Personalized Concept-based Search and Exploration on the Web of Data using Results Categorization

Melike Sah and Vincent Wade
Centre for Next Generation Localisation, KDEG, Trinity College Dublin, Dublin, Ireland
{Melike.Sah, Vincent.Wade}@scss.tcd.ie

Abstract. As the size of the Linked Open Data (LOD) increases, searching and exploring LOD becomes more challenging. To overcome this issue, we propose a novel personalized search and exploration mechanism for the Web of Data (WoD) based on concept-based results categorization. In our approach, search results (LOD resources) are conceptually categorized into UMBEL concepts to form *concept lenses*, which assist exploratory search and browsing. When the user selects a concept lens for exploration, results are immediately personalized. In particular, all concept lenses are personally re-organized according to their similarity to the selected concept lens using a similarity measure. Within the selected concept lens; more relevant results are included using results re-ranking and query expansion, as well as relevant concept lenses are suggested to support results exploration. This is an innovative feature offered by our approach since it allows dynamic adaptation of results to the user's local choices. We also support interactive personalization; when the user clicks on a result, within the interacted lens, relevant categories and results are included using results re-ranking and query expansion. Our personalization approach is non-intrusive, privacy preserving and scalable since it does not require login and implemented at the client-side. To evaluate efficacy of the proposed personalized search, a benchmark was created on a tourism domain. The results showed that the proposed approach performs significantly better than a non-adaptive baseline concept-based search and traditional ranked list presentation.

Keywords: Concept-based search, personalized search/exploration, linked open data, UMBEL, query expansion, results re-ranking, interactive personalization.

1 Introduction

With the adoption of the LOD by a wider Web community, large volumes of semantic data are being generated. The challenge now is finding and exploring relevant information on the WoD. This is crucial for the uptake of the LOD by applications in order to support both ordinary Web and Semantic Web users with innovative user interfaces. In this context, LOD search engines play a vital role for providing efficient access mechanisms. However, current approaches (e.g. Sindice [1], Watson [2]) adopt keyword-based search and ranked result lists presentation of traditional Information Retrieval (IR), which is not very efficient for large volumes of data [3]. In ranked lists, users cannot understand “what the resource is about” without opening and investigating the LOD resource itself. There is a need to investigate search problems on WoD.

Another search paradigm for the LOD is faceted search/browsing systems, which provide facets (categories) for interactive search and browsing [4]. Facets assist results filtering and exploration. However, the main limitation of faceted search is that facet creation depends on specific data and schema properties of underlying metadata and it can be difficult to generate useful facets to large and heterogeneous WoD [5]. Based on the existing work, it is evident that LOD search mechanisms need improvement, which is our main objective. This is crucial for exploration of WoD data/datasets and uptake of LOD by wider community not just Semantic Web experts.

Traditional IR has been investigating efficient search mechanisms for decades; results clustering and personalized search are two popular methods for enhancing search effectiveness. In clustering search, results are organized into categories for assisting users in results exploration and in disambiguation of the query (Snaket [6], Vivisimo.com, carrot2.org). For example, the query “tiger” may match to animal, computer or golf result categories. The user can disambiguate the query by selecting the correct category. Results categorization is used widely, such as Google categories, Yahoo Directories and Open Directory Project (ODP). Although clustering search and faceted search seems similar, the latter filters results based on schema/metadata, whereas the former clusters results based on their meaning (language model).

On the other hand, personalized search aims to improve retrieval efficiency by adapting results to context/interests of individual users; thus the user can explore personally relevant results. It is a popular research topic and commercial interest (i.e. Google). However, personalized search gained very little focus on the Semantic Web. This could be because of isolated and low volumes of metadata created in early linked data initiatives. As the size of LOD increases, personalized search and interactions become more important. We innovatively combined results categorization and personalized IR to introduce a novel personalized search and exploration mechanism.

1.1 Contributions

In our approach, users access to the WoD with (keyword or Uniform Resource Identifier (URI)) queries. UMBEL conceptual vocabulary (umbel.org) is used to categorize the retrieved LOD resources (search results) into concepts. UMBEL provides a hierarchy of ~25,000 broad concepts that are organized into 32 top-level supertype categories. UMBEL is also interconnected to linked datasets (i.e. DBpedia, GeoNames, Opencyc, schema.org), which can be used for results presentation. Results categorization is achieved by the proposed fuzzy retrieval model [8], which works on any linked dataset, scalable and reasonably accurate (~90% on 10,000 mappings). Alternatively, other methods can be utilized for categorization. For each query, our engine provides results and their UMBEL concepts. On the client-side, results with the same concepts are grouped to form *concept lenses*. Concept lenses favour results exploration and help to disambiguate the meaning of the query. In particular, concept lenses support informational queries (i.e. the intent is to acquire information). It is estimated that ~80% of Web queries are informational [10].

In our approach, personalization is applied in two phases: (i) When a user select a concept lens from the result lists for exploration, immediate personalization is applied; all concept lenses are re-organized according to their similarity to the selected concept lens using a similarity measure. This is a novel method, which allows re-organization of all results based on conceptual and syntactic similarity to a particular lens. In addition, within the selected concept lens, immediately more relevant results are included using results re-ranking and query expansion as well as relevant categories are suggested for results exploration. (ii) We also support interactive personalization. To achieve this, last *N* clicks of the users within a search session are monitored. When the user clicks on a result, within the interacted lens, immediately personalization is applied. Such as, relevant results and lenses are added by query expansion and results re-ranking. The adapted concept lenses are referred as *personal lenses*.

