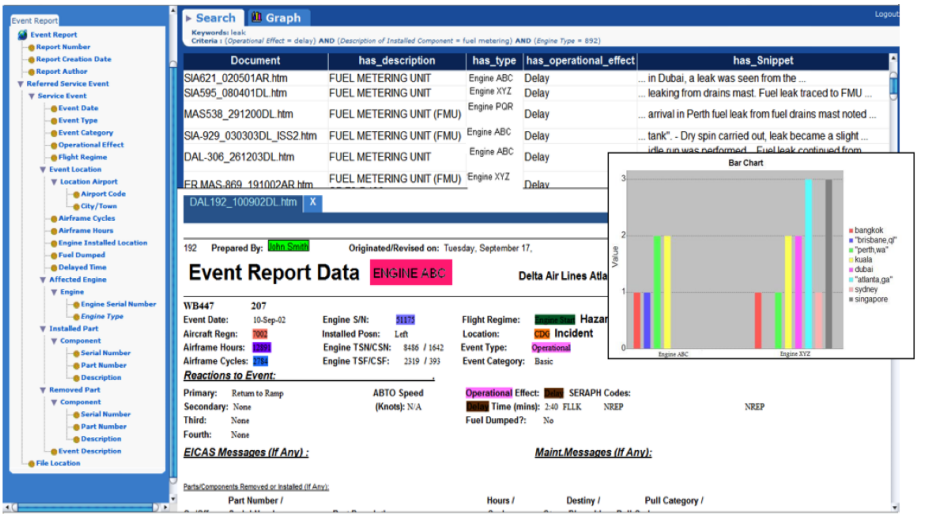

Fig. 2. The interface showing the list of documents returned (centre top), an annotated document and a graph produced from the results (image modified to protect confidential data)

the document and the values in the metadata that satisfy the semantic search. Clicking on one item in the list opens the corresponding document on the bottom right. The document is presented in its original layout with added annotations via colour highlighting; advanced features or services are associated to annotations [12, 13], including refining the current query. Multiple documents can be opened simultaneously in different tabs.

As for knowledge retrieval, K-Search provides two ways of inspecting the returned metadata. On the one hand the triples extracted are visible in the document list, because the values that satisfy the semantic query are listed for each document (middle panel). This enables an exhaustive and user-friendly inspection of the content of the triples and re-ranking of results according to these values. On the other hand, K-Search enables the creation of bi-dimensional graphs via selection of style (pie or bar chart) and variables to plot (e.g. engine Vs affected component). The graph in Figurre 2 plots the results of the previous query by location and engine type. Each graphic item (each bar in the example) is active and can be clicked to focus on the subset of documents that contains that specific occurrence. All retrieved triples can be exported in RDF or in CSV format for further statistical processing.

4.2 Indexing and Annotation

In order to make available document metadata and indexes, K-Search uses: (i) SolR for indexing documents and (ii) a generic semantic annotation plugin . Plugins currently exist for AktiveMedia (manual and semi-automatic annotation [6]) and some information extraction tools (T-Rex, an ontology-based IE tool [15] and Saxon, a rule-based extraction system[5]). Extracted information (ontology-based annotations)

[5] http://nlp.shef.ac.uk/wig/tools/saxon/

is stored in the form of RDF triples according to OWL or RDF ontologies into a triple store. K-Search provides plugins for Sesame and 3store; query languages supported are SPARQL and SeRQL.

5 Evaluation

Tests were carried out to evaluate the effectiveness and the user acceptance of the HS paradigm. The evaluation was performed using the K-Search Event Reports application (developed for Rolls-Royce plc) in two separate steps:

- *in vitro*: First of all the precision and recall of the IE system used in the specific case were evaluated; then 21 user-defined topics were translated into queries using three options: keyword-based searching, ontology-based searching and hybrid searching and the performances were recorded; these tests enabled us to evaluate the effectiveness of the method in principle;
- *in vivo:* 32 Rolls-Royce plc employees were involved in a usability test and commented on efficiency, effectiveness, and satisfaction; this evaluation enabled measuring the extent to which users understand the HS paradigm and feel that it returns appropriate results.

5.1 *In Vitro* Evaluation

The *in vitro* evaluation is composed by two parts, one to evaluate the effectiveness of the IE, the second to compare HS to keyword-based and semantic search.

