Crosslingual Ontology-Based Document Retrieval

Eelco Mossel
University of Hamburg
Department of Informatics
Vogt-Kölln-Str. 30
D-22527 Hamburg, Germany
mossel@informatik.uni-hamburg.de

Abstract

An approach for crosslingual ontology-based document retrieval has been devised and is being implemented. It allows the user to enter a query in any language that is part of the system and retrieve documents in selected languages. A domain ontology and term-concept lexicons, containing synonymous terms where applicable, are used to overcome discrepancies between the search query and the words occurring in the documents, in a monolingual situation for the individual languages as well as in a crosslingual setting.

The ontology is used in two different ways. First, concepts relevant for a search query are found automatically and used to retrieve documents. Second, relevant parts of the ontology are displayed to the user, who can navigate further starting from the displayed part of the ontology, and explicitly select concepts to continue the search with.

Keywords

multilinguality, document retrieval, search, ontologies, eLearning

1. Introduction

The most well-known kind of document retrieval is based on full-text indexing, and the retrieval results heavily depend on the correct choice of words and word forms in the query. With the increasing availability of digital documents, also in languages other than English, it is important to have more targeted means to find the wanted document from an available collection. This applies particularly to the multilingual case, where the user would have to translate his search query into all the languages he wants to retrieve documents in, which requires an active command of the involved languages and in particular knowledge about the habitual terms in the respective domain.

In this paper, I describe a setup for using domain ontologies to facilitate crosslingual search in an available document collection. The approach was developed in the framework of the *BIS-21++* project[1] and implemented within the project *LT4eL* (Language Technology for eLearning)[2]. The design fits closely to the resources and needs of LT4eL, but is in principle applicable to other situations with similar preconditions.

The project LT4eL aims at improving the retrieval and usability of learning materials in multiple languages. Among other things, learning materials are semi-automatically annotated with keywords and concepts, and a domain ontology has been created.

The relevance of crosslingual retrieval in eLearning lies in the fact, that learning materials might be available in a different language than the student's native language. Instead of having to come up with a good search query in foreign languages, the user will be presented immediately with relevant documents in languages he knows.

In the framework of LT4eL, the search functionality described in this paper is being integrated in the Learning Management System ILIAS[3].

Section 2 describes the general idea for the crosslingual ontology-based search, the underlying assumptions and a user scenario. Based on all this, the search functionality was developed, as described in Section 3. Section 4 explains how the approach will be integrated and applied in the project LT4eL. Section 5 contains the conclusion and information about future work.

2. Basis for the document retrieval

The goal of the approach is a search functionality with the following three characteristics:

1. Improved access to documents. The search mechanism must exploit semantic characteristics of search queries and documents, and be able to find relevant documents that would not be found by a simple full-text search[4].
2. Multilinguality. With one implementation, the approach should work for multiple languages.

[1] Bulgarian IST Centre of Competence in 21 Century (BIS-21++, http://bis-21pp.acad.bg/): one-month research stay of Eelco Mossel at the Bulgarian Academy of Sciences, Institute for Parallel Processing, Sofia.

[2] www.lt4el.eu - The LT4eL project is supported by the European Community under the Information Society and Media Directorate, Learning and Cultural Heritage Unit.

[3] www.ilias.de

[4] Or rather, the relevant documents must be found in a more efficient way than having to try numerous full-text searches, hoping to capture the words that occur in a relevant document.

3. Crosslinguality. It should be possible to retrieve documents in languages other than the language that is used for representing the ontology or formulating/selecting terms.

2.1 Assumptions

The starting point for designing an ontology-based search functionality have been the following assumptions, which hold for LT4eL and are realised as described in [2].

