A Novel Concept-based Search for the Web of Data using UMBEL and a Fuzzy Retrieval Model

Melike Sah and Vincent Wade
Knowledge and Data Engineering Group, Trinity College Dublin, Dublin, Ireland
{Melike.Sah, Vincent.Wade}@scss.tcd.ie

Abstract. As the size of Linked Open Data (LOD) increases, the search and access to the relevant LOD resources becomes more challenging. To overcome search difficulties, we propose a novel concept-based search mechanism for the Web of Data (WoD) based on UMBEL concept hierarchy and fuzzy-based retrieval model. The proposed search mechanism groups LOD resources with the same concepts to form categories, which is called *concept lenses,* for more efficient access to the WoD. To achieve concept-based search, we use UMBEL concept hierarchy for representing context of LOD resources. A semantic indexing model is applied for efficient representation of UMBEL concept descriptions and a novel fuzzy-based categorization algorithm is introduced for classification of LOD resources to UMBEL concepts. The proposed fuzzy-based model was evaluated on a particular benchmark (~10,000 mappings). The evaluation results show that we can achieve highly acceptable categorization accuracy and perform better than the vector space model.

Keywords: Categorization, concept-based search, data mining, semantic indexing, fuzzy retrieval model, linked open data, UMBEL concept hierarchy.

1 Introduction

A key research focus in Web technology community is Linked Data. The term Linked Data describes best practices for creating typed links between data from different sources using a set of Linked Data principles. This ensures that published data becomes part of a single global data space, which is known as “Web of Data” (WoD) or “Linked Open Data” (LOD). Since the data is structured and relationships to other data resources are explicitly explained, LOD allows discovery of new knowledge by traversing links. However, as the number of datasets and data on the LOD is increasing, current LOD search engines are becoming more important to find relevant data for further exploration. This is analogous to the problem of the original Web [1]. However, current LOD search mechanisms are more focused on providing automated information access to services and simple search result lists for users [2, 3]. For example, they present search results in decreasing relevance order based on some criterion (i.e. relevance to class names). However, result list based presentations of retrieved links/resources do not provide efficient means to access LOD resources since URIs or titles of LOD resources are not very informative. More efficient access and discovery mechanisms on the WoD are crucial for finding starting points for browsing and exploring potential data/datasets by Web developers and data engineers. Currently, there are few approaches, which investigate this problem [1, 4, 11].

Our objective is to improve current search mechanisms on the WoD with a novel concept-based search method. The dictionary definition of *concept* is "*a general notion or idea*". Concept-based search systems provide search results based on the meaning, general notion of information objects so that search results can be presented in more meaningful and coherent ways. Key challenges in supporting concept-based search are: (1) the availability of a broad conceptual structure, which comprises good concept descriptions, (2) extraction of high-quality terms from LOD resources for the representation of resource context and categorization under the conceptual structure, and (3) a robust categorization algorithm. In this paper, we focus on these issues in order to introduce a novel concept-based search mechanism for the WoD.

1.1 Our Approach and Contributions

We introduce a novel concept-based search mechanism for the WoD based on the UMBEL concept hierarchy (http://umbel.org/), a fuzzy-based retrieval model and a categorical result list based presentation. The proposed search mechanism groups LOD resources with the same concepts to form categories, which we call *concept lenses*. In this way, search results are presented using categories and concept lenses, which can support more efficient access to the WoD. Such categorization enables the generation of much more human intuitive presentations, aggregations and concept-based browsing of retrieved links aligned to the users' intent or interests. It can offer a considerable improvement over a single ranked list of results. Such presentations of links can allow more effective personalized browsing and higher user satisfaction [9].

There are three unique contributions of our approach: (1) For the first time, UMBEL is used for concept-based Information Retrieval (IR). UMBEL provides a rich concept vocabulary that is linked to DBpedia and designed to enable reasoning and browsing. (2) A second contribution is in novel semantic indexing and fuzzy retrieval model, which provides efficient categorization of search results in UMBEL concepts. This approach extends the traditional vector space model, which uses term frequency (*tf*) and inverse document frequency (*idf*) ($tf \times idf$) approach in indexing and retrieval to enable fuzzy relevancy score calculation according to relevancy of a term to semantic elements (structure) of concept(s). This significantly extends traditional $tf \times idf$ calculations. (3) A minor contribution is the realization of a concept-based search approach to WoD exploration. Concept-based search has only traditionally occurred in Web IR rather than in the realm of linked data.

The remainder of the paper is organized as follows: Section 2 discusses the related work. Section 3 discusses conceptual vocabularies for concept-based IR and explains why UMBEL was chosen rather than other well-known conceptual structures. Section 4 introduces the proposed concept-based search. In particular, term extraction from LOD resources, a novel semantic indexing and a novel fuzzy retrieval model is introduced for the categorization of LOD resources in UMBEL concepts. Section 5 presents evaluations prior to conclusions. Specifically, the proposed fuzzy retrieval model was evaluated on a particular benchmark from DBpedia to UMBEL mappings (~10,000), which achieved high categorization accuracy (~89%) and outperformed the vector space model (~33%), which is crucial for the correct formation of concept lenses. Moreover, time evaluations were performed to measure system performance.

2 Related Work

2.1 Concept-based and Clustering-based Information Retrieval (IR) Systems

Most of the current search engines utilize keyword-based search algorithms (i.e. full-text search) for information retrieval [2, 3]. Although this method is simple and fast, it often produces the problem of high recall and low precision, because it mainly finds documents that contain query keywords. Concept-based IR aims to improve retrieval effectiveness by searching information objects based on their meaning rather than on the presence of the keywords in the object. In concept based search, query context and context of information objects are represented with a reference concept hierarchy. Thus relevant information objects can be retrieved and results can be re-ranked (personalized) based on user's query context [5, 6]. Different than existing concept-based IR systems, which is based on results re-ranking [5, 6], our objective is to use concepts of LOD resources for categorical links presentation (concept lenses), which also has a key benefit of supporting concept-based browsing and discovery. On the other hand, since it is expensive and difficult to create broad conceptual structures, concept-based search approaches use existing conceptual structures, such as Yahoo Directory [7] and Open Directory Project (ODP) [5, 6]. With the increasing number of LOD ontologies and metadata, interests in using these taxonomies are decreasing.

