Discovery Hub: on-the-fly linked data exploratory search

Nicolas Marie
INRIA Sophia-Antipolis
Alcatel-Lucent Bell Labs
France
nicolas.marie@inria.fr

Fabien Gandon
INRIA Sophia-Antipolis
2004 route des Lucioles
06902 Sophia Antipolis, France
fabien.gandon@inria.fr

Myriam Ribière, Florentin Rodio
Alcatel-Lucent Bell Labs
Route de Villejust
91620, Nozay, France
firstname.lastname@alcatel-lucent.com

ABSTRACT
Exploratory search systems help users learn or investigate a topic. The richness of the linked open data can be used to assist this task. We present a method that selects and ranks linked data resources that are semantically related to the user's interest. The objective is to focus the user's attention on a meaningful subset of highly informative resources. We extended spreading activation to typed graphs and coupled it with a graph sampling technique. The results selection and ranking is performed on–the-fly and doesn't require pre-processing. This allows addressing remote SPARQL endpoints. We describe first implementation on top of DBpedia. It is used by the Discovery Hub exploratory search system to select interesting resources, to support faceted browsing of the results, to provide explanations and to offer redirections to third-party services. Results of a user evaluation conclude the article.

Categories and Subject Descriptors
G.2.2 [**Mathematics of Computing**]: Graph Theory – *Graph algorithms*; E.1 [**Data**]: Data Structures – *Graphs and networks*

General Terms
Algorithms, experimentation

Keywords
semantic web, linked data, DBpedia, semantic spreading activation, exploratory search system, discovery engine

1. INTRODUCTION
Exploratory search is information-seeking in an open-ended context and following an opportunistic, iterative, and multitactical process [27]. [18] makes a distinction between lookup and exploratory search activities. The lookup tasks are performed to satisfy precise information needs (*e.g.* known item search, fact checking). In this case the user's query keywords are well-defined. Exploratory search refers to cognitive consuming search tasks such as learning or topic investigation. In that case the information need is fuzzy: the keywords are a-priori unknown, vague and evolving. The actual search engines are very efficient for lookup queries but less for exploratory search due to their keyword-oriented paradigm. There is a need to complete the existing solutions with systems or functionalities optimized for exploratory search.

Some of these systems make use of formal knowledge sources including the linked open data datasets. Linked data-based approaches involve the use of (1) semantic web models, formalisms (RDF/S,OWL) and published schemas (2) *Linked open data cloud*[1] (LOD) interconnecting public datasets published using this standards. Among all the LOD datasets DBpedia [1] is the most central one in the LOD cloud and results from the extraction of data from Wikipedia[2].

We want to ease the exploration of a topic of interest by suggesting related and informative resources. In this paper we propose a novel method for performing this selection in linked data datasets. It relies on extending semantic spreading activation to the typed graphs of the semantic web formalisms and combining it with a sampling technique to compute result *"on-the-fly" i.e.* without any preprocessing. It allows supporting exploratory search on remote data using their SPARQL endpoints and let users influence the results by specifying their queries or adding serendipity to the process.

The research questions addressed in this paper are the following: (1) in a dense and heterogeneous linked data graph, how to select and rank a meaningful subset of resources related to the user's interest? (2) As linked data are distributed, how to perform this selection on remote LOD sources? (3) On the interface side, how to present and explain the selection to the user?

Section 2 compares related works. Section 3 details our formal extension of spreading activation to typed graphs. Section 4 describes the implementation and introduces the graph sampling technique we couple to the algorithm. Then it details the algorithm behavior over DBpedia and its configuration. The section 5 presents the Discovery Hub application that makes use of the method and presents the results in an interface optimized for exploratory search. Section 6 presents the hypotheses, protocols and the results of a complete user's evaluations.

2. RELATED WORK
In the field of exploratory search, the linked-data based faceted search systems have proven to be efficient: important examples are [8], [9] and [10]. These systems present a view on the data according to the set of facets currently activated. One drawback of these approaches is that it sometimes necessitates complex interaction sequences that are not adapted to casual users [9].

A different or complementary approach is to directly suggest resources that are strongly related to the user's interest using a relatedness or similarity measure. The aim is to ease the exploratory search tasks by focusing user's attention on resources and associations that convey a lot of knowledge. It can also unveil unknown and unexpected results. In the past 3 years several research initiatives demonstrated the value of using DBpedia to

ISEM '13, September 04 - 06 2013, Graz, Austria

[1] http://linkeddata.org
[2] http://wikipedia.org

	Aemoo	Kaminskas & al.	LED	MORE	Seevl	Yovisto
Purpose	Exploratory search	Cross-domain recommendation	Exploratory search on ICT domain	Film recommendation	Musical recommendation	Video exploratory search
Data	DBpedia EN + external services	DBpedia EN subset	DBpedia + external services	DBpedia EN subset	DBpedia EN subset	DBpedia EN+DE subset
Multi-domain	Yes	Cross two domains	No	No, cinema	No, music	Yes
Query	Entity search	Entity selection in a pre-processed list	Entity search	Entity search	Entity recognition from Youtube.	Entity recognition in keywords
Algorithm	EKP filtered view	weighted activation	DBpedia Ranker	sVSM algo.	DBrec algorithm	Set of heuristics
Ranking	No	Yes	Yes	Yes	Yes	Yes
Explanations	Wikipedia-based	Path-based	No	Shared prop.	Shared properties	No
Offline proc.	Yes , EKP part	Yes	Yes	Yes	Yes	Yes

Table 1. Related works summary

compute relatedness and similarity measures for recommendation and exploratory search purposes (summarized in the table 1).