IE evaluation
We analyzed a corpus of 18,097 reports on operational conditions of jet engines provided by Rolls-Royce plc. They are semi-structured Word documents containing tables and free text. As these documents are generated as part of the same management process, they all contain broadly the same relevant information but tables are user defined, so in principle each document can contain different types of table. However, some regularity occurs in tables across documents as users tend to re-use previously generated documents as template. The documents were converted into XML and HTML then indexed using SolR and metadata were generated using T-Rex. The ontology included concepts like the location where the event occurred, installed component(s), removed component(s), event details, what was the operational effect on the flight (delay, cancellation etc.), location, author, etc. The evaluation of the IE system was performed in order to understand which metadata were recognisable with an acceptable accuracy. Information in tables tends to be captured reliably by the IE system. This is because, although tables are irregular (e.g. sometimes the semantics is on the rows, sometimes on the columns, sometimes the information is spread over multiple cells, sometimes multiple information is compressed in one single cell), they roughly contain the same information and derive from evolution of common tables. T-Rex's learning curve assumed an asymptotic shape after learning from about 200 manually annotated documents. The combined evaluation results on all fields obtained in a two-cross folder test using 400 documents were Precision=98%,

Recall=99%, (harmonic) F-Measure=98%. Information in tables contained most of the metadata required in the ontology with the exception of the event cause.

As for the information contained in the free text (which was mainly describing the event cause), instead, accuracy was not at a level adequate to the user expectations (which was – according to our studies very close to 100% for recall and >90% for precision) therefore it was not made available to semantic search; it was however still available for searching via keywords.

Hybrid Search Comparative Evaluation

The goal of the comparative evaluation was to show that HS can provide better results than the pure keyword based or semantic search, by combining their reciprocal strengths. The evaluation was done considering a set of 21 topics generated on the basis of observed tasks, sequences of user queries recorded in the event corporate database or as elaboration of direct input from users (i.e. examples of their recent searches). Each topic represents a realistic information-seeking task of Rolls-Royce engineers, which typically could be previously answered only via repeated searches and extensive manual work. As it turned out, some topics, like *"How many events happened during maintenance in 2003*", can be answered using pure semantic search (because all the relevant metadata was captured by the IE system), others, like "*What events happened during maintenance in 2003 due to control units?*" can only be answered by combining annotations and keywords (in this case due to the lack of metadata about the cause of the event). Finally one topic could only be answered using keyword-based search, as no parts of it are covered by metadata.

During evaluation, topics were transformed into queries by manually translating the topics into semantic, hybrid and keyword-based queries. An example of hybrid query is ((flight-regime maintenance) AND (event-date year-2003)) + (keywords-contained "control unit" OR "control" OR "unit").

Precision and Recall were computed on the first 20 and 50 documents returned by each modality. We used standard Precision and Recall measures.

$$Precision = \frac{Correct\ System\ Answers}{System\ Answers} \qquad Recall = \frac{Correct\ System\ Answers}{Expected\ Answers}$$

Evaluation of results on such a large amount of documents is quite a difficult issue. The problem comes in computing recall's *Expected Answers* without manually matching all the 18,097 documents against all 21 topics. Therefore, we decided to approximate *Expected Answers* with the cardinality of the set of all the relevant documents returned by any of the three modalities. We believe that this measure is enough for the purpose of this evaluation because our goal is to demonstrate that HS outperforms the other two approaches in terms of precision and recall in returning relevant documents in the first 20 and 50 returned results; this means HS must show the ability:

(i) not to omit relevant documents identified by the other methodologies when intersecting the two sets (high recall)
(ii) to rank the relevant documents in the first 20 and 50 respectively (high precision)

Query	POS	Keyword 20			Ontology 20			Hybrid 20 General		
		COR	ACT	EXP	COR	ACT	EXP	COR	ACT	EXP
Q1	84	16	20	20	20	20	20	20	20	20
Q2	22	16	20	20	0	0	20	16	20	20
Q3	25	1	20	20	11	20	20	11	20	20
Q4	63	19	20	20	19	20	20	19	20	20
Q5	27	9	20	20	12	20	20	12	20	20
Q6	5	4	8	5	0	0	5	4	8	5
Q7	7	6	6	7	0	0	7	6	6	7
Q8	1	1	1	1	0	0	1	1	1	1
Q9	5	3	3	5	0	0	5	5	5	5
Q10	83	12	20	20	0	0	20	20	20	20
Q11	2	1	1	2	0	0	2	1	1	2
Q12	3	3	3	3	0	0	3	3	3	3
Q13	7	6	6	7	0	0	7	6	6	7
Q14	145	19	20	20	19	20	20	20	20	20
Q15	40	8	20	20	0	0	20	20	20	20
Q16	11	1	16	11	11	11	11	11	11	11
Q17	13	3	20	13	0	0	13	4	4	13
Q18	7	1	4	7	0	0	7	4	20	7
Q19	25	10	17	20	0	0	20	11	11	20
Q20	53	3	20	20	20	20	20	20	20	20
Q21	37	18	20	20	0	0	20	20	20	20
TOTAL	665	160	285	281	112	131	281	234	276	281
		PREC	REC	F-MEAS	PREC	REC	F-MEAS	PREC	REC	F-MEAS
		0.56	0.57	0.57	0.85	0.40	0.54	0.85	0.83	0.84

