This is a preprint of an article accepted for publication in: IEEE 8th International Conference on Semantic Computing (ICSC'14), Newport Beach, California, USA, June 2014 (http://ieee-icsc.org/icsc2014/).

Post-Analysis of Keyword-based Search Results using Entity Mining, Linked Data and Link Analysis at Query Time

Pavlos Fafalios and Yannis Tzitzikas
Institute of Computer Science, FORTH-ICS, GREECE, and
Computer Science Department, University of Crete, GREECE
{fafalios, tzitzik}@ics.forth.gr

***Abstract*—The integration of the classical Web (of documents) with the emerging Web of Data is a challenging vision. In this paper we focus on an integration approach during *searching* which aims at enriching the responses of non-semantic search systems (e.g. professional search systems, web search engines) with semantic information, i.e. Linked Open Data (LOD), and exploiting the outcome for providing an overview of the search space and allowing the users (apart from restricting it) to explore the related LOD. We use named entities (e.g. persons, locations, etc.) as the "glue" for automatically connecting search hits with LOD. We consider a scenario where this entity-based integration is performed at query time with no human effort, and no a-priori indexing, which is beneficial in terms of configurability and freshness. To realize this scenario one has to tackle various challenges. One spiny issue is that the number of identified entities can be high, the same is true for the semantic information about these entities that can be fetched from the available LOD (i.e. their properties and associations with other entities). To this end, in this paper we propose a Link Analysis-based method which is used for (a) ranking (and thus selecting to show) the more important semantic information related to the search results, (b) deriving and showing top-K semantic graphs. In the sequel, we report the results of a survey regarding the marine domain with promising results, and comparative results that illustrate the effectiveness of the proposed (PageRank-based) ranking scheme. Finally, we report experimental results regarding efficiency showing that the proposed functionality can be offered even at query time.**

I. Introduction

The Web has evolved from an information space of interconnected web pages to one where both unstructured documents and structured data in various forms coexist. An important question is how typical web users, who mainly use keywords in searching, can access and exploit this increasing body of knowledge. In addition, most search methods are appropriate for *focalized search*, i.e. they make the assumption that users can accurately describe their information need using a small sequence of words and that they are interested only in the top hits. However, a high percentage of search tasks are *exploratory* and focalized search very commonly leads to inadequate interactions and poor results [1]. Our objective is to enable effective exploratory search services which can bridge the gap between the responses of non semantic search systems (e.g. professional search systems, web search engines) and semantic information, i.e. Linked Open Data (LOD) [2]. An important observation is that *entity names* (e.g. persons, locations, organizations, etc.) occur in all kinds of artifacts: documents, database cells, RDF triples, etc. Therefore, a basic hypothesis that we investigate is whether and how we can exploit named entities for offering a kind of *entity-based integration* method; the *named entities* are used as the "glue" for automatically connecting documents (i.e. search results) with data and knowledge. For being configurable, and for tackling the constant evolution of published LOD, we investigate a scenario where these services are provided as meta-services, and the entity-based integration is performed at *query time* with no human effort, making the LOD accessible to the end users.

Consider the following scenario from the *marine* domain[1]: a biologist seeks information about *marine species* and submits to a professional search system a query for requesting information about a particular fish species. Past systems that provide a kind of semantic enrichment of search results (like [3], [4]) present to the user only the detected entities (e.g. several species identified in the search results) allowing the user to narrow the search space to a set of results that contain a particular species. However, the structured knowledge that is available for these entities is not exploited. For instance, an identified 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

[1]Which is a real scenario related to the iMarine project (http://www.i-marine.eu/).

sophisticated overview and to make better sense of the results.

However, the number of identified entities can be high and the amount of structured information that is available for these entities can be very high too (i.e. their associations, properties and categories[2]). 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 entities, associations and properties.

To tackle the above challenges, in this paper, we propose a method founded on Link Analysis. Specifically, we introduce 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 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. We report the results of a survey and of a comparative evaluation with other ranking methods that demonstrate the usefulness and the effectiveness of this approach. We also report experimental results that support the feasibility of this approach. For example and regarding the marine domain, by analyzing the snippets of the top-100 results that Bing web search engine returns for the query *yellowfin tuna* (with *fish species* as the entities of interest and exploiting DBpedia[3] at real-time), in the top semantic graphs we get 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. We get all this information in only 3 seconds without performing any additional query. Figure 1 depicts an example of a top-5 semantic graph.[4]

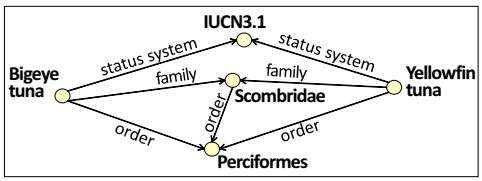

Fig. 1: A top-5 semantic graph

In a nutshell, the key contributions are the following:

- We introduce a novel search paradigm in which the search results are connected with data and knowledge at query time with no human effort (Section II).
- We present and detail a biased PageRank algorithm for ranking entities and properties which can identify and promote the important semantic information (Section IV).
- We present the results of a survey regarding the marine domain which demonstrate the usefulness of the proposed approach (Section V-A).
- We report the results of a comparative evaluation with other ranking methods that illustrate the effectiveness of the proposed ranking scheme (Section V-B).
- We report experimental results that reveal the applicability and the efficiency of the proposed approach, and we discuss how we can achieve scalability (Section V-C).