Seevl [21] is a DBpedia-based band recommender for Youtube based on the DBrec algorithm. The DBrec algorithm ranks the similar bands according to shared direct and indirect properties. The ranking is processed offline and stored in RDF. MORE [6] is a DBpedia-based film recommender. It uses a semantic adaptation of the vector space model called sVSM . The more features two movies share the more similar they are. The ranking is processed offline. [11] proposed a method to perform cross-recommendations on at least two chosen domains (e.g. recommending musical artists starting from tourists' attractions). The recommendation computation is operated offline and uses a weighted spreading activation algorithm. The positive results of the users' evaluation confirm the potential of DBpedia to perform cross-domain and cross-type recommendations.

Yovisto [26] is an academic video platform providing an exploratory search feature. It proposes a ranked list of related topics besides the search results. The selection and the ranking of the resources (corresponding to topics) are computed offline thanks to a set of eleven heuristics. A user evaluation showed that the exploratory search feature significantly improved the search experience. *Aemoo* [20] is a DBpedia-based exploratory search system making use of Encyclopedic Knowledge Patterns (EKP) which identify typical classes used to describe instances of a specific class. Starting from a resource of interest, *Aemoo* presents its neighborhood filtered with its corresponding EKP or inverted EKP using the *curiosity* function. In [19] the authors present the Lookup Explore Discover exploratory search system. The application recommends a set of tags that are strongly related to the named entities recognized in the user query (e.g. *RDFa*, *micro-data* for the *microformat* query).The user is assisted by the tags during the exploration and can use it to refine the query.

In table 1, we observe that the approaches all use an offline preprocessing step. It considerably limits the type and the range of retrieved results. It narrows the range of information needs that can be supported by the applications. Such approaches are then sufficient for building domain-specific applications but do not fully exploit the potential of linked data for exploratory search:

- They give results for a subset of chosen resources only ([6][11][19][21]): there is only a limited sub-set of resources that are pre-processed and stored locally.
- They exclude the user from the results computation and propose a fixed ranking scheme ([19][20][21][26]). The user might want to influence the recommendations/results.
- They compute the results by addressing only one local dataset replicated from public LOD sources. They might also retrieve outdated results: the preprocessing needs to be performed regularly if the knowledge base evolves. This is an issue for some uses such as data journalism.

3. PROPOSITION

3.1 On-the-fly linked data based exploration

We propose a method that selects and ranks on-the-fly a meaningful subset of resources in a targeted LOD dataset:

- Every resource available in the targeted dataset can constitute a potential topic for exploration/discovery, not only a type or domain-dependant subset.
- As the results are not pre-computed the users can influence them by tuning several parameters (e.g. for advanced search, customization, contextualization).
- It is possible to address remote SPARQL endpoints to retrieve results. By querying the public SPARQL endpoints and not a local replica we also ensure the freshest data.

The method has to be sufficiently fast to be performed on-the-fly (in few seconds maximum). To reach this goal we propose a semantic spreading activation algorithm and a sampling technique that makes it applicable on remote LOD graphs.

3.2 Spreading activation basis

Spreading activation comes from cognitive psychology and works related to the memory [4]. Later it inspired a lot of algorithms in various fields and was successfully employed in information retrieval. Early and important works include [3] and [5].

The core functioning is always the same: first a stimulation value is assigned to one or several node(s) of interest. Then this value is propagated to the neighbor's node(s). The value assigned to neighbors depends on the algorithm purpose and settings. During the next iterations the propagation continues from newly activated nodes. This process is repeated till a stop condition is reached *e.g.* maximum number of nodes activated or iterations.

The spreading activation technique was applied over RDF for creating context-inference models, extending and refining ontologies, predicting social annotations [23]. It was also studied in the context of semantic search. [22] proposes a hybrid search approach combining a classical keyword-based search method with a spreading activation over a domain ontology. [24] is combining concepts-based similarity, text-based similarity and spreading activation for document retrieval. [13] retrieves content associated to one or several resources in a socio-semantic

network. [14] performs semantic association based search over an ontology thanks to specificity and generality measures. [7] answers natural language queries with DBpedia.

These works rely on a weight mapping step applied on nodes or properties using local and global graph measures [7][15][22][24] or manual intervention [13]. Our approach differs as the propagation controlling pattern is a type-based semantic weight that is function of the stimulated origin node. In other words the origin node semantic pattern plays a significant role in the distribution of activation even in "*distant*" parts of the graph. It is particularly adapted to dense and strongly typed graphs that are common in the LOD context. Indeed in linked data sources there is a strong need to constraint the propagation in order to target relevant part of the graph only and minimize the amount of data processed. In our approach the types of the nodes is a fundamental criteria taken in account for the propagation distribution. The fact the semantic pattern can be easily computed on the fly is leveraged in our implementation to address remote LOD data sources (section 4).

3.3 Formal proposition

The algorithm presented below aims to explore the graph obtained from one or more SPARQL end-points of the linked data cloud. It identifies a meaningful subset of resources strongly related to the user's interest (e.g. *The Rolling Stones*). Then these results are presented and explained to the user through an interface optimized for exploratory search (section 5).

The graph semantics is exploited thanks to a weight that is function of the origin node. It constrains the propagation to certain node types (see definition 9) and integrates a triple-based similarity measure (see definition 10). At the end of the algorithm execution the activation values of the nodes determine their ranks. Prior to the algorithm presentation, we introduce several necessary definitions on RDF triples and the classic graph functions we used:

Definition 1. (RDF triple, RDF graph). Given U a set of URI, L a set of plain and typed Literal and B a set of blank nodes. An RDF triple is a 3-tuple $(s,p,o) \in \{U \cup B\} \times U \times \{U \cup B \cup L\}$. s is the node subject of the triple, p the predicate of the triple and o the node object of the triple. An RDF graph is a set of triples.