Fig. 3. Comparative Evaluation of KS, pure semantic search, and HS on 20 queries. POS are the possible correct answers. COR is the number of correct answers returned by the system. ACT is the number of total answers provided by the system. EXP=min(POS, 20) and is the number of correct answers expected, given the limitation to 20 hits.

Moreover, consider that HS is defined as ranked intersection of results from the other two methodologies; therefore it cannot discover more documents than the other two methods. So its recall is a direct function of the recall of the other two modalities.

The experiments showed that semantic search has very high precision, but the lowest recall in identifying relevant documents in the first 20 returned results (Figure. 3).

This is because the metadata did not cover completely 6 of the topics. Keyword-based search has the lowest precision and fair recall in the same task. Hybrid Search reports very high precision (same as OS, +51% with respect to KS), and the highest recall (+46% with respect to keywords and +109% with respect to ontology-based search). (weighted harmonic) F-Measure is +49% with respect to keywords and +55% with respect to ontology-based. In conclusion, in our experiment HS outperforms the other methods in ranking relevant documents within the first 20 results. Experimental results for the first 50 returned documents are largely equivalent.

5.2 In-Vivo Evaluation

A user evaluation was carried on with 32 users (recruited from a number of departments of Rolls-Royce plc) that individually tested the system. The goal was to evaluate K-Search as a means for both document retrieval and knowledge retrieval. The individual sessions lasted an average of 90 minutes. After a short introduction to the system participants were asked to carry out an assisted training task to familiarise with the features of K-Search and the idea of HS. Then they were asked to carry out a second task without assistance. Finally they were asked to propose and carry out a

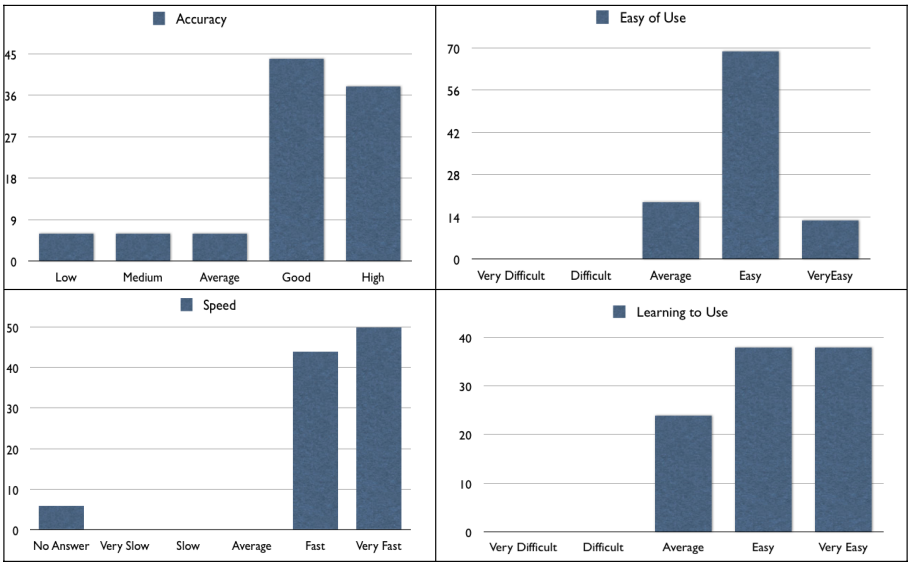

Fig. 4. Results of evaluation of K-Search by 32 users (values are in %)

task that reflected their work experience and interests. A user satisfaction questionnaire was filled in at the end of the test; a short interview on the experience closed the session.

The data collected allows assessing the validity of the HS paradigm as well as the usability of K-Search (Figure 4):

- **Use of HS:** all users appeared to have grasped the concept of HS. Users adopted different strategies: some started querying using keywords and added conditions on metadata in a second iteration; others instead composed conditions on metadata and keywords in a single search; others used metadata search initially and added keywords later to refine the task. This means that different user's searching strategies can be accommodated within the framework.
- **Learning:** 75% of users found easy or very easy to learn to use the system, 25% found it average.
- **System accuracy** (system reliability in retrieving relevant documents)**:** 82% of the users judged K-Search reliable or highly reliable; although this could seem a feature of the system rather than of HS, in our view the comment refers to the fact that with HS the searches were effective.
- **Searching experience:** 82% of users found K-Search easy or very easy to use; the ease of use was often commented about in the interviews;
- **System Speed:** the system was judged fast or very fast in executing the queries allowing a quick task completion by 98% of users.