Other relevant work is clustering search engines that group pages into topics (e.g. Carrot[2] – http://search.carrot2.org/stable/search) or hierarchical topics (e.g. Clusty – http://search.yippy.com/) for efficient information access/discovery. In these methods, challenge is the creation of useful topics that is achieved by clustering the retrieved pages using complex clustering algorithms [10]. Whereas in our work, a taxonomy is used for topic labels (which solve vocabulary problem)and our focus is categorization of individual resources to the known concepts (i.e. reverse of clustering techniques).

2.2 Search Mechanisms on the WoD

Current WoD search engines and mechanisms, such as Sindice [2] and Watson [3], utilize full-text retrieval, where they present a list of search results in decreasing relevance. However, users cannot understand "what the resource is about" without opening and investigating the LOD resource itself, since the resource title or example triples about the resource are not informative enough. More efficient search mechanisms on the WoD are crucial for finding starting points and exploration of potential data/datasets by Web developers and data engineers. Sig.ma attempts to solve this problem by combining the use of Semantic Web querying, rules, machine learning and user interaction [1]. The user can query the WoD and Sig.ma presents rich aggregated mashup information about a particular entity. Our approach is however focused on a novel concept-based presentation of search results using categories, which is different than mashup-based presentation of a particular resource.

Another related work is faceted search/browsing systems [4, 11], which is also known as exploratory search systems [14]. A key important difference between our "concept lenses" and faceted search is that we generate concept lenses based on a reference concept hierarchy rather than particular datatype or object properties of LOD resources. Thus our approach can be applied to heterogeneous LOD resources that use different schemas. In contrast, faceted search systems are generally bound to specific schema properties and it can be difficult to generate useful facets for large and heterogeneous data of the WoD [12]. In addition, scalability and system performance is another issue for faceted systems over LOD [13].

3 Conceptual Structures and Vocabularies for Concept-based IR

In concept-based IR, existence of a conceptual structure for representing context of an information object and query is crucial. The conceptual structures can range from simple thesaurus, dictionaries to more complex semantically rich ontologies. Yahoo Directory (http://dir.yahoo.com/), ODP (http://www.dmoz.org/), Proton (http://proton.semanticweb.org/), SUMO (http://www.ontologyportal.org/) and Sensus (http://www.isi.edu/natural-language/projects/ONTOLOGIES.html) are examples that have been utilized for concept-based search. SUMO and Proton have relatively sparse subject concepts and their penetration into general Web is quite limited. Sensus is a concept ontology derived from WordNet. However, Sensus does not include much semantic information, which can be very useful for concept-based IR. Yahoo and ODP are by far the most commonly used taxonomies for concept-based IR [5-8]. However, with the increasing number of LOD ontologies and metadata, usage of Yahoo and ODP has significantly decreased. Therefore, we looked for candidates in LOD ontologies such as DBpedia (http://dbpedia.org/), Yago (http://www.mpi-inf.mpg.de/yago-naga/yago/), OpenCyc (http://www.opencyc.org/) and UMBEL.

DBpedia and Yago's named entity coverage is good but their content does not have a consistent backbone structure[1], which makes it difficult to use and reason. OpenCyc is another upper ontology generated manually over the last twenty years. It captures common sense knowledge and a broad number of concepts. In addition, OpenCyc is purposefully created to support inferencing such as it uses WordNet for concept disambiguation and it captures subject relationships between concepts to enable reasoning. However, OpenCyc's top ontology concepts are obscure and contain many domain specific concepts that are developed for project purposes.

UMBEL is a cleaner and simpler sub-set of OpenCyc with the specific aim of promoting interoperability with external linked datasets. UMBEL provides a coherent framework of broad subjects and topics (around 28,000 concepts) with useful relationships and properties drawn from OpenCyc (i.e. broader, narrower, external, equivalent classes and preferred, alternative, hidden labels). In addition, UMBEL concepts are organized into 32 super type classes, which make it easier to reason, search and browse. Moreover, UMBEL is connected to/from OpenCyc and DBpedia (which is also linked to Yago through DBpedia).

On the other hand, recently Google, Yahoo and Bing announced schema.org, which is a markup vocabulary for annotating Web pages, so that search engines can improve presentation of search results. The vocabulary contains broad concepts, which can be useful if they publish the training data in future.

Based on current data, UMBEL's broad concept coverage, rich representation of concept descriptions and powerful reasoning capabilities stand out among other LOD conceptual ontologies. Thus, we propose to use UMBEL for concept-based search.

[1] DBpedia and Yago provide rich structures for linking instance data. However they do not have a consistent framework of concepts (topics) for representing those instances

4 Proposed Concept-based Search on the Web of Data

The proposed concept-based search mechanism is fully implemented[2] and its system architecture is shown in Figure 1. Users can provide keyword or URI based queries to the system. Using these input queries, our system search the WoD by utilizing Sindice search API [2] and initial search results from the Sindice search are presented to users with no categorization. Then, for each search result (LOD URI), parallel requests are sent to the server for categorization of LOD resources under UMBEL concepts. First, for each LOD resource, its RDF description is cached to a Jena model using the Sindice Cache (i.e. http://any23.org/) at the server. Using the RDF description, terms from different semantic parts of the LOD resource are mined and the extracted terms are weighted for matching to UMBEL concept representations. In particular, UMBEL concepts are represented by a semantic indexing model (see Section 4.2). The obtained terms from the LOD resource are matched to the inverted concept index by a fuzzy-based retrieval algorithm (see Section 4.3) and categorized LOD resources are sent back to the client. Since dynamic categorization can be time consuming, search results are shown incrementally by using Asynchronous JavaScript (AJAX) to enhance user experiences with the system. Finally, LOD resources with the same concepts (i.e. resources with same categories, e.g. Organization) are grouped together to form concept lenses for coherent presentation of search results. LOD resources with no categorization are presented without a category in the result lists.