Definition 2.(RDF typing triple, RDF non-typing triple.) An RDF typing triple is a 3-tuple$(s,p,o) \in \{U \cup B\} \times \{rdf{:}type\} \times \{U \cup B \cup L\}$. An non-typing triple is a 3-tuple$(s,p,o) \in \{U \cup B\} \times \{U \setminus rdf{:}type\} \times \{U \cup B \cup L\}$.

Let KB be the set of all the triples in the triple store:

Definition 3. The node degree is the number of edges involving node j: $degree_j = |\{(j,p,x) \in KB\} \cup \{(x,p,j) \in KB\}|$

Definition 4. (Type depth) $depth(t)$ uses the subsumption schema hierarchy (as in RDFS or OWL) to compute the depth of a type t and identify the most precise type(s) available for a node.

$$depth(\mathrm{t}) = \begin{cases} depth(\mathrm{t}) = 0 \; if \; t = \mathrm{T} \text{ the root of the hierarchy,} \\ \mathrm{depth}(t) = 1 + Min_{\,s_t;(t,rdf:subClassOf,s_t) \in KB} depth(s_t) \\ otherwise \end{cases}$$

Where type t is a class in the hierarchy of the RDFS schema and S_t is a direct super class of t in this hierarchy before any transitive closure is computed.

Definition 5. (Node neighborhood) $Neighbor(i)$ is the set of neighbors of the node i in the graph retrieved from the targeted linked data sources:

$$Neighbor(i) = \{x\,; ((i,p,x) \in KB \vee (x,p,i) \in KB) \wedge p \neq rdf{:}type \wedge x \in U \cup B\}$$

Definition 6. (Semantic Spreading Activation algorithm,) The following formula determines the nodes' ranks along the iterations:

$$a(i,n+1,o) = s(i,n,o) + w(i,o) * \sum_{j \in Neighbor(i)} \frac{a(j,n,o)}{degree_j}$$

Where:

- o is the origin node *i.e.* the instance of interest initially stimulated (e.g. *The Rolling Stones*);
- i is an arbitrary instance node of the graph;
- j iterates over the neighbors of i ;
- n is the current number of iterations;
- $a(i,n+1,o)$ is the activation of node i at iteration $n+1$ for an initial stimulation at o;
- $s(i,n,o)$ is the stimulation value of the node i at n, the value that is redistributed to its neighbor's. The node having a positive stimulation at initial time is the origin/seed node *i.e.* here $s(i,n,o)$ =1 if $i = o$ and $n = 0$;
- $a(j,n,o)$ is the activation from a neighbor node j of i for a propagation origin o at iteration n;
- degree_j returns the degree of the node j (def. 3);
- $w(i,o)$ is a semantic weighting function which takes into account the semantics of the nodes i and o. First, it aims to identify the propagation domain: the nodes are activated or not depending on their types (def. 9). Second, it encourages the activation of the nodes that are similar to the origin o using others semantics attributes (def.10). $w(\mathrm{i,o})$ is explained below.

The class-based propagation domain $CPD(o)$ is the set of types through which the propagation spreads. To be precise, the propagation spreads through all the nodes which have at least one type present in $CPD(o)$. It aims to increase the results relevance by focusing the activation distribution on a consistent subset of nodes and limit the amount of data processed. The propagation domain is identified on run-time before the propagation starts and is based on types of the origin node's neighbors.

Definition 7. $Tmax(x)$ is the set of the deepest types t of a given node x according to their $depth(t)$ (def. 4):

$$Types(x) = \{t; (x, rdf{:}type, t) \in KB\}$$

$$Tmax(x) = \begin{cases} t \in Types(x)\,; \\ \forall t_i \in Types(x)\,; \\ depth(t) \geq depth(t_i) \end{cases}$$

Definition 8. $NT(o)$ is a multi-set counting the occurrences of the deepest types in the seed node's neighborhood (def 5.).

$$NT(o) = \{(t,c); t \in Tmax(n); n \in Neighbor(o); c = |\{n \in Neighbor(o); t \in Tmax(n)\}|\}$$

Definition 9. $CPD(o)$ is the classes propagation domain, it constitutes the class-based "*semantic pattern*" used all along the propagation. A threshold-based filtering can be applied to exclude the less prevalent types present in $NT(o)$. This threshold can be used to restrains the propagation domain size for performance purpose. After this last operation we obtain the classes'

propagation domain $CPD(o)$ *i.e.* the nodes with a type included in $CPD(o)$ and that will be activated during the propagation. It allows limiting the propagation to a subset of types that are consistent with the activation origin:

$$CPD(o) = \left\{ t; (t,c) \in NT(o); \frac{c}{\sum_{(n_i,c_i)\in NT(O)} c_i} \geq threshold \right\}$$

Definition 10. $commontriple(i,o)$ is an additional measure that aims to improve the algorithm relevance by favoring activation of node i having similar properties with the origin o. It is a triple-based comparison: the more a node is a subject of triples that share a property p and an object v with triples involving the origin node o as a subject, the more it will receive activation:

$$w(i,o) = \begin{cases} 0 \; if \; \nexists t \in Types(i); t \in CPD(o) \\ 1 + |commontriple(i,o)| \; otherwise \end{cases}$$

Where $commontriple(i,o) = \{(i,p,v) \in \text{KB}; \exists (o,p,v) \in \text{KB}\}$

Definition 11. Semantic Spreading Activation algorithm Serendipitous Mode retrieves results adding randomness:

$$a_{random}(i, n+1, o, r) = (1-r) * a(i, n+1, o) + r * random()$$

Where r is the level of serendipity, comprised between 0 and 1, and $random()$ produces a random value between 0 and 1.

