This is a preprint of a book chapter accepted for publication in:
Professional Search in the Modern World, Lecture Notes in Computer Science (LNCS), Vol. 8830, Springer, 2014

Exploratory Professional Search through Semantic Post-Analysis of Search Results

Pavlos Fafalios and Yannis Tzitzikas

Institute of Computer Science, FORTH-ICS, and
Computer Science Department, University of Crete, Greece
{fafalios,tzitzik}@ics.forth.gr

Abstract. Professional Search is usually a recall-oriented problem. For helping the user to get efficiently a concise overview, to quickly restrict the search space and to make sense of the results, in this article we present an exploratory strategy for professional search that is based on semantic post-analysis of the classical search results (of keyword based queries). The described strategy can exploit the metadata that are already available, as well as the results of textual clustering and entity mining that can be performed at query time. The outcome of this process (i.e. metadata, clusters and entities grouped in categories) complement the ranked list of results produced from the core search engine with useful information for the user. This extra information is useful not only for providing a concise overview of the search results, but also for supporting a faceted and session-based interaction scheme that allows the users to restrict their focus gradually and to explore other related information. To tackle the corresponding configuration requirements of this process, we show how one can exploit the (constantly evolving) Linked Data for specifying the entities of interest and for providing further information about the identified entities. In this article, apart from detailing the steps of this process, we present applications of this approach in the *marine* domain and in the domain of *patent* search.

Keywords: exploratory search, professional search, entity mining and exploration, linked data, faceted search

1 Introduction

In professional search (e.g. medical search, patent search, bibliography search), it is often unacceptable to miss relevant documents, therefore the retrieval (and inspection) of nearly all relevant documents is sometimes necessary. In that context, the grouping of the numerous search results through *facets* that correspond to various kinds of *metadata* or *extracted entities* can help the user to get efficiently a concise overview, to quickly restrict the search space, and to make better sense of the results. Moreover, the integration of unstructured documents that usually appear in the search results, with the emerging Web of Data can bring significant benefits and is nowadays a challenging vision.

In this chapter, we describe a search process for recall-oriented information needs (i.e. focusing on retrieving as much as possible relevant documents) that is based on post-processing of the search results. It can also be considered as an integration approach that takes place during *searching* and aims at enriching the responses of non-semantic professional search systems with semantic information. Specifically, we analyze an exploratory search process which combines the following methods:

- Exploitation of the static metadata of the (top) search results
- Textual Clustering of the (top) search results
- Named Entity Extraction in the (top) search results
- Semantic enrichment and exploration of the identified entities using external (online) Knowledge Bases (KB), i.e. Linked Data.
- Session-based interactive exploration of the above information

The results of this process (metadata, clusters and entities grouped in categories) complement the query answers with useful information for the user which is further exploited in a faceted and long session interaction scheme that allows users to restrict their focus gradually as their information need is better defined[1]. This is important because a high percentage of search tasks are *exploratory* and focalized search very commonly leads to inadequate interactions and poor results [58, 42]. This interaction scheme can also save a lot of time in professional searches as it allows locating very quickly hits which are low ranked.

We could say that the issue of *integrated professional search systems* is addressed from two perspectives. From an *information integration* perspective, we can say that entity names are used as the "glue" for automatically connecting documents with data (and knowledge). This approach does not require designing or deciding on an integrated schema/view (e.g. [55]), nor mappings between concepts as in KBs (e.g. [54, 33]), or mappings in the form of queries as in the case of databases (e.g. [32]). Entities can be identified in documents, data, database cells, metadata attributes and KBs.

From an *information seeking* process perspective we present the tight integration of different search tools for a) faceted search using existing metadata, b) entity extraction and c) textual clustering, with the main retrieval engine which produces ranked lists of documents in response to a query. This integration allows different search interfaces to coexist in an information seeker's search system.

Correlation to the MUMIA Cost Action Part of the work in this chapter has been done in the context of MUMIA Cost Action[2]. Specifically, the Working Group 4 *(Semantic Search, Faceted Search and Visualization)* has the objective to identify and critically review the aspects of next generation search related

[1] Faceted search is a technique for accessing information organized according to an analytic-synthetic classification scheme, allowing users to explore a collection of information by applying multiple filters [49].

[2] `http://www.mumia-network.eu/`

to Semantic and Faceted search, as well as to study visualization techniques that can be applied as multiplying "gain" factor to both types of search. In this chapter, we present an approach for *semantic search* which is based on the post-analysis (by performing entity mining at query-time) of the results of a professional search system. In addition, we show how the result of the above process, as well as the result of textual clustering and of metadata-based grouping, can be integrated and exploited in a *faceted interaction scheme*. In a nutshell, and in correspondence to the objective of this Working Group, this chapter proposes a generic model of how Semantic and Faceted Search can be applied in next generation professional search.

The rest of this chapter is organized as follows: Section 2 describes in detail the post-analysis process. Section 3 analyzes the interaction model that can support this functionality. Section 4 reports experimental results regarding effectiveness and efficiency, ways of improving the efficiency of the proposed approaches and limitations that arise when exploiting online KBs. Section 5 presents two application examples that demonstrate how this process can be applied in the *marine* domain as well as in the domain of *patent* search. Finally, Section 6 concludes and identifies directions for future research.

2 Post-Analysis of Search Results

This section first presents the motivation, the context, some related works, and the steps of the overall process. Then, it describes the main post-analysis functions (metadata-based grouping, textual clustering, entity mining) and discusses issues of ranking and connectivity that arise in such context.

2.1 Motivation

The *analysis of search results* is a useful feature as it has been shown by several user studies. For instance, the results in [36] show that categorizing the search results improves the search speed and increases the accuracy of the selected results. A user study [35] shows that categories are successfully used as part of users' search habits. Specifically, users are able to access results that are located far in the rank order list and formulate simpler queries in order to find the needed results. In addition, the categories are beneficial when more than one result is needed like in an *exploratory* or *undirected search* task. According to [40] and [59], recall-oriented information can play an important role not only in understanding an information space, but also in helping users select promising sub-topics for further exploration.

Recognizing entities and grouping hits with respect to entities is not only useful to public web search, but is also particularly useful in *professional search* that is, search in the workplace, e.g. in industrial research and development [38]. A user study [47] indicated that categorizing dynamically the results of a search process in a *medical* search system provides an organization of the results that is clearer, easier to use, more precise, and in general more helpful than simple

relevance ranking. As another example, in professional *patent search*, in many cases one has to look beyond keywords to find and analyse patents based on a more sophisticated understanding of the patent's content and meaning [34]. We should also stress that professional search sometimes requires a long time. For instance, in the domain of patent search, the persons working in patent offices spend days for a particular patent search request. The same happens in bibliographic and medical search.

Technologies such as entity identification and analysis could become a significant aid to such searches and can be seen, together with other text analysis technologies, as becoming the cutting edge of information retrieval science [15]. Analogous results have been reported for search over collections of *structured* artifacts, e.g. ontologies. For instance, [8] showed that making explicit the relationships between ontologies and using them to structure (or categorize) the results of a Semantic Web Search Engine led to a more efficient ontology search process.

Finally, the usefulness of the various analysis services (over search results) is subject of current research, e.g. [19] comparatively evaluates *clustering* versus *diversification* services.

2.2 Context and Related Works

The idea of enriching the classical *query-and-response* process of current Web search engines, with *static* and *dynamic* metadata for supporting *exploratory search* was proposed in [45] and it is described in more detail (enriched with the results of a user-based evaluation) in [44]. In that work the notion of *dynamic metadata* refers to the outcome of *results clustering* algorithms which take as input the *snippets* of hits, where snippets are *query word dependent* (and thus they cannot be extracted, stored and indexed a-priori). Note that the result of entity mining if applied over the textual snippets also falls into the case of *dynamic metadata*.