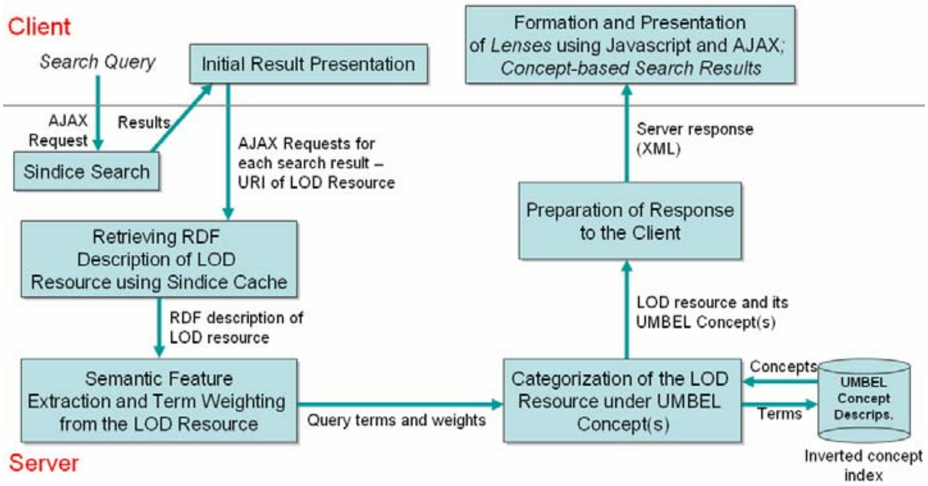

Client
Search Query
Initial Result Presentation
Formation and Presentation of *Lenses* using Javascript and AJAX; *Concept-based Search Results*
AJAX Request
Results
Sindice Search
AJAX Requests for each search result – URI of LOD Resource
Server response (XML)
Preparation of Response to the Client
Retrieving RDF Description of LOD Resource using Sindice Cache
RDF description of LOD resource
LOD resource and its UMBEL Concept(s)
Semantic Feature Extraction and Term Weighting from the LOD Resource
Query terms and weights
Categorization of the LOD Resource under UMBEL Concept(s)
Concepts
Terms
UMBEL Concept Descrips.
Inverted concept index
Server

Fig. 1. System Architecture

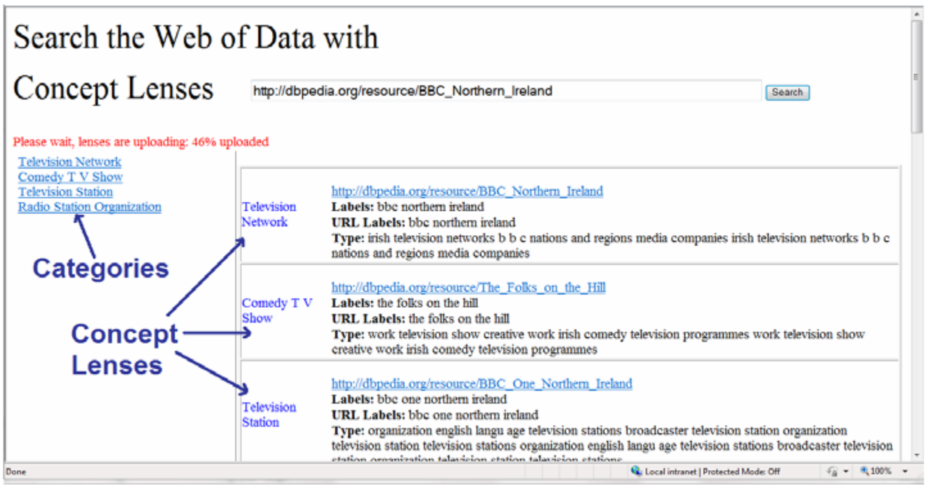
Given that indexing and caching of WoD is very expensive, our approach is based on existing 3rd party serives. In particular, we use Sindice search for querying the WoD and Sindice Cache for retrieving RDF descriptions of LOD resources [2]. Lucene IR framework is utilized for indexing of concepts and at the implementation of the fuzzy retrieval model. The server side is implemented with Java Servlets and uses Jena for processing RDF. The client side is written using Javascript and AJAX.

[2] It will be made public. A video demo is available at http://www.scss.tcd.ie/melike.sah/concept_lenses.swf

In Figure 2, a screen shot of the concept-based search interface is presented. The user interface presents the list of categories (concepts) at the left of the screen for quick navigation access to concept lenses (LOD resources grouped based on context).

4.1 Recognizing Context of Linked Open Data Resources

In order to generate concept-based search results, first the retrieved LOD resources from the Sindice search need to be categorized under UMBEL concepts. To achieve this, the concepts of LOD resources should be understood, where lexical information about LOD resources can be used to mine such knowledge. One option is to extract all lexical information from the URI, labels, properties and property values of the LOD resources that are retrieved by Sindice search. However, in such a process, many misleading words may also be extracted. For example, a LOD resource about a TV broadcasting organization may include information about broadcasting network but it may also include other information about programs, coverage, etc., which may lead to an incorrect categorization. Thus, the challenge is to identify important features of LOD resources for correct categorization.


Search the Web of Data with Concept Lenses

http://dbpedia.org/resource/BBC_Northern_Ireland

Please wait, lenses are uploading: 46% uploaded

Television Network
Comedy T V Show
Television Station
Radio Station Organization

Television Network
http://dbpedia.org/resource/BBC_Northern_Ireland
Labels: bbc northern ireland
URL Labels: bbc northern ireland
Type: irish television networks b b c nations and regions media companies irish television networks b b c nations and regions media companies

Comedy T V Show
http://dbpedia.org/resource/The_Folks_on_the_Hill
Labels: the folks on the hill
URL Labels: the folks on the hill
Type: work television show creative work irish comedy television programmes work television show creative work irish comedy television programmes

Television Station
http://dbpedia.org/resource/BBC_One_Northern_Ireland
Labels: bbc one northern ireland
URL Labels: bbc one northern ireland
Type: organization english langu age television stations broadcaster television station organization television station television stations organization english langu age television stations broadcaster television station organization television station television stations

Fig. 2. The user interface of the concept-based search mechanism – categories (concepts) at the left of the screen and concept lenses in the main panel of the screen.