4. IMPLEMENTATION

In this section we present the first implementation of the algorithm using DBpedia. We notably detail the sampling technique that is a key component. It is coupled with the spreading activation algorithm and makes it applicable "*on-the-fly*" over remote LOD datasets using their SPARQL endpoints.

4.1 Dataset

We decided to set up a first implementation on top of DBpedia due to the amount of topics it covers. It also offers an interesting ground for user evaluations as it contains common-knowledge items such as films or musical artists.

For comparative purpose we used the DBpedia 3.7 version (section 6). As we needed to query the endpoint millions of times during the analysis we set up a local version. Our version contains the *wikiPageWikilink*[3] triples. This property indicates that a hypertext link exists in Wikipedia between the 2 resources but that the semantics of the relation was not captured. It provides an amount of valuable extra-links for connectionist methods like spreading activation. DBpedia 3.7 graph with *wikiPageWikiLink* triples is large (3.64 million nodes, 270 million triples) and heterogeneous since there are 319 classes in the DBpedia 3.7 ontology schema.

4.2 Architecture

The algorithm is coded in JAVA. Each time a query is processed a Kgram[4] inference engine instance is created. This *local* instance imports a limited sub-graph sampled from the targeted SPARQL endpoint. In other words, we apply the spreading activation algorithm only on a limited and defined sub-graph per query. The Kgram instance iteratively imports subparts of DBpedia using `INSERT` queries and `SPARQL <service>`.

The samples are imported iteratively according to the nodes activation values. At the beginning the neighborhood of the origin node (filtered by its *CPD*) is loaded and a first round of propagation is performed. During the next iterations the neighborhoods of the top activated nodes are imported into the Kgram instance till a maximum limit of triples is reached (such limit is discussed further).

4.3 Settings

We set up some variables in order to implement our formula:

- The propagation spreads in both directions to take into account incoming and outcoming neighbors. From a spreading activation point of view the orientation is arbitrary and depends on a modeling choice.
- The $threshold$ filtering the propagation domain is set to a low value of 0.01, as we do not want to restrain too much the CPD in our exploratory search context.
- In DBpedia, instances are linked to their category through the $dcterms{:}\,subject$[5] property. We make use of *this* property to compute $commontriple(i,o)$. The categories constitute a topic taxonomy which is very informative on the resources. It constitutes a valuable basis for computing $commontriple(i,o)$. The nodes reached by the propagation and belonging to the same categories as the origin receive a greater amount of activation.
- The maximum number of iterations has still to be fixed and is discussed further.

4.4 Limiting the response time

At this point two parameters still need to be discussed: (1) the *maximum number of iterations* and (2) *the size of the sample imported*. To set up these variables we observed the behavior of the algorithm over DBpedia. We processed a large amount of queries and analyzed the results lists variations along the iterations and according to the sample size. We needed to know (1) if the results converge quickly and (2) what is the impact of the sample size on the retrieved result lists. For the system to remain usable, the response time cannot exceed a few seconds.

4.4.1 Analysis method

According to [16] the best method to select a representative subset of a large graph is a random walk. We followed these recommendations and computed a 100 000 nodes sample from DBpedia. To compare the rankings of the results lists obtained with various configurations we used the Kendall's tau-b coefficient τ_B. τ_B is comprised between -1 and 1 where -1 means a total rank discordance and 1 a total concordance [12]. Our configuration for the tests was an application server (8 Intel Xeons CPU E5540 @2.53GHz 48 Go RAM) and a SPARQL endpoint (2 Intel Xeons CPU X7550 @2.00GHz 16Go RAM)

4.4.2 Setting the maximum number of iterations

As spreading activation is an iterative algorithm we have to set a stop condition. To determine this parameter we studied the algorithm convergence on DBpedia data. We processed 100 000 queries taking as inputs the nodes of the DBpedia subset. We measured the τ_B correlation coefficient and the number of common results between the top 100 results at iteration n and at iteration $n+1$ (n being comprised between 1 and 99). The Kendall-Tau is calculated by considering the common results in

[3] http://dbpedia.org/ontology/wikiPageWikiLink

[4] http://wimmics.inria.fr/node/26

[5] http://purl.org/dc/terms/subject

the two lists. The triples loaded limit has not been studied yet and is experimentally fixed to 10.000.

For clarity purpose the figure 1 shows only the twenty first iterations, after 16 iterations the percentage of average shared results exceeds 99% and the average τ_B is superior to 0.99 from one iteration to another. It is visible that the results change very slowly after a few iterations. Thus we decided to fix the maximum pulse at 6 to obtain a good trade-off between response time and results stability. A propagation visualization video using Web Import plugin for Gephi[6] has been published[7].

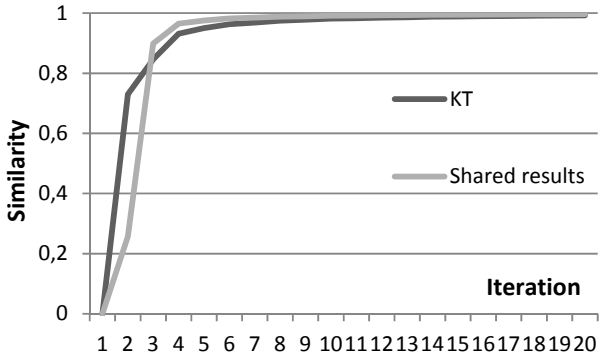

Figure 1: τ_B and percentage of shared results from one iteration to another, top 100 results

4.4.3 Setting the triples loading limit

In order to control the size of the sub-graph imported and consequently the response time, we introduce a limit of triples loaded per query. Along the iterations the neighbors of the most activated nodes are imported in the Kgram instance. When the imported graph overtakes the triples limit, no more neighborhoods of nodes are imported, the sample is considered complete. We also introduce an experimental loading threshold of 0.1 *i.e.* nodes having an activation value under 0.1 are not taken into account during the loading process. This threshold allows distributing the loading process among the iterations and reaching distant nodes.