There is also a plethora of works and systems that offer a kind of entity search. `EntityCube`[3] generates *summaries of entities* from Web pages and allows the exploration of their relationships. However it supports only three categories of entities (people, locations and organizations) and for complex or long queries that do not contain the entity of interest the results are poor. In [21], the authors propose a framework with two indexing and partition schemes for efficient entity search in which users formulate queries that directly describe what types of entities they are looking for (using the prefix `#`, e.g. `#phone`). [11] presents `ESTER`, a modular search system that combines full-text and ontology search. `ESTER` supports entity recognition by assigning words or phrases in the corpus to the entities from the ontology they refer to. It is domain-specific and elaborates well on a small set of predefined categories of entities (ontology classes). Finally, [57] discovers (offline) *entity structures* in Web pages regarding the computer science

[3] http://entitycube.research.microsoft.com/

domain, and constructs a heterogeneous network of bibliographic information (which is then analyzed) for offering keyword-based entity search.

In contrast to the works listed in the previous paragraph (which focus on entity search and retrieval), the approach that we describe in this article does not change the (user-friendly) way users search for information, but acts as a mediator between any search system and semantic information; users still get documents as search results, but also get and interact with semantic information that helps them locate and explore fast possible useful results.

2.3 The Steps of the Post-Analysis Process

Hereafter we consider the following, quite general, process for post-analysis of search results (also depicted in Figure 1):

1) The user submits a keyword query to the professional search system.
2) The search system retrieves the top-K results that correspond to the submitted query together with the static metadata of each result.
3) The search system uses a component, we call it `PProc` from "Post-Processor", which derives (ideally at real-time) the *cluster labels* and the *entities* that correspond to the top-K results (Sections 2.5 and 2.6 detail this step).
4) `PProc` *groups* the metadata values according to their category and the entities according to their class.
5) `PProc` *ranks* the metadata values of each category, the clusters and the entities of each class (cf. Section 2.7).
6) The results (i.e. groups of ranked metadata values, clusters and entities) are visualized and exploited in a faceted and session-based interaction scheme [49] that allows the user to restrict his/her focus or information need *gradually*, and exploits the results of the previous steps (cf. Section 3). Apart from gradual exploration, the user can also retrieve more information about an identified entity by exploiting the Linked Open Data (LOD) [14].

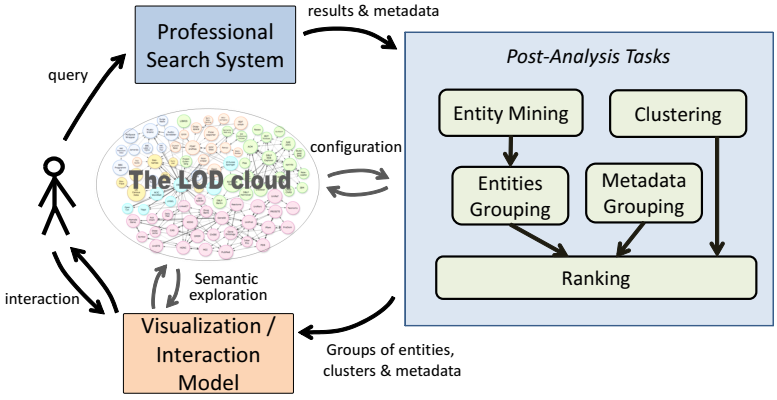

Fig. 1. Semantic post-analysis of search results

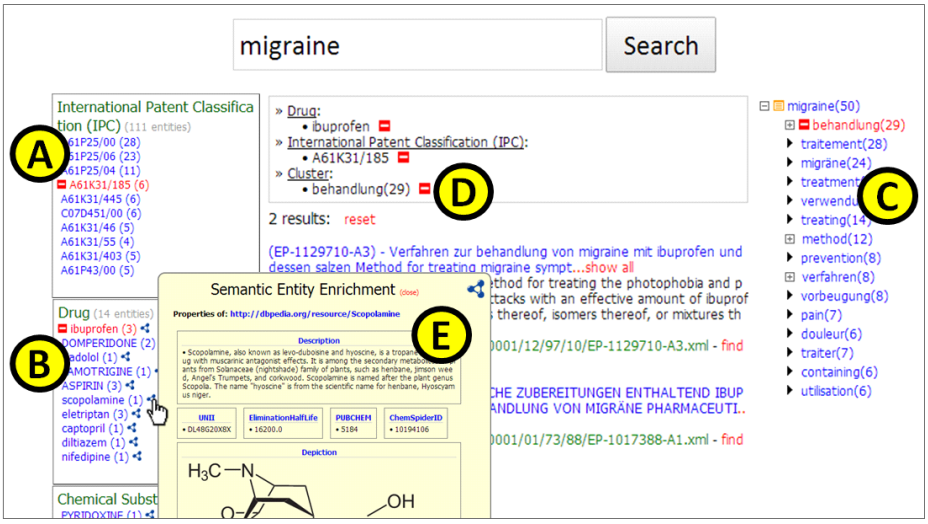

Fig. 2. A prototype offering real-time exploratory patent search

To grasp the idea and what the outcome of this process can be, Figure 2 depicts an indicative screen dump of a prototype patent-search system that offers the aforementioned functionality. Note that the user has many options for restricting the search space by selecting one or more metadata values of the category International Patent Classification (A), one or more entities of type Drug (B), or one or more cluster labels (C). For instance, in the current example the user has focused on the Drug ibuprofen, the International Patent Classification A61531/185, and the Cluster behandlung, restricting the search space to only 2 results which correspond to the selected facets (D). Furthermore, the user is able to retrieve (at real-time) more information about an entity (E) by exploiting online semantic KBs (e.g. DBpedia [10]).

2.4 Metadata-based Grouping

For grouping the results according to their static metadata values, one has to decide which metadata elements are important in a professional search process, i.e. which of them are the more useful and likely to be used in a faceted search-like process for narrowing the search space. For example, in the context of patent searching, a patent document has numerous metadata elements. Indicatively, a patent document of the *Matrixware Research Collection* (`MAREC`) data corpus[4] may contain more than 20 metadata elements including *document identification fields, concerned parties, filing and priority information, national and international classification codes, titles, abstracts and descriptions (in many languages), citations, related applications, claims*, etc.

[4] http://www.ir-facility.org/prototypes/marec

For selecting the fields which are most useful and likely to be used in a faceted search-like interface, one approach is to gather opinions from professionals in the corresponding domain. For example, [27] gathered opinions during interviews with patent examiners from the *Industrial Property Organization of Greece*[5], in a visit aiming to observe patent examiners searching in their working environment. They also did an one-on-one interview with a very experienced patent examiner specifically to learn about their attitudes and beliefs accompanying the usefulness of different types of metadata in patent search. The expert mentioned the following nine metadata fields as being important in a faceted patent search: *International Patent Classification (IPC), European Classification (ECLA), Applicant, Inventor, publication number, publication country, publication year, application country, application year*. Thus, they decided to offer the above metadata fields for faceted exploration in their patent search system.

2.5 Textual Clustering

Results clustering aims at grouping the search results into topics (called clusters), with predictive names (labels), aiding the user to locate quickly one or more documents that otherwise it would be laborious to find, especially if they are low ranked. Results clustering is very useful in cases where there are no metadata elements (e.g. in cases where only textual snippets of the hits are available), or in cases of metadata elements that contain long textual descriptions.

There are many algorithms for results clustering. In our applications we adopt the clustering method described in [39] which is actually a variation of the Suffix Tree Clustering (STC) algorithm [60], called `NM-STC` (No-Merge STC), that derives hierarchically organized labels and is able to favor occurrences in a specific part of the result (e.g. in the title). Figure 3 depicts the result of the `NM-STC` algorithm applied in the top-200 snippets returned by Bing Search Engine for the query `tuna species`. By clicking for example the cluster label "seafood (7)", the user can inspect the seven results that contain information about tuna species and seafood.

Fig. 3. Top-10 cluster labels for the query `tuna species`

[5] http://www.obi.gr/

The main advantage of the STC-based algorithms is that they do not rely on external resources or training data, and thus they have broad applicability (e.g. for different natural languages). For supporting a different language, the user/administrator must only provide a list with the *stop words* of this language. This feature is particularly useful in patent search where the patent descriptions may be in different languages. In addition, the *invention title* is usually the most important and descriptive part of a patent, therefore `NM-STC` can favor occurrences of labels (topics) in the title.

2.6 Discovering Entities

Named Entity Extraction (NEE), also referred as *semantic annotation*, is the process of identifying entities in texts and linking them to relevant semantic resources. NEE often consists of two main sub-processes: *named entity recognition* (or *entity mining*) which is the task of identifying entities belonging to a set of class labels (such as Person, Location, Organization, etc.), and *entity linking* which tries to link a named entity with a resource in a KB.