Term Extraction and Enrichment. We chose the following common features of LOD resources that may be used to represent context of the resource – *URI* (*u*), *label* (*l*), *type* (*t*), *subject* (*s*) and *property names* (*p*). We chose these features for the following reasons: the *URI* of a resource may contain keywords relevant to the context of the resource. Titles (dc:title), names (foaf:name) and labels (rdfs:label), so called *label* features usually include informative information about the resource. *property names* typically provide further knowledge about "what type of the resource is". For example, birth place, father, mother, spouse are properties associated with persons. However, some property names are generic (i.e. rdfs:label, rdf:type, foaf:name, etc.) and do not provide information about the context. To overcome this, we compiled a list of generic property names and if a property name matches any of these, it is not accepted. On the other hand, *type* (rdf:type and dc:type) and *subject* (dc:subject) provides the most discriminative features about the context of a LOD resource. For instance, *type* and *subject* values can provide useful knowledge about concepts (general notion or idea) of the resource for correct categorization. For instance, for label “ocean” the context is not clear. But if we know *type* is album, then we can understand that “ocean” is a label of an “album”.

From each LOD resource, keywords are extracted from the features as explained above. Then, qualifiers and propositions are removed from the extracted keywords, to enhance categorization accuracy. For instance, for the context “hockey games in Canada”, the main concept is “hockey games” and not the “country” Canada. Thus, we remove qualifiers/ propositions and the words after them for better term extraction. Qualifier removal is based on keyword matching of *from*, *in*, *of* and *has*. To ensure correct matching, there must be a space before/after the qualifiers, e.g. words in italic are removed after the qualifiers: people *from Alaska*, reservoirs *in Idaho*, mountains *of Tibet*, chair *has four legs*. This has the effect of generalizing the concepts, which is perfectly reasonable for our purpose of categorizing and browsing based on higher level of concepts.

On the other hand, after initial experiments, we observed that many LOD resources do not have information about *label*, *type* and *subject*. To improve lexical data mining, we also apply a semantic enrichment technique, where more lexical data is gathered from the linked data graph of the resource by traversing owl:sameAs and dbpedia:WikiPageRedirect links. If a resource has such links, first we obtain RDF description of these resources and apply the feature extraction techniques explained above. Finally, from the obtained and enriched terms, stop words are removed and stemming is applied, where we obtain the final terms, which we call *LOD terms*.

LOD Term Weights based on Features. Since different *LOD terms* have comparative importance on the context of the LOD resource, terms are weighted. For example, *type* and *subject* features provide more discriminative terms. Therefore terms which appear in these features should be weighted higher. To achieve this, we divided LOD resource features into two groups: Important features (*I*) and Other features (*O*). For important features terms from *type* and *subject* features are combined to form a vector. For other features terms from *URI*, *label* and *property name* are combined to form a vector. We use the normalized term frequency (*tf*) of these features for term weighting as given below,

$$if\ t \in I,\ w(t) = 0.5 + 0.5 \times \left(\frac{tf\,(t)_I}{\max(tf_I)} \right), \quad if\ t \in O,\ w(t) = \frac{tf\,(t)_O}{\max(tf_O)} \tag{1}$$

where $w(t)$ is weight of term t, and $tf(t)_I$ and $tf(t)_O$ is term frequency of term t in important and other features respectively. $\max(tf(t)_I)$ and $\max(tf(t)_O)$ represent maximum term frequency in those features. For terms that appear in important features (*I*), a minimum weight threshold of 0.5 is used to encourage syntactic matches to these terms within UMBEL concept descriptions for categorization. On the other hand, inverse document frequency (*idf*) can also be used together with term frequency. However, *idf* calculation for each LOD term is expensive. For dynamic *idf* calculations, dynamic search on the LOD is required for each term, which is computationally expensive. For offline calculations of *idf*, we need to continuously index the LOD, which is a resource-intensive task. Thus, we did not use *idf*.

4.2 Representation of Umbel Concept Descriptions

In UMBEL version 1.0, there are 28,000 UMBEL concepts, which are categorized under 32 top-level *Supertype* classes (i.e. Events, Places, etc.) and classified into a taxonomy using super and sub-concepts. Each UMBEL concept description contains preferred label and alternative labels. Alternative labels usually include synonyms, quasi-synonyms, lexical variations, plural, verb derivations and semantic related words. In summary, UMBEL provides highly structured descriptions of a broad number of concepts, which can be used to represent the context of LOD.

Semantic Indexing Model. The formal representation of concept descriptions plays a crucial role in the effectiveness of the IR system. In general, concepts' representations are based on extraction of keywords and the usage of a weighting scheme in a vector space model ($tf \times idf$). This provides a simple but robust statistical model for quick retrieval [8]. For indexing of UMBEL concept descriptions, $tf \times idf$ weighting scheme is used similar to other concept-based IR models [5-7]. Typically these IR models utilize vector space representations of categories (concepts), where the terms inside the category description and the terms inside sub-categories are indexed and retrieved using $tf \times idf$ scheme. However, such formal representations are extremely simple and do not discriminate the terms that are semantically more important to the concept based on the semantic structure of concept hierarchy.

In our opinion, structured concept descriptions of UMBEL can be indexed more efficiently by exploiting the semantic structure of the concept descriptions. For instance, where the term appears in a structured concept description (i.e. in a URI label, preferred labels, alternative labels, sub-concepts labels or super-concepts labels), should have a certain impact on the associated weight of the term to the concept. Therefore, to produce more effective concept representations, we propose a semantic indexing model based on the different semantic parts of the concept.