In order to set this limit we used again the DBpedia subset. For each node we processed 10 times the query with a loading limit ranging from 2000 to 20000 (by steps of 2000). It allowed comparing the variations of the response time as well as the changes in the top 100 results list according to the triples limit. We wanted to observe the cost of importing more triples and its impact on the results.

The figure 2 shows that the algorithm response time is linear to the triples loading limit. It also presents the top 100 results Kendall-Tau variation from one loading limit to another (2000 by 2000). After 6000 triples the top results evolve slowly from one sample size to another. In other words, loading more triples does not impact very much the top results list compared to its cost. The figure 3 shows the response times for the 6000 triples loading limit. A large proportion of queries are processed in less than 4 seconds with this configuration.

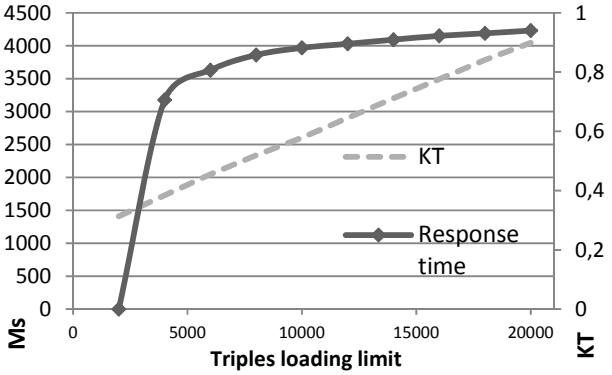

Figure 2: response time and τ_B correlation coefficient from one loading limit to another, top 100 results

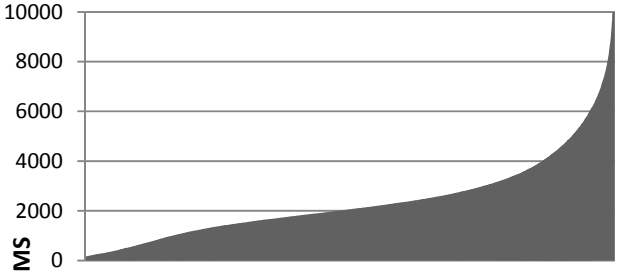

Figure 3: response times histogram of the 100.000 queries

4.4.4 Multilingualism

Being able to address remote SPARQL endpoint is very interesting to deal with the LOD in general but also in the context of DBpedia as more and more local chapters emerge[8]. These chapters are related to a language and considerably differ in what they describe and how it is described. In august 2012 there were already 20.8 million of resources described in the local chapters but only 10.5 million overlaps with concepts from the English DBpedia[9]. The interest of enabling exploratory search on these knowledge bases is obvious.

We tested the response time of our method when applied to the French and Italian SPARQL endpoints. We run 10000 queries corresponding to the 10000 first resources of the DBpedia subset that are described in the English, French and Italian versions of DBpedia. The equivalence among resources in the different DBpedia chapters is declared thanks to the *wikiPageInterLanguageLink* properties. We applied the same parameters as described above for the English-speaking version. The response times were satisfying for all the SPARQL endpoints: an average of 2.05, 1.63 and 1.99 seconds for respectively the English, French and Italian endpoints[10].

5. DISCOVERY HUB PROTOTYPE

Discovery Hub[11] (Figure 4) uses the algorithm and the sampling method previously described to suggest resources of interest. It is an exploratory search engine which helps the user to discover things he might like or might be interested in starting from one or

[6] http://wiki.gephi.org/index.php/SemanticWebImport

[7] http://semreco.inria.fr/hub/videos/

[8] http://dbpedia.org/internationalization

[9] http://blog.dbpedia.org/2012/08/06

[10] Will be fully available in the V2, currently in development

[11] http://semreco.inria.fr

several topics of interest. It proposes redirections to third-party platforms for extending the search process: services are proposed according to the type of the resource *e.g.* music services for a *Band* or tourism platforms for a *Museum*. Online videos are available[12].

The exploration starts after selecting the topic of interest thanks to a DBpedia lookup[13]. Then the algorithm proposes a set of highly related and informative resources (section 3 and 4). Discovery Hub enables faceted browsing over these results. The classes identified in $\boldsymbol{CPD(o)}$ are used to build the facets proposed on the left (e.g. *Album, Band, Film*). A set of 20 filters (or sub-facets) is also proposed per facet. For instance the filters "*american rock music film*" and "*films directed by Martin Scorsese*" are proposed in the *Film* facet of *The Rolling Stones* results. These filters correspond to prevalent DBpedia categories that have been identified in the facet's results. The filters having a lower degree are put in evidence and presented with clearer colors. It aims to drive the user in unexpected browsing paths and consequently augments the discovery potential of the application.

Explaining the results is fundamental in our approach. The resources selection is valuable only if users can deeply understand their relation to their initial interest (query). When a user is interested or intrigued by an item he can use 3 different explanatory features[14]. The first one shows the common properties shared by the query-resource and the result. The second one identifies and highlights the crossed references in Wikipedia pages between them (when existing). The third one presents a set of direct and indirect connections between the result and the query in a graph format. From this graph it is possible to use the second functionality to get more explicit information on the nodes connections thanks to their Wikipedia pages.

