Federated semantic search using terminological thesauri for learning object discovery

Dimitrios Koutsomitropoulos, Georgia Solomou and Katerina Kalou
Department of Computer Engineering and Informatics, University of Patras, Patras, Greece

Received 17 June 2016
Revised 16 March 2017
Accepted 27 March 2017

Abstract

Purpose – The purpose of this paper is to propose a framework and system to address the inability to discover new and authentic learning material and the lack of a single access point for search and browsing of remote learning object repositories (LORs).

Design/methodology/approach – The authors develop a framework for keyword-based query expansion using SKOS domain terminologies and implement a federated search mechanism integrating various disparate LORs within a learning management system (LMS).

Findings – The authors show that the expanded query achieves improved information gain and it is applied for federated information access, by simultaneously searching within a number of repositories. Results can be seamlessly aggregated back within the LMS and the course context.

Practical implications – It is possible to retrieve additional learning objects (LOs) and achieve a corresponding increase in recall, while maintaining precision. SKOS expansion behaves well in a scholarly setting, which, combined with federated search, can contribute toward LOs' discovery at a balanced cost. The system can be easily integrated with other platforms as well, building on open standards and RESTful communication.

Originality/value – To the authors' knowledge, this is the first time SKOS-based query expansion is applied in a federated setting, and for the discovery and alignment of learning objects residing within LORs. The results show that this approach can achieve considerable information gain and that it is possible to strike a balance between search effectiveness, query drift and performance.

Keywords Federated search, SKOS, Learning object repositories, Learning objects, Reasoning, Query expansion

Paper type Research paper

1. Introduction

A major reason for the wide growth and success of learning object repositories (LORs) is that they make available and allow the reuse of high-quality educational resources for addressing multifaceted didactic goals. Ochoa and Duval (2009) have studied more than 39 repositories with over 400K available LOs and an average steady growth of over 3K LOs per year each. The Open Educational Resources initiative presents another reason for making LORs attractive, by allowing unhindered and open access to learning content (Atenas and Havemann, 2013). Additionally, easy componentization of LOs within LORs makes it possible to glean learning elements and reuse them within larger settings, such as MOOCs (Piedra *et al.*, 2014).

These benefits however come at the cost of two important drawbacks instructors and learners face when in need for educational resources. In most cases, management and communication of course resources tend to be made through a learning management system (LMS). Instructors often strive to search and acquire additional content that would complement and enhance the backbone of their course; the same also holds for

This work has been partially supported by the project "Information System Development for Library Functional Services" of the Democritus University of Thrace, co-financed by Greece and the European Union, in the context of Operational Program "Digital Convergence" of the National Strategic Reference Framework (NSRF) 2007-2013.

Journal of Enterprise Information Management
Vol. 30 No. 5, 2017
pp. 795-808

1741-0398
DOI 10.1108/JEIM-06-2016-0116

learners requesting supplementary material. The common practice of browsing bluntly through the internet yields poor results, since focused extracurricular material is hard to discover and depends heavily on the instructor's (and learner's) expertise, up to whom it is to devise and submit educated queries. As a result, online courses in an LMS end up with a majority of unauthoritative and uncurated external pointers, if not missing whatsoever.

On the other hand, human searches through the multitude of available LORs require too much manual labor, not only for making the individual queries, but also for selecting and combing through results. What is more, the context of the original course container, the LMS, is lost and recommendations have to be manually added.

Therefore, in this paper we propose a framework and a service for mitigating these problems and aiding the discovery of LO within LORs. We employ the idea of keyword-based query expansion to automatically and transparently submit, expand and refine queries toward LORs, based on the set of keywords originally describing a particular course within an LMS. To this end, expert terminological knowledge is offered by two domain thesauri, thus shifting the weight from searchers' competency to the specialists' expertise. The thesauri are implemented using SKOS (Miles and Bechhofer, 2009) and OWL, in order to be able to apply reasoning operations during expansion.

Expanded queries are then addressed toward a number of LORs, including MERLOT (McMartin, 2006) and PubMed (Europe PMC Consortium, 2014), in a federated manner. The federated search mechanism replicates the expanded keywords which are distributed to the LORs search engines and seamlessly aggregates the results back within the LMS and the course context. Our results show that this approach can achieve considerable information gain and that it is possible to strike a balance between search effectiveness, query drift and performance.

Our framework has been applied for the benefit of the online courses of the Democritus University of Thrace (DUTH) in Greece and integrated with the eClass LMS platform. Open eClass (GUnet asynchronous eLearning group), initially a spin-off of Claroline[1], is a widely used LMS by higher education institutions worldwide and is the solution offered by the Greek Academic Network GUnet to support asynchronous eLearning services in universities. A prototype is available[2] and a production version has been already incorporated within the official eClass deployment of DUTH (http://eclass.duth.gr).

The main contributions of this paper can be summarized below:

- a framework for extending learning objects' metadata with external LO references;
- a framework for the discovery of learning objects;
- SKOS-based multilingual thesauri implementations following standards (National Documentation Center);
- a federated search mechanism integrating various disparate LORs; and
- a pay-as-you-go, query expansion algorithm using reasoning operations and SKOS OWL ontologies that improves recall and guarantees precision (stays unaffected).