Our contributions are: (i) We propose a novel personalized concept-based search and exploration mechanism for the WoD. To the best of our knowledge, no such previous work exists. (ii) We suggest the use of results categorization as a tool for personalized concept lenses re-ranking, results re-ranking and query expansion. The evaluations have indicated that the use of these personalization and lenses approach greatly enhances retrieval precision. In particular, the key idea is that the user clicks on a concept lens that best suits his/her information needs. Given the selection, our approach personalizes the order of concept lenses. In addition, within the selected lens; the ranked list is personalized to push up the relevant results and the category label is used to generate an expanded query to retrieve more relevant results. We think that this is an innovative feature offered by our approach since it allows dynamically adaptation of results to the user's local choices. In addition, we support interactive personalization following user clicks onto results. (iii) Our personalization approach is non-intrusive, privacy preserving and scalable, since it does not require an explicit login by the user and the personalization is implemented completely at the client-side. (iv) Our approach is adaptable and can be plugged on top of any Linked Data search engine; in this paper, we use Sindice [1]. It only requires UMBEL categorizations, which can be achieved by number of methods such as the fuzzy retrieval model [8].

Section 2 discusses related work. In section 3, the system architecture is introduced. Section 4 introduces personalized concept-based search methods. Section 5 shows evaluations on a benchmark dataset. Section 6 provides conclusions and future work.

2 Related Work

2.1 Search Mechanisms - Clustering, Faceted and WoD Search Engines

Clustering or concept-based search (conceptual search) aims to improve retrieval effectiveness by organizing search results based on their meaning [6]. Open Directory Project and Yahoo Directory for instance use manual categorization, which is not scalable. Conversely, automatic clustering of results is scalable but challenging. Approaches usually use data mining, NLP and statistical techniques (e.g. k-means clustering) to calculate document similarities, form/label clusters and present flat or hierarchical result categories ([6], vivisomo.com, carrot2.org). In contrast to IR approaches, we use LOD resources rather than documents; we extract semantic data from the context of resources for categorization in UMBEL concepts (see section 3 for details).

Faceted search [4] allows interactive filtering of results based on shared schema properties. Generally, faceted search uses labeled graph [18][19] or textual overviews (semantic properties as browsable facets). In both cases, usability/efficiency decreases as the complexity of information space increases. To increase usability of information visualization on huge repositories, [20] describes "*overview first, zoom and filter, then details on-demand*" fashion. [18] uses both statistical knowledge and graph structure (subject and broader topics) to estimate resource popularity for graph presentation in DBpedia. Whereas, [19] utilizes clustering and personalization in a multimedia domain to decrease visualization complexity. Faceted search is typically applied in closed domains since it requires high data completeness and consistent markup across the whole corpus. Considering the varying data quality and heterogeneous vocabularies of the WoD [5], it can be challenging to generate consistent facets for the whole LOD. Moreover, applying dynamic conjunctive clauses on large datasets significantly increases complexity of faceted search. Our approach works on open corpus of LOD resource thanks to the use of the proposed fuzzy retrieval and UMBEL [8].

Finally considering the large body of work on clustering or faceted search, current WoD search mechanisms (Sindice [1] and Watson [2]) utilize traditional full-text retrieval and ranked result lists, which are not focusing on data exploration problems. Users cannot understand "what the resource is about" without opening and investigating the LOD resource, since title/triples are not informative enough. Sig.ma [3] attempts to solve this issue using querying, rules, machine learning and user interaction. However, Sig.ma's focus is on data aggregation. Another relevant aspect of semantic search is the way users express their information needs. Keyword queries are the simplest and widely used approach [1][2][13]. Natural language queries increase expressiveness such as linguistic analysis can be applied to extract syntactic information [17]. Controlled natural language queries are also utilized, where query can be expressed by values/properties of an ontology [14]. Finally, the most formal systems use ontology query languages (i.e. SPARQL), which demands high expertise and impractical from usability point of view. A trade-off between expressivity and usability should be achieved. Compared to existing work, we propose a unique concept-based personalized search and exploration for the WoD. In our approach, we use keyword queries and results are categorized into concept lenses to support exploration. Categorization acts as a tool for personalized lenses/results re-ranking and query expansion.

2.2 Personalized Information Retrieval

Personalized IR is a popular topic in traditional Web. Generally, personalized IR comprises of: (1) User data gathering, (2) user profile representation and (3) personalization techniques. User profiles can be created from [11]: explicit/implicit user relevance feedback, desktop, social Web or user's context. In our work, we use user's context. The advantages and disadvantages are discussed further in section 4.1.

General user profile representation methods in personalized IR are: weighted keywords, semantic network of terms or semantic network of concepts [12]. The simplest model is weighted vector of keywords. However, keyword-based representation does not capture semantics of related terms. Ontology-based profile representation techniques try to overcome this problem. [12] utilizes the entire ontology for representing user profiles. Extracted keywords from the browsed pages are matched to ontology concepts and concepts are represented as weighted vector of keywords. Generally user profiles are utilized for results re-ranking. In contrast to the general approach, our personalized search approach is driven by results categorization. We need to represent user interests as concept lenses for lenses re-organization and capture user's information needs from clicked results for results re-ranking. Lenses are re-organized based on user's local user choices, hence correct personalized re-ordering of categories significantly affects precision. For this purpose, we represent concept lenses with three rich sources of data for similarity comparison. First, all results within the concept lens are combined to create; a vector of UMBEL concepts (specific user interests); a vector of supertype concepts (broad user interests); and a vector of terms (for language comparison). In addition, we track last *N* user clicks within the current search session to represent user's interests for specific concepts, broad concepts and terms for results re-ranking. We represent user interests using combined ontology-based and keyword-based vectors. Usually either one of these representations is used.