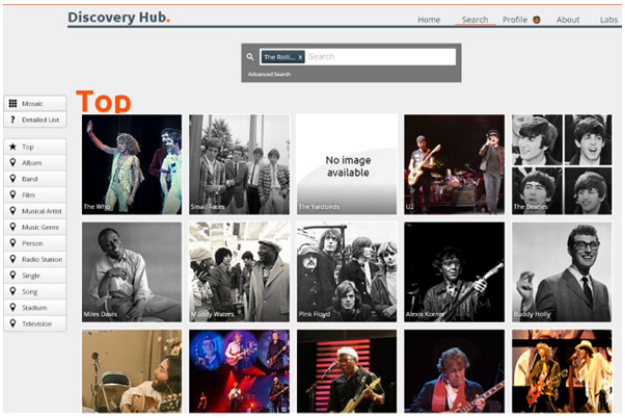

Figure 4: Discovery Hub mosaic results page

The user can influence the results in Discovery Hub thanks to the *advanced search* mode. First, he has the possibility to select the aspects that interest him the most about the query-resource. These aspects are the categories the query-resource belongs to (e.g. *Blues-rock musicians*, *Musical_groups_from_London* for *The Rolling Stones*). In this case the $commontriple(i,o)$ (def. 10) is not computed by taking in account all the categories of the query-resource but only the subset explicitly specified by the user.

The user can also add randomness in the results selection by manipulating the "*surprise*" mode slide bar (from 0 to 1). 0 means that there is no randomness in the processing; it corresponds to a normal query. The closer the value is to 1 the more the results will be influenced by randomness (def 11). It can be very useful and playful for users that already have knowledge on the topic explored and that want to retrieve unexpected, surprising information about it.

6. EVALUATION

As mentioned in [14] and [26] the research initiatives in the exploratory search domain suffer from the lack of evaluation standardization. We evaluated the interest of the users towards the results selected by the algorithm. Indeed, this selection step is critical in our approach. A full evaluation of the application through task-based scenarios will be envisioned in future work.

First, we performed an evaluation on a neutral interface. Second we evaluated the influence of the Discovery Hub explanatory features on the user's judgments. We evaluated the results of our semantic spreading activation algorithm (SSA) against the sVSM algorithm used in the MORE recommender. The main reason of this choice is that MORE is the only linked-data based system of the related works list that has been compared to another: Seevl [6]. As MORE only recommends films, only the *Film* results were taken into account during the comparison.

6.1 Resources recommendation evaluation

During the first round of evaluation we evaluated both the relevance and the discovery potential of our proposition regarding the sVSM baseline. We wanted to verify the following hypothesizes:

- **Hypothesis 1**: SSA gives results at least as relevant as sVSM (even if it is not dedicated to the cinema domain).
- **Hypothesis 2**: the SSA algorithm has a less strong degradation than sVSM algorithm. In other words, the end-list results are better for SSA.
- **Hypothesis 3**: there is a greater chance that results are less relevant but newer to users at the end of the lists.
- **Hypothesis 4:** the advanced search functionality gives better results compared to the standard query ones.

The hypothesis 1 and 2 aim to verify that the SSA algorithm retrieves relevant results compared to a domain specific approach. The hypothesis 3 aims to verify that the algorithm correctly ranks the results by retrieving the most relevant first. The hypothesis 4 aims to verify that the advanced search functionalities help the user to find more interesting resources.

The participants evaluated alone both algorithms on a neutral interface set up with the online tool Limesurvey[14]. They had to judge 5 lists of films' recommendations. These lists were composed of the top 20 results from the 2 algorithms. Each list was generated starting from one seed-film. The lists were fully randomized, the participants were not aware of the results provenance. The seed films used to generate the lists were randomly chosen in the « *50 films to see before you die* »[15] list. It was chosen because of its diversity:"*each film was chosen as a paragon of a particular genre or style*"[20]. The selected films were: *2001: a space odyssey*, *Erin Brockovich, Terminator 2: judgment day*, *Princess Mononoke* and *Fight club*. Two Likert scale [17] questions were asked:

[12] http://semreco.inria.fr/hub/videos/

[13] http://lookup.dbpedia.org

[14] http://www.limesurvey.org

[15] http://en.wikipedia.org/wiki/50_Films_to_See_Before_You_Die

- *With the film [result] I think I will live a similar cinematic experience as with [seed film]? Strongly agree, agree, disagree, strongly disagree*.
- *You and [result]? Seen, Known but not seen, Not known*

This formulation was chosen because similarity is a very important factor of relevance for Discovery Hub (def. 10). Thus it can be used to get recommendations.

To analyze the relevance and the discovery potential a 2 (SSA vs sVSM) * 5(Film 1 vs Film 2 vs Film 3 vs Film 4 vs Film 5) * 2 (1-10 ranks vs 11-20 ranks) analysis of variance (ANOVA) test was realized. In statistics, an ANOVA [25] is a method used to compare more than two means simultaneously and determine if their differences are substantial or reflects natural sampling fluctuation. As we study several factors at the same time and as the users participate in all conditions, we performed a factorial ANOVA with repeated measure. The survey was filled by 15 persons (i.e. 3750 votes): 13 males, 2 females, average age of 31.7 years, mainly computer scientists. The average number of movies seen on any support monthly was 10.4 (standard deviation = 8.66). In the following results 0 corresponds to *strongly disagree*, 1 to *disagree*, 2 to *agree*, 3 to *strongly agree* for the relevance score. 0 corresponds to *seen*, 1 to *known but not seen*, 2 to *not known* for the discovery score. The score for each user were very stable because it was computed over a large number of user responses (250 per user) and over two identified major sources of variation (ranking and film). This procedure allows us to increase the reliability on our measurement setting and thus had a positive impact on the power of our statistical testing [2].