The rest of this paper is organized as follows: Section 2 surveys related work in the fields of federated search and query expansion for LOs' discovery. Section 3 gives an overview of our system's architecture and its main components. The development of the SKOS thesauri and their usage as the basis for query expansion is presented in Section 4. Section 5 describes the matching and expansion process and introduces the implementing algorithm, followed by an evaluation using various thesauri in Section 6. Section 7 shows a usage example and the integration of results within a course's context. Finally, section 8 summarizes our concluding remarks.

2. Background and related work

2.1 Federated search in LORs

It is no surprise that the wide availability of educational content has given rise to many efforts for federated search over LORs. For example, in the work of De la Prieta *et al.* (2014), the necessity and significance of aligning learning content providers are recognized and the authors describe a cloud-based architecture for the integration of federated search services.

Ternier *et al.* (2008) investigated the dire need for interoperability between learning repositories. As a result, they proposed a protocol language, building on top of the simple query interface (SQI) (Van Assche *et al.*, 2006) that would facilitate federated searching, at the overhead of supporting an additional standard.

Adhering to a common metadata application profile and querying protocol for aggregating learning resources was also the purpose of the CELEBRATE project (Massart and Le, 2004). The project aimed at unifying LMSs across European schools by registering with a brokerage system facilitating contracts between parties. The ASPECT Project (2009) also promoted federated search among LORs by supporting the SQI specification, given that many providers supported it already. However, it is identified that it is not always evident for repositories to implement another standard.

In view of the benefits of LORs federation, a special case of LORs (Ochoa and Duval, 2009) are actually meta-search engines for LOs, by fetching results from other sources on the web. Typical examples of this form include MERLOT and ARIADNE (Duval *et al.*, 2001). Thankfully, they are also exposing simple search interfaces through appropriate web services, either REST or SPARQL, beyond the SQI that is SOAP-based and therefore unattractive (Mazo *et al.*, 2012). These services can naturally act as very rich intermediaries for federated search.

Finally, semantic technologies and Linked Data can play an important role in improving interoperability and visibility of LORs. A related survey (Dietze *et al.*, 2013) points out the possibilities for automated enrichment and discovery of LOs, by means of semantic web approaches.

2.2 Query expansion

Ontology- and SKOS-based query expansion have been proposed before. Bhogal *et al.* (2007) review several techniques for query expansion using WordNet and other domain-specific ontologies and generally attest improved query effectiveness. It is also acknowledged that query expansion works best when short queries are involved, as is the case with keyword-based searches, rather than complex and detailed ones.

Haslhofer *et al.* (2013) present an approach for SKOS-based query expansion, close to our work, and show evidence for improved results in web search. They also identify query drift as a possible side effect of greedy term expansion, which means that the expanded queries may be too specific to express the user's information need any more.

Even without using SKOS, Segura *et al.* (2011) study the effects of query expansion based on the Gene ontology and targeting a single repository only, MERLOT. The authors confirm expanded queries allow the user to retrieve relevant objects, which might not be obtained without expansion.

However, to our knowledge, this is the first time SKOS-based query expansion is applied in a federated setting, and for the discovery and alignment of learning object metadata residing within multiple LORs, further boosted by the knowledge gain brought in by reasoning.

An important restriction this imposes is that there is no access to the data set corpora that would allow affecting document indexing a priori, nor is there control on the retrieval models implemented by the search engines of the various LORs. Therefore, all implementation must remain transparent and oblivious to the actual repositories

accessed and their respective search mechanisms (black boxes), in contrast to Haslhofer *et al.* (2013). Reasoning on the other hand accommodates the full exploitation of a SKOS vocabulary, by giving the chance to amend possible omissions and to complement the thesaurus using inferences.

3. Architecture

Our service consists of three main components: thesauri development is administered by the thesaurus/ontology authoring and editing subsystem that is largely based on WebProtégé (Horridge *et al.*, 2014); course management and importing of external LOs is undertaken by the external resources management subsystem that is directly implemented within eClass; finally, the Semantic Middleware is at the heart of the service and its purpose is to detect semantic relevance between keywords and thesauri terms, to perform query expansion and to conduct federated search in remote repositories.

Semantic Middleware communicates using RESTful interfaces over HTTP, first as a query client to the LORs and then as a results server to eClass, which sends in the initial set of keywords. The main advantage of this approach is that the middleware would be interoperable with additional services and LMSs, other than eClass. It would also be capable of extending transparently and seamlessly in the future, just by implementing additional methods of the RESTful API.

Ontology storage is managed by WebProtégé, which supplies the middleware with the thesauri. Other than MERLOT and PubMed (Europe Central), we also include openarchives.gr, a meta-search engine and harvester for Greek academic and research repositories. A SPARQL connector to ARIADNE has also been implemented. Figure 1 depicts the general architecture of the service as well as the basic functionality and communication between the various components.