II. Context

We consider the following process:

(1) The user submits a keyword query to a search system (e.g. to a professional search system or to a web search engine).
(2) The search system uses a component, we call it `SPP` from "Semantic Post Processor", which exploits a named entity recognition tool (in our prototype Gate Annie[5]) for identifying entities in the (top) search results (either in their textual snippets or in their full contents or in their metadata fields). For configuring the *entities of interest* (in a preprocessing step), we exploit the LOD, i.e. we can define that the entities of interest are the names of entities returned by a given SPARQL query or those that belong to a particular RDF class (thereby each entity is accompanied by its URI). Specifically, we have automated the procedure of creating a new supported category of entities in Gate Annie (as well of updating a category), i.e. we can easily configure and update the entity names that are interesting for the application at hand.
(3) `SPP` exploits the LOD for getting *more* information about the entities (their properties and related entities). For instance, by running SPARQL queries we can retrieve all the incoming and outgoing properties of each entity URI.
(4) For the identification of the most *important* entities and properties, a PageRank-like ranking scheme is used (it is analyzed in detail in Section IV).
(5) Apart from returning to the user the top-K entities as derived by the PageRank-like algorithm, we can return the top-K semantic graph, allowing the user to gradually increase or reduce the value of K. This is very important for showing how the entities are connected. This functionality can be also offered *on-demand* as a complementary representation of the identified entities. Several user actions could be supported over this graph. For example, the user can inspect how two entities are associated, explore the properties of a particular entity, narrow the search space by selecting an entity, etc. $\diamond$

We should stress that the above process is *fully configurable*. The user/admininstrator can configure the entities of interest (i.e. the categories of entities for which the system can identify entities) and the underlying Knowledge Bases (accessible through SPARQL endpoints) that are used for retrieving more information about the identified entities. Thereby, one can configure it for different domains. For example, for the *marine* domain, the useful categories include *fish species*, *countries*, *water areas*, *ecosystems*, etc., while for the *medical* domain

[2]We call *association* a relationship between two entities, *property* an attribute of an entity, and *category* a class (or classification) of that entity.

[3]http://dbpedia.org/

[4]A proof-of-concept prototype (configured for the marine domain) is available at http://139.91.183.72/x-ens-2/

[5]http:/gate.ac.uk/ie/annie.html

drugs, *diseases*, *proteins*, etc. are interesting categories. As regards the underlying Knowledge Bases, the LOD cloud contains numerous datasets covering many domains. For example, GeoNames[6] can be exploited for *geographic data*, DrugBank[7] for *drugs*, DBpedia contains data related to many domains, etc. For reasons of homogeneity, all the examples in this paper concern the *marine* domain and we consider *fish species* (from DBpedia) as the entities of interest.

III. Related Work

Semantic Post-Processing of Search Results. [3], [4] presented a method to enrich the classical (keyword-based) web searching with entity mining that is performed at query time over the *snippets* of the search hits. The results of entity mining (entities grouped in categories) complement the query answer and can be further exploited in a faceted and session-based interaction scheme. In comparison to our work, the aforementioned category of works does not exploit the structured knowledge that is available for these entities, i.e. their *properties* and the *associations* with other entities.

Keyword queries with entity-based markup. Works like [5], [6] propose frameworks for entity search in which users formulate queries that directly describe what types of entities they are looking for (using the prefix #, e.g. `#professor`). Again, the structured information about entities is not exploited.

Link Analysis for Entity Search. There are several works that exploit *link analysis-based* methods for *ranking* the results of an *entity search* process. [7] and [8] combine classical search techniques and *spread activation* techniques for ranking keyword search results, while [9] and [10] propose a *PageRank-like* method for ranking RDF resources and take into account the *data sources*. [11] adopts a modification of *Kleinberg's HITS algorithm* [12] for estimating the importance of RDF resources, [13] uses *PageRank* and *HITS* as features for ranking query-independent resources, while [14] applies link analysis methods based on a *rational surfer model* for ranking the importance of RDF documents. Finally, a most recent work [15] elaborates on *entity-relationship* queries and exploits the idea of *spread activation* for scoring the answers. However, the above works focus on retrieving and ranking resources from a semantic collection, and users get as output directly entity resources that match the query, not documents, therefore such works are quite distant from the way users search for information.

Exploiting Semantic Data in Web Search. *Google Knowledge Graph*[8] (GKG) also evidences the increasing interest of exploiting semantic data in Web searching. GKG tries to understand the submitted query and presents a semantic description (in a right sidebar) of *one* entity, the entity that the user is maybe looking for. However, for a bit more complex queries the user does not get any semantic information. For instance (and for the time being), for the query *"Barack Obama and Honolulu"*, GKG does not return any semantic information, although *Honolulu* is the birth place of *Barack Obama*, i.e. the two entities are highly connected.

Synopsis. The approach that we propose does not change the (user-friendly) way users search for information, but acts as a *mediator* between any search system and semantic information (LOD); users still get documents as search results, but also get and interact with semantic information that is highly related to the results (providing a way of making the LOD accessible to the end-users). In addition, the derived *semantic graphs* show how the entities are connected and their context, and deter the disengagement of the users from their initial task (since users can instantly inspect semantic information that may be laborious and time consuming to detect).

IV. A Link Analysis-based Approach

We focus on the problem of selecting and ranking the identified entities and their related structured information, i.e. on steps 2, 3 and 4 of the process described in Section II. The main idea is to construct *dynamically* an RDF graph about the identified entities, and then to analyze it probabilistically. Below we describe the approach in detail, by defining the required notions and notations.