In our approach, we divided concept descriptions into different parts: *concept URI labels (uri)*, *concept labels (cl)*, *sub-concept labels (subl)*, *super concept labels (supl)* and *all labels (al)*. A *uri* contains terms that appear in the URI of the concept, where terms that appear in a *uri* can be particularly important to the concept. *cl* contain terms that appear in the preferred and alternative labels of the concept. Hence most lexical variations of the concept description are captured by *cl*. In concept-based IR, typically sub-concept labels are also accepted as a part of the concept [5-7]. For instance, for the concept "sports", sub-concepts baseball, basketball, football, etc. may provide further relevant lexical terms about "sports". Instead of accepting sub-concepts as a part of the concept, we separately index sub-concepts labels as *subl*. The *subl* include terms that occur in all inferred sub-concepts' URIs, preferred and alternative labels. In addition, we observed that many LOD resources contain links to super-concepts. For example, a resource about a writer also contains information that writer is a person. Thus, we index super-concept labels for more robust descriptors, where *supl* contain terms that occur in all inferred super-concepts' URIs, preferred and alternative labels. Finally, *al* contain all the terms that appear in all parts.

UMBEL is formatted in RDF N-triple format and we load the triples into a triple store (Jena persistent storage using Mysql DB) to extract the terms from UMBEL concepts. Each concept is divided into semantic parts of *uri*, *cl*, *subl*, *supl* and *al* using SPARQL queries, where concept descriptions are extracted from each semantic part.

From the concept descriptions, stop words are removed, as they have no semantic importance to the description, and words are stemmed into their roots using the Porter stemmer. The resultant words are accepted as *concept terms*. Finally, the extracted concept terms from the semantic parts are indexed. To do this, we consider each concept as a unique document and each semantic part is separately indexed as a term vector under the document (concept) using Lucene IR framework. In addition, the maximum normalized term frequency and inverse document frequency term value of each semantic part is calculated (which is subsequently used by the fuzzy retrieval model) and indexed together with the concept for quick retrieval. The inverted concept index is used for categorization of LOD resources.

It should be also noted that concept descriptions can be enhanced with lexical variations using WordNet. However, typically UMBEL descriptions include such word variations in alternative labels. This is the advantage of UMBEL being built upon on OpenCyc since OpenCyc contains rich lexical concept descriptions using WordNet. For the UMBEL concept <http://umbel.org/umbel/rc/Automobile> for instance, preferred label is *car* and alternative labels are *auto*, *automobile*, *automobiles*, *autos*, *cars*, *motorcar* and *motorcars*. This demonstrates a rich set of apparent lexical variations. Since these rich lexical descriptions are available in UMBEL, we did not use other lexical enhancement techniques because accuracy of the automated enhancements may affect categorization performance significantly.

4.3 Categorization of LOD Resources using a Novel Fuzzy Retrieval Model

In this step, we match the extracted *LOD terms* to UMBEL concept descriptions. In traditional IR, $tf \times idf$ is used to retrieve relevant concepts, i.e. each term in a concept has an associated value of importance (i.e. weight) to that concept and important terms are those that are frequently occur inside a text but infrequent in the whole collection ($tf \times idf$) [8]. Since we represent each UMBEL concept as a combination of different semantic parts (rather than one document), we need to calculate term relevancy to each semantic part. Then individual part relevancies can be used for a final relevancy score calculation. For this purpose, we propose a fuzzy retrieval model, where relevancy of a term, *t*, on concept, *c*, is calculated by a fuzzy function, $\mu(t,c) \in [0,1]$, using semantic parts of the concept and an extended $tf \times idf$ model.

Fuzzy Retrieval Model. First, UMBEL concept candidates are retrieved by searching *LOD terms* in all labels (*al*) of concepts. Then, for each LOD term, *t*, a relevancy score to every found UMBEL concept, *c*, is calculated by using a fuzzy function, $\mu(t,c) \in [0,1]$, on *uri, cl, subl* and *supl* semantic parts (since different parts have relative importance on the context of the concept *c*). Thus, $\mu(t,c)$ shows the degree of membership of the term *t* to all semantic parts of the concept *c*; where high values of $\mu(t,c)$ show that *t* is a good descriptor for the concept *c* and , $\mu(t,c) = 0$, means that the term *t* is not relevant for *c*. For membership degree calculation of $\mu(t,c)$, first membership degree of the term, *t*, to each part, *p*, should to be computed:

Definition 1: The membership degree of the term, *t*, to each part, *p* = [*cl*, *subl*, *supl*, *uri*], is a fuzzy function, $\mu(t,c,p) \in [0,1]$, which is based on $tf \times idf$ model. First, we calculate normalized term frequency (*ntf)* of *t* in *cl*, *subl* and *supl* of the concept *c*,

$$ntf(t,c,cl) = 0.5 + 0.5 \times \left(\frac{tf(t,c,cl)}{\max(tf(c,cl))} \right), \; \forall subl, supl \in p, ntf(t,c,p) = \frac{tf(t,c,p)}{\max(tf(c,p))} \quad (2)$$

where $tf(t,c,p)$ represents term frequency of the term *t* in the part *p* of the concept *c* and $\max(tf(c,p))$ represents maximum term frequency in the part *p* of the concept *c*. We calculate local normalized term frequencies for each semantic part, rather than calculating normalized term frequency using all terms of the concept in all semantic parts. In this way, term importance for a particular semantic part is obtained and frequent terms have higher value. For *cl* a minimum threshold value of 0.5 is set, since *cl* contains preferred/alternative terms of the concept, which is important for the context of the concept *c*. Then, for each part, *p*, we calculate the *idf* value of *t* in *p*,

$$\forall cl, subl, supl \in p, \;\; idf(t,c,p) = \log\left(\frac{C}{(n : t \in p) + 1} \right) \quad (3)$$

here, $n : t \in p$ is the number of semantic parts that contain the term *t* in *p* (i.e. $n : t \in cl$) and *C* is the total number of concepts in the collection. Again *idf* of a term in a particular semantic part is calculated instead of *idf* of a term in the whole corpus. In this way, rare terms that occur in a particular semantic part are assigned with higher values, which mean that rare terms are more important for the semantic part *p*. Next, $tf \times idf$ value of the term *t* in the semantic part *p* is computed,

$$\forall cl, subl, supl \in p, \;\; tf \times idf(t,c,p) = ntf(t,c,p) \times idf(t,c,p) \quad (4)$$