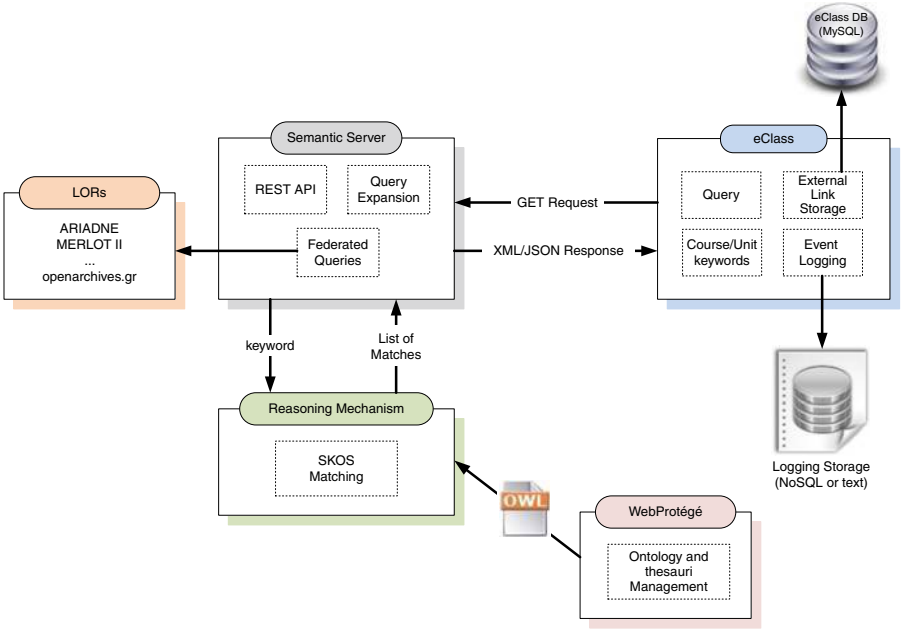

Figure 1.
The architecture of the federated semantic search framework

4. SKOS thesauri implementation and usage

SKOS (Miles and Bechhofer, 2009) is a model for expressing knowledge organization systems (KOS), including thesauri, in machine readable format, namely RDF(S) and OWL. It provides a uniform representation of a set of terms and hence a common mechanism for the thematic indexing and retrieval of information. Concept is the basic structural element of SKOS. Concepts are abstract entities, which are independent of their names (i.e. the labels). A set of properties is offered by the model in order to be able to characterize the notions represented by SKOS concepts (see Figure 2 for an example). These concepts and relations are used by our system as a basis upon which term expansion is performed.

To this end, we proceeded with the creation of two thesauri – initially not in SKOS format – that cover two very common fields of knowledge: maths and medicine. These thesauri were actually extracted from the Thesaurus of Greek Terms (TGT), a bilingual (Greek, English) controlled vocabulary published by the National Documentation Center in Greece (EKT)[3]. The latter covers a very broad field of knowledge and was created in order to facilitate libraries, museums, information centers and other institutions in Greece in characterizing and managing their digital material.

The Maths Thesaurus is comprised of 76 terms, making reference to 17 other related terms, whereas the Medicine Thesaurus contains 54 terms and makes reference to 71 additional terms. Although both of these thesauri cover specific fields of knowledge, they are generic enough and thus sufficient for the characterization of the most common subjects met in these thematic areas. Through an appropriate mapping process (Solomou and Papatheodorou, 2010), we achieved the SKOS transformation of these two thesauri, from their initial XML format into OWL.

For the matching and expansion process, we take advantage of the following relations (Figure 2):

- skos:narrower/skos:broader: these two inverse relations connect a concept with its refinements. To expand the initial user keyword, possible refinements of the keyword's matching concept are found and used to expand the query;

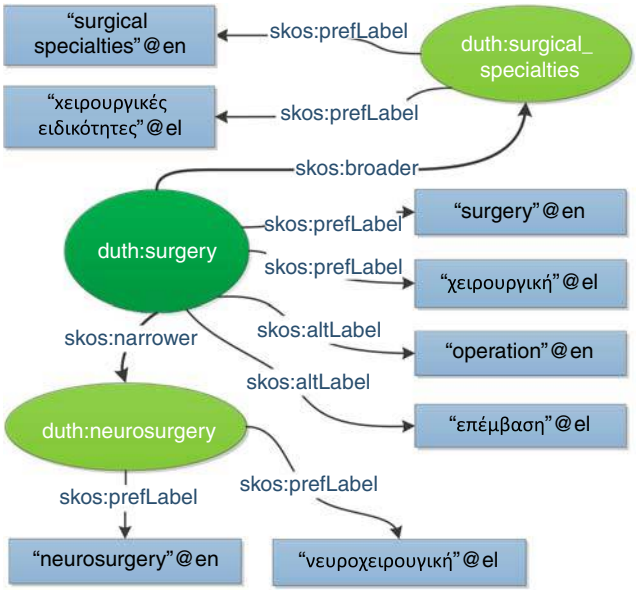

Figure 2.
Example of SKOS relations for medicine

- skos:preflabel: this is the preferred lexical representation of a concept. There may be multiple preferred labels – by design, one for each language. This is the approach followed in the thesauri used. Preferred labels are counted in for expanding the initial keyword; and
- skos:altlabel: this denotes an alternative lexical representation for the concept, other than the preferred label. There can also be many alternative labels for a concept, but not necessarily translations. Alternate labels are also counted in for keyword expansion.

5. Semantic query server (Semantic Middleware)

5.1 Reasoning operations

In the case of SKOS, reasoning allows to remedy incomplete terminologies given the semantic implications of the SKOS schema. For example, in SKOS terms, `skos:narrower`, which is naturally used to refine a concept, is the semantic inverse of `skos:broader`, but this is not always modeled in thesauri that may contain only one of the sides of this relationship. This is true, e.g., for the MeSH thesaurus SKOS implementation (Van Assem *et al.*, 2006) that only contains broader relations. However, this fact can be modeled in an OWL ontology and the implied bidirectional relations can be inferred using appropriate operations.