Search Results and Identified Entities. For a given keywords query submitted by the user, let A be the set of the top-L hits (e.g. $L = 200$) returned by the underlying search system. For a hit $a \in A$, let $ent(a)$ denote the set of entities that have been identified in a by applying entity mining (e.g. over its snippet, its full content or over some of its metadata fields). Inversely, let $docs(e) = \{\ a \in A \mid e \in ent(a)\}$ denote the elements of A in which e has been identified. Let $E = \cup_{a \in A} ent(a)$, i.e. E is the set of all entities identified in A.

The SEGIE. Here we describe how to enrich the set of entities E by exploiting the structured knowledge enclosed in one or more RDF graphs, in order to construct what we call SEGIE (Semantically Enriched Graph of Identified Entities), which is an RDF graph, denoted by $\mathcal{X}$.

Let us first formalize the structured knowledge available as LOD or queryable through a SPARQL endpoint. Consider an infinite set U of RDF URI references, an infinite set B of blank nodes[9] and an infinite set L of literals. A triple $(s, p, o) \in (U \cup B) \times U \times (U \cup B \cup L)$ is called an *RDF triple* (s is called the *subject*, p the *predicate* and o the *object*). An *RDF graph* G, is a set of RDF triples. For an RDF Graph G_i we shall use U_i, B_i, L_i to denote the URIs, blank nodes and literals that appear in the triples of G_i respectively. The *nodes* of G_i are the values that appear as subjects or objects in the triples of G_i. Figure 2(a) depicts a simple RDF graph; node 2 (_:b) represents a blank node, node 4 (*Thunnus albacares@en*) is a literal (specifically a string in English), while the other nodes represent URIs (for improving readability we have omitted the namespaces).

Now we describe how from E and an RDF Graph G_i, we will define the SEGIE $\mathcal{X}$. Let $out(e) = \{o \mid (e, p, o) \in G_i\}$, i.e. all objects that are *pointed by* an entity $e \in E$, and $in(e) =$

[6] http://www.geonames.org/

[7] http://www.drugbank.ca/

[8] http://www.google.com/insidesearch/features/search/knowledge.html

[9] http://www.w3.org/2005/rules/wg/wiki/bNode_Semantics.html

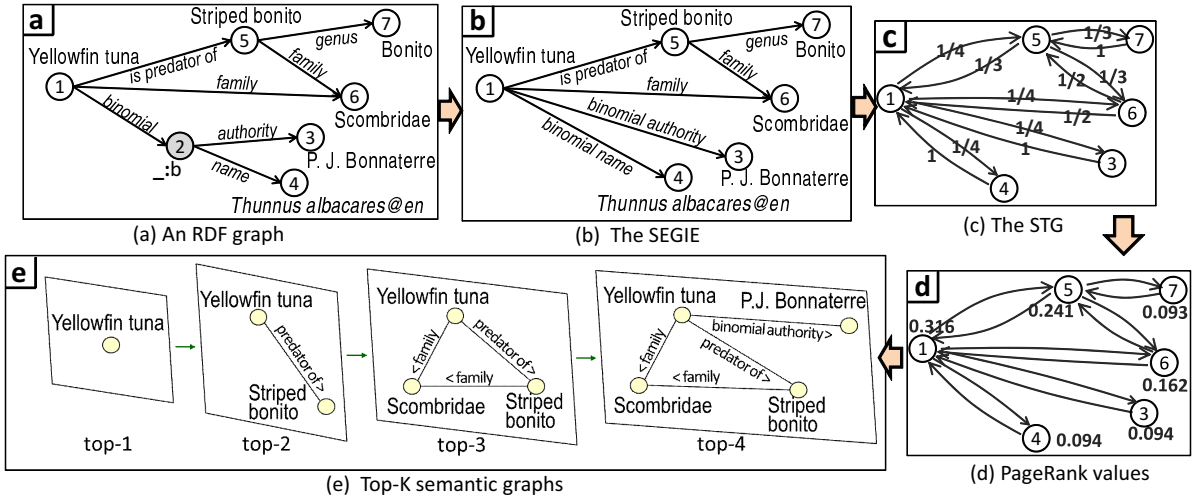

Fig. 2

$\{s \mid (s,p,e) \in G_i\}$, i.e all subjects that *point to* an entity $e \in E$ (for simplicity, we consider subjects and objects as entities in our setting). We first define the *base set* S as follows $S = E \cup (\cup\{out(e)|e \in E\}) \cup (\cup\{in(e)|e \in E\})$. Note that one could also add to S the sets $out(e')$ and $in(e')$, for each $e' \in out(e) \cup in(e)$, and so on, i.e. subjects and objects in any *radius*. For the moment, we consider radius equal to 1. For example, in Figure 2(a), we can consider that the nodes 1 and 5 correspond to entities detected in the search results (i.e. entities in E), while the remaining nodes correspond to semantic information derived by exploiting a Knowledge Base (i.e. an RDF Graph G_i).