- There is a multilingual document collection (this is not a requirement, but with documents in only one language, the full potential of the approach will not be exploited).
- There is a (language independent) ontology that includes a domain ontology on the domain of the documents. There may be more than one domain ontology. A domain ontology consists of a set of concepts, belonging to the same domain, and various kinds of relations between the concepts are possible. Between two separate domain ontologies, there are no relations, because they have been created independently. But as they can all be connected to the same upper ontology and become a part of one large ontology in this way, it is not necessary to iterate over a list of ontologies – the descriptions in this paper always just refer to "the ontology".
- For each of the addressed languages, there is a lexicon with words or phrases that are mapped to concepts of the ontology.
- The concepts have a name and/or description in each of the addressed languages, which can be used for presenting them in the user interface.
- The documents are annotated with concepts. In the simplest case, for each document there is a set of concepts that are relevant for the contents of the document. In LT4eL, more detailed information is available: the annotation of a concept is attached to every place in the document where this concept is mentioned. It can thus be derived, how often each concept occurs. If this information and also the length of the document are available, they are used for ranking the found documents.
- For each document, its language is available (requirement).
- It is known, which languages the user can use to formulate his query (e.g. by means of a user profile). This is used to determine which lexicons to use; however, if it is not known, all available lexicons could be consulted by default.
- It is known, in which languages the user wants to retrieve documents (if this is not known, all relevant documents in any available language could be shown by default).

Having those assumptions in mind, I proceeded to define a search scenario (see Section 2.2) from the point of view of the user, showing what the user will (have to) do and what he will see. On the basis of this scenario, the search functionality (see Section 3) was specified.

The basic idea of the ontology-based search (or *semantic search*) is, that concepts from the ontology lead to the documents to retrieve. The search will probably work best, when the user selects exactly the concept(s) from the ontology that he wants as a topic for the retrieved documents. However, there are two reasons to start with a free-text query by the user.

First, the semantic search will "compete" with other search strategies such as *full-text search* (all words from the text are considered when looking for a match for the query) and *keyword search* (matching with words that are annotated as keywords in the documents – to avoid confusion, the words the user types in are called *search words* and not keywords). The user, who is used to Google, wants his results fast, with not too many intermediate steps where he has to choose things. Therefore, we would like to invoke semantic search as soon as the user has entered his search words, and give first results simultaneously with the other search methods.

Second, it is good to give the user a starting point for finding the desired concept in the ontology, so that he does not have to start at the root of the ontology. The search words are used to find the starting point in the ontology. In a second step, the search can be refined by selecting concepts from the ontology. Because of this approach, two different ways of semantic search occur in the search scenario.

2.2 Search scenario

1. **Submit query**
   User enters free text and submits this as a query.
2. **See document list**
   A list of retrieved documents is displayed with some meta information, for example:
   - title;
   - length;
   - original language;
   - keywords and concepts that are common to both the query and the document;
   - other keywords and concepts that are related to the document but not to the query.
3. **See concept browsing units**
   Each concept that is assumed to be related to the search query, is presented to the user together with its neighbourhood from the ontology (related concepts and relations between the concepts). Call such a displayed part of the ontology a "browsing unit" (for an example see Figure 1). If no concept related to the search query is found, the root of the ontology with its neighbourhood is chosen as the browsing unit. The user can browse the ontology: by clicking on a concept or a related button, the user will see related

concept. If there are many relations, a possibility is, to let the user select the kind of relation he would like to use to explore the ontology (e.g. "show only concepts that are related to ApplicationProgram by a part-of relation").