Reasoning operations are implemented using the OWL API (Horridge and Bechhofer, 2011) rather than a dedicated reasoner: first, the operations required can be kept lightweight and easily implemented using the API, since there is no need for full KB classification or other complex and costly operations; and second, the use of an external inference engine would impose too much communication and computational overhead, as it is well known that expressive reasoning algorithms do not scale well (Bock *et al.*, 2008).

Reasoning operations implemented and used by our model include:

- Instance checking on property assertions, i.e., check if T entails $R < x, y >$, where T is the thesaurus/ontology and R, a refinement relationship between SKOS concepts (broader/narrower).
- Calculating property inverses, i.e., given $P \in I \times I$ and $R^{-} \equiv P$ find all $< x, y > \in R$, where R, P are the inverse properties and I is the individuals domain. It is evident that this operation is a prerequisite for the instance checking computation to be complete.

5.2 Semantic matching and query expansion

To perform semantic matching and query expansion, Semantic Middleware implements the algorithm shown in Table I.

The algorithm works as follows: given an original keyword t, there might be multiple matching SKOS concepts (lines 1-2). To keep the number of matches to a reasonable limit, in our implementation we consider only exact lexical matches, although this can be easily configured. For a matching concept x, the thesaurus is being traversed in BFS order in the direction of the skos:narrower property, either asserted or inferred, in order to find the refining concepts. The algorithm finds first all the direct children of the matching concept (i.e. depth 1) and adds their number to the set of total descendants, $\mathcal{R}(x)$ (initially empty). The algorithm does not proceed to the next depth, unless $\#\mathcal{R}(x) \leqslant$ threshold. Then, each child in order is being expanded and its children are added to $\mathcal{R}(x)$, until the threshold is reached or exceeded (lines 3-5).

The algorithm takes care to avoid duplicate queries (line 8). There is the chance that different concepts may have labels with identical lexical forms or share common descendants.

l: $<c, l> \in$ skos:prefLabel $\cup$ skos:altLabel, $c \in$ skos:Concept
t: initial query term
$\mathcal{R}$: the set of concepts matching the query term
$\mathcal{R}(x)$: the set of concept x and its refinements
$\mathcal{Q}(y)$: the set of label-language tuples for concept y
$\mathcal{K}(v)$: the set of results for label v
$\mathcal{V}$: the set of labels queried (used)
$\mathcal{L}(v)$: the label v, its language, its translations and alternatives and their languages
$\mathcal{K}(x)$: the set of unique result-tuples for all concepts in the hierarchy of concept x
$\mathcal{K}'(v)$: the set of unique results for label v
$\mathcal{A}$: the server response

1. $\forall\ l$: $t \cong l$ **and** $\# \mathcal{R} \leq threshold$
2. $\quad \mathcal{R} \leftarrow \mathcal{R} \cup \{x \mid <x, l> \in$ skos:prefLabel $\cup$ skos:altLabel $\}$
3. $\forall\ x \in \mathcal{R}$ **and** $\# \mathcal{R}(x) \leq threshold$ //recursion until threshold
4. $\quad \mathcal{R}(x) \leftarrow \{x\}$
5. $\quad \mathcal{R}(x) \leftarrow \mathcal{R}(x) \cup \{z \mid <z, x> \in$ skos:broader **or** $<x, z> \in$ skos:narrower$\}$
6. $\forall\ x \in \mathcal{R}, \forall\ y \in \mathcal{R}(x)$
7. $\quad \mathcal{Q}(y) \leftarrow \{<v, lang> \mid <y, v@lang> \in$ skos:prefLabel $\cup$ skos:altLabel $\}$
8. $\quad \forall\ v: <v, lang> \in \mathcal{Q}(y)$ **and** $v \notin \mathcal{V}$ //only unique labels
9. $\quad\quad \mathcal{K}(v) \leftarrow$ **get_results** (v)
10. $\quad\quad \mathcal{V} \leftarrow \mathcal{V} \cup \{v\}$
11. $\quad\quad \mathcal{L}(v) \leftarrow \{<v, lang, \mathcal{Q}(y) / <v, lang>>\}$ //keep the label and its translations/alternatives
12. $\quad\quad \mathcal{K}(x) \leftarrow \mathcal{K}(x) \cup \{<r, v> \mid r \in \mathcal{K}(v)$ **and** $<r, *> \notin \mathcal{K}(x)\}$ //only unique results
13. $\forall\ t$: $t \ncong l$ **and** $t \notin \mathcal{V}$ //get results for non-matching query terms
14. $\quad \mathcal{L}(t) \leftarrow \{<t, *, \emptyset>\}$
15. $\quad \mathcal{K}'(t) \leftarrow$ **get_results** (t)
16. $\quad \mathcal{V} \leftarrow \mathcal{V} \cup \{t\}$
17. $\forall\ v \in \mathcal{V}, \forall\ x \in \mathcal{R}$
18. $\quad \mathcal{K}'(v) \leftarrow \mathcal{K}'(v) \cup \{r \mid \exists\ x: <r, v> \in \mathcal{K}(x)\}$ //organize (already unique) results per label
19. $\quad \mathcal{A} \leftarrow \mathcal{A} \cup \{<\mathcal{L}(v), \mathcal{K}'(v)>\}$

Table I.
The semantic matching and query expansion algorithm

The algorithm keeps track of the labels (lexical forms) already used for querying and avoids using the same label twice.