Finally, the membership degree of the term *t* to each part *p* is a fuzzy value,

$$\forall cl, subl, supl \in p, \;\; \mu(t,c,p) = \frac{tf \times idf(t,c,p)}{\max(tf \times idf(c,p))} \quad (5)$$

where $\mu(t,c,p) \in [0,1]$ equals to normalized $tf \times idf$ value of the term *t* in the part *p*. In this way, a fuzzy relevancy score is generated, where the term that has the maximum $tf \times idf$ value in the part *p*, $\mu(t,c,p) = 1$ and $\mu(t,c,p)$ reduces as the term importance decreases. As we discussed earlier, the maximum $tf \times idf$ value for each semantic part is calculated and indexed during the semantic indexing for better algorithm performance. Last, $\mu(t,c,uri) \in [0,1]$ equals to normalized term frequency,

$$\mu(t,c,uri) = 0.5 + 0.5 \times \left(tf(t,c,uri) \Big/ \sum_{i=1}^{n} tf(t_i,c,uri) \right) \quad (6)$$

here, term frequency of *t* in *uri* is divided by total number of terms in the *uri*. A minimum threshold value of 0.5 is set, since *uri* terms are important. If *uri* contains one term, $\mu(t,c,uri) = 1$, means the term is important for *uri*. The term importance decreases as the number of terms in the *uri* increases.

Definition 2: Relevancy of the term *t* to the concept *c*, is calculated by, $\mu(t,c)$, where membership degrees of the term *t* to the parts *uri*, *cl*, *subl* and *supl* are combined,

$$\mu(t,c) = \frac{w_{uri} \times \mu(t,c,uri) + w_{cl} \times \mu(t,c,cl) + w_{subl} \times \mu(t,c,subl) + w_{supl} \times \mu(t,c,supl)}{w_{uri} + w_{cl} + w_{subl} + w_{supl}} \quad (7)$$

where, $\mu(t,c) \in [0,1]$ and, w_{uri}, w_{cl}, w_{subl} and w_{supl} are constant coefficients that aid to discriminate features obtained from different parts. For example, the terms obtained from the *uri* and *cl* can be weighted higher than *subl* and *supl*. The parameter values were experimentally determined as we discuss later in the evaluations section.

Definition 3: Finally, relevancy of all *LOD terms*, $T = \{t_1,...,t_m\}$, to the concept *c* is,

$$\mu(T,c) = \sum_{i=1}^{m}\left(\mu(t_i,c)\times w(t_i)\right) \Big/ \sum_{i=1}^{m} w(t_i) \tag{8}$$

where, $\mu(T,c)\in[0,1]$ and $w(t_i)$ is term importance of LOD term t_i, which is calculated by Equation 1. Using term weights, $w(t_i)$, term matches to the important LOD terms are encouraged (i.e. terms with higher weights). In addition, concepts that have a good coverage of *LOD terms* especially in important semantic parts (which is determined by coefficients in Equation 7), will have a higher relevancy score.

Definition 4: Finally, the concept with the maximum $\mu(T,c)$ is selected as the categorization of the LOD resource. In cases where there are two or more concepts that have the maximum value, categorization is decided based on an ontological relationships driven voting algorithm. The algorithm is as follows: For each concept with the maximum $\mu(T,c)$; (1) we find the number of sub-concepts (*n*), (2) sub-concepts (*sc*) count, *k*, that have non-zero relevancy value to *LOD terms*, $\mu(T,sc)>0$, (3) total membership degree, ψ, of all sub-concepts that $\mu(T,sc)>0$. The voting score (*v*) of the concept is computed as, $v=\psi\times(k/n)$, which means we encourage the concept whose sub-concepts have higher scores as well as the concept with a greater number of sub-concept matches. After the voting, the concept with the maximum *v* is accepted as the categorization. If there is still more than one maximum, we accept all concepts as categorizations, since a LOD resource may belong to one or more concept, e.g. Kyoto is a city as well as a district.

5 Evaluations

This section discusses the evaluation setup and the experiments undertaken to test the performance of our approach. A particular benchmark was created to evaluate: (1) performance of different LOD features, (2) categorization accuracy of the fuzzy retrieval model against the vector space model, (3) efficiency of system performance.

5.1 Setup

The performance of the proposed concept-based search depends on the accuracy of the fuzzy-based categorization algorithm, because incorrect categorizations can degrade user experience with the search mechanism. Thus, categorization accuracy is crucially important and it needs to be evaluated. Fortunately, many DBpedia LOD resources (~900,000) have mappings to UMBEL concepts and DBpedia publishes these links in RDF N-Triple format (http://dbpedia.org/Downloads). Since our aim is to test performance of various features and different algorithms, the use of whole mappings is computationally expensive. Instead, we created a particular benchmark of ~10,000 mappings from the provided DBpedia links. The selection procedure for the benchmark was as follows: We randomly selected from different types of UMBEL concept mappings (e.g. umbel:HockeyPlayer, umbel:Plant, etc.) and those DBpedia resources that are cached by the Sindice Cache. This resulted with 10227 benchmark resources. Then, RDF descriptions of 10,227 benchmark resources and resources that are linked from those resources using owl:sameAs and dbpedia:WikiPageRedirect links (~2 per resource) were cached to a local disk for context extraction. In addition, during the feature extraction, links to UMBEL concepts were ignored.