Hypothesis 1. To verify the hypothesis 1, we observed the difference between the SSA and the sVSM recommendation relevance scores. The figure 5 shows that overall SSA (mean m = 1.42, standard deviation sd = 0.27) outperforms sVSM (m = 1.18, sd = 0.24). The ANOVA test being statistically significant ($F(1,14) = 113.85$, $p < .001$) the hypothesis 1 is verified.

Hypothesis 2. To verify the hypothesis 2, we observed the difference between the SSA and the sVSM relevance scores at the end of the results list (rank 11-20). The table 2 presents the average scores of relevance and discovery for the beginning and the end of results lists. SSA has a better relevance score (m = 1.28, sd = 0.243) than sVSM (m = 0.93, sd = 0.228) for results at the end of the list. The ANOVA test being statistically significant ($F(1,14) = 20.23$, $p = .001$) the hypothesis 2 is validated.

Hypothesis 3. To validate the hypothesis 3 we compared both the relevance and discovery scores of the 2 two algorithms for the beginning and the end of results lists. Results are less relevant in the second half of the list (beginning m = 1.48, sd = 0.299, end m = 1.10, sd = 0.235) but have a higher discovery score (beginning m = 1.12, sd = 0.249 vs end m = 1.355, sd = 0.216). The ANOVA test being statistically significant for relevance ($F(1,14) = 134.02$, $p < .001$) and discovery ($F(1,14) = 64.30$, $p < .001$), thus hypothesis 3 is validated. The discovery score difference between the beginning and the end of the results lists is slight for SSA. One explanation is that the algorithm computes a relatedness which does not reflect any kind of popularity. It is noticeable that sVSM has a better discovery score than SSA at the end of the list but at the same time its relevance decreases considerably. SSA can be considered as more balanced.

Hypothesis 4. To validate the hypothesis 4 we performed a complementary experimentation. We asked to the users to choose the 3 most interesting aspects (DBpedia categories) and the less interesting ones about *Fight Club*. *Fight Club* was chosen because it was the most viewed by 86.66% of the participants (it was the highest rate). Ten persons participated to this experimentation extension. The participants showed interest in narrow categories (e.g. *American black comedy films*) and disinterested in broad ones (e.g. *1999 films*). The 3 specified criteria of interest were used to compute the results using the advanced search mode. We asked the participants to judge the ten first results. The average relevance score of the top ten results significantly raised compared to the "*classic*" queries ones: 1.94 (sd=0.55) versus 1.42 (sd=0.39) previously. The hypothesis 4 is validated.

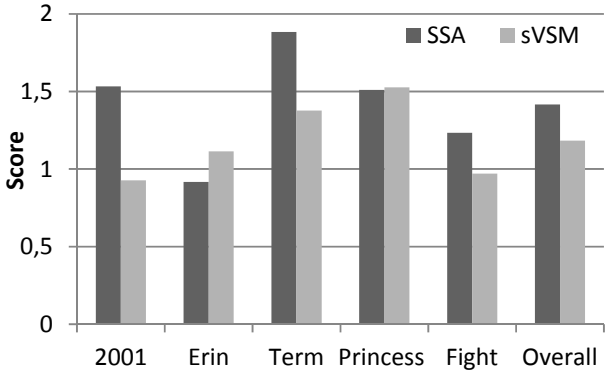

Figure 5: relevance score of algorithms (between 0 and 3)

Measure	Algo	Rank	Mean	St. Dev.
Relevance	**SSA**	1-10	1.54	0.305
		11-20	1.28	0.243
	sVSM	1-10	1.42	0.294
		11-20	0.93	0.228
Discovery	**SSA**	1-10	1.10	0.247
		11-20	1.21	0.228
	sVSM	1-10	1.14	0.251
		11-20	1.50	0.205

Table 2: scores for partial lists

6.2 Explanatory features evaluation

We also evaluated the influence of the explanatory features on the users' judgments:

- **Hypothesis 5**: explanatory features increase user's overall judgments positivity.

To analyze the impact of the explanatory features on the users' perception of the results, the participants of the first experimentation were asked to evaluate again 20 results with the Discovery Hub interface. These 20 results were randomly selected in the SSA results list of the first evaluation. We wanted to estimate if the explanatory features are efficient to increase the users' interest.

Hypothesis 5. To verify the hypothesis 5, we observed the difference between the relevance scores obtained with and without the explanatory features. The relevance score rose significantly: previously m = 1.26, sd = 0.40, with the features: m = 1.50, sd = 0.26. The average number of positive judgments reached 9.4 versus 7.34 previously. A Student test [25] was performed. It is used instead of the ANOVA in the case of only two mean are

compared [25]. The Student test being statistically significant ($t(14) = 3.872$), $p = 0.002$) the hypothesis 5 is verified.

We asked the participants to give their opinion about the three explanations. For the 3 features we asked "*the feature X helped me understand the relation between the films and to make a choice?"* and one more general question: "*overall, I feel that these three features can help me to make new discoveries*". 0 corresponds to *strongly disagree*, 1 to *disagree*, 2 to *agree* and 3 to *strongly agree*.

The common properties and graph-based features helped significantly the participants (average scores: 2.13 for both) whereas the benefit of the Wikipedia-based feature was less evident (average score: 1.86). The more general question received the high average score of 2.53. The results are not uniform and show the interest to propose different explanatory features.