Results are grouped by label. Each label is accompanied by all its translations and alternatives so that it would be easy for receiving applications to distinguish and order results (lines 17-19). Each concept x in $\mathcal{R}$ is traversed by its insertion order (BFS). Similarly, the set of labels $\mathcal{V}$ is insertion ordered. This means that search result-sets per label in the server response are also ordered in the way the concept hierarchy is traversed: as far as we draw away from the parent concept, results are ranked lower.

Additionally, duplicate results for a matching concept and its hierarchy are ignored (line 12). Although unique, labels in a single hierarchy may have overlapping results, considering that they are translations of each other, refinements of each other and that they are targeted to (possibly) overlapping data sets. For this purpose, an LO's URL is used as a key r and only unique URLs per label v are counted in the final result set. Overlapping results occurring in different hierarchies (other $\mathcal{K}(x)$'s) are preserved, because they originate from different matching concepts and thus are answers to virtually different queries.

Because of the fact that a concept is expanded first in breadth rather in depth and that there is an insertion-ordered set of expanded concepts up to query time, the threshold would act as a guard against concept drift. However, the threshold alone does not protect from a costly and increased number of queries, since expanded concepts can have an arbitrary number of preferred and/or alternative labels, all of which are considered during querying. Therefore, we also introduce an upper limit on the number of queries that are going to be submitted eventually.

5.3 Example

To see an example, consider the input keyword "operation" and the terminology shown in Figure 2. When the threshold equals 2 or greater, the matching and expansion process for this keyword would result in:

{"surgery"@en, "χειρουργική"@el, "operation"@en, "επέμβαση"@el, "neurosurgery"@en, "νευροχειρουργική"@el},

i.e., the concept duth:surgery is matched first, then its refinement, duth:neurosurgery, is added and the input keyword is expanded into six queries. In case there is 1threshold the set would be:

{"surgery@en", "χειρουργική"@el, "operation"@en, "επέμβαση"@el}, because now only one concept, the matching one (duth:surgery), is added. Finally, in case there is a othreshold, i.e. no concept expansion and matching takes place, then the query to be sent would just be {"operation"}.

6. Augmenting course external links

After successful authorization with eClass, the logged instructor can select the "Link" module from the navigation menu of his course and then the newly added "Add Learning Objects" option. A search form appears with a unique field that has a predefined set of keywords (see Figure 3). These keywords, separated by a comma, include the keywords that the instructor has already registered for his own course with the LMS. However, the instructor is free to set a different set of keywords each time.

Next, the keywords are sent to the Semantic Middleware. Once the application completes the loading process, the web interface presents a table of learning object metadata grouped into categories (see Figure 4). These categories are in fact the labels of the matching and refining concepts found during the expansion process or the original keywords themselves, if no matches exist. The labels and their translations/alternatives are by default included in the response. Based on them, search results are ordered so that translating and alternate labels appear first and then move on to the next label.

For each learning object, the URL, the title and the description are available to the end user. For clarity reasons, there is a pagination capability of the results' categories. Besides, the categories are shown by default collapsed and not the entire list of results is presented at once.

Instructors can traverse through the results' pages using the navigation buttons and select what to store by clicking the checkbox near the result's title. In case they desire to select all the results for a category, they can do it at once by clicking the checkbox near the category title. The full list view of results for a particular category/keyword can be toggled by clicking on the plus/minus button or on the category title, which expands or collapses category results, respectively. Added LOs are then available for students' reference in the "external links" section (Figure 5).

7. Evaluation

7.1 Configuration

To evaluate our system, we used the two domain thesauri we developed and described above. In addition, we experimented with larger thesauri, Medical Subject Headings (MeSH), already converted into SKOS (Van Assem *et al.*, 2006) and comprising of 23,883 concepts and the complete TGT, that we have implemented in SKOS previously (Solomou and Papatheodorou, 2010), comprising of 5,227 terms. Experiments were performed on standard virtualized hardware, using a single CPU and 2GB RAM. Times reported are in milliseconds.

We partition queries into four sets, each containing ten keywords related to medicine or mathematics, depending on the ontology used, and report on their average. We generally

Figure 3.
Communicating search keywords to the semantic middleware

Figure 4.
List of results for input keyword "surgery" categorized per concept labels

Figure 5.
Available links to external LOs for a course

use keywords in Greek, except for MeSH, where we use English keywords. For TGT, we use the medicine keywords.

Set C1 contains terms matching concepts at the top of the thesaurus hierarchy that would naturally have a lot of refinements; set C2 contains terms matching concepts near the

middle; set C3 contains terms-leafs, i.e. they have no refinements, but have alternate labels and translations; set C4 contains keywords irrelevant to the ontology used, i.e. they have no matches whatsoever.