6 Comparison with the State of the Art

Most of the methodologies proposed for semantic search [1, 2, 3, 4, 9] consider accessing documents using metadata only and do not consider the cases where metadata is unavailable. However, the HS idea can in principle be implemented with all the main types of semantic search described in literature. In [1] Uren et al classify the approaches according to: keywords-based approaches, view-based approaches, natural language approaches and form-based approaches. **Keyword-based approaches** [3, 16] are based on the interpretation of keywords according to an underlying ontology. They require translating all the keywords in order to perform the query. These methods could implement HS by replacing keywords in the query with concepts in the ontology when possible while leaving the rest for pure keyword-based searching. A **View-based approach** [17] is based on querying by building visual graphs exploring the ontology. This is quite effective and appealing method to query; however as [2] noted, experimentally this is one of the least preferred methods for querying. These approaches could easily support HS by just adding a new arc labelled e.g. *document-contains* that sends the query to the indexer rather than to the knowledge base engine. A **natural language approach** [18] addresses querying the knowledge base using natural language. A parser and a semantic analyser interpret the query and transform it into formal queries to the knowledge base. These methods are quite appealing to users, but generally suffer from limitations in the expressiveness of the underlying supported language [2]. They could implement the approach quite naturally by recognising expressions like ("and the document contains…"). Finally, we have seen HS can be implemented in a **form-based approach.** The model could be easily built into other models of this type presented in literature [19]. We have chosen to implement a form-based approach in K-Search because user analysis for the use cases at hand showed that this was the way of interaction preferred by our users.

Concerning other hybrid models, Rocha et al [14] presented a hybrid approach where users input a set of keywords which are sent to a search engine. The results of the search engine are re-ranked using semantic information associated to fields in the ontology. For example, they use the (generally long) provenance text of some annotations to decide that some documents are semantically more relevant than others. They have a spreading mechanism to reach also concepts not explicitly mentioned in the document. This method is similar in spirit to the keyword-based semantic searches such as [3] but it allows retrieval of the cases where there is no metadata available. So it centres on some of our initial objectives. However, the method is equivalent to the use of our keyword searching plus keyword-in-context searching. There is no way to address unique concepts and relations directly as in our model. Also, the keywords-first approach does not solve the issue of synonymity mentioned for keyword-based searches.

KIM [9] provides keyword-based search and ontology-based search as alternative options, i.e. a query is either based on keywords or on metadata but it does not enable mixing them. This is quite reductive with respect to the full HS mechanism.

LKMS [5] enables integration of keyword-based search and ontology-based search, but the actual functionality, the way the combination is performed, the expressive power of the formalism used and a number of details are unclear in the literature. It appears that their annotation is limited to named entities and that their form of HS

reduces to searching for the presence of a concept in a document or in the metadata. They do not seem to provide any facility for Boolean queries. Also, even if the system has dozens of real world users, it is unclear how and to what extent they actually use the HS mechanism.

7 Conclusions and Future Work

In this paper we have proposed HS, a mixed approach to searching based on a combination of keyword-based and semantic search. We believe that hybrid search is interesting because it overcomes an implicit limitation of most of the current literature that is that semantic search must rely on metadata only. We have given a formal definition of the method and we have shown experimentally that HS outperforms both keyword-based search and pure semantic search in a real case scenario. We have also shown how the strategy is compatible with most of the current models presented in literature. We believe that this is because:

- Hybrid search performs equally well as pure semantic search when metadata is fully available for a specific query;
- When metadata does not cover the whole information need, HS reaches higher recall than pure semantic search via the use of keywords for sections not covered by IE. Recall is boosted with limited loss in precision;
- HS outperforms keyword-based search in terms of both precision and recall. Higher precision is obtained by the use of metadata when available. Higher recall is obtained thanks to better ranking capabilities due to the use of metadata.
- In cases where the metadata is unavailable, HS is equivalent to keyword-based search.

HS has been implemented in a working system, K-Search, and two real world applications have been developed for Rolls-Royce plc. Such applications are currently deployed for real users to (i) access event reports and (ii) retrieve document and knowledge about requests of product technical variances. User studies carried out before the launch of the applications have shown appreciation for the system and the hybrid approach in general. More applications are being planned. A University spin-out company has been created to exploit the HS approach and its applications.