There are several tools that support NEE and that could be exploited by a professional search system. Below we briefly describe some of them.

DBpedia Spotlight: DBpedia Spotlight [43] is a tool for annotating mentions of DBpedia resources in text, providing a solution for linking unstructured information sources to the LOD. It performs NEE, including entity resolution. It finds and returns entities that exist in a text, ranks them depending on how relevant they are with the text content, and links them with URIs from DBpedia.

AlchemyAPI: AlchemyAPI [1] is a Natural Language Processing (NLP) service which provides cloud-based and on-premise text analysis infrastructure. AlchemyAPI eliminates the expense and difficulty of integrating NLP systems into an application, service or data processing pipeline. It provides a platform for analyzing Web pages, documents and tweets along with APIs for integration. In addition, the named entity extractor is able to disambiguate the detected entities, link them to various datasets on the LOD and resolve co-references.

OpenCalais: Calais [3] is a toolkit of capabilities that allows incorporating semantic functionality within a blog, content management system, Web site or application. The OpenCalais Web Service automatically creates semantic metadata for the submitted content. Using NLP, Machine Learning (ML) and other methods, Calais analyzes a document, finds the entities within it and gives them a score based on their text relevance. It also supports automatic connection to the LOD.

Wikimeta: Wikimeta [7] is a NLP semantic tagging and annotation system that allows incorporating semantic knowledge within a document, Web site or content management system. It tries to link each detected named entity to some entity in DBpedia based on a disambiguation process that is described in [20]. The dataset used to train the NLP tools of Wikimeta, are derived from Wikipedia and are also available to download to build customized applications, gazetteers (i.e. dictionaries) or training corpora.

Lupedia: Lupedia [2] uses a gazetteer which is a list of surface forms that are associated to a subset of entities in DBpedia and LinkedMDB (a dataset that contains movies descriptions). The default configuration takes the longest sequence of consecutive words that corresponds to some entry in the gazetteer and annotates it with the corresponding entity in the KB.

Gate ANNIE: Gate ANNIE [22, 16] is a ready-made information extraction system which contains several components (e.g. Tokeniser, Gazetteer, Sentence Splitter, etc.) and supports both gazetteers and NLP functions.

We should note that we are interesting only in the result of the NEE process. Thus, the approach that we propose can use any NEE system, independently of the underlying NLP algorithm. This also means that the support of several capabilities involved in the entity mining process (e.g. support of multiple languages, stemming, lemmatization, etc.) are in the responsibility of the NEE system.

Configuring the Entities of Interest

The useful entities are not the same in every domain. To tackle the corresponding configuration requirements of the considered process, here we show how one can exploit the (constantly evolving) LOD for specifying and updating the entities of interest, and for providing further information about the identified entities.

In case a NEE system supports *gazetteers* for named entity recognition (apart from NLP and ML techniques), like `Gate ANNIE`, we can exploit the LOD for creating new (supported) categories of entities. Specifically, we can query a semantic KB (that is accessible through a SPARQL [6] endpoint) for retrieving a list of names that belong to a particular *resource class*[6] or which are described by a *SPARQL query*.

For example, Figure 4 shows an example of a SPARQL query that returns a list of Fish names (using DBpedia's SPARQL endpoint as the underlying KB). The URI "`http://dbpedia.org/ontology/Fish`" is the resource class of the category Fish. Likewise, we can create a complex SPARQL query that describes a set of entities with some specific characteristics, or we can use *federated SPARQL queries* [5] and gather information from multiple KBs. Thereby, we can also support the identification of entities in any language, e.g. we can run a query that returns a list of entity names in a specific language by exploiting the `FILTER` operator of SPARQL 1.1. In addition, by exploiting the LOD, and since the LOD constantly changes and increases, we can keep "fresh" the entities of the supported categories or update (at any time) a category with names of entities coming from a new KB.

Note that there are many tools that can facilitate the construction of the SPARQL queries, without requiring any advanced knowledge in SPARQL (like [9] and [48]). Furthermore, there are natural language approaches that guide users in formulating queries in a language seemingly akin to English and translate them to SPARQL [24].

[6] A resource class actually represents a category of entities.

```
SELECT DISTINCT str(?label) WHERE {
  ?uri rdf:type <http://dbpedia.org/ontology/Fish> .
  ?uri rdfs:label ?label }
```

Fig. 4. Example of a SPARQL query for retrieving a list of Fish names from DBpedia

Figure 5 depicts the result of entity mining applied in the top-200 snippets returned by Bing Search Engine for the query `tuna species`. In this example, the NEE system has been configured to identify fish species, countries and water areas. By clicking for example the entity "yellowfin tuna (9)" from the category Species, the user can inspect the nine results that contain information about the species "yellowfin tuna".

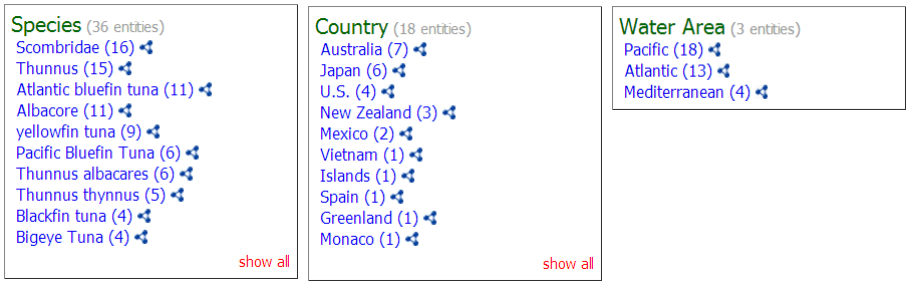

Fig. 5. The result of entity mining (configured to identify fish species, countries and water areas) for the query `tuna species`

2.7 Ranking

The identified entities, the clusters and the metadata values may be numerous. Thereby, we need an effective method for ranking them and promoting the most important. [25] and [27] proposed a ranking scheme for entity mining that promotes the entities that have been identified in the top positions of the ranked list of results. We can adapt this formula for ranking also the clusters and the metadata values. However, most clustering algorithms rank the derived clusters, thus in this case we can avoid ranking them. Specifically, consider that the user submits a keyword query q, and let R be the set of the top-K results (e.g. $K = 200$) returned by the underlying search system. For a $r \in R$, let $rank(r)$ be its position in the answer (the first result has rank equal to 1, the second 2, and so on). We apply entity mining and clustering in R, and get a set of entities E and a set of cluster labels C. We also have a set of metadata values M. Now, we can rank each element e in $E \cup C \cup M$ according to the formula:

$$Score(e) = \frac{\sum_{r \in hits(e)} ((|R| + 1) - rank(r))}{\frac{|R|(|R|+1)}{2}} \quad (1)$$

where $hits(e)$ denotes the results (i.e. the elements of R) in which an entity e has been identified. We can see that the elements (i.e. metadata values, entities and

cluster labels) occurring in the top results are promoted. The rational behind this ranking formula is that the top hits in the ranked list probably contain more useful elements that the last hits since they are considered "better" results. On this account (and considering that we analyze only the top-K results), the ranking algorithm of the underlying search system is very important.

[25] comparatively evaluated with users three formulas for entity ranking in the context of Web searching. The first ranking formula is the one described above (Formula 1). The second formula takes into account the words of the entity names and of the query, and promotes the entities that exist in the query string. The third formula considers both perspectives, i.e. it promotes both the entities identified in the top search results and the entities that exist in the query string. The results showed that the string similarity between the query and the entity names did not improve entity ranking.

An issue is how to also rank the categories of entities (or the metadata categories) if they are numerous. In this case, the system must decide which categories to promote (e.g. in order for the user to see them without needing to scroll). Various methods can be applied, however this is an issue that is worth further research.