If the object o (or subject s) is a *blank node* b, and in order to avoid losing information that may be important, we include in the graph the set $out(b)$ (or $in(b)$ correspondingly) and not the blank node b. In that case, and for labeling the edge that connects an entity e with an entity e' in $out(b)$ (or in $in(b)$ correspondingly), we *concatenate* the names of the properties. For example, if e *birth place* b and b *city* e', then the name of the edge is "*birth place city*". Figure 2(b) depicts the RDF graph of Figure 2(a) without blank nodes (the blank node "_:b" has been removed and the incident edges have been merged). The SEGIE $\mathcal{X} = (\mathcal{E}_\mathcal{X}, \mathcal{P}_\mathcal{X})$, where $\mathcal{E}_\mathcal{X} = S$ and $\mathcal{P}_\mathcal{X}$ is the directed links that connect S, of our running example is shown in Figure 2(b).

The next step is to apply on SEGIE a PageRank-like [16] algorithm, for identifying the *more important* entities. We prefer to follow a PageRank-inspired method because the underlying theoretical framework is solid (random walks and stochastic processes) and it can be customized (biased) according to the needs of different types of applications (as it will be shown later). The intuition behind PageRank (which was proposed and has been successful in web search), is that the *important* web pages are pointed by several other *important* web pages. Analogously, in our problem an entity is considered *important* (and thus it is worth presenting to the user) if several other *important* entities point to it. This is evident from the following example: suppose that we submit a query to a search system regarding the marine domain. SPP analyzes the snippets of the returned results and identifies five *fish species* on them. Now, by exploiting the LOD we find out that four of them belong to the same *genus*. However, that particular *genus* does not occur in the snippets of the results and thus has not been identified. Nevertheless, this node is useful for showing how the four fish species are related, and consequently it is reasonable to expect that a ranking method should rank it highly. Below we detail the probabilistic analysis.

The State Transition Graph (STG). We will define a STG $\mathcal{G} = (\mathcal{E}, \mathcal{P})$ over which a *random walk model* can be applied, i.e. its nodes $\mathcal{E}$ correspond to *states* and its edges $\mathcal{P}$ to (state) *transitions*. From each node in $\mathcal{X}$, we create a node in $\mathcal{G}$. For each directed edge $(e_1 \rightarrow e_2)$ in $\mathcal{X}$, we create *two* directed edges in $\mathcal{G}$; one of the same direction $(e_1 \rightarrow e_2)$ and one of the opposite direction $(e_2 \rightarrow e_1)$. We do that because we consider that if a property connects two entities, then the two entities are *semantically biconnected*. For example, in Figure 2(a) the entities *Yellowfin tuna* and *Scombridae* are connected by the relation *family*, meaning that *Yellowfin tuna "belongs to the family" Scombridae* and equivalently that *Scombridae "is the family of" Yellowfin tuna*, i.e. the difference lies in how we name the property. In addition, two nodes in $\mathcal{X}$ may be connected with multiple directed edges. In that case, in $\mathcal{G}$ we collapse the common directed edges in one directed edge, but we also specify accordingly the *edge weights*. Note that the weights of the outgoing edges from a given entity must represent transition probabilities, i.e. they must sum to 1. For a given entity $e \in \mathcal{E}$, let $o(e) \subseteq \mathcal{P}$ be the set of outgoing (directed) edges of e, and $i(e) \subseteq \mathcal{P}$ the set of incoming (directed) edges. Let also $props(e, e') \subseteq o(e)$ be the set of (directed) edges that connect e with e' (i.e. the properties that connect the two entities). Note that in our setting $|o(e)| = |i(e)|$ and $|props(e, e')| = |props(e', e)|$. The weight of the single outgoing edge that connects e with e' is $\frac{|props(e,e')|}{|o(e)|}$. Figure 2(c) illustrates the STG that corresponds to the SEGIE of Figure 2(b) and also shows the edge weights as described above.

Analyzing the STG. Here we describe how we analyze the above graph for identifying the important entities. The PageRank-like value $r(e)$ of an entity e is defined as:

$$r(e) = q \cdot Jump(e) + (1-q) \cdot \sum_{e' \in i(e)} \frac{|props(e', e)|}{|o(e')|} r(e') \quad (1)$$

where q is a decay factor (typically set to 0.1 - 0.2), while $Jump(e)$ expresses the probability of random jumps to e, and thus it can be defined as $Jump(e) = \frac{1}{|S|}$ if we assume uniform distribution (the initial PageRank value of each entity also equals $\frac{1}{|S|}$).

Note that the value of an entity e is the sum of two components: one part of the value is equal for all entities (and expresses the probability of a random jump to e), and the other part comes from entities that point to e. The values can be computed iteratively and iterations should be run to convergence. According to [16], the number of iterations required for convergence is empirically O(log$n$), where n is the number of links (i.e. edges).

Figure 2(d) shows the PageRank values regarding the STG of Figure 2(c) (with decay factor 0.15 and performing 10 iterations). The nodes 1 (*Yellowfin tuna*) and 5 (*Striped bonito*) have the highest PageRank values because they have many connections and are also interconnected. On the contrary, the node 7 (*Bonito*) has the lowest value.

Biased Jumps for Promoting the Entities of the Top-ranked Hits. So far we have ignored the rank of the hits in which an entity occurred. However, it is reasonable to consider that the top results in the ranked list of A will probably contain more useful entities than the last results. To capture this, here we introduce a *biased* version of the scoring scheme. Instead of assuming a uniform distribution for the random jumps, we will now bias it. Specifically:

$$Jump(e) = \frac{HitScore(e)}{\sum_{e' \in E} HitScore(e')} \quad (2)$$

where (as in [3]):

$$HitScore(e) = \sum_{a \in docs(e)} ((|A| + 1) - rank(a)) \quad (3)$$

where $rank(a)$ stands for the position of an $a \in A$ in the answer (the first hit has rank equal to 1, the second 2, etc). This means that the probability of a random jump to e is higher if e has been identified in the top ranked documents. Also, the probability of a random jump to an entity that has not been identified in the search results is zero.