Query disambiguation, query expansion, result re-ranking, results filtering, hybrid methods and collaborative adaptation are common personalized IR techniques [11]. Two popular techniques are query expansion and results re-ranking. Query expansion methods augment the query with terms that are extracted from interests/context of the user so that more personally relevant results can be retrieved. A general limitation is that if expansion terms are not selected carefully, it may degrade the retrieval performance. Conversely, in result re-ranking (rank biasing), the initial set of results are retrieved and the results are re-ranked based a user profile (i.e. profile similarity [12][13]). The aim is to push personally relevant results up in the result list.

In personalized IR, generally user's activities with the retrieval system are continuously monitored for results adaptation (i.e. Google, amazon). This approach requires explicit login by the user and storage of the user information at the server-side, which raises privacy issues. However, relying on all past user interests is tricky and often a correct subset of past interests needs to be identified for correct personalization based on the current information needs. Therefore, in our approach, we only use the current search context, hence it does not require user login. As a result, our approach provides personalization according to local choices of the user based on results categorization. A similar work is [6], which uses hierarchical page snippet clustering for personalized search. Results are categorized into hierarchical folders using gapped sentences and ODP. The user need to select a list of relevant labels related to his/her information needs. Then relevant results are filtered, and the query is expanded. In our approach, all relevant lenses are implicitly re-organized when the user selections a concept lens. This is different from the approach in [6], as it requires explicit selection of all relevant labels. In addition, we apply results re-ranking, query expansion and category suggestion within the selected concept lens as well as present the results using concept lenses rather than ranked lists of [6].

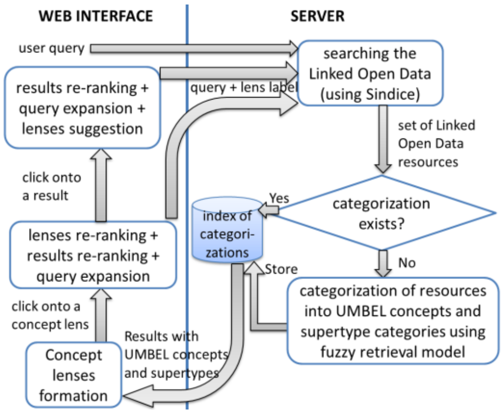
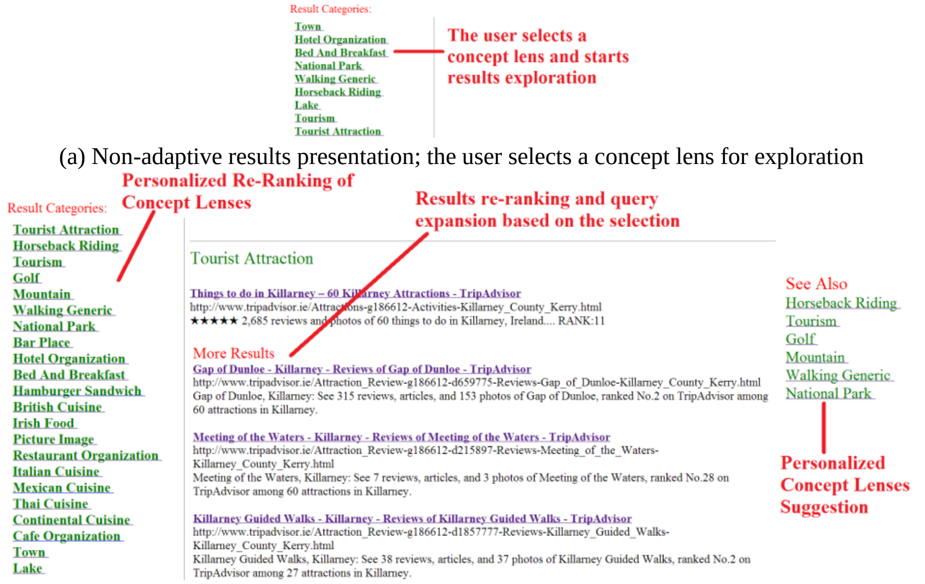

WEB INTERFACE
SERVER
user query
searching the Linked Open Data (using Sindice)
results re-ranking + query expansion + lenses suggestion
query + lens label
set of Linked Open Data resources
click onto a result
Yes
index of categori-zations
categorization exists?
lenses re-ranking + results re-ranking + query expansion
Store
No
categorization of resources into UMBEL concepts and supertype categories using fuzzy retrieval model
click onto a concept lens
Results with UMBEL concepts and supertypes
Concept lenses formation

Figure 1. The architecture and workflow.

3 Proposed Personalized Concept-based Search Framework

The proposed system architecture is shown in Figure 1. Users can provide keyword or URI queries to the system. Using the input queries, the WoD is searched. Our approach can be plugged on top of any LOD search engine (currently using Sindice search API). After receiving results, our system augments the results with UMBEL categorizations, which can be performed offline or dynamically [9]. For example, using a crawler and Sindice, LOD resources can be categorized offline by the proposed fuzzy retrieval model [8], or other clustering methods (also UMBEL linked data mappings can be used). New LOD resources are incrementally categorized and indexed at the server-side for a scalable performance [9]. In particular, we use the whole 5-depth UMBEL hierarchy (~25,000 concepts); a LOD resource may match to any concept, which is different than many personalized IR methods. Generally the top level or top 2 levels of the ontology (~200 categories of ODP) are used to represent search results. However, such an approach can only model general user interests. To achieve categorization, we extract various semantic information from context of LOD resources; type, subject, labels, property names and URL labels. Our experiments [8] on 10,000 mappings indicate that subject and type properties provide the best information for categorization, while property names add significant noise. Extracted semantic information is mapped to UMBEL concepts using a fuzzy retrieval model [8]. In order to utilize the semantic relationships and similarity present in the UMBEL vocabulary, we use the hierarchical relationships between concepts to form the vectors to represent the concepts. Vector space representation of concepts is an accepted method [13][14][16], which allows scalable performance. This provides a simple way of encoding key semantic knowledge into IR retrieval model. We use only hierarchical relationships in UMBEL as the ontology does not contain the semantic relatedness relationships between the concepts. We alleviate this issue to an extent by extracting data from subjects of LOD resources. For example, semantically related resources may share common subjects, e.g. Pope and Vatican might share Christianity and Catholic subjects. To include semantically related concepts into the categorization, we associate each LOD resources to 3 UMBEL concepts. We use the most confident concept (categorization with the highest score) for lenses creation (e.g. Pope) and the rest for semantic similarity comparison. Moreover, textual content is extracted from abstract/labels of resources to generate term vectors for combined semantic and syntactic similarity. Combining semantic and syntactic similarity provides better results [17] when the data is incomplete or poor quality (i.e. varying LOD quality).