2.8 Inspecting the Connectivity of the Identified Entities

So far, the user can only inspect a *list* of entities identified in the search results. The structured knowledge that may be available as LOD for the identified entities is not exploited. For instance and regarding the marine domain, an identified fish species (e.g. the *yellowfin tuna*) may have many properties (e.g. *family*, *genus*, *kingdom*, etc.) and related entities (e.g. *predators*, *binomial authority*, etc.), and can belong to multiple categories (e.g. *Fish*, *Eukaryote*, *Fish of Hawaii*, etc.). Moreover, some species may share one or more common properties or related entities (e.g. two species belong to the same *genus* or *family*). All this information should be exploitable as it can provide useful information about the *context* of these entities. In addition, it allows the user to instantly inspect information that may lie in different places and that may be laborious and time consuming to locate, e.g. how the detected species *papuan seerfish* and *kanadi kingfish* are related, why the species *pacific bonito* was detected in the search results for the query *tuna*, etc. Furthermore, all this information can be integrated in the search process helping the user (apart from restricting the search space) to get a more sophisticated overview and to make better sense of the results.

However, the amount of structured information that is available for these entities can be very high (i.e. their associations and properties). Therefore, there is a need for methods for ranking all this semantic information in order to promote and present to the end-users the most important associations and properties.

To tackle the above challenges, [29] proposes a method founded on Link Analysis. Specifically, this work introduces an appropriately biased PageRank-like algorithm for ranking entities and properties, which is also exploited for producing (and showing to the user) *top-K semantic graphs*. A top-K semantic graph can complement the query answer with useful information regarding the

connectivity of the identified entities. The keypoint is that this approach can exploit associations and it is quite general and configurable. Moreover, it promotes the entities identified in the top ranked results, as well as the semantic information that is linked with many important (i.e. highly ranked) entities.

For example and regarding the marine domain, by analyzing the snippets of the top-100 results that Bing returns for the query *yellowfin tuna* (with *fish species* as the entities of interest), and exploiting DBpedia at real-time for retrieving the properties of the identified entities, in the top semantic graphs the user gets information about the taxonomy of the *yellowfin tuna* (family, order, etc.), other tuna species that belong to the same family or the same conservation status system (e.g. the *bigeye tuna*), how all these entities are connected, etc. The user gets all this information in only 3 seconds without performing any additional query. Figure 6 depicts an example of a top-5 semantic graph.

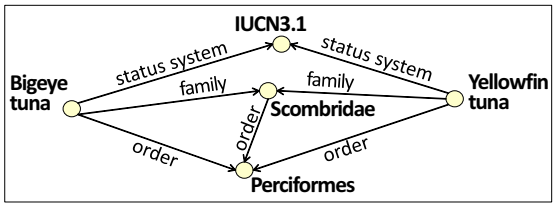

Fig. 6. A top-5 semantic graph

3 Interaction Model

This section focuses on how the user can interact with the result of the post-analysis process.

3.1 Faceted Search-Like Exploration of the Results

The results of entity mining, clustering and metadata-based grouping can be visualized and exploited according to the *faceted exploration* interaction paradigm [49]; when the user clicks on an entity, cluster or metadata value, the hits are restricted to those that contain that entity, cluster or metadata value. The user is able to gradually select elements from one or more categories and refine the answer set accordingly (the mechanism is session-based). Figure 7 depicts an indicative example regarding the marine domain in which the user has selected to inspect the results containing information for three species.

There are several approaches for supporting this functionality. For instance, if such selections belong to the same category, they can have disjunctive (`OR`) semantics and if they belong to separate categories they can have conjunctive (`AND`) semantics [27]. In addition, in order to avoid overloading the interface, we can display only the top-L (e.g. L=5) values of each category, and by clicking a hyperlink (e.g. a "show all" button), the user will be able to inspect all of them.

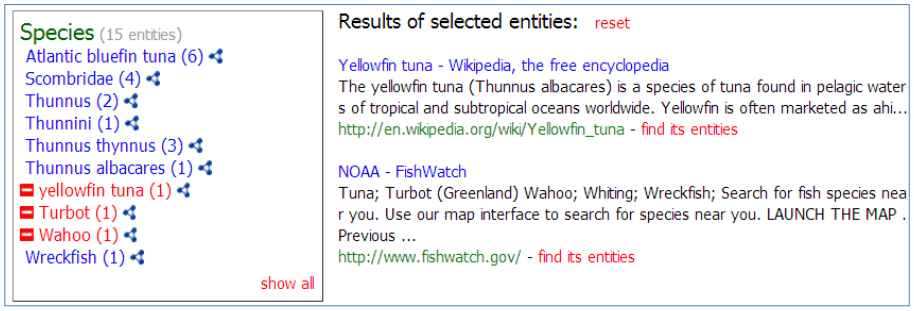

Fig. 7. Example of faceted exploration of the results

3.2 Semantic Exploration of the Identified Entities

There are already vast amounts of structured information published according to the principles of LOD. The availability of such datasets enables not only to configure easily the entity names that are interesting for the application at hand (as described in Section 2.6), but also the enrichment of the entities with more information about them. In this way, users not only can get *useful information* about one entity without having to submit a new query, but they can also start *browsing* the entities that are linked to that entity. Note that many of the *static metadata* can also be considered entities (e.g. the Author of a document).

Another important point is that exploiting LOD is more dynamic, affordable and feasible, than an approach that requires each search system to keep stored and maintain its own KB of entities and facts. Returning to our setting, a question is which LOD dataset(s) to use. An approach is to identify and specify one or more appropriate dataset(s) for each category of entities. For example, GeoNames[7] can be exploited for *geographic data*, DrugBank[8] for *drugs*, DBpedia, YAGO [52] and FactForge [13] contain data related to many domains, etc.

Running one (SPARQL) query for each entity would be a very expensive task, especially if the system has discovered a lot of entities. For this reason, one can offer this service *on demand*. Specifically when the user requests to inspect more information about an entity, e.g. by clicking a button, the system *at that time* can collect semantic resources that "match" the name of the selected entity by querying one or more SPARQL endpoints.

For example, Figure 8 shows an example of a SPARQL *template* query which tries to find a resource of type (`rdf:type`) Fish whose label (`rdfs:label`) contains the name of the selected entity (ignoring case). Note that the SPARQL template query contains the character sequence `[ENTITY]`. At request time, the system reads the endpoint and the corresponding template query of the category in which the identified entity belongs, replaces each occurrence of `[ENTITY]` in the template query with the entity's name, and finally runs the query. For in-

[7] `http://www.geonames.org/`
[8] `http://www.drugbank.ca/`

stance, for the entity name "salmon" (of type Fish) and having defined DBpedia as the underlying KB, the matching resources are 16. Some of them follow:

- `http://dbpedia.org/resource/Chum_salmon`
- `http://dbpedia.org/resource/Coho_salmon`
- `http://dbpedia.org/resource/Giant_salmon_carp`
- `http://dbpedia.org/resource/Salmon_shark`
- `http://dbpedia.org/resource/Chinook_salmon`

```
SELECT DISTINCT ?uri WHERE {
  ?uri rdf:type <http://dbpedia.org/ontology/Fish> .
  ?uri rdfs:label ?label
     FILTER(regex(str(?label), '[ENTITY]', 'i')) }
```

Fig. 8. Example of a SPARQL *template* query for matching an identified Fish name with resources in DBpedia

The derived semantic information can be visualized in a popup window as shown in Figure 2 (E). Then, the user is able to continue browsing by further exploring the properties of the related resources. For example, Figure 9 shows an example of a SPARQL template query for retrieving all the outgoing properties of a resource. The SPARQL template query contains the character sequence `[URI]` (including the [and]) which at request time is replaced by the resource's URI. Alternatively, someone could provide a query that retrieves only a subset of the properties, literals in a specific language, images, etc. Figure 10 depicts an example of a pop-up window showing some of the properties (from DBpedia) of the entity "Chum salmon" (resource class: `http://dbpedia.org/resource/Chum_salmon`).

```
SELECT DISTINCT ?propertyName ?propertyValue WHERE {
  <[URI]> ?propertyName ?propertyValue }
```

Fig. 9. Example of a SPARQL *template* query for retrieving the outgoing properties of resource

4 Experimental Results

Several works have pointed out the value (for the end users) of categorizing the search results in both Web and Professional search. We should note that the post-analysis approaches that we have described in this chapter also fall into this case, i.e. they are ways of categorizing search results.