Future work will clarify some outstanding issues. The major issue concerns the use of IE in tasks where it does not perform at a very high standard. In those cases, the findings could change; because it could be no longer true that semantic search provides high precision. All the findings above are based on this important aspect. With lower precision, the strategy of designing hybrid search as applying semantic search when possible and resorting to keyword for the uncovered parts could actually prove to be not the most effective strategy. Experiments have to be carried out to understand the consequences of reduced precision and recall in the annotation process.

Acknowledgments. This work was supported by IPAS, a project jointly funded by the UK DTI (Ref. TP/2/IC/6/I/10292) and Rolls-Royce plc and by X-Media (www.x-media-project.org), an Integrated Project on large scale knowledge management across media, funded by the European Commission under the IST programme, (IST-FP6-026978). Thanks to Colin

Cadas (Rolls-Royce) for the constant support in the past two years. Thanks to all the users for their very positive attitude and the helpful feedback.

References

1. Uren, V., Lei, Y., Lopez, V., Liu, H., Motta, E., Giordanino, M.: The usability of semantic search tools: a review, Knowledge Engineering Review (in press)
2. Kaufmann, E., Bernstein, A.: How Useful are Natural Language Interfaces to the Semantic Web for Casual End-users? In: Proceedings of the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference, Busan, Korea (November 2007)
3. Lei, Y., Uren, V., Motta, E.: SemSearch: A Search Engine for the Semantic Web. In: Staab, S., Svátek, V. (eds.) EKAW 2006. LNCS (LNAI), vol. 4248, Springer, Heidelberg (2006)
4. Guha, R., McCool, R., Miller, E.: Semantic Search. In: 12th International Conference on World Wide Web (2003)
5. Gilardoni, L., Biasuzzi, C., Ferraro, M., Fonti, R., Slavazza, P.: LKMS – A Legal Knowledge Management System exploiting Semantic Web technologies. In: Proceedings of the 4th International Conference on the Semantic Web (ISWC), Galway (November 2005)
6. Chakravarthy, A., Lanfranchi, V., Ciravegna, F.: Cross-media Document Annotation and Enrichment. In: Proceedings of the 1st Semantic Authoring and Annotation Workshop, 5th International Semantic Web Conference (ISWC 2006), Athens, GA, USA (2006)
7. Ireson, N., Ciravegna, F., Califf, M.E., Freitag, D., Kushmerick, N., Lavelli, A.: Evaluating Machine Learning for Information Extraction. In: Proceedings of the 22nd International Conference on Machine Learning (ICML 2005), Bonn, Germany (2005)
8. Kiryakov, A., Popov, P., Terziev, I., Manov, D., Ognyanoff, D.: Semantic annotation, indexing, and retrieval. Journal of Web Semantics 2(1), 49–79
9. Shneiderman, B.: Designing the User Interface, 3rd edn. Addison-Wesley, Reading (1997)
10. Dzbor, M., Domingue, J.B., Motta, E.: Magpie - towards a semantic web browser. In: Fensel, D., Sycara, K.P., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, Springer, Heidelberg (2003)
11. Lanfranchi, V., Ciravegna, F., Petrelli, D.: Semantic Web-based Document: Editing and Browsing in AktiveDoc. In: Proceedings of the 2nd European Semantic Web Conference, Heraklion, Greece (2005)
12. Rocha, R., Schwabe, D., Poggi de Aragão, M.: A Hybrid Approach for Searching in the Semantic Web. In: The 2004 International World Wide Web Conference, New York, May 17-22 (2004)
13. Iria, J., Ciravegna, F.: A Methodology and Tool for Representing Language Resources for Information Extraction. In: Proc. of LREC 2006, Genoa, Italy (May 2006)
14. Tran, T., Cimiano, P., Rudolph, R., Studer, R.: Ontology-based Interpretation of Keywords for Semantic Search. In: Proceedings of the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference, Busan, Korea (November 2007)
15. Catarci, T., Di Mascio, T., Franconi, E., Santucci, G., Tessaris, S.: An Ontology Based Visual Tool for Query Formulation Support. In: 16th European Conference on Artificial Intelligence (ECAI 2004), Valencia, Spain (2004)
16. Kaufmann, E., Bernstein, A., Zumstein, R.: Querix: A natural language interface to query ontologies based on clarification dialogs. In: 5th ISWC, Athens, GA, pp. 980–981 (2006)
17. Corby, O., Dieng-Kuntz, R., Faron-Zucker, C., Gandon, F.: Searching the Semantic Web: Approximate Query Processing Based on Ontologies. IEEE Intelligent Systems 21(1) (2006)