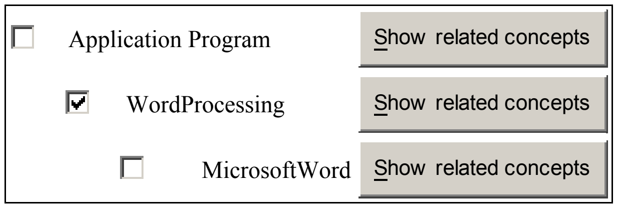
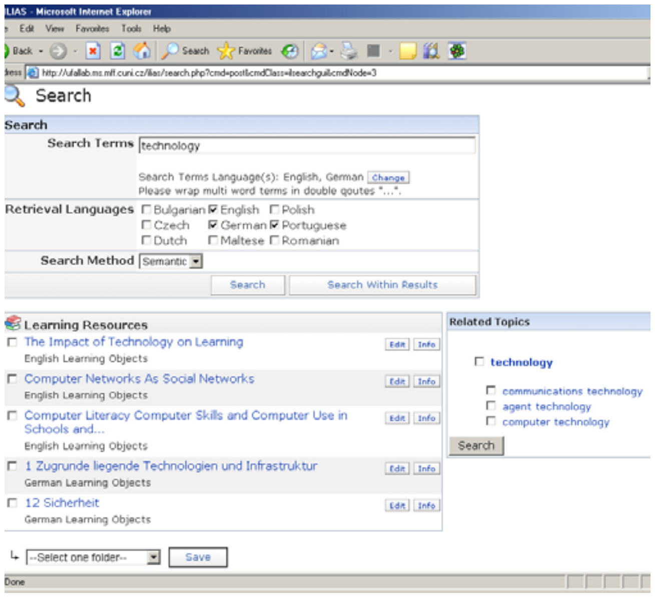


Figure 1 – Example of a possible representation of a browsing unit, where only taxonomical relations are present.

4. **View documents**
   User looks into the documents from the list.
5. **Browse ontology**
   User browses the ontology: starting from the presented concepts, he can proceed to related concepts, and concepts that are related to those again, etc.


ILIAS - Microsoft Internet Explorer
Edit View Favorites Tools Help
Back
Search
Favorites

http://ufallab.ms.mff.cuni.cz/ilias/search.php?cmd=post&cmdClass=ilsearchgui&cmdNode=3

Search

Search

Search Terms: technology

Search Terms Language(s): English, German Change
Please wrap multi word terms in double qoutes "...".

Retrieval Languages: ☐ Bulgarian ☑ English ☐ Polish ☐ Czech ☑ German ☑ Portuguese ☐ Dutch ☐ Maltese ☐ Romanian

Search Method: Semantic

Search | Search Within Results

Learning Resources

- ☐ The Impact of Technology on Learning — English Learning Objects — Edit Info
- ☐ Computer Networks As Social Networks — English Learning Objects — Edit Info
- ☐ Computer Literacy Computer Skills and Computer Use in Schools and... — English Learning Objects — Edit Info
- ☐ 1 Zugrunde liegende Technologien und Infrastruktur — German Learning Objects — Edit Info
- ☐ 12 Sicherheit — German Learning Objects — Edit Info

Related Topics

- ☐ technology
  - ☐ communications technology
  - ☐ agent technology
  - ☐ computer technology

Search

--Select one folder-- Save

Done

Figure 2 – User interface for search in ILIAS. In the upper part, search words can be entered and languages can be selected. In the lower left part, the resulting documents are listed. In the right part, concepts can be selected.

6. **Select concepts**
   User selects a set of concepts ontology fragments (sets of related concepts, possibly only indirectly related) from the presented browsing units.
7. **Select search option**
   User selects an option about how to use the ontology fragments for search. Options might include "disjunctive" (find documents, in which any of the selected concepts occur) and "conjunctive" (find documents, in which all of the selected concepts occur).
8. **See new document list**
   A new list of documents is displayed, based on ontological search. If also new search terms have been specified, the documents based on the selected concepts come first, followed by the ones that are found only by the new search terms and not by the selected concepts.
9. **See browsing units including shared concepts**
   As in step 3, but now, also *shared concepts* are presented: concepts, that are common to more than N of the found documents; this includes the concepts that were used as the search key but might include further concepts. The number of documents, specified by the threshold parameter N, can be relative (a percentage of the number of found documents) or absolute.
10. **Repeat steps from step 6 (Select concepts)**
    User selects another set of related concepts and submits it as the search key, etc.

The search scenario is now realised in LT4eL by integrating the functionality into the Learning Management System ILIAS. Figure 2 shows the user interface.