To grasp the effect of the biased approach, in the example of Figure 2(b) consider that we have performed entity mining in the top-10 results returned by a search system and got the following results: *Striped bonito (node 5) was detected in the 1st, 2nd and 3rd result, Bonito (node 7) was detected in the 1st and 3rd result, Yellowfin tuna (node 1) was detected in the 8th result only, while P. J. Bonnaterre (node 3), Thunnus albacares (node 4) and Scombridae (node 6) were not detected in the top-10 results (but derived by exploiting the LOD).* By running the biased version of PageRank as described above (with decay factor 0.15 and performing 10 iterations), we get the following values:

Striped bonito ≈ 0.331 ($\uparrow$ 1), *Yellowfin tuna* ≈ 0.260 ($\downarrow$ 1), *Bonito* ≈ 0.150 ($\uparrow$ 2), *Scombridae* ≈ 0.149 ($\downarrow$ 1), *P. J. Bonnaterre* and *Thunnus albacares* ≈ 0.055 ($\downarrow$ 1)

Now the entity with the highest value is *Striped bonito*, *Bonito* has gained two ranks, while *Yellowfin tuna*, *Scombridae*, *P. J. Bonnaterre* and *Thunnus albacares* have lost one rank. We notice that the entities identified in the top search results have been promoted.

One could exploit the biased version for supporting also various other kinds of *personalization*, e.g. promotion of entities of one or more particular categories (RDF classes) or those coming from particular Knowledge Bases, etc.

Top-K Graphs. Apart from producing and returning the top-K entities, say $\mathcal{E}_\mathcal{K}$ ($\mathcal{E}_\mathcal{K} \subseteq \mathcal{E}$), as derived by the biased PageRank algorithm, we can return (at query time or on-demand) the *top-K graph* $\mathcal{G}_\mathcal{K} = (\mathcal{E}_\mathcal{K}, \mathcal{P}_\mathcal{K})$ for any K from 1 to $|\mathcal{E}_\mathcal{K}|$, allowing the user to increase or reduce the value of K. The set of edges $\mathcal{P}_\mathcal{K}$ of this graph consists of those elements of $\mathcal{P}$ that connect elements of $\mathcal{E}_\mathcal{K}$, i.e. it is the *restriction* of $\mathcal{P}$ on $\mathcal{E}_\mathcal{K}$, i.e. we can write $\mathcal{P}_\mathcal{K} = P_{|\mathcal{E}_\mathcal{K}}$. This is very important for showing how the entities are connected. Several user actions could be supported over this graph. Figure 2(e) depicts several top-K semantic graphs of our running example.

V. Evaluation

A. Usefulness

In order to get a first feedback for the usefulness of the proposed approach, we performed a survey regarding the *marine* domain. The objective is to study whether the depiction of associations among the derived semantic information (through a semantic top-K graph related to the search results) can help the users in an exploratory search process.

The survey is based on a questionnaire (Google Form) in which we ask participants related to the marine domain to answer a few questions related to five particular queries (each query corresponds to a different *query type*). At first, for each query we derive the top-5 semantic information (i.e. entities and properties) by applying the proposed approach (described in Section IV), specifically by performing entity mining in the top-100 snippets as returned by Bing search engine, with *Fish Species* as the entities of interest and using DBpedia as the underlying Knowledge Base. Then, we depict this semantic information in two different ways: as a *top-5 list* (as proposed in [3]) and as *top-5 graph*. We select to show the top-5 list and graph (and not for example the top-10 or the top-20) because we do not want the quality of the visualization of the graph to affect the participants' opinion (since this is not the focus of this paper). Then, the participant must answer the following question:

Q1. *"In an exploratory search process regarding the query* `<here the query>`*, how would you prefer to see the Top-5 entities and properties related to that query?"*.

The participant can select one of the following options: *Only the LIST is enough*, *Only the GRAPH is enough*, *I would like to see BOTH*, *I do not want to see neither the list nor the graph*. The participant must answer the above question for

TABLE I: Results of Q1

QUERY	ONLY LIST	ONLY GRAPH	BOTH	NO LIST, NO GRAPH
yellowfin tuna	23%	30%	43%	3%
jack fishes	13%	37%	47%	3%
chum salmon genus	17%	37%	43%	3%
zander and walleye	13%	43%	40%	3%
fishing in Hawaii	23%	37%	30%	10%

TABLE II: Results of Q2

QUERY	YES	MAYBE YES	MAYBE NO	NO
yellowfin tuna	23%	63%	10%	3%
jack fishes	27%	67%	7%	0%
chum salmon genus	33%	57%	7%	3%
zander and walleye	33%	53%	13%	0%
fishing in Hawaii	30%	43%	23%	3%

each one of the five queries. In the next step (in a new page), we ask the participant to answer the following question (again for each one of the five queries):

Q2. *"In an exploratory search process regarding the query* `<here the query>`*, do you believe that the appearance of a graph of semantic information related to the search results can help the user during his/her search process?"*

The participant can select one of the following options: *Yes*, *Maybe Yes - it depends on the interaction model and the quality of the visualization of the graph*, *Maybe No*, *No*.