In this section, we first review the results of several experimental evaluations that demonstrate the *effectiveness* and the *usefulness* of such approaches (Section 4.1). In the sequel, we report experimental results regarding the *efficiency*

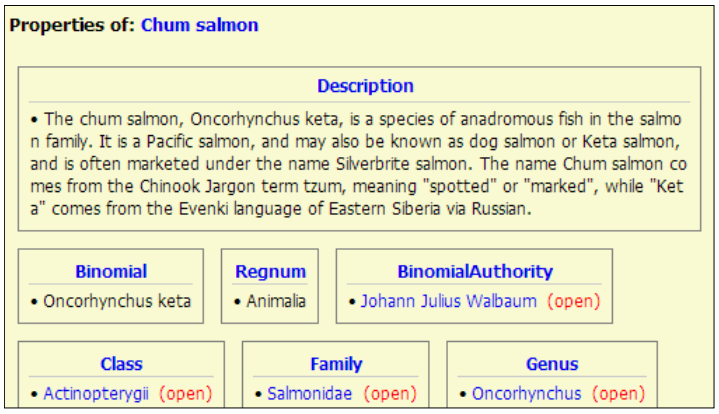

Fig. 10. Pop-up window showing some of the properties of the fish "Chum salmon"

of the post-analysis processes and we also discuss ways for improving it (Section 4.2). Finally, we present experimental results regarding the efficiency of the semantic exploration of the identified entities, we discuss limitations that arise when exploiting online KBs at query time, and also we show ways to improve the reliability of this process (Section 4.3).

4.1 Effectiveness and Usefulness

In [36], an experiment with 20 participants was conducted to compare an interface that categorizes the search results to the de facto standard solution (ranked list, 10 results per page interface). This interface provides an overview of the results by presenting a list of the most frequent words and phrases as categories next to the actual results. The results showed that the users were 25% faster and 21% more accurate with a system that categorizes the search results. In more details, the results showed that it is possible to browse through more results because the searching speed is higher. This is important, since many times the search results are unreliable and it is thus desirable to be able to access alternative results quickly. In addition, categorizing the search results not only gives the users more options but gives them more relevant options. The results showed that the increase in the number of results was due to the increase in the number of relevant results, while the speed of finding relevant results was about 40% higher. Furthermore, the users found the first relevant result earlier (with fewer selections).

In another experimental evaluation [35], the same interface was provided to 16 users for a two-month period. The interactions with the system were logged and the users' opinions were elicited with two questionnaires. The results showed that categories are successfully used as part of users' search habits. Specifically, categories are helpful when the result ranking of the search engine fails. In this case, the users are able to access results that locate far in the rank order list. Moreover, by exploiting the categories, there were fewer cases where user did

not find any results. This means that when the query formulation fails, the user may still be able to find results using the categories. Finally, the results showed that the categories are beneficial when more than one result is needed like in an *exploratory* search task.

The user study in [40] examined how searchers interacted with a web-based, faceted library catalog when conducting exploratory searches. It applied eye tracking, stimulated recall interviews and direct observation to investigate important aspects of gaze behavior in a faceted search interface. Three facets (i.e. categories of results) were used in the evaluation: *Subject*, *Region* and *Time Period*. The results showed that facets played a major role in the exploratory search process, accounting for about one-half the amount of time spent looking at actual results.

[59] presented the results of a four-week longitudinal study investigating the use of both exploratory and keyword forms of search within an online video archive, where both forms of search were available concurrently in a single user interface. The results showed that there was a balance of exploratory and keyword searches and that they were often used together. Specifically, the facets were used as often as keyword searches, and also they were used both passively to understand the structure of the collection and actively to produce more expressive queries.

Finally, [47] conducted a task-based user study of a *medical* search system for evaluating a dynamic categorization technique. The goal was to determine whether this technique for organizing search results is more useful than two existing techniques: relevance ranking (i.e. ordered list of search results) and SONIA document-clustering [50]. Fifteen users completed query-related tasks using all three tools. The authors measured the time it took the subjects to accomplish their tasks, the number of answers to the query that the subjects found in four minutes, and the number of new answers that they could recall at the end of the study. Subjects also completed a user-satisfaction questionnaire. The results showed that users could find significantly more answers in a fixed amount of time and were significantly more satisfied with their search experience when they used the dynamic categorization tool. In addition, the users indicated that this categorization provided an organization of search results that was more clear, easy to use, accurate, precise, and helpful.

4.2 Efficiency of Real-Time Post-Analysis

[27] measured the average time required for a) grouping the top-K results according the their metadata values (each result is actually a patent document), b) applying clustering (at real-time) on the title and abstract of the top-K results (using `NM-STC`), c) applying entity mining (at real-time) on the title and abstract of the top-K results (using `Gate ANNIE`), for several values of K. The results showed that the metadata-based grouping requires about 0.8 ms per result, the clustering about 3 ms per result, while entity mining is the most time consuming task requiring about 10 ms per result. The total time for analyzing the top-200 results is about 3 seconds. However, the three tasks can be performed

in parallel, i.e. the results of a task are not required for running another task. Note also that the time depends not on the size of the underlying data sources but on the number of the top results that we want to analyze; the more results we analyze, the more time is required for grouping, clustering and mining them.

In addition, [25] showed that performing real-time entity mining (using Gate ANNIE) in the *full contents* of the top-50 results returned by a Web search system (Google) costs about one minute (including the time for downloading the content of each result). Nevertheless, in that case one can adopt a distributed approach. A scalable method for entity-based summarization of Web search results at query time using the MapReduce programming framework [23] is described in [37]. That work shows how to decompose a sequential entity mining algorithm into an equivalent distributed MapReduce algorithm (the logical decomposition is sketched in Figure 11) and deploy it on the cloud for speedup the process.

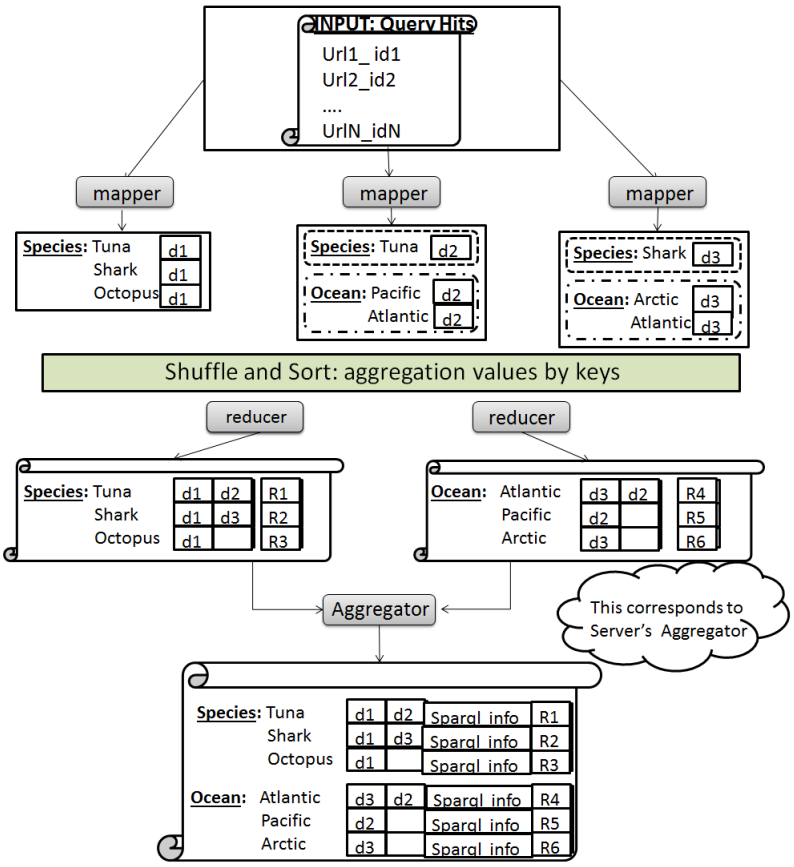

Fig. 11. Example of distributed entity mining processing using MapReduce

Alternatively, instead of offering *real-time* entity mining of the snippets or the full contents of the top hits of the answer, one could analyze the entire corpus

offline (assuming that the corpus is available), and build an appropriate index (or database) for using it at run time. Then, for each incoming query, the entities of the top-K (e.g. $K = 1,000$) hits of the answer are fetched from the index, and are given to the user. An important observation is that the size of the entity index in the worst case could be in the scale of the corpus. Also note that this approach cannot be applied at meta (uncooperative) search level.