In the experiments, the parameter values of w_{uri} = 2, w_{cl} = 2, w_{subl} = 1 and w_{subl} = 1 which were experimentally determined to achieve the best results. We randomly selected 500 mappings from the benchmark and chose the values that gave the best precision/recall. With these parameter values, URI and concept labels have the highest impact and, sub-concept and super-concept labels contribute moderately.

5.2 Categorization Accuracy – Precision and Recall

By accepting the DBpedia to UMBEL mappings as ground truth, precision (P) and Recall (R) of the automatically generated categorizations are calculated. Precision equals to the number of correctly predicted (*CPr*) divided by total number of predictions (*Pr*), $P = CPr / Pr$. Recall equals to the number of unique correct predictions (*UPr*) per resource (since our algorithm may predict one or more categorizations for each resource) divided by total number of mapings (*T* = 10227), $R = UPr / T$. In evaluations, if the predicted categorization directly matches to any of UMBEL concept mappings or the predicted categorization is a super-concept of a UMBEL concept mapping, then the categorization is accepted as correct. Super-concept mappings are also accepted as correct since it is intuitively and logically true. For example, if *x* is a umbel:Cyclist, it is also true that *x* is a umbel:Athlete or if *x* is a umbel:Bird, it is also true that *x* is a umbel:Animal.

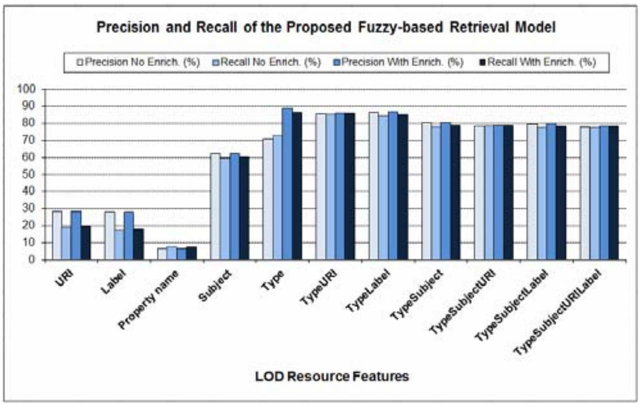


Fig. 3. Categorization accuracy of the proposed fuzzy retrieval model with respect to different LOD resource features and the semantic enrichment technique

Proposed Fuzzy Retrieval Model. Figure 3 shows precision and recall of the proposed fuzzy retrieval model: (1) with different LOD resource features and (2) with and without the semantic enrichment technique. The results show that among all LOD resource features, *type* feature alone gave the best precision of 70.98% and 88.74% without and with the enrichment respectively. This is because most resources contain *type* feature, which provides knowledge about the context of resources. *subject* feature performed a precision of ~62%, *uri* and *label* features alone did not perform well (~28%) and *property names* performed the worse accuracy. Among combinations of different LOD resource features, *type*+*uri* and *type*+*label* provided the best accuracy without the semantic enrichment with a precision of 85.55% and 86.38% respectively. Other combinations did not improve the overall accuracy despite more *LOD terms* being used in the categorization. Another interesting outcome is that the semantic enrichment technique did not have a significant impact on the categorization accuracy (~1% improvement) except the *type* feature. In the *type* feature, the enrichment technique improved the precision and recall ~18%. In addition, we noticed that in some cases all possible mappings from DBpedia to UMBEL are not included, e.g. a volcano mountain is mapped as umbel:Mountain, but not as umbel:Volcano. Besides, DBpedia uses more general mappings, for example, a science fiction writer is mapped as umbel:Writer, despite the existence of umbel:ScienceFictionWriter. This could be because of human error since manual mapping process[3] is involved, which can be error-prone. Although these particular cases affected categorization accuracy, the proposed fuzzy retrieval model achieved high accuracy on the benchmark. Especially high performance is achieved by using the *type* feature and the *type*+*uri* and *type*+*label* features (with and without the enrichment). The results are promising because typically LOD resources contain data about type and labels of the resource, which can be used to provide high quality categorization.

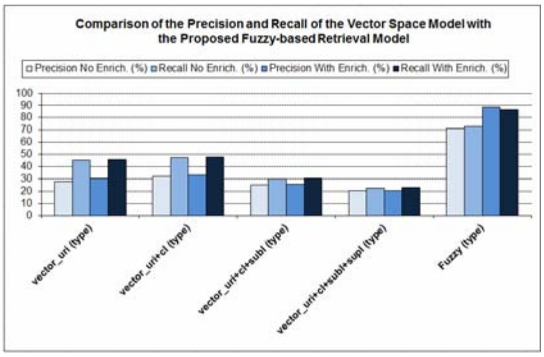


Fig. 4. Comparison of the vector space model with the proposed fuzzy retrieval model

Vector Space Model. Since the proposed fuzzy retrieval model extends *tf* × *idf* with a fuzzy relevancy score calculation using semantic structure of concepts, we compared the categorization accuracy against the *tf* × *idf* model. In vector space model, concept descriptions can be represented in a number of ways; using (1) only *uri*, (2) *uri*+*cl*, (3) *uri*+*cl*+ *subl*, and (4) *uri*+*cl*+*subl*+*supl*. On the concept representation alternatives, we applied *tf* × *idf* retrieval model on the benchmark. For fair comparison, the same clean-up steps are applied to the vector space model (i.e. stemming, stop word and qualifier removal) and the same voting algorithm is used if there is more than one maximum categorization. In contrast, concept weights to all *LOD terms* are calculated using the *tf* × *idf* scheme. In Figure 4, the best results are shown, which is achieved by the *type* feature. Results show that the vector space model did not perform well. The best precision and recall is obtained by using *uri*+*cl* with a precision and recall of

[3] http://umbel.googlecode.com/svn/trunk/v100/External%20Ontologies/dbpediaOntology.n3

31.77% and 47.42 without the semantic enrichment and with a precision and recall of 33.0.9% and 47.75 with the semantic enrichment. When using all semantic parts, the precision of the vector space is decreased to 20.37% compared to 88.74% precision of the proposed fuzzy retrieval model.