For each experiment we increment the value of the query limit at a step of 4, and keep the expansion threshold at 500, so for example "4-500" means a query limit of four and threshold at 500. We also include results with no query limit and zeroing the expansion threshold (inf-0), which amounts to a single query only and no expansion, thus forming our baseline. This particular case could possibly be considered identical with 1-500, since only a single query is eventually permitted. However, in the case of 1-500, query expansion is indeed performed and the first term in the insertion-ordered expansion set is selected for querying – likely a preflabel or altlabel – that can differ from the original input keyword.

7.2 Precision and recall

Precision is guaranteed by the soundness of the thesauri terminology: each concept searched for, say C, is a conceptual refinement of the primary one, say T in the semantic network of concepts. Therefore, search results that are valid for C would be precise for T also. This means that the precision of the federated search depends on the respective precision of the search engines (black boxes) of the repositories participating in the federation.

Given an average stable such precision, p_{rep}, our system performs additional queries and fetches an additional percent a of retrieved results, out of which $a \cdot p_{rep}$ would be correct. Then the precision of federated semantic search, p_{fed}, would be:

$$p_{fed} = \frac{\text{correct} + a \cdot p_{rep} \cdot \text{retrieved}}{\text{retrieved} + a \cdot \text{retrieved}} \tag{1}$$

and:

$$p_{rep} = \frac{\text{correct}}{\text{retrieved}}. \tag{2}$$

By replacing correct from (2) into (1) we get:

$$p_{fed} = \frac{p_{rep} \cdot \text{retrieved} + a \cdot p_{rep} \cdot \text{retrieved}}{\text{retrieved} + a \cdot \text{retrieved}} \Rightarrow p_{fed} = p_{rep}. \tag{3}$$

For recall, it holds that:

$$r_{fed} = \frac{\text{correct} + a \cdot p_{rep} \cdot \text{retrieved}}{\text{total}} \tag{4}$$

and:

$$r_{rep} = \frac{\text{correct}}{\text{total}}. \tag{5}$$

Therefore:

$$r_{fed} = r_{rep} \frac{\text{correct} + a \cdot p_{rep} \times \text{retrieved}}{\text{correct}}. \tag{6}$$

Replacing (2) in (6) gives:

$$r_{fed} = r_{rep} \frac{\text{correct} + a \cdot \text{correct}}{\text{correct}} \Rightarrow r_{fed} = r_{rep}(1 + a), \tag{7}$$

i.e. recall is increased by the factor of a.

7.3 Results and discussion

Our results for the two domain thesauri are summarized in Tables II-III. Notice that for the baseline, performance and recall are almost constant no matter query partitioning, which is to be expected because only the input keyword is queried. For 1-500, times are also not much affected (count also loading of the XML response), but there is a chance for a multilingual match in the two thesauri, thus fetching results from the non-Greek speaking repositories also.

The expansion starts to pay off is when more queries are permitted; then, the higher the keyword in the hierarchy, the more results are fetched. In contrast C4, containing terms that cannot be expanded, is constant in all experiments and equals the baseline. C3 grows stable after 4-500, apparently because it contains keywords that can hardly be expanded beyond 4 (they are leaves). Similarly, higher-ranked partitions would tend to stabilize for higher query limit values (e.g. C2 steadies after 12-500).

Table IV presents load times for all four thesauri, based on the arithmetic mean of the four query partitions, and summarizes likewise the increase in recall as a factor. Note that these metrics assume an equal distribution of the four possible events for keyword matches in a thesaurus. However, the "no-match" scenario (C4) is highly unlikely in practice, because the thesaurus has been developed using expert knowledge; at the same time keywords to the university courses are also assigned by experts – the course instructors themselves. As a result, a set of keywords for a given course will probably be within C1-3 rather than in C4, thus containing at least one thesaurus match.

	C1		C2		C3		C4	
Med	Time	Res	Time	Res	Time	Res	Time	Res
inf-0	4,645	20	2,767	23	2,697	20	2,573	20
1-500	5,792	40	5,351	38	3,131	30	2,721	20
4-500	12,784	138	11,928	135	7,003	76	3,103	23
8-500	26,108	246	17,615	181	7,396	72	2,893	20
12-500	36,511	364	26,569	202	8,284	72	2,785	20

Table II. Results and load times for the medicine thesaurus

	C1		C2		C3		C4	
Math	Time	Res	Time	Res	Time	Res	Time	Res
inf-0	2,958	23	2,807	22	2,635	12	2,428	17
1-500	4,367	37	5,751	39	4,149	27	2,487	17
4-500	16,257	140	15,969	124	9,966	78	3,158	18
8-500	32,216	278	23,082	195	8,967	79	2,781	17
12-500	38,770	403	25,291	223	8,877	79	2,637	17

Table III. Results and load times for the mathematics thesaurus

	Med		Math		TGT		MeSH	
	Time	Gain r_{fed}	Time	Gain r_{fed}	Time	Gain r_{fed}	Time	Gain r_{fed}
inf-0	3,170	0	2,707	0	7,782	0	35,556	0
1-500	4,249	0.5	4,188	0.6	10,977	0.5	36,531	0
4-500	8,704	3.5	11,338	3.9	14,909	3.6	47,501	2.7
8-500	13,503	5.2	16,762	6.7	17,046	5.3	59,662	4.9
12-500	18,537	6.9	18,894	8.8	20,509	7.4	74,190	6.9

Table IV. Load times and recall increase for the various thesauri

Regarding performance, it is easily seen that the complexity of our algorithm, given the BFS traversing is, in the case of thesauri trees, polynomial to the number of axioms in the ontology. The fact that we dully implement reasoning operations – like computing broader/narrower inverse relationships – without using a dedicated reasoner, is a plus. Therefore, relatively small thesauri, as the ones used by our system, pose no serious overhead in performance. Even for larger ontologies, such as MeSH, additional time is almost constant, no matter how deeply a keyword is expanded into the concept hierarchy. To see this, consider MeSH in Table IV and compare the load between the baseline (inf-0, no expansion) and 1-500 (full expansion), in both of which cases only a single query per input keyword is actually submitted.