Clearly, the type of result expected depends on the *type* of the query. For example, a query such as *tuna species* is looking for instances of a class of entities, while a query like *yellowfin tuna* is looking for information for one particular entity, in this case a certain tuna species. Pound et al. [17] proposed a classification of queries from a semantic search point of view by expected result:

- *Entity query:* the intention of the query is to find information about a particular entity.
- *Type query:* the intention of the query is to find entities of a particular type or class.
- *Attribute query:* the intention of the query is to find values of a particular attribute of an entity or type.
- *Relation query:* the intention of the query is to find how two or more entities or types are related.
- *Other keyword query:* the intention of the query is described by some keywords that do not fit into any of the above categories.

The existence of these query types is very important for our problem, since each type requires a different type of result, and thus must be evaluated differently by the human judge. Thereby, the participants must answer the aforementioned two questions for five different queries, each one belonging to a different type. Specifically, we selected the following queries: *yellowfin tuna* (entity query), *jack fishes* (type query), *chum salmon genus* (attribute query), *zander and walleye* (relation query), *fishing in Hawaii* (other keyword query).

We distributed the questionnaire to marine biologists and to persons working on marine-related projects (who have a basic knowledge on marine species). In the top of the questionnaires, we also included a brief description of the proposed functionality.

Results. 30 subjects participated in the user study (22 to 60 years old), from 6 countries and 12 organizations. Table I depicts the results of Q1 and Table II of Q2.

As regards Q1, we notice that the majority of the participants (67% to 84%) would like to see a graph representation (ONLY GRAPH or BOTH) of the top-5 entities. As expected, the biggest percentage is for the query *zander and walleye*, which is a relation query, and the smallest is for the query *fishing in Hawaii* which does not belong to a particular type. In addition, we notice that for the first three types of queries (*entity*, *type* and *attribute*), more participants prefer to see both a list and a graph, which means that the graph representation can be offered *on-demand* as a complementary representation which enables users to inspect the connectivity of the identified entities.

Regarding Q2, we notice that the majority of the participants (73% to 94%) believe that the appearance of a graph of semantic information related to the search results can help users during a search process (YES or MAYBE YES). In addition, most of them consider that the success of this visualization approach depends on the interaction model and the quality of the visualization of the graph. This is a strong rationale for elaborating in the future on the interaction model and the quality of the top-K graphs.

B. Effectiveness: Comparative Results on Ranking

We performed a user study regarding the marine domain. The objective is to evaluate the effectiveness of the proposed (PageRank-based) ranking scheme (described in Section IV). Specifically, we comparatively evaluated the proposed Biased PageRank algorithm (`BiPR`) with i) the plain PageRank algorithm (`PR`), and ii) a Spreading Activation [18] algorithm (`SA`). Regarding `SA`, we adopted a similar approach with [7] and [15]. However, we set the initial activation of a node that represents an identified entity to be the score of this entity as derived by Formula 2, while the remaining nodes have zero activation. In addition, we set the decay factor α to equal 0.85 which in our setting appears to produce the best results (the decay factor corresponds to the percentage of activation that is lost every time an edge is processed). We also set the *firing threshold* to equal a very small real number (0.00001) because we want all the nodes that represent entities identified in the results to fire even if their ranking score is very small.

We deployed a Web application which implements the proposed functionality. Specifically, the system accepts keyword-based queries and performs entity mining in the top-100 snippets as returned by Bing, with *Fish Species* (from DBpedia) as the entities of interest and using DBpedia for retrieving the properties of the identified entities. We allowed the users to submit their own queries (so that they can better judge the results), and we also stored the results and various statistics for each submitted query (number of identified entities, number of retrieved triples, etc.).

For each submitted query, the system presents three top-10 lists (the one next to the other with random display order) of ranked semantic information (entities and properties) related to the results, each one produced by one of the aforementioned ranking schemes (`BiPR`, `PR` and `SA`). The user can evaluate

the ranking of each list by selecting one of the following options: 1 (poor), 2 (not bad), 3 (good), 4 (very good), 5 (excellent). In addition, the user can inspect many top-K lists, for several values of K ($5 \leq K \leq 50$), in order to better judge each ranking. We also included guidelines and a brief description of the proposed functionality allowing the participants to better understand the context of the evaluation.

Results. 17 subjects performed the evaluation (part of those who completed the survey) and totally 51 queries were submitted. For each submitted query, in average 11.5 entities were detected in the search results, 4,687 triples were derived from DBpedia, and 2,031 entities and properties had to be ranked.

The average score of `BiPR` was $3.53/5.0$ (good to very good), of `PR` $2.47/5.0$ (not bad to good), and of `SA` $2.90/5.0$ (almost good).[10] From these results, we can conclude that promoting the entities identified in the top search results affects positively the ranking of the semantic information, and thus the elements of the top-K graphs. In addition, a PageRank-inspired method appears to produce a better ranking than a Spreading Activation algorithm. Thus, we can also support that the semantic information that is accessible by many (highly ranked) entities is useful for the users.

Discussion. We should stress that there is not any standard evaluation procedure and collection for our purpose. Approaches like the SemSearch Challenge[11] and the SEALS Project[12] (i.e. retrieve resources from the underlying semantic collection regarding a query) cannot be applied in our problem because the proposed approach does not retrieve resources that much the criteria described by a query, but it *semantically post-analyzes* the results of a search system. Thus, there are many parameters that affect the results like the quality of the hits, the quality of the underlying Knowledge Bases, the underlying entity mining algorithm, the categories of entities for which the system identifies entities, etc.