(a) Non-adaptive results presentation; the user selects a concept lens for exploration

Tourist Attraction

Things to do in Killarney – 60 Killarney Attractions - TripAdvisor
http://www.tripadvisor.ie/Attractions-g186612-Activities-Killarney_County_Kerry.html
★★★★★ 2,685 reviews and photos of 60 things to do in Killarney, Ireland.... RANK:11

More Results

Gap of Dunloe - Killarney - Reviews of Gap of Dunloe - TripAdvisor
http://www.tripadvisor.ie/Attraction_Review-g186612-d659775-Reviews-Gap_of_Dunloe-Killarney_County_Kerry.html
Gap of Dunloe, Killarney: See 315 reviews, articles, and 153 photos of Gap of Dunloe, ranked No.2 on TripAdvisor among 60 attractions in Killarney.

Meeting of the Waters - Killarney - Reviews of Meeting of the Waters - TripAdvisor
http://www.tripadvisor.ie/Attraction_Review-g186612-d215897-Reviews-Meeting_of_the_Waters-Killarney_County_Kerry.html
Meeting of the Waters, Killarney: See 7 reviews, articles, and 3 photos of Meeting of the Waters, ranked No.28 on TripAdvisor among 60 attractions in Killarney.

Killarney Guided Walks - Killarney - Reviews of Killarney Guided Walks - TripAdvisor
http://www.tripadvisor.ie/Attraction_Review-g186612-d1857777-Reviews-Killarney_Guided_Walks-Killarney_County_Kerry.html
Killarney Guided Walks, Killarney: See 38 reviews, articles, and 37 photos of Killarney Guided Walks, ranked No.2 on TripAdvisor among 27 attractions in Killarney.

(b) Results are immediately personalized; the order of lenses are adapted, more relevant results are added by results re-ranking and query expansion, also relevant concept lenses are suggested

Figure 2. Personalized concept-based search for the query “killarney sightseeing” [7]

For a scalable performance, LOD resources’ UMBEL concepts and supertypes are also indexed at the server-side as discuss in [9]. Subsequently, uncategorized LOD resources can be dynamically categorized using asynchronous parallel requests between the client and server [9]. Categorized LOD resources (results) are sent back to the client and the results with the same concepts are grouped to form *concept lenses*. Specifically, we only use the most confident categorization to form lenses.

The user is required to select a concept lens in order to start exploring results. When the user clicks onto a concept lens, all the concept lenses are personally reorganized based on conceptual and syntactic similarity to the selected concept lens. In addition, following a lens selection, more relevant results are included based on: (i) results re-ranking using concept similarity and (ii) query expansion using the concept lens label. Moreover, when the user interacts with the results, result re-ranking, query expansion and concept lenses suggestions are provided. With our approach, personalized and conceptual result exploration is supported. This is especially useful in complex information needs, such as information gathering in an unfamiliar domain.

The client side is written using Javascript and AJAX. To support non-intrusive user modeling and adaptation, personalization is completely implemented at the client-side. Thus, it does not require user login since only user’s click data within the current search session is used. Client-side personalization is also scalable and computationally efficient since the workload is distributed to the clients and network traffic is significantly reduced. We use Sindice Search API to search the WoD and Lucene for indexing/fuzzy retrieval model. The server side is implemented with Java Servlets and uses Jena. In Figure 2, our search interface is shown. A demo can also be found at [7].

4 User Data Gathering and Search Results Personalization

4.1 Context-based User Data Gathering

User profiles can be generated from relevance feedback, implicit relevance feedback, desktop data, social Web or user's context. Generally the effectiveness of relevance feedback is limited since users are often reluctant to manually provide information. Implicit relevance feedback thus uses interactions with the system such as previous browsing activity, time spent on pages, etc. as an indication of implicit user interests. In both cases, the system needs time to gather enough information about user's all past interests. To overcome this issue, some approaches utilize desktop data or recently social web data [12], which often contains enough information about general user interests. However, relying on all past user interests is tricky and often a correct subset of past interests needs to be identified, which can be very challenging. This is because not all past interests may be important in the current context and an incorrect personalization may annoy the user experience, e.g. a user looking for hotels in Florence will not be interested to get Florence hotels in results after booking a hotel. Thus, approaches based on all past interests require fine-tuning, such as threshold selection for similarity/time decay, which may differ from various users or search scenarios. Moreover, in long-term user profiling, the extracted user interests are usually stored at the server side. This means users are required to register and login to get benefit of the personalization, which often raises privacy issues. An alternative to this approach is, no login/no storage or client storage. Client-side storage has its own issues; users may have multiple access mechanisms to the internet (especially with growing mobile access devices). Thus the user profile may be dislocated to multiple devices and the user may get different personalization experience based on the device s/he used. Finally, in the context-based user modeling, only the current available information within the current search context is utilized (i.e. query, query context, clicked results, etc.). A benefit is system only deals with few number of interests hence performance is scalable. The drawback is past user interests are lost but not all past interests are useful or identification of related interests can be challenging as we mentioned earlier.