The additional time imposed by query expansion in all cases is due to the initial loading of the ontology, as it is evident when comparing performance across the different sized thesauri. This can be easily amortized for example using prefetching. The actual time load comes from the federated queries. The more expanded queries submitted to the repositories, the longer it takes for the process to complete and merge the search results. We have configured an upper limit of four expanded queries per keyword, which seems to produce satisfactory information gain without giving up much of performance.

It is also clear that richer and more thoroughly developed thesauri, containing a large number of terms, do not necessarily have a greater impact on improvement. Rather, we need to increase the query limit to take advantage of expansion and we have shown this to be costly. Therefore, dense, medium-sized concept hierarchies are also capable of maintaining a balance between improved retrieval and performance.

8. Conclusions

Advances in interoperability in the educational sector combined with an increasing interest on distant learning have caused an explosion to the availability of learning material to the point of information overload. Oddly enough though, LMSs and computer-aided courses suffer a poverty of highly quality additional resources other than the course itself and few steps have been made to alleviate the traditional, strenuous handpicking. There is also a gap between the LORs and the systems that can actually reuse the learning objects, such as an LMS.

8.1 Findings and implications

In this paper, we have presented a framework and system for remedying this situation that revolves around two main axes: first, keyword-based query expansion using SKOS domain terminologies and second, federated search in LORs. We have shown that it is possible to retrieve additional LOs and achieve a corresponding increase in recall, while maintaining precision. SKOS expansion behaves well in a scholarly setting. Query expansion, when combined with federated search, can contribute toward LOs discovery at a balanced cost: we have proposed a semantic matching and expansion algorithm that exhibits safety switches able to control time load. It is possible to contain query drift by guarding term expansion with an appropriate threshold, which we have introduced into the algorithm. The benefits of using domain thesauri to enrich federated queries about LOs can be transparent to end users. Based on interoperability interfaces, search results can transparently integrate with an LMS, without the user being involved in the process, other than initiating a search.

8.2 On the use of vocabularies and thesauri

Clearly terminological expansion of queries can contribute toward improved recall. Our experimental results suggest that it is not necessary to adhere to complex and deep terminologies including a multitude of terms. The same, if any effect can also be achieved with smaller and dense thesauri trees that are also capable of improving retrieval, while maintaining performance at the same time. Reasoning is of key importance for the full

exploitation of thesauri, especially when they are incomplete or include implied knowledge of use to the expansion process. For this kind of expressivity, evidence has been presented that the implementation of reasoning operations instead of using a dedicated reasoner can avoid overhead and improve overall performance.

8.3 Limitations and future work

While SKOS expansion behaves well in a scholarly setting, this might not hold in all cases. For example, for informal, everyday queries this might not be the case. Both the thesauri used, as well as the search keywords for the annotation of learning material are provided by experts. Therefore, it is much more likely for matches to occur between the two, rather than when using generic web searches. The latter case is also prone to the effects of semantic and syntactic ambiguity which are almost nonexistent in expert domains, because of the specific terminology used. Lexical semantic networks would then be more appropriate instead of domain thesauri. However, there can be a gain even in the average case, as suggested by our results.

We have also demonstrated integration with a contemporary LMS, namely, eClass, which is the standard for higher education institutions in Greece and widely used elsewhere, through Claroline. It would be straightforward to consider other thesauri to also accommodate additional domains. Our system can be easily integrated with other platforms as well, building on open standards and RESTful communication. Finally, it would be worth investigating instructor feedback on the expansion process. This can be made implicit, by keeping track of what search results have been actually selected for annotation and then implement the most frequent semantic shortcuts that may occur between the initial term and the thesaurus terms that led to the chosen results. This can be the basis for an emerging ontology of collaborative expert intelligence.

Notes

1. www.claroline.net
2. http://snf-630087.vm.okeanos.grnet.gr:8888/SemanticMiddleware-1.0/results?q=keyword
3. www.ekt.gr/en/

References

ASPECT Project (2009), "ASPECT approach to federated search and harvesting of learning object repositories", Deliverable D2.1, ECP 2007 EDU 417008, available at: http://storage.eun.org/resources/upload/780/20170727_120401472_780_ASPECT_D2p1.pdf (accessed November 8, 2017).

Atenas, J. and Havemann, L. (2013), "Quality assurance in the open: an evaluation of OER repositories", *International Journal for Innovation and Quality in Learning*, Vol. 1 No. 2, pp. 22-34.

Bhogal, J., Macfarlane, A. and Smith, P. (2007), "A review of ontology-based query expansion", *Information Processing & Management*, Vol. 43 No. 4, pp. 866-886.