7. CONCLUSION

To ease the exploratory search tasks we propose to draw users' attention to resources and associations that convey a lot of knowledge regarding their initial interest. Therefore the research questions addressed in this paper are: (1) in a dense and heterogeneous linked data graph, how to select and rank a meaningful subset of resources related to the user's interest? (2) As linked data are distributed, how to perform this selection on remote LOD sources? (3) On the interface side, how to present and explain the selection to the user? To select and rank a set of meaningful resources related to the user's interest we proposed a novel method based on a semantic spreading activation. To address remote SPARQL endpoints we coupled it with a graph sampling technique. We studied its behavior over DBpedia in order to set its main parameters. Finally we introduced the Discovery Hub application that offers faceted browsing and explanation features that help the user understand the results.

We observed that the results converged quickly and that processing a limited amount of triples was sufficient. The queries were processed in a few second and confirmed the validity of the approach implemented in the Discovery Hub prototype available online. Our user's evaluations showed the method efficiency for the suggestion of resources of interest against the sVSM baseline. They also confirmed the efficiency of explanatory features and of the advanced search functionality.

We now want to study the behavior of the algorithm on graphs having diverse properties (e.g. structure, degree, size, diameter) in order to determine to what extend the parameters used for DBpedia are generic and can be reused for others data sources. We will also study how several linked data sources can be combined with the proposed approach. Finally we intend to take into account the learning process in the exploratory for instace declaring as new interests and seeds as the users explore.

8. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z.Ives, DBpedia, A nucleus for a web of open data, *ISWC,* 2007.

[2] RL Brennan, *Generalizability Theory*, Springer-Verlag 2001

[3] Cohen, P and Kjeldsen, R. Information Retrieval by Constrained Spreading Activation on Semantic Networks. *Information Processing and Management, 23(4): 1987.*

[4] A. Collins and E. Loftus. A spreading activation theory of semantic processing. *Psychological Review, 1975.*

[5] Crestani, F. Application of Spreading Activation Techniques in Information Retrieval. *Artificial Intelligence Review, 1997.*

[6] T. Di Noia, R. Mirizzi, V. C. Ostuni, D. Romito, and M. Zanker. Linked open data to support content-based recommender systems. *In 8th I-SEMANTICS 2012.*

[7] A. Freitas, J.G. de Oliveira, S. O'Riain, E. Curry, and J.C. Pereira da Silva. Querying Linked Data using semantic relatedness: A vocabulary independent approach, 2011.

[8] Andreas Harth. Visinav: A system for visual search and navigation on web data. We Semantics: Science, Services and Agents on the World Wide Web, 8(4):348 – 354, 2010.

[9] Heim, P., Ziegler, J. and Lohmann, S. gFacet: A Browser for the Web of Data. In Proc. SAMT Workshop: IMC-SSW, CEUR-WS 2008, 49-58

[10] Huynh DF, Karger DR., Parallax and companion: set-based browsing for the data web. *In: Proc. of WWW* 2009.

[11] Kaminskas M. and al, Knowledge-based Music Retrieval for Places of Interest, in Proceedings of *MIRUM'12, 2012.*

[12] Kendall, Maurice, Rank Correlation Methods. London: *Charles Griffin and Co*., 1948.

[13] Kinsella, S., Harth, A., Troussov, A., Sogrin, M., Judge, J., Hayes, C., & Breslin, J. G.). Navigating and annotating semantically-enabled networks of people and associated objects. In *Why Context Matters*,pp. 79-96, 2008

[14] Kules, B. and Shneiderman, B., Users can change their web search tactics: Design guidelines for categorized overviews. *Information Processing & Management*, 2007

[15] Lee, M., & Kim, W. (2009, December). Semantic association search and rank method based on spreading activation for the Semantic Web. IEEM 2009. IEEE International

[16] J. Leskovec and C. Faloutsos. Sampling from large graphs. In KDD '06: *Proceedings of the 12th ACM SIGKDD*

[17] Likert, R., A technique for the measurement of attitudes. *Archives of Psychology.* 1931

[18] G. Marchionini, Exploratory search: From finding to understanding. *Comm. ACM,* 2006

[19] Roberto Mirizzi and Tommaso Di Noia. From exploratory search to web search and back. In Proc. of the 3rd workshop on Ph.D. students in information and knowledge management, PIKM '10, 2010.

[20] A. Musetti, A. G. Nuzzolese, F. Draicchio, V. Presutti, E. Blomqvist, A. Gangemi, and P. Ciancarini. 2012. Aemoo: Exploratory search based on knowledge patterns over the semantic web. *Semantic Web Challenge.*

[21] A. Passant,. 2010. dbrec – music recommendations using dbpedia. *ISWC 2010*

[22] Rocha, C., Schwabe, D., and de Aragão, M. P.: A Hybrid Approach for Searching in the Semantic Web. In *Proc. of the 13th International World Wide Web Conference*, 2004

[23] Rodríguez, J. M. Á., Gayo, J. E. L., & Ordoñez de Pablos, P. (2012). An Extensible Framework to Sort out Nodes in Graph-Based Structures Powered by the Spreading Activation Technique: The ONTOSPREAD Approach. *International Journal of Knowledge Society Research*

[24] Scheir, P., Ghidini, C., & Lindstaedt, S. N., Improving search on the semantic desktop using associative retrieval techniques. *Proceedings of I-MEDIA*, 221-228. 2007

[25] Sprinthall, R.. Basic Statistical Analysis. *Prentice-Hall, New Jersey.* 1997

[26] Waitelonis, J., Sack, H., Augmenting Video Search with Linked Open Data. *Int. Conf. on Semantic Systems 2009*

[27] White Ryen, Roth Resa A., Exploratory Search: Beyond the Query-Response Paradigm, Morgan and Claypool Publishers, March 2, 2009, ISBN-10: 159829783X