In our approach, we use context-based user modeling rather than background knowledge. Only click data within the current search session is used to adapt to user's local choices. Search 'session start' is when the user opens the retrieval interface and 'session ends' when she closes it. The system is able to cope with changes of search domain from categorization. Suppose the user refined a query; it is probable that similar concepts/supertypes will occur in new search results. However, if the search topic changes completely, categorization in ~25,000 UMBEL concepts will not be the same, thanks to the use of whole concepts. Fortunately, supertypes can be used to understand user's general interests even if search topic change. This approach is complemented by categorization and interactive personalization. Suppose the user is interested in concept *x* and clicked onto a promising result in this lens. After a quick investigation, she deems the result irrelevant. However, this negative feedback is still very valuable thanks to categorization. By analyzing the last *N* clicks of the user on concepts/supertype concepts and the system can find similar LOD resources that share related concepts. In addition, we developed an interactive personalization where any feedback can be used. On click to a concept lens or a result, immediate personalization is supported such as lenses re-ranking, results re-ranking and query expansion.

4.2 User Profile Representation

For profiling, we track: (i) click onto a concept lens and (ii) clicks onto last *N* results.

User Concept Lens Choices: When a user clicks onto a concept lens, the results are adapted based on user's local choices. Accurate personalized re-ordering of lenses significantly affects precision @Top *N* results. Thus, robust and efficient similarity measure is essential for personalized lenses re-ranking. Similarity measures play an important role in IR, such as measuring relevance between the user's keyword query and set of pages. A majority of these measures are statistical or linguistic models for unstructured text documents. With the Semantic Web, semantic similarity measures are proposed to compare concepts and/or concept instances. They can be classified into structure and information based approaches. The structure-based methods use ontology hierarchical structure, such as edge distance between concepts. Information based methods use the shared content between concept features, e.g. comparing concepts' textual data using cosine similarity. Hybrid approaches combine both methods. For semantics-based IR, appropriate similarity measures depend on many factors, such as concepts representation (e.g. bag of words, logic predicates, etc.), search context and concept expressivity [15]. Description logic based approaches allow full expressivity but complexity can be high [22]. Overall, similarity measures depend on the application area. In our approach, we use a hybrid similarity measure combining hierarchical structure of the UMBEL vocabulary and shared statistical data between resources. We adopted vector space representation of the ontology [13][14][16] that allows efficient and scalable similarity compared to more complex description logic-based approaches. Since, performance is vital for on-time personalization.

To represent user interests, first information about all results under a concept lens are used to represent concept lens with; (a) *vector of UMBEL concepts*, i.e. user interests to specific concepts in a 5-depth hierarchy of 25,000 concepts. Unlike general approach, we represent results with very specific UMBEL categorizations. (b) *vector of supertype categories*, i.e. top-level categories to represent broad user interests. (c) *vector of terms*, i.e. terms extracted from results snippets such as title, url keywords and descriptions of the concept lens. Stop words are removed and terms are stemmed for comparison. User's interest for a concept lens is represented with three vectors. Suppose search results contain *m* concept lenses, *l*. Each concept lens, *l*, contains *n* results, *r*. Each result is represented with up to *k* UMBEL concepts, *c*, and their associated supertypes, *sc* (*k*=3 in experiments). *Vector of concepts* is calculated as:

$$\sum_{z=1}^{m} l_z = \sum_{i=1}^{n}\sum_{j=1}^{k} r_i c_j \rightarrow Vc(\vec{l_z}) = (w(c_1,l_z), w(c_2,l_z),\ldots,w(c_t,l_z)) \tag{1}$$

where it is the sum of all the concepts that all results contain under a concept lens. Here, each dimension of $w(c_1, l_z)$ corresponds to a separate UMBEL concept or supertype category and its weight. A similar method is used for calculating *vector of supertypes*. We use concept/supertype frequency as weight, i.e. if a concept does not occur in the concept lens, the value is 0. Generally term frequency, inverse document frequency ($tf \times idf$) is used for weighting. However, our studies show that the frequency, tf , works better $tf \times idf$. idf weights rare terms (or concepts) higher. This works well for retrieval, but not for similarity comparison as we compute the shared information between the lenses. In the same manner, results' snippets are combined to form a *vector of terms* of the concept lenses (eq. 2). Each dimension corresponds to a unique term and its weight. Again we use the term frequency as weight.

$$\sum_{z=1}^{m} l_z = \sum_{i=1}^{n} r_i t \rightarrow \vec{Vt}(l_z) = (w(t_1,l_z), w(t_2,l_z), \ldots, w(t_s,l_z)) \qquad (2)$$

User Interests: For interactive personalization of the results, we need to capture the user's information need from the clicked results, which are then used for results reranking and query expansion. For this purpose, we track last *M* clicks of the user, *u*, and generate three types of vectors to represent the user's information need: vector of concepts (specific interests), vector of supertypes (broad interests), vector of terms (language model). In this case, vectors are extracted from the last *M* results clicks;

4.3 Re-Organization of Concept Lenses

Dynamic adaptation of results to the local user choices is the most innovative personalization provided by our system. This allows dynamic results adaptation to local user choices, which moves conceptually relevant concept lenses to the top of the list. To achieve this, we compare the similarity of the selected concept lens to other concept lenses using the cosine similarity of concept, supertype and term vectors:

$$sim(l_1,l_2) = \vec{V}_1(l_1).\vec{V}_2(l_2) / |\vec{V}_1(l_1)||\vec{V}_2(l_2)| \qquad (3)$$

where $sim(l_1,l_2) \in [0,1]$, numerator is the inner product of the vectors and the denominator is the multiplication of the vector magnitudes. We generate three similarity scores for each concept lens, namely *c_sim*, *s_sim* and *t_sim* according to their similarity to the selected concept lens, l_s . The concept similarity (*c_sim*) compares similarity of shared specific concepts, i.e. if lenses share more specific concepts, it is more likely that they are relevant. The supertype similarity (*s_sim*) computes shared broad concepts. For example, "mountain" and "lake" concept lenses have the same supertype category and they broadly related. Finally, the term similarity (*t_sim*) allows comparing language models of the lenses. This information can be noisy since different resources may share similar meanings but may use different terms. However, still term similarities can be used to guarantee some level of similarity between lenses.