3. Design of the search functionality

The search functionality as a whole has as input parameters:

- Possible languages of search query (determines which lexicons to use for lookup)
- Retrieval languages (find documents in those languages)
- Search terms, entered by user
- Concepts, selected by user
- Method for combining the concepts (conjunctive, disjunctive)
- Option indicating whether to also retrieve documents that do not contain the desired concept, but do contain a superconcept or a subconcept of it

and as output:

- the ID of the found documents
- the initial contents for the browsing units (parts of the ontology, see further Section 3.5), including shared concepts

To support the search scenario outlined in Section 2.2, the search function has to incorporate the following steps.

1. Find terms in the term-concept-lexicons that match the search query.
2. Find corresponding concepts for derived terms.
3. Find relevant documents for concepts.
4. Create a ranking for a set of found documents.
5. Support displaying and selecting concepts from the ontology.
6. Find "shared concepts": concepts occurring in multiple documents.

In the subsections 3.1 through 3.6, each of those points is explicated.

3.1 From query to term-concept mappings

A few issues have to be regarded when trying to transform a query to terms that could be found in a term-concept-lexicon, namely:

- Uppercase/lowercase differences
- For languages other than English, words that officially contain letters with diacritics might be entered in the search query using simple characters without diacritics, depending on the keyboard layout and settings.
- Dashes, spaces and apostrophes might be added or omitted by the user.

A thorough way to find all terms that are relevant for an entered search query is the following.

1. Tokenise the query.
2. Create combinations for multi-word terms. Not only the individual words could refer to concepts, but also sequences of words. Depending on the language, they can be compound words, such as “school year”, or expressions and collocations that might denote a concept, such as “high-speed connection”. There are several ways to form compounds. Besides treating them as a whole with just a space in between, also concatenating them with a dash or with nothing in between is possible.

Currently, the lexicon lookup is just simple string matching, so all the possible multi-word terms have to be generated from the query and looked up one by one. The basis for the combination are all the subsequences of the query; for example, for query “a b c d”, the lists of terms to be combined would be [a,b] [b,c] [c,d] [a,b,c] [b,c,d] [a,b,c,d].

3. Find a normalised form for each term, as not all inflected forms are included in the lexicons. The LT4eL lexicons mainly contain lemmas. The best recall will be achieved, if for every addressed language, a lemmatiser is used. Furthermore, depending on the matching mechanism, normalisation can comprise replacement of diacritics and uppercase letters by their simple, lowercase variants.
4. The set of strings for lookup consists of the original terms, the normalised terms and the created multi-word terms.

The current prototype supports the creation of multi-word terms, but no lemmatisation.

Instead of adding lemmatisation, the following alternative approaches can be considered:

- Expand the term lexicon, by generating word forms.
- Allow partial matches: take lexicon terms into account, that only match part of a token of the query, and the other way around: lexicon terms of which only a part matches a token from the query. Or even more flexible: take into account such partial matches, where a token of the query and a lexicon term have a part in common, but both have an unmatched part. Obviously, this solution has disadvantages. It can result in false matches, while at the same time it does not work for an inflected word form whose stem is different from the stem of the corresponding lemma. And decisions will have to be taken on how large the overlapping part must be. Perhaps this parameter should even be chosen language-independently.

The partial-matching solution would not only work for single words: it could also be used instead of generating possible multi-word terms from the query.

Related to the mechanism of automatic partial matching, an interactive way of term selection can be thought of: the user will see a dynamically changing list of all available terms that contain the entered characters as a substring[5]. Every time a character is added or deleted, the list is updated to reflect the change. When this works fast, it is a convenient way of selecting terms. However, it is conflicting with the idea of just typing a free-text query, which can be used for non-ontological search at the same time.

3.2 From terms to concepts

When trying to find a concept for a certain term, in principle, the following situations can occur:

- The corresponding concept is missing from ontology.
- Unique result: the term is a lexicalisation of exactly one concept.
- The term denotes multiple concepts from one domain, for example:
  - Key (from keyboard).
  - Key (in database).
- The term denotes concepts from more domains, for example:
  - Window (graphical representation on monitor).
  - Window (part of a building).
- The term denotes different concepts for different languages:
  - “Kind” (English: sort/type)
  - “Kind” (German: child)

By presenting all the possibilities to the user and let him choose, the ambiguities do not have to be resolved automatically.