C. Efficiency

Performing real-time entity mining (using Gate Annie) in the top-100 snippets returned by a web search system costs about 1 second [3], [19] (in general the cost is about 10 ms. per snippet). Here we measure the average time for i) creating the SEGIE (with DBpedia as the underlying Knowledge Base), ii) creating the STG, iii) running PageRank, and iv) creating the top-500 graph, for various numbers of randomly selected entities. We run the experiments for entities belonging to 10 randomly selected RDF classes (i.e. categories of entities): *Tennis Player*, *Boxer*, *Country*, *Philosopher*, *Drug*, *Disease*, *Chemical Substance*, *Bacteria*, *Fish*, and *Golf Player*. In a real setting, the randomly selected entities correspond to entities discovered in the search results or entities related to the detected entities (if we consider radius > 1). For achieving accuracy, we repeated the experiments 20 times and we computed the average values.[13]

Regarding i), i.e. the creation of the SEGIE, we decided to access DBpedia at *real-time* (and not to download its datasets and load them in one or more servers) because we want to preserve the dynamic nature of our approach (since the LOD constantly changes and increases). For each entity, DBpedia offers its data (properties, associations and related entities) online in various forms: JSON, XML, triples and N3/Turtle. We access the data in the N3/Turtle form. As regards the objects in the triples that represent literals, we retrieve only those in English language. Although DBpedia offers a SPARQL endpoint, we decided to access its data by directly parsing the N3/Turtle pages because this fits our problem (for each entity we want only its properties, associations and related entities) and because the parsing of pages turned out to be much more efficient than running SPARQL queries (allowing us to run concurrent threads). Table III reports the average values and also includes the main characteristics of the graphs in order to better understand how these characteristics affect the times.

As expected, the most time consuming task is the creation of the SEGIE, since for each entity we access DBpedia at *real-time* and retrieve its related LOD. We notice that for up to 100 detected entities (which is the common case for snippet-mining), the time is in average less that 3 seconds. For bigger number of detected entities the time is higher, e.g. for 1,000 entities about 30 seconds are required. We should stress that this is acceptable in professional search, e.g. persons working in *patent* offices spend many hours for a particular patent search request (the same is true in *bibliographic* and *medical* search).

The rest of the tasks requires around one order of magnitude less time. We can conclude that the proposed functionality can be offered at real-time for about 100 identified entities even if we query an online (publicly available) Knowledge Base (like DBpedia). In addition, as we have already said, this functionality can be also offered *on-demand*, so the user can decide if he/she wants to pay the cost.

Scalability and Reliability. We have seen that for a big number of detected entities the process can be time consuming (although this is often acceptable in professional search). One reasonable (default) policy would be to retrieve LOD (step 3 of the process described in Section II) only for the top-m (e.g. $m = 100$) detected entities as returned by the plain entity mining approach [3]. The top entities as returned by that approach are those that lie in most of the top-ranked results, therefore they are probably the more important. In this way, we can bound the maximum response time.

In addition, from our experience, the existing online Knowledge Bases (like DBpedia) are not reliable since they mainly serve demonstration purposes. The fact that everyone can

[10]The full results, the identified entities and the RDF triples retrieved from DBpedia for each submitted query, and the top-200 rankings as produced by each one of the three ranking algorithms, are available to download through http://139.91.183.72/x-ens-2/fullEvalResults.zip.

[11]https://km.aifb.kit.edu/ws/semsearch11/

[12]http://www.seals-project.eu/

[13]The experiments were carried out using an ordinary laptop with processor Intel Core i5 @ 2.4Ghz CPU, 4GB RAM and running Windows 7 (64 bit). The implementation is in Java 1.7 and for the creation and the management of the graphs we use the Java Universal Network/Graph Framework (JUNG) (http://jung.sourceforge.net/).

TABLE III: Graphs statistics and creation time, and PageRank execution time for several number of (randomly selected) entities.

#entities	SEGIE #vertices	SEGIE #edges	STG #edges	Top-500 Graph #edges	SEGIE creation time	STG creation time	Time for Running PageRank	Top-500 creation time
50	2,573	3,790	7,580	889	1.4 sec	28 ms	194 ms	42 ms
100	4,133	6,193	12,386	1,493	2.9 sec	95 ms	329 ms	68 ms
500	20,743	34,816	69,632	3,471	13 sec	298 ms	1.7 sec	343 ms
1,000	49,954	84,893	169,786	3,411	27 sec	480 ms	3.9 sec	552 ms
10,000	528,815	995,981	1,991,962	3,421	258 sec	8 sec	58 sec	22 sec

query them affects their efficiency and availability. Of course, in a real application the underlying Knowledge Bases may not be publicly available, or a dedicated warehouse can be constructed that will only serve the intended application (like the marineTLO-based warehouse [20]), or a *KnowledgeStore* [21] can be built (in case the underlying corpus is available) that enables the (entity-based) storage, management and retrieval of documents, entities and semantic information. The Knowledge Bases can also be distributed in many servers, so the system can apply a load balancing technique [22] for serving requests. Finally, as proposed in [23], we can 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.

VI. Conclusion

We have proposed a general method for semantic post-processing of search results which is based on entity mining and Linked Data. For selecting the entities that better characterize the search results and their context, we proposed a Link Analysis-based method. The produced top-K semantic graphs allow the users to instantly inspect information that may lie in different places and that may be laborious and time consuming to locate (avoiding thereby the disengagement of the users from their initial task). In addition, they provide useful information about the context of the identified entities and allow the users to get a more sophisticated overview and to make better sense of the results.