Our evaluations on the benchmark dataset showed that the concept vector similarity of lenses provided the best precision @top N concept lenses compared to supertype and term similarities (see section 5). In addition, when different similarity scores were combined, precision was improved. In particular, when the influence of the *c_sim* was weighted higher than the *s_sim* and *t_sim*, the best precision @top N concept lenses was obtained. Especially the best results were obtained when $\alpha = 2, \beta = 1$ and $\delta = 1$. If $sim(l, l_s) > 0.2$, the concept lens is suggested for exploration as shown in Figure 2.

$$sim(l, l_s) = \left(\alpha * c\_sim(l, l_s) + \beta * s\_sim(l, l_s) + \delta * t\_sim(l, l_s)\right) / \left((\alpha + \beta + \delta)\right) \qquad (4)$$

Finally, concept lenses are re-ranked in decreasing $sim(l, l_s)$ order. By default, the selected lens came on top of the list since cosine similarity of a vector to itself is 1.

4.4 Results Re-ranking and Concept Lenses Suggestion

For results re-ranking, each result is represented with a vector of concepts, supertypes and terms: $\overrightarrow{Vc}(r) = (w(c_1, r), \ldots, w(c_x, r))$, $\overrightarrow{Vsc}(r) = (w(sc_1, r), \ldots, w(sc_w, r))$, $\overrightarrow{Vt}(r) = (w(t_1, r), \ldots, w(t_y, r))$. We apply results re-ranking in two cases: (a) when the user selects a concept lens from the results list for exploration and (b) when the user clicks onto a result (LOD resource) within a concept lens. In both cases, the re-ranked results are included in the context of the interacted concept lens. This allows in context results exploration thanks to the use of concept lenses. In case (a), we compare concept vector, of the selected concept lens (l_s) with the top *K* results using eqs. (3), (5);

$$\sum_{i=1}^{K} sim(\overrightarrow{Vc}(r_i), \overrightarrow{Vc}(l_s)) \qquad (5)$$

We compare concept vectors since results matching at specific UMBEL concepts are more likely to be relevant compared to supertype or term similarities (we only have user's interest for a concept lens). In our experiments, *K*=100, for a scalable performance. Results, where $sim(\overrightarrow{Vc}(r), \overrightarrow{Vc}(l_s)) > \alpha$, are re-ranked in decreasing order and added into the interacted concept lens. $\alpha = 0$; any match was considered because of specific concept vectors comparison. Later, α can be determined experimentally.

In case (b), we use the click history of the user within the current search session to re-rank results. In particular, from the last *M* results clicks of the user, user's specific concept, supertype and term interests are represented as vectors. These vectors are compared with top *K* result vectors using eqs. (3), (6);

$$\sum_{i=1}^{K} \frac{\alpha * sim(\overrightarrow{Vc}(r_i), \overrightarrow{Vc}(u)) + \beta * sim(\overrightarrow{Vsc}(r_i), \overrightarrow{Vsc}(u)) + \delta * sim(\overrightarrow{Vt}(r_i), \overrightarrow{Vt}(u))}{\alpha + \beta + \delta} \qquad (6)$$

where three similarity scores are combined for re-ranking of the results in decreasing order. Especially, $\alpha = 2$, $\beta = 1$ and $\delta = 1$ gives better results. Again, a threshold can be used to select relevant results conservatively, i.e. higher thresholds. If a relevant result belongs to another concept, then the concept lens is suggested for exploration.

4.5 Query Refinement using Concept Labels

Query adaptation is applied in two cases: (i) when the user selects a concept lens from the results list for exploration and (ii) when the last two consecutive result clicks share the same concept. In both cases, we assume that the user is interested in this concept and we expand the original query with the concept label that the user is interested. It is a simple approach, but works well since UMBEL categorizations provide very specific concept names and it can be used to clarify the meaning of the query with the user feedback. In both cases, more results are included in the context of the interacted concept lens, so that the user can explore more relevant results in context.

5 Evaluations

In traditional IR, there are public benchmarks for standardized evaluation and comparison (i.e. TREC). However, there are no standard evaluation benchmarks for semantic search evaluations [13]. Current semantic search methods are based on user-centered evaluation, which tend to be high-cost, non-scalable and difficult to repeat. [13] proposes to use TREC for cross-comparison between IR and ontology-based search models. They annotate TREC collections with instances of ontology concepts. However, it was found that only 20% TREC search topics have semantic matches in 40 public ontologies. Thus it can be difficult to apply this technique in many topic domains. Although a similar approach of [14] can be adopted, they rely on semantic annotation of *documents*. In our approach, we focus on categorization of LOD resources as the basis of the personalization and visualization rather than semantic annotations of documents. Thus, we created a benchmark dataset using LOD resources, which is available online for validation and comparison [21]. We measured personalized search efficacy using precision @top *M* concept lenses and @top *N* results. We focused on precision since our aim is to improve precision on the top results/lenses.

Dataset. For the experiment, we selected tourism domain. Because our aim is not just providing direct answers to a search query but to support results exploration with categorization and personalization. The tourism domain suits such data gathering and informational queries, since the user has a vague idea about queries and gradually refines queries to gather/explore more information. This scenario is also fits for WoD search, since developers usually explore WoD to gather data about a specific domain.