Another approach is to process the top hits of the answer of only the *frequent queries*. In that case, for each frequent query of the log file (e.g. for those which are used for query suggestions), we compute its answer, fetch the top-K hits, apply textual clustering and entity mining and finally save its results as they should be shown using the approach and indexes described at [28, 26]. The benefit of this approach (apart from the *instant response*) is that here we do not have to process the entire collection but only the top hits (e.g. top-200) of the most frequent queries. This significantly reduces the required computational effort and storage space. The downside of this approach is that if a user submits a query which does not belong to the frequent queries, and thus it has not been processed, then the system cannot offer results. In that case the system could offer to the user the "real-time" approach as it was described earlier. Finally, we should note that this approach is applicable also at a meta search level, but periodically the index has to be refreshed, mainly incrementally.

4.3 Efficiency of Semantic Exploration

According to [27], the time for matching an entity with semantic resources (by querying the LOD) highly depends on the SPARQL endpoint (i.e. the underlying KB) and the SPARQL template query. The authors noticed that the more data (i.e. labels of entities) a category of entities contains in the underlying KB, the more time is required for matching an entity that belongs to this category.

Indicatively (and for the time being), DBpedia's endpoint contains about 1 million labels of type Geographical Area. For retrieving information about a geographical area, about 5 seconds are required (including network delay time). However, for retrieving information about a Physical Entity (DBpedia contains about 6 millions labels of this type), the time required is about 20 seconds. On the contrary, the time for retrieving the properties of a semantic resource is very low because we already know its URI (no string comparisons are required like in the case of entity matching).

Limitations The existing publicly available online KBs (like DBpedia) are not reliable since they mainly serve demonstration purposes. The fact that everyone can query them affects their efficiency and availability. They also do not serve multiple concurrent requests in order to avoid overloading their systems.

In addition, if an entity belongs to a category with millions of entities then the time for retrieving related resources (i.e. URIs) can be high. The same is true in case the underlying application requires to retrieve semantic information for numerous entities at once, i.e. when this functionality is not offered on-demand.

In such cases, adopting a *caching mechanism* or *indexing* a part of the underlying KB (with the cost of loosing the freshness of the results) will highly improve the response times and the throughput that can be served.

Of course, in a real application the underlying KBs may not be publicly available, or a *dedicated Warehouse* can be constructed that will only serve a particular application (like the *MarineTLO-based warehouse* described in [53]). The KBs (or the Warehouse) could also be *distributed* in many servers, so the system can apply a load balancing technique [18] for serving the requests. Furthermore, as it is proposed in [56], one could keep a local copy of data that hardly changes and offer a hybrid query execution approach for improving the response time and reducing the load on the endpoints, while keeping the results fresh. All the above can highly improve the performance and the scalability of the underlying professional search system.

Finally, we should stress that even if the post-analysis services require some time to complete, and therefore are not "real-time", in various kinds of professional search, this time is really low. For instance, in the domain of patent search, the persons working in patent offices spend days for a particular patent search request. In bibliographic search, a few minutes is a rather short period considering the time that could be saved if a useful hit corresponding to a low ranked document gets retrieved because a mined entity allowed the user to locate it.

5 Application Examples

This section presents two applications of the described approach. Specifically, Section 5.1 describes an application for the marine domain, while Section 5.2 describes an application for patent search.

5.1 X-Search: Exploring Marine Resources

`X-Search` is a meta-search engine that reads the description of an underlying search source (OpenSearch [4] compliant), queries that source, analyzes the returned results in various ways and also exploits the availability of semantic repositories. It also has a gCube version in which the underlying search system is gCube Search. gCube [17] is a service-oriented application framework that supports the on-demand sharing of resources for computation, content and application services. gCube enables the realization of e-infrastructures that support the notion of Virtual Research Environments (VREs), i.e. collaborative digital environments through which scientists, addressing common research challenges, exchange information and produce new knowledge.

`X-Search` has been developed in the context of the iMarine project[9]. iMarine exploits gCube and offers an operational distributed infrastructure that serves hundreds of scientists from the marine domain. The key features of `X-Search` are the following:

[9] `http:/www.i-marine.eu/`

- *Provision of textual clustering of the results* (supporting the following algorithms: `STC`, `STC+`, `NM-STC`, `STC++` and `NM-STC+` [39]). Clustering is performed on the textual snippets of the returned results, but clustering of the entire contents is also supported.
- *Provision of entity mining of the results.* Entity mining can be performed either over the textual snippets or over the entire contents. It also supports ranking of the identified entities [25].
- *Faceted search-like exploration of the results.* The results of clustering and entity mining are visualized and exploited according to the faceted exploration interaction paradigm: when the user clicks on a cluster or entity, the results are restricted to those that contain that cluster or entity.
- *On-click semantic exploration of a KB.* `X-Search` provides the necessary linkage between the mined entities and semantic information. In particular, by exploiting the *MarineTLO-based Warehouse* [53], the user can retrieve more information about an entity by querying and browsing over this KB. The *MarineTLO-based Warehouse* integrates information coming from FishBase [31], WoRMS[10], ECOSCOPE[11], FLOD[12] and DBpedia, and currently contains information (more than 4M triples) about marine species (40,000), ecosystems, water areas, vessels, etc.
- *Entity discovery and exploration during plain Web browsing.* `X-Search` also offers entity discovery and exploration while user is browsing on the Web. Specifically, the user is able to inspect the entities of a particular Web page by simply clicking a *bookmarklet*[13] and then to semantically explore the properties of the identified entities. Namely, the user can at real-time exploit the aforementioned functionality while browsing.

Figure 12 depicts an indicative screen shot of `X-Search` in gCube. We notice that for a particular query, the user can see the top results and the metadata of each result (A), the identified entities (B) and the result of textual clustering (C). The user can also inspect semantic resources that match an identified entity (D) and explore their properties (E). Figure 13 depicts a screen shot of an annotated Web page. Specifically, the Wikipedia page of *Thunnus* has been analyzed (using the bookmarklet provided by `X-Search`), the identified entities have been annotated and the user can start exploring them (A).

`X-Search` is fully configurable in terms of the supported categories of entities, the underlying KBs and the way the system queries the KBs. Specifically, the user/administrator can add a new category of entities or update an existing one by accessing online semantic KBs (accessible through SPARQL endpoints), and

[10] `http://www.marinespecies.org/`
[11] `http://www.ecoscopebc.ird.fr/EcoscopeKB/ShowWelcomePage.action`
[12] `http://www.fao.org/figis/flod/`
[13] A bookmarklet is a bookmark stored in a Web browser that extends the browser's functionality (`http://en.wikipedia.org/wiki/Bookmarklet`). In `X-Search`, the bookmarklet sends the current URL (of the Web page the user is viewing) to a server. The server then analyzes the contents of the Web page and presents to the user a new (annotated) Web page.

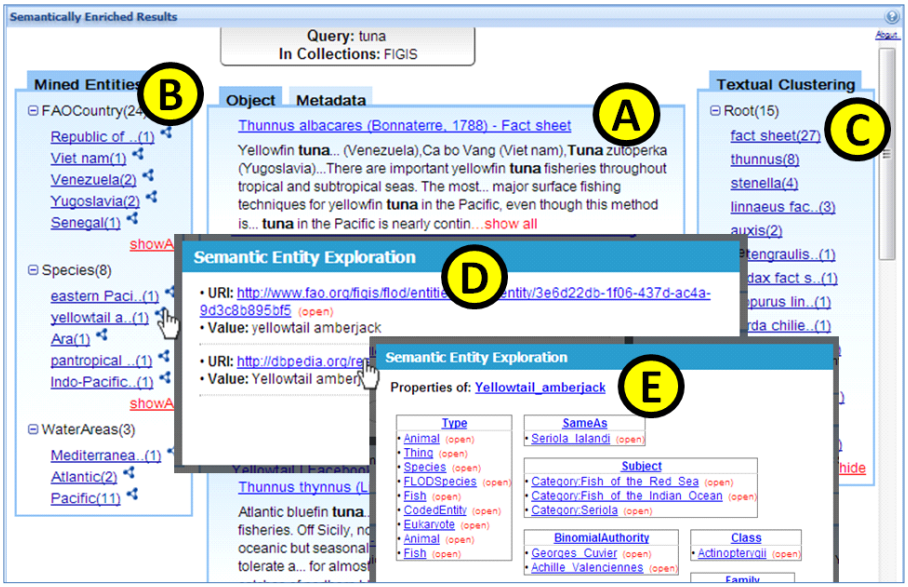

Fig. 12. The `X-Search` system

Fig. 13. An annotated Web page (using the bookmarklet provided by `X-Search`)

specify how to semantically link and enrich the identified entities. This enhanced configurability allows `X-Search` to be lightly (and dynamically) configured for different contexts, for building domain-specific applications. In the context of iMarine, `X-Search` has been configured to identify Fish Species, FAO Countries, Water Areas and Regional Fishery Bodies. It links the identified entities with resources from the MarineTLO-based warehouse [53] and enrich them by retrieving their outgoing properties.