Bock, J., Haase, P., Ji, Q. and Volz, R. (2008), "Benchmarking OWL reasoners", *Proceedings of the ARea2008 Workshop*, *Tenerife*.

De la Prieta, F., Gil, A.B., Martín, A.J.S. and Zato, C. (2014), "Learning object repositories with federated searcher over the cloud", in Mascio, T., Gennari, R., Vitorini, P., Vicari, R. and de la Prieta, F. (Eds), *Methodologies and Intelligent Systems for Technology Enhanced Learning. Advances in Intelligent Systems and Computing*, Vol. 292, Springer, Cham.

Dietze, S., Sanchez-Alonso, S., Ebner, H., Yu, H., Giordano, D., Marenzi, I. and Pereira Nunes, B. (2013), "Interlinking educational resources and the web of data – a survey of challenges and approaches", *Program*, Vol. 47 No. 1, pp. 60-91.

Duval, E., Forte, E., Cardinaels, K., Verhoeven, B., Van Durm, R., Hendrikx, K., Forte, M.W., Ebel, N., Macowicz, M., Warkentyne, K. and Haenni, F. (2001), "The Ariadne knowledge pool system", *Communications of the ACM*, Vol. 44 No. 5, pp. 72-78.

Europe PMC Consortium (2014), "Europe PMC: a full-text literature database for the life sciences and platform for innovation", *Nucleic Acids Research*, Vol. 43 No. 2015, pp. D1042-D1048.

GUnet asynchronous eLearning group, "Platform description (Open eClass 2.10)", available at: http://docs.openeclass.org/en:detail_descr (accessed November 8, 2017).

Haslhofer, B., Martins, F. and Magalhães, J. (2013), "Using SKOS vocabularies for improving web search", *Proceedings of the 22nd International Conference on World Wide Web Companion. International World Wide Web Conferences Steering Committee*, pp. 1253-1258.

Horridge, M. and Bechhofer, S. (2011), "The OWL API: a Java API for OWL ontologies", *Semantic Web*, Vol. 2 No. 1, pp. 11-21.

Horridge, M., Tudorache, T., Nuylas, C., Vendetti, J., Noy, N.F. and Musen, M.A. (2014), "WebProtege: a collaborative web based platform for editing biomedical ontologies", *Bioinformatics*, Vol. 30 No. 16, pp. 2384-2385.

McMartin, F. (2006), "MERLOT: a model for user involvement in digital library design and implementation", *Journal of Digital Information*, Vol. 5 No. 3, available at: https://journals.tdl.org/jodi/index.php/jodi/article/view/143 (accessed November 8, 2017).

Massart, D. and Le, T.D. (2004), "Federated search of learning object repositories: the CELEBRATE approach", *RIVF*, pp. 143-146.

Mazo, S., Otón, S., de-Marcos, L., García, A. and García, E. (2012), "RESTful service oriented architecture for querying and publishing learning objects in repositories", *Proceedings of the Fourth International Conference on Mobile, Hybrid, and On-line Learning (eLmL)*, pp. 20-23.

Miles, A. and Bechhofer, S. (2009), "SKOS simple knowledge organization system reference", W3C recommendation, 18, W3C, available at: www.w3.org/TR/skos-reference (accessed November 8, 2017).

Ochoa, X. and Duval, E. (2009), "Quantitative analysis of learning object repositories", *IEEE Transactions on Learning Technologies*, Vol. 2 No. 3, pp. 226-238.

Piedra, N., Chicaiza, J.A., López, J. and Tovar, E. (2014), "An architecture based on linked data technologies for the integration and reuse of OER in MOOCs context", *Open Praxis*, Vol. 6 No. 2, pp. 171-187.

Segura, N.A., García-Barriocanal, E. and Prieto, M. (2011), "An empirical analysis of ontology-based query expansion for learning resource searches using MERLOT and the Gene ontology", *Knowledge-Based Systems*, Vol. 24 No. 1, pp. 119-133.

Solomou, G. and Papatheodorou, T. (2010), "The use of SKOS vocabularies in digital repositories: the DSpace case", *IEEE Fourth International Conference on Semantic Computing (ICSC)*, pp. 542-547.

Ternier, S., Duval, E., Massart, D., Campi, A., Guinea, S. and Ceri, S. (2008), "Interoperability for searching learning object repositories: the ProLearn query language", *D-Lib Magazine*, Vol. 14 Nos 1/2, available at: www.dlib.org/dlib/january08/ceri/01ceri.html (accessed November 8, 2017).

Van Assche, F., Duval, E., Massart, D., Olmedilla, D., Simon, B., Sobernig, S., Ternier, S. and Wild, F. (2006), "Spinning interoperable applications for teaching & learning using the simple query interface", *Journal of Educational Technology & Society*, Vol. 9 No. 2, pp. 51-67.

van Assem, M., Malaisé, V., Miles, A. and Schreiber, G. (2006), "A method to convert thesauri to SKOS", in Sure, Y. and Domingue, J. (Eds), The Semantic Web: Research and Applications, ESWC Lecture Notes in Computer Science, Vol. 4011, Springer, Berlin, Heidelberg, pp. 95-109.

Corresponding author
Dimitrios Koutsomitropoulos can be contacted at: kotsomit@ceid.upatras.gr