We performed a survey regarding the marine domain which showed that the majority of the participants a) would like to see a graph representation of the top-5 entities regardless the type of the submitted query, and b) believe that the appearance of a graph of semantic information related to the search results can help them during an exploratory search process. We also reported comparative results which illustrated that the proposed (PageRank-based) ranking scheme produces more preferred rankings compared to other (link analysis-based) algorithms. As regards efficiency, we have analyzed experimentally the costs of all steps and we have seen that for up to 100 detected entities (which is the case for snippet-mining), we can offer the proposed functionality at real-time even if we query an online Knowledge Base like DBpedia. Finally, we have seen approaches on how we can achieve scalability and reliability.

We believe that the result of this (ongoing) research can provide a general, flexible and adaptive method for enriching search results with structured knowledge in the context of a session-based exploratory search interaction. In future, amongst others, we plan to elaborate on the interaction model and on the visualization of the semantic graphs.

Acknowledgements. We thankfully acknowledge the support of *iMarine* (FP7 Research Infrastructures, 2011-2014) and *MUMIA* COST action (IC1002, 2010-2014). We would also like to thank the members of the following organizations who willingly participated in the user study: *Hellenic Centre for Marine Research, Foundation for Research and Technology - Hellas, University of Crete, University of Carthage, University of Salento, University of Patras, University of Montpellier, TEI of Western Greece, University of the Aegean, Agro-Know Technologies, CNRS-NCSM, EIHP.*

References

[1] G. Marchionini, "Exploratory Search: from Finding to Understanding," *Communications of the ACM*, 2006.
[2] C. Bizer, T. Heath, and T. Berners-Lee, "Linked Data - the Story so far," *Semantic Web and Information Systems*, vol. 5, no. 3, 2009.
[3] P. Fafalios, I. Kitsos, Y. Marketakis, C. Baldassarre, M. Salampasis, and Y. Tzitzikas, "Web Searching with Entity Mining at Query Time," in *5th Information Retrieval Facility Conference*, 2012.
[4] P. Fafalios and Y. Tzitzikas, "X-ENS: Semantic Enrichment of Web Search Results at Real-Time," in *SIGIR'13*, 2013.
[5] T. Cheng and K. Chang, "Beyond Pages: Supporting Efficient, Scalable Entity Search with Dual-Inversion Index," in *EDBT'10*, 2010.
[6] T. Cheng, X. Yan, and K. C.-C. Chang, "EntityRank: Searching Entities Directly and Holistically," in *VLDB'07*, 2007.
[7] C. Rocha, D. Schwabe, and M. Aragao, "A Hybrid Approach for Searching in the Semantic Web," in *WWW'04*, 2004.
[8] M. Ciglan, K. Nørvåg, and L. Hluchỳ, "The SemSets Model for Ad-hoc Semantic List Search," in *WWW'12*, 2012.
[9] A. Harth, S. Kinsella, and S. Decker, "Using Naming Authority to Rank Data and Ontologies for Web Search," in *ISWC'09*, 2009.
[10] R. Delbru, N. Toupikov, M. Catasta, G. Tummarello, and S. Decker, "Hierarchical Link Analysis for Ranking Web Data," in *ESWC'10*, 2010.
[11] B. Bamba and S. Mukherjea, "Utilizing Resource Importance for Ranking Semantic Web Query Results," *Semantic Web and Databases*, 2005.
[12] J. Kleinberg, "Authoritative Sources in a Hyperlinked Environment," *Journal of the ACM (JACM)*, 1999.
[13] L. Dali, B. Fortuna, T. Duc, and D. Mladenic, "Query-Independent Learning to Rank for RDF Entity Search," in *ESWC'12*, 2012.
[14] L. Ding, R. Pan, T. Finin, A. Joshi, Y. Peng, and P. Kolari, "Finding and Ranking Knowledge on the Semantic Web," in *ISWC'05*, 2005.
[15] A. Agrawal, S. Sudarshan, A. Sahoo, A. Sandalwala, and P. Jaiswal, "Entity Ranking and Relationship Queries Using an Extended Graph Model," in *COMAD'12*, 2012.
[16] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," 1999.
[17] J. Pound, P. Mika, and H. Zaragoza, "Ad-hoc Object Retrieval in the Web of Data," in *WWW'10*. ACM, 2010, pp. 771–780.
[18] F. Crestani, "Application of Spreading Activation Techniques in Information Retrieval," *Artificial Intelligence Review*, vol. 11, no. 6, 1997.
[19] 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*, 2013.
[20] 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 *MTSR'13*, Nov. 2013.
[21] F. Corcoglioniti, M. Rospocher, R. Cattoni, B. Magnini, and L. Serafini, "Interlinking Unstructured and Structured Knowledge in an Integrated Framework," in *ICSC'13*. IEEE, 2013.
[22] V. Cardellini, M. Colajanni, and P. S. Yu, "Dynamic Load Balancing on Web-Server Systems," *Internet Computing, IEEE*, vol. 3, no. 3, 1999.
[23] J. Umbrich, M. Karnstedt, A. Hogan, and J. X. Parreira, "Hybrid SPARQL Queries: Fresh vs. Fast Results," in *The Semantic Web–ISWC 2012*. Springer, 2012, pp. 608–624.