Discussion of Results. Our fuzzy retrieval model performs outstandingly better than the vector space model for the following reason. *tf* × *idf* is a robust statistical model, which works well with good training data. Traditional concept-based IR systems [5,6,7] use the top 2-3 levels of a concept hierarchy (few hundred concepts) with hundreds of training documents. In contrast, we use the whole 28,000 UMBEL concepts. Moreover each concept contains few lexical information in different semantic parts of the concept, such as in URI, preferred/alternative labels and super/sub-concept(s) labels to describe that concept. *tf* × *idf* cannot discriminate terms only using combined terms and often few LOD terms are matched to many concepts (sometimes hundreds) with the same *tf* × *idf* scores. We propose a more intuitive approach, where our fuzzy retrieval model extends *tf* × *idf* with a fuzzy relevancy score calculation based on semantic structure of concepts, i.e. terms from the concept, sub-concept(s) and super-concept(s) have certain importance in retrieval. Besides, relevancy scores are combined according to their importance to the concept. Hence, this more intuitive approach performs astoundingly better than *tf* × *idf*, which do not discriminate term importance based on semantic structure of a concept hierarchy.

5.3 Computational Efficiency of Dynamic Categorization

In addition to high accuracy, dynamic categorization performance is an important factor for the proposed concept-based search. To provide fast categorizations, each search result (resource) is processed in parallel using AJAX. In addition, to give an idea of dynamic (online) categorization times, we measured average processing times based on number of *LOD terms* per resource using a laptop with Windows 7 operating system, 4 GB RAM, Intel Core 2 Duo CPU (2.53 GHz) and 54Mbps Internet connection. Without the enrichment, average processing times vary between 1-1.5 secs for the proposed approach compared to 0.1-0.5 secs of the vector space model. With the enrichment, processing times increase for both model, because of dynamic caching from LOD graphs. We found that an average of twelve *LOD terms* are extracted from the benchmark resources, which means we can perform categorization within ~1 secs and ~1.5 secs with and without the enrichment respectively.

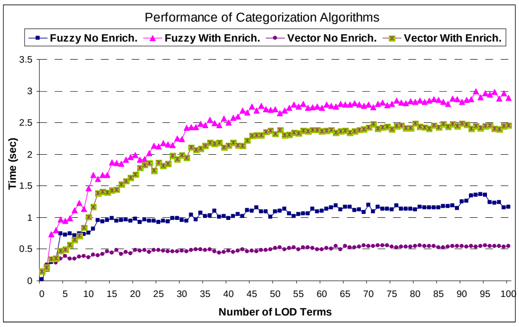


Fig. 5. Dynamic algorithm performance with respect to number of LOD terms

7 Conclusions and Future Work

We have presented a novel approach for concept-based search on the Web of Data. The proposed innovative search mechanism is based on UMBEL concept hierarchy, fuzzy-based retrieval model and categorical result list presentation. Our approach groups LOD resources with same concepts to generate concept lenses that can provide efficient access to the WoD and enables concept-based browsing. The concept-based search is achieved using UMBEL for representing context of LOD resources. Then, a semantic indexing model is applied for efficient representation of UMBEL concept descriptions. Finally a fuzzy-based retrieval algorithm is introduced for categorization of LOD resources to UMBEL concepts. Evaluations show that the proposed fuzzy-based model achieves highly acceptable results on a particular benchmark and outperforms the vector space model in categorization accuracy, which is crucial for correct formation of concept lenses.

The introduced semantic indexing and fuzzy retrieval model are not inherently dependent on UMBEL vocabulary and should be applicable to multiple vocabularies. Moreover, in UMBEL, we are using all sub-concepts and super-concepts of a concept, but other vocabularies can be explored for relating concepts with different semantic relationships other than hierarchical structures. In future work, we will incorporate personalization into our concept-based search methodology in order to personalize results to the context and individual needs of the user, and perform user-based studies.

References

1. 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, vol, 8, no. 4, pp. 355-364 (2010)
2. Delbru, R., S., Campinas, G., Tummarello: Searching Web Data: an Entity Retrieval and High-Performance Indexing Model. Journal of Web Semantics, vol. 10, pp. 33-58, (2012)
3. 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)
4. Heim, P., T. Ertl and J. Ziegler.: Facet Graphs: Complex Semantic Querying Made Easy, Extended Semantic Web Conference, LNCS, vol. 6088, pp. 288-302 (2010)
5. Chirita, P. A., Nejdl, W., Paiu, R., and Kohlschütter, C.: Using ODP metadata to personalize search. International ACM SIGIR Conference (2005)
6. A. Sieg, B. Mobasher, R. Burke.: Web Search Personalization with Ontological User Profiles. International Conference on Information and Knowledge Management (2007)
7. Labrou, Y., T. Finin: Yahoo! As An Ontology – Using Yahoo! Categories to Describe Documents, International Conference on Information and Knowledge Management, (1999)
8. Salton, G. and M. J. McGill. Introduction to Modern Information Retrieval, (1983)
9. Steichen, B., O'Connor, A., and Wade, V. (2011). Personalisation in the Wild – Providing Personalisation across Semantic, Social and Open-Web Resources. ACM Hypertext (2011)
10. Carpineto, C. and G. Romano. Optimal Meta Search Results Clustering. SIGIR, (2010)
11. Erling, O. Faceted Views over Large-Scale Linked Data. Linked Data on the Web (LDOW) Workshop, co-located with International World Wide Web Conference, (2009)
12. Teevan, J., S. T., Dumais and Z. Gutt: Challenges for Supporting Faceted Search in Large, Heterogeneous Corpora like the Web. Workshop on HCIR, (2008)
13. Shangguan, Z. and D. L. McGuinness: Towards Faceted Browsing over Linked Data. AAAI Spring Symposium: Linked Data Meets Artificial Intelligence, (2010)
14. White, R.W., Kules, B., Drucker, S.M., and schraefel, m.c. Supporting Exploratory Search, Introduction to Special Section of Communications of the ACM, 49(4), 36-39, (2006)