3.3 From concepts to documents

Given the annotation of concepts in the documents, it is a trivial task to find the documents dealing with a certain

[5] This mechanism is used in electronic dictionaries, with the restriction, that substrings are matched only from the beginning of the word.

concept. Two non-trivial issues, however, are the following.

First, what further role can the ontology play in finding relevant documents? The concepts are not just labels for the documents; they have their place in the used ontology. Intuitively, a document dealing with a subtopic of the desired topic (thus containing a subconcept of the central concept) partially satisfies the information need of the user, and should therefore be included in the result. This brings us to the question, whether this also holds for the subconcept of this subconcept. And it might also apply for other kinds of relations, such as part-of. But every step down in the taxonomy changes the level of detail, and every step following a different relation might take us further away from the desired topic (e.g. a computer has a processor, a processor is made of silicon, silicon is produced in Australia, …)

In order to make use of the relations in the ontology and yet avoid very unpredictable results, we employ the following strategy. The user can select for each available kind of relation (currently: superconcept and subconcept) whether he wants to include the concepts reached by this kind of relation in the search. Two restrictions are used: only related concepts are used that are one step away in the ontology, and this is done only, when a search concept occurs in less than N documents (determined by a threshold parameter). In addition, however, the user can select an option to include the related concepts in any case, independent of the threshold.

The usefulness of this kind of query expansion also depends on the quality and granularity of the ontology. In the current LT4eL ontology on the domain of computer science for non-experts, for example, "Computervirus" is a subconcept of "Program", but most people will probably not be interested in computer viruses when searching for "program". On the other hand, adding results for "web portal" (a subconcept of "web site") to the results for "web site" is very useful.

The second issue concerns how to deal with the selection of more than one concept from the ontology. Two obvious possibilities, which are also easy to understand for the user and do not involve explicit specification of boolean operators, are *conjunctive search* (find only documents in which all of the selected concepts occur) and *disjunctive search* (find documents in which any of the selected concepts occur). However, when using the ranking criterion that is described in Section 3.4, the top-ranked results of the disjunctive search will be exactly the same as the results for conjunctive search.

Baumann et al. [1] use an ontology for automatic query expansion. They state a relationship between the issues of related concepts and disjunctive/conjunctive search in the following way: "we are generating boolean queries on the fly based on the assumption that part-of edges can be interpreted as every sub-concept should be contained in one document, while edges of the type is-a allow for alternatives." This is in line with our approach where a document is returned if a subconcept occurs instead of the original concept; with other relations, we cannot experiment yet in LT4eL (see also Section 4).

3.4 Ranking

As a measure to rank the retrieved documents, we use the following two ranking criteria:

1. The number of different concepts that are used for searching *and* occur in the document. The intuition is, that a document serves the search query better, if a larger part of the search query is matched.

   Example:

   - User enters terms *create* and *folder*.
   - *create* denotes concept *CREATE*
   - *folder* denotes concept *DIRECTORY*

   So the search is done with concepts *CREATE* and *DIRECTORY*.

   - Document 1 deals with concepts *CREATE*, *DIRECTORY* and *LINUX*.
   - Document 2 deals with concepts *FILE*, *DIRECTORY*, *LINUX* and *SYMBOLICLINK*.
   - Document 1 contains two concepts that were used for searching, so it is ranked higher than document 2, which only has *DIRECTORY* in common with the search query.

2. Normalised *annotation frequency*: the number of times that the looked up concepts are annotated in the documents, divided by the length of the document. This criterion is applied to documents that got the same ranking by the first criterion. The normalisation (dividing by the document length) is done because otherwise, long documents would be favoured, as they are more likely to mention the concept more often. However, if the occurrences are concentrated only in certain parts of the document, the document as a whole will have a low score because of the division by a relatively large length, even though it contains very relevant parts. This low score can be justified by the fact that indeed not the whole document is very relevant. Another option could be to take not the length of the whole document but the length of the section in which the concept occurs several times. However, this makes things more complicated than it seems at first sight: if the concept happens to occur also outside of the relevant part, the entire (much larger) distance between the first and this last occurrence would be taken as divisor, so the score will be penalised just because of the additional occurrence.