5.2 PerFedPat: Pluggable Platform for Personalized Multilingual Patent Search

The `PerFedPat`[14] project aims to research into a new generation of advanced patent search systems for the patent related industries and the whole spectrum of patent users by designing a framework for integrating multiple patent data sources, patent search tools and UIs.

The `iPerFedPat` system [51], which is the main result of the project, is based on the `ezDL` framework [12] and has a pluggable architecture, providing core services and operations being able to integrate multiple patent data sources and patent related data streams, thus providing multiple patent search tools and UIs while hiding complexity from the end user. `iPerFedPat` currently integrates the results of four patent search systems: Clef-IP 2011 [46], Espacenet[15], Google Patents[16] and WIPO PatentScope[17]. It can post-process the federated results in various ways using pluggable tools, and supports all the functionalities described in Section 2, i.e. *metadata-based grouping*, *entity mining* and *textual clustering*.

Figure 14 depicts an indicative screen shot of `iPerFedPat`. We notice that the interface is split in several windows, each corresponding to a different tool. In this example, the user has submitted the query `migraine` using the "Advanced Query" tool (A) and the top results are shown in the "Results" tool (B). The user can also see the details of a particular result (C) and to inspect the entities and the metadata (grouped in categories) that exist in the search results (D), as well as a clustering of the search space (E). Thereby, the user can narrow the search space by a selecting one or more entities, metadata values or clusters.

6 Conclusion and Future Research

We have presented an exploratory method for professional search that exploits the available metadata plus the results of *textual clustering* and *entity mining* in a *faceted* and *session-based* interaction scheme that allows the users to get an overview of the search space and to restrict their focus gradually. We have seen that *Linked Data* can be exploited for specifying the entities of interest and for providing further information about the identified entities. This functionality

[14] `http://www.perfedpat.eu/`
[15] `http://www.epo.org/searching/free/espacenet.html`
[16] `https://www.google.com/?tbm=pts`
[17] `http://patentscope.wipo.int/search/en/search.jsf`

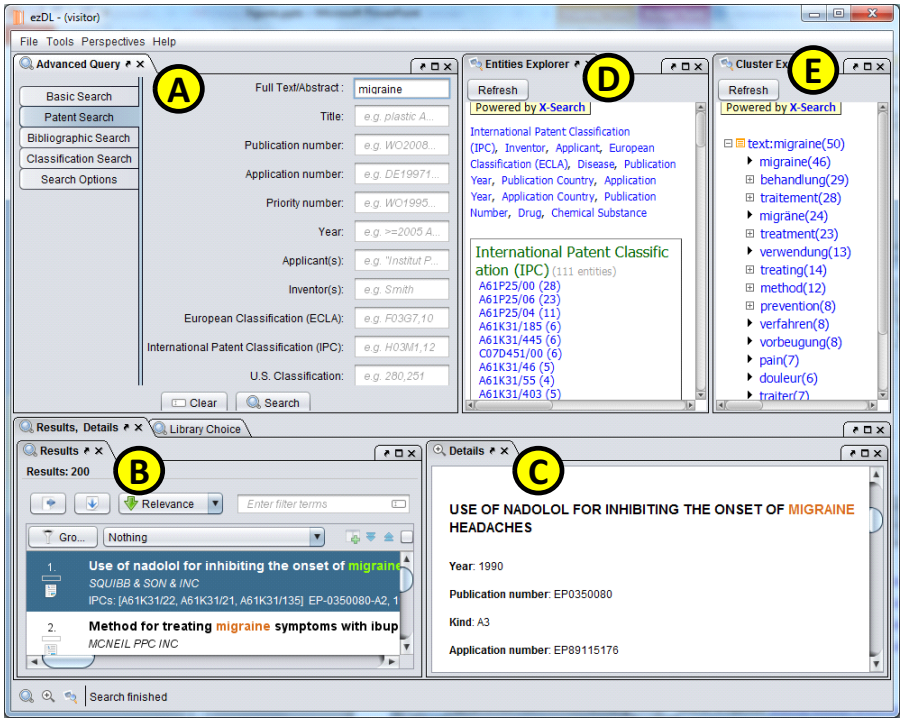

Fig. 14. The `iPerFedPat` system

essentially offers an *entity-based integration* of search results, metadata and other external (semantic) resources.

In particular, the described approach offers the ability to a) restrict the focus using static metadata values that are important for the searchers, b) restrict the focus using entity values and important topics (clusters) that were discovered in the search results, c) inspect and explore the properties of the identified entities by exploiting KBs that are accessible through SPARQL endpoints. Furthermore, showing values and their count gives an overview (e.g. percentage of patents published in a particular country). Note also that the described functionality can be exploited by any professional search system.

Experimental results have showed that this functionality can be efficiently offered at real-time however the time that we have to pay is proportional to the number of the top results that we want to "explore" (pay-as-you-go). Furthermore, the time for semantically exploring the identified entities highly depends on the efficiency and reliability of the underlying KBs. Ultimately, we should stress that the post-analysis services that we described can considerably reduce the time that a user must devote in a professional search context.

The long term vision is to be able to mine not only correct entities but probably entire *conceptual models* that describe and relate the identified entities (plus other external entities) and are appropriate for the context of the user's information need. After reaching that objective the exploratory process could

support the interaction paradigm of faceted search over such (crispy or fuzzy) semantic models, e.g. [30] for plain RDF/S, or [41] for the case Fuzzy RDF.

Acknowledgement

This work was supported by *iMarine* (FP7 Research Infrastructures, 2011-2014) and *MUMIA* COST action (IC1002, 2010-2014).

References

1. Alchemyapi. `http://www.alchemyapi.com/`.
2. Lupedia enrichment service, ontotext. `http://lupedia.ontotext.com/`.
3. Opencalais, thomson reuters. `http://www.opencalais.com/`.
4. Opensearch. `http://www.opensearch.org/`.
5. Sparql 1.1 federated query, w3c recommendation, 21 march 2013. `http://www.w3.org/TR/sparql11-federated-query/`.
6. Sparql query language for rdf, w3c recommendation, 15 january 2008. `http://www.w3.org/TR/rdf-sparql-query/`.
7. Wikimeta. `http://www.wikimeta.com/`.
8. C. Allocca, M. dAquin, and E. Motta. Impact of using relationships between ontologies to enhance the ontology search results. In *The Semantic Web: Research and Applications*, pages 453–468. Springer, 2012.
9. O. Ambrus, K. Möller, and S. Handschuh. Konduit vqb: a visual query builder for sparql on the social semantic desktop. In *Workshop on Visual Interfaces to the Social and Semantic Web*, 2010.
10. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.
11. H. Bast, A. Chitea, F. Suchanek, and I. Weber. Ester: efficient search on text, entities, and relations. In *30th international ACM SIGIR conference on research and development in information retrieval*, 2007.
12. T. Beckers, S. Dungs, N. Fuhr, M. Jordan, S. Kriewel, and V. T. Tran. ezdl: An interactive search and evaluation system. In *SIGIR 2012 Workshop on Open Source Information Retrieval*, pages 9–16, 2012.
13. B. Bishop, A. Kiryakov, D. Ognyanov, I. Peikov, Z. Tashev, and R. Velkov. Factforge: A fast track to the web of data. *Semantic Web*, 2(2):157–166, 2011.
14. C. Bizer, T. Heath, and T. Berners-Lee. Linked data-the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22, 2009.
15. D. Bonino, A. Ciaramella, and F. Corno. Review of the state-of-the-art in patent information and forthcoming evolutions in intelligent patent informatics. *World Patent Information*, 32(1), 2010.
16. K. Bontcheva, V. Tablan, D. Maynard, and H. Cunningham. Evolving gate to meet new challenges in language engineering. *Natural Language Engineering*, 10(3-4):349–373, 2004.
17. L. Candela, D. Castelli, and P. Pagano. gcube: a service-oriented application framework on the grid. *ERCIM News*, 72:48–49, 2008.
18. V. Cardellini, M. Colajanni, and P. S. Yu. Dynamic load balancing on web-server systems. *Internet Computing, IEEE*, 3(3):28–39, 1999.