Our dataset is about "tourism in Killarney Ireland" and it was created as follows: One option was to use Sindice for dynamic querying. However, Sindice search results may change due to dynamic indexing. Thus, we decided to index a particular dataset for stable and comparative evaluations. First, we investigated popular search queries about the domain from Google search trends. Then, these queries were used to query WoD with Sindice to gather data about available URIs. Particularly ~500 URIs from DBpedia, GeoNames, Trip Advisor and ookaboo domain were selected. RDF descriptions of the URIs, their UMBEL/supertype categorizations were indexed offline by the proposed fuzzy retrieval model [8] to carry the experiments. Then, we selected 20 queries, which do not have a direct answer, i.e. navigational queries were not selected, such as "Killarney Victoria Hotel". Although 20 queries is a small sample set, such sizes have been used before to determine indicative results in semantic search [13] [14]. Top concept lenses and results returned by the queries, were manually assigned relevant or irrelevant. For non-adaptive baseline systems, we used the same dataset.

5.1 Personalization Time Overhead

Results categorization is applied offline during the indexing of LOD resources. Thus, we computed average time required to generate personalized results (i.e. lenses re-organization, results re-ranking and query expansion following a lens selection). For each query, average of 5 runs used. Results showed that personalized results were obtained within an average of 0.26 seconds compared to 0.16 of non-adaptive case. Our personalization is scalable thanks to complete client-side implementation. The results were run on Windows 7 computer, 2.2GHz CPU and 7.90GB RAM.

5.2 Performance of Personalization Strategies

In the experiments, personalization was performed after the user's concept lens selection following a query. To evaluate the efficiency of personalized lenses re-ranking, precision at top *M* concepts was measured, which was adopted by [6]. Precision at top *M* concepts is: *P@M=R@M / M*, where *R@M* is the number of concept lenses which have been manually tagged relevant among top *M* concept lenses. For ambiguous concept lenses, if the majority of results under the lens were relevant, then we judge relevant. We use P@1, P@3, P@5, P@10 and P@15, since lazy users browse top concept lenses. The results in Figure 3 (left) show that for lenses re-ranking, lenses' concept vector similary provided the best precision compared to supertype and term vector similarities. When various similarity scores were combined, the precision was improved (in the experiment, influence of concept similarity is higher than others, e.g. $\alpha = 2, \beta = 1$ and $\delta = 1$ in eq. (8)). The best personalized lenses re-ranking was obtained when concept, supertype and term vector similarities were combined.

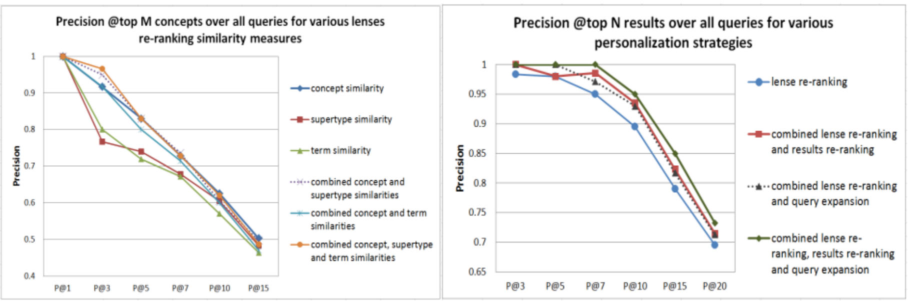


Figure 3. Precision @top N concepts over all queries for lenses re-ranking similarity measures (left). Precision @top *N* results over all queries for various personalization strategies (right)

In a similar manner, we measured precision at top *N* results for different personalization strategies: *P@N = R@N / N*, where *R@N* is the number of results which have been manually tagged relevant among top *N* results as shown in Figure 3 (right). The results showed that lenses re-ranking significantly improve precision @top *N* results. Combined lenses re-ranking with results re-ranking or query expansion improve lenses re-ranking performance. This also shows that personalized re-ranking of results and query expansion with concept lens label work well. When all personalization strategies were combined, the best results were obtained, where 100% precision at P@3, P@5 and P@7 were obtained on the tested 20 queries.

5.3 Comparison with Non-Adaptive Concept-based Search and Ranked Lists

We compared personalized search performance against non-adaptive concept-based search and ranked list presentation. Here, the non-adaptive concept-based search present the results without adaptation to the user's selected concept lens, i.e. there is no lenses re-ordering, results re-ranking and query expansion. Whereas, the ranked result lists uses the original rank of the result and present it without categorization. First, we evaluated non-adaptive concept lenses ordering. Lenses can be ordered based on; (a) the minimum result rank within a lens, or (b) average of all results' ranks within it. Results in Figure 4 (left), show that both cases provide similar results. However, for the minimum rank order, P@1 is slightly better than the average rank. Thus, we used the minimum order for comparison with personalized lenses re-ranking. The personalized re-ordering of lenses significantly improved precision @top *M* concept lenses compared to the non-adaptive concept lenses as shown in Figure 4 (right). In a similar manner, we compared our personalized search on precision @top *N* results against non-adaptive concept-based search and traditional ranked list presentation (Figure 5). The results showed that our personalized search outperforms precision at all levels compared to non-adaptive concept-based search and traditional rank lists.