19. C. Carpineto, M. DAmico, and G. Romano. Evaluating subtopic retrieval methods: Clustering versus diversification of search results. *Information Processing and Management*, 48(2):358–373, 2012.
20. E. Charton, M. Gagnon, and B. Ozell. Automatic semantic web annotation of named entities. In *Advances in Artificial Intelligence*, pages 74–85. Springer, 2011.
21. T. Cheng and K. Chang. Beyond pages: supporting efficient, scalable entity search with dual-inversion index. In *13th International Conference on Extending Database Technology*, 2010.
22. H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, 2002.
23. J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.
24. M. T. Enrico Franconi, Paolo Guagliardo. Quelo: a nl-based intelligent query interface. In *Procs of the Second Workshop on Controlled Natural Languages (CNL 2010)*, 2010.
25. P. Fafalios, I. Kitsos, Y. Marketakis, C. Baldassarre, M. Salampasis, and Y. Tzitzikas. Web searching with entity mining at query time. In *Proceedings of the 5th Information Retrieval Facility Conference*, Vienna, Austria, July 2012.
26. P. Fafalios, I. Kitsos, and Y. Tzitzikas. Scalable, flexible and generic instant overview search. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 333–336. ACM, 2012.
27. P. Fafalios, M. Salampasis, and Y. Tzitzikas. Exploratory patent search with faceted search and configurable entity mining. In *1st International Workshop on Integrating IR technologies for Professional Search (ECIR'13 Workshop)*, 2013.
28. P. Fafalios and Y. Tzitzikas. Exploiting available memory and disk for scalable instant overview search. *Web Information System Engineering–WISE 2011*, pages 101–115, 2011.
29. P. Fafalios and Y. Tzitzikas. Post-analysis of keyword-based search results using entity mining, linked data and link analysis at query time. In *2014 IEEE Eighth International Conference on Semantic Computing (ICSC 2014)*, Newport Beach, California, USA, June 16-18 2014. IEEE.
30. S. Ferré and A. Hermann. Semantic search: reconciling expressive querying and exploratory search. *The Semantic Web–ISWC 2011*, pages 177–192, 2011.
31. R. Froese and D. Pauly. Fishbase. `http://www.fishbase.org/`.
32. A. Y. Halevy. Answering queries using views: A survey. *The VLDB Journa–The International Journal on Very Large Data Bases*, 10(4):270–294, 2001.
33. E. Jiménez-Ruiz, B. C. Grau, I. Horrocks, and R. Berlanga. Ontology integration using mappings: Towards getting the right logical consequences. In *The Semantic Web: Research and Applications*, pages 173–187. Springer, 2009.
34. H. Joho, L. Azzopardi, and W. Vanderbauwhede. A survey of patent users: an analysis of tasks, behavior, search functionality and system requirements. In *Procs of the 3rd symposium on Information interaction in context*. ACM, 2010.
35. M. Käki. Findex: search result categories help users when document ranking fails. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2005.
36. M. Käki and A. Aula. Findex: improving search result use through automatic filtering categories. *Interacting with Computers*, 17(2):187–206, 2005.

37. I. Kitsos, K. Magoutis, and Y. Tzitzikas. Scalable entity-based summarization of web search results using mapreduce. *Distributed and Parallel Databases*, pages 1–42, 2013.
38. A. Kohn, F. Bry, A. Manta, and D. Ifenthaler. *Professional Search: Requirements, Prototype and Preliminary Experience Report*, pages 195–202. 2008.
39. S. Kopidaki, P. Papadakos, and Y. Tzitzikas. Stc+ and nm-stc: Two novel online results clustering methods for web searching. *Web Information Systems Engineering*, 2009.
40. B. Kules, R. Capra, M. Banta, and T. Sierra. What do exploratory searchers look at in a faceted search interface? In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pages 313–322. ACM, 2009.
41. N. Manolis and Y. Tzitzikas. Interactive Exploration of Fuzzy RDF Knowledge Bases. *Proceedings of the 8th Extended Semantic Web Conference (ECSW'2011)*, pages 1–16, 2011.
42. G. Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 2006.
43. P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8. ACM, 2011.
44. P. Papadakos, N. Armenatzoglou, S. Kopidaki, and Y. Tzitzikas. On exploiting static and dynamically mined metadata for exploratory web searching. *Knowledge and Information Systems*, 30:493–525, 2012.
45. P. Papadakos, S. Kopidaki, and Y. Armenatzoglou, N.and Tzitzikas. Exploratory web searching with dynamic taxonomies and results clustering. In *Proceedings of the 13th European Conference on Digital Libraries*, September 2009.
46. F. Piroi, M. Lupu, A. Hanbury, and V. Zenz. Clef-ip 2011: Retrieval in the intellectual property domain. In *CLEF (notebook papers/labs/workshop)*, 2011.
47. W. Pratt and L. Fagan. The usefulness of dynamically categorizing search results. *Journal of the American Medical Informatics Association*, 7(6):605–617, 2000.
48. A. Russell, P. R. Smart, D. Braines, and N. R. Shadbolt. Nitelight: A graphical tool for semantic query construction. In *Semantic Web User Interaction Workshop (SWUI 2008)*, April 2008.
49. G. Sacco and Y. Tzitzikas. *Dynamic Taxonomies and Faceted Search: Theory, Practice, and Experience*, volume 25. Springer, 2009.
50. M. Sahami, S. Yusufali, and M. Q. Baldonaldo. Sonia: A service for organizing networked information autonomously. In *Proceedings of the third ACM conference on Digital libraries*, pages 200–209. ACM, 1998.
51. M. Salampasis and A. Hanbury. A generalized framework for integrated professional search systems. In *Multidisciplinary Information Retrieval*, pages 99–110. Springer, 2013.
52. F. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *Procs of the 16th World Wide Web conf.*, pages 697–706, 2007.
53. Y. Tzitzikas, C. Alloca, C. Bekiari, Y. Marketakis, P. Fafalios, M. Doerr, N. Minadakis, T. Patkos, and L. Candela. Integrating heterogeneous and distributed information about marine species through a top level ontology. In *Proceedings of the 7th Metadata and Semantic Research Conference (MTSR'13)*, Thessaloniki, Greece, November 2013.
54. Y. Tzitzikas and C. Meghini. Ostensive automatic schema mapping for taxonomy-based peer-to-peer systems. In *Cooperative Information Agents VII*, pages 78–92. Springer, 2003.

55. Y. Tzitzikas, N. Spyratos, and P. Constantopoulos. Mediators over taxonomy-based information sources. *The VLDB Journal–The International Journal on Very Large Data Bases*, 14(1):112–136, 2005.
56. J. Umbrich, M. Karnstedt, A. Hogan, and J. X. Parreira. Hybrid sparql queries: fresh vs. fast results. In *The Semantic Web–ISWC 2012*, pages 608–624. Springer, 2012.
57. T. Weninger, M. Danilevsky, F. Fumarola, J. Hailpern, J. Han, T. Johnston, S. Kallumadi, H. Kim, Z. Li, D. McCloskey, et al. Winacs: construction and analysis of web-based computer science information networks. In *ACM SIGMOD international conference on Management of Data*, 2011.
58. R. White, B. Kules, S. Drucker, and M. Schraefel. Supporting exploratory search. *Communications of the ACM*, 49(4), 2006.
59. M. Wilson et al. A longitudinal study of exploratory and keyword search. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '08, pages 52–56. ACM, 2008.
60. O. Zamir and O. Etzioni. Web document clustering: A feasibility demonstration. In *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval*. ACM, 1998.