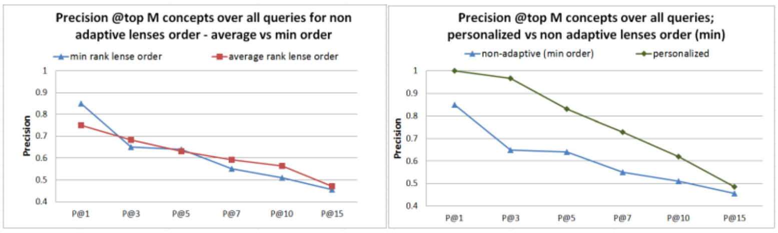


Figure 4. Precision @top *M* concepts over all queries: Left; non-adaptive lenses ordering using minimum vs. average rank. Right; comparison of personalized vs. non-adaptive lenses ordering

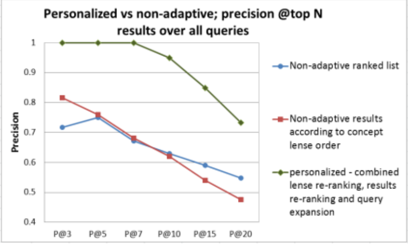


Figure 5. Precision @top N results over all queries for personalized and non-adaptive search

6 Conclusions and Future Work

We introduced a novel personalized search and exploration mechanism for the Web of Data based on concept-based results categorization. In our approach, search results (LOD resources) are conceptually categorized to form *concept lenses*, which assist exploratory search/browsing. When the user selects a concept lens, results are immediately personalized; lenses are re-organized, more relevant results are included using results re-ranking and query expansion, as well as, relevant lenses are suggested for exploration. This is an innovative feature offered by our approach since it allows dynamic results adaptation to the user's local choices. Our personalization is privacy preserving, non-intrusive and scalable since it does not require user login and implemented at the client-side. Evaluations showed that the proposed approach significantly enhances precision compared to non-adaptive concept-based search and ranked list. In future, we will perform user studies to evaluate usability of our approach. In addition, data quality, trust and graph popularity can be considered in rankings.

Acknowledgments. This research is supported by the SFI part of EMPOWER research fellowship (07/CE/I1142) and part of the CNGL (www.cngl.ie) at University of Dublin.

7 References

1. Delbru, R., S., Campinas, G., Tummarello: Searching Web Data: an Entity Retrieval and High-Performance Indexing Model. Journal of Web Semantics, 10, 33-58, 2012.
2. D'Aquin, M., E., Motta, M., Sabou, S. Angeletou, L., Gridinoc, V., Lopez and D., Guidi: Toward a New Generation of Semantic Web Applications. IEEE Intelligent Systems, 2008.
3. Tummarello, G., R. Cyganiak, M. Catasta, S. Danielczyk, R. Delbru and S. Decker: Sig.ma: live views on the Web of Data, Journal of Web Semantics, 8(4), 355-364, 2010.
4. Heim, P., T. Ertl and J. Ziegler: Facet Graphs: Complex Semantic Querying Made Easy, Extended Semantic Web Conference, LNCS, 6088, 288-302, 2010.
5. Hogan, A., J. Umbrich, A. Harth, R. Cyganiak, A. Polleres, S. Decker: "An empirical survey of Linked Data conformance. Journal of Web Semantics, 14, 14–44, 2012.
6. Ferragina, P., and A., Gulli: 2005. A Personalized Search Engine Based on Web-Snippet Hierarchical Clustering. International World Wide Web Conference (WWW),801-810,2005.
7. Demo: `https://www.scss.tcd.ie/melike.sah/ESWC2013demo.swf`
8. Sah, M., V. Wade: A Novel Concept-based Search for the Web of Data using UMBEL and a Fuzzy Retrieval Model. Extended Semantic Web Conference, 7295, 103-118, 2012.
9. Sah,M.,V.Wade: A Novel Concept-based Search for the Web of Data.I-SEMANTICS, 2012.
10. Jansen, B.J., D.L., Booth, A. Spink: Determining the User Intent of Web Search Engine Queries. International Conference on World Wide Web, 1149-1150, 2007.
11. Ghorab, M.R., D. Zhou, A. O'Connor, V. Wade: Personalised Information Retrieval: Survey and Classification. Journal of User Modeling and User-Adapted Interaction (to appear).
12. Sieg, B. Mobasher, R. Burke: Web Search Personalization with Ontological User Profiles. International Conference on Information and Knowledge Management (CIKM), 2007.
13. Fernandez, M.,V. Lopez, M. Sabou, V. Uren, D. Vallet, E. Motta, P. Castells: "Using TREC for cross-comparison between classic IR and ontology-based search models at a Web scale. Workshop on Semantic Search (SemSearch 2009), WWW 2009.
14. M. Fernandez, I. Cantador, V. Lopez, D. Vallet, P. Castells, E. Motta: "Semantically enhanced Information Retrieval: an ontology-based approach". JWS, 9(4), 434-452, 2011.
15. Janowicz, K., M. Raubal, W. Kuhn, "The semantics of similarity in geographic information retrieval", Journal of Spatial Information Sciences, No. 2, 29-57, 2011.
16. Tous R.,and J. Delgado. "A vector space model for semantic similarity calculation and OWL ontology alignment". Int.Conf. on Database and Expert Systems Applications(DEXA),2006.
17. Giunchiglia, F., U. Kharkevich, I. Zaihrayeu: Concept Search, ESWC, 429-444, 2009.
18. Mirizzi,R., A. Ragone, T. Di Noia, and E. Di Sciascio. "Semantic wonder cloud: exploratory search in DBpedia". International Conference on Web Engineering (ICWE), 138-149, 2010.
19. Tvarozek, M. and M. Bielikova. "Factic: personalized exploratory search in the semantic web". International Conference on Web Engineering (ICWE), 527-530, 2010.
20. Shneiderman, B. "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations". IEEE Symposium on Visual Languages, 1996.
21. Benchmark: `https://www.scss.tcd.ie/melike.sah/tourismdataset.zip`
22. D'Amato, C., N. Fanizzi, and F. Esposito. "A semantic similarity measure for expressive description logics". Convegno Italiano di Logica Computazionale (CILC), 2005.