If a subconcept or superconcept is counted instead of the original concept, its will get a lower weight (determined by a factor) in the calculation. This holds for both of these ranking criteria.

This is a relatively easy ranking mechanism; experiments will show whether it is satisfying. Vallet et al. [4], who describe a similar approach for semantic search, although not multilingual, use a more advanced ranking algorithm, based on similarity vectors between queries and documents.

3.5 Browsing Units and Ontology Fragments

For the second phase of the search (searching documents by selecting concepts), ontology fragments (parts of the ontology) will be displayed to the user as *browsing units* (cf. point 3 in Section 2.2).

For every concept that is found through the search terms, the neighbourhood of the concept is looked up in the ontology, and the result is passed to the search function as an ontology fragment. In order to not display the same part of the ontology twice, overlapping fragments are optionally merged, and displayed within one browsing unit. For example, if one fragment contains “A has subclass B”, and another contains “A has subclass C”, then the merged fragment will be displayed as A having subclasses B and C.

From the architecture perspective, the browsing units are on the border between the search functionality and the graphical user interface. The search function determines which of the concepts are displayed initially, but the GUI determines, besides from the fact how they are displayed, what will be presented when navigating through the browsing units.

3.6 Shared concepts

As described in step 9 of the user scenario, a set of concepts that are common to a certain part of the found documents is calculated. Ideally, this includes concepts that were not found by means of the search terms, and it can guide the user to other documents he is interested in but could not figure out the direct way to them. For example, the user enters terms leading to the concept denoted by “Report”. Some documents on academic writing are found, and they share the concept “Publication”.

For the first experiments, we will use a threshold of 50%: the set of shared concepts includes each concepts, that occurs in at least 50% of the found documents (the 50% of the documents can be a different subset for each shared concept).

The idea of using the contents of retrieved documents to adapt/expand a search query and find additional relevant documents is applied by Song et al. [3] on the level of terms rather than concepts, and in an automatic way.

4. Architecture and use within LT4eL

In the project LT4eL, learning objects (LOs) have been collected in eight languages: Bulgarian, Czech, Dutch, English, German, Polish, Portuguese, Romanian. The size of the collection varies from around 30 to 80 documents per language, with a total of approximately 200,000 words per language.

One domain, that is covered by LOs in all of those languages can be called “computer or information science for non-experts”. Based on the contents of those LOs, a domain ontology for this domain was created. It contains 700 concepts and only taxonomic relations.

Furthermore, for the languages mentioned above plus Maltese, term lexicons are being generated, containing lexicalisations for all concepts of the ontology. The lexicons contain a few synonymous terms: terms in the same language, that are mapped to the same concept. Some of the terms consist of multiple words. The occurrences of terms that denote a concept from the ontology have been semi-automatically annotated in the documents.

The lower part of Figure 3 gives a schematic view of the relation between terms, ontology and LOs.

The search functionality, as described in Section 3, uses the following functions during runtime (not necessarily in this order – the following numbers 1 through 8 are referred to in Figure 3, while the order of steps is given in the beginning of Section 3):

1. For a specified term and a specified language, find all the concepts denoted by this term in this language.
2. Find the superconcepts for a specified concept.
3. Find the subconcepts for a specified concept.
4. For a specified concept, find all the concepts that are reachable by at most one step by following any relational edge in the ontology.
5. Find all the documents in a certain language that contain the specified concept(s).
6. For a specified document, return all the concepts that occur in it.
7. For a specified document and a specified concept, return the number of times the concept occurs in the document (annotation frequency).
8. Return the length of a document (used for normalisation).

These functions are provided by three software components, developed by IPP-BAS[6]:

- Lexicon Tool (LEX).
- Ontology Management System (OMS). It allows for more complex reasoning than is currently used by the search functionality.

[6] Institute for Parallel Processing at the Bulgarian Academy of Sciences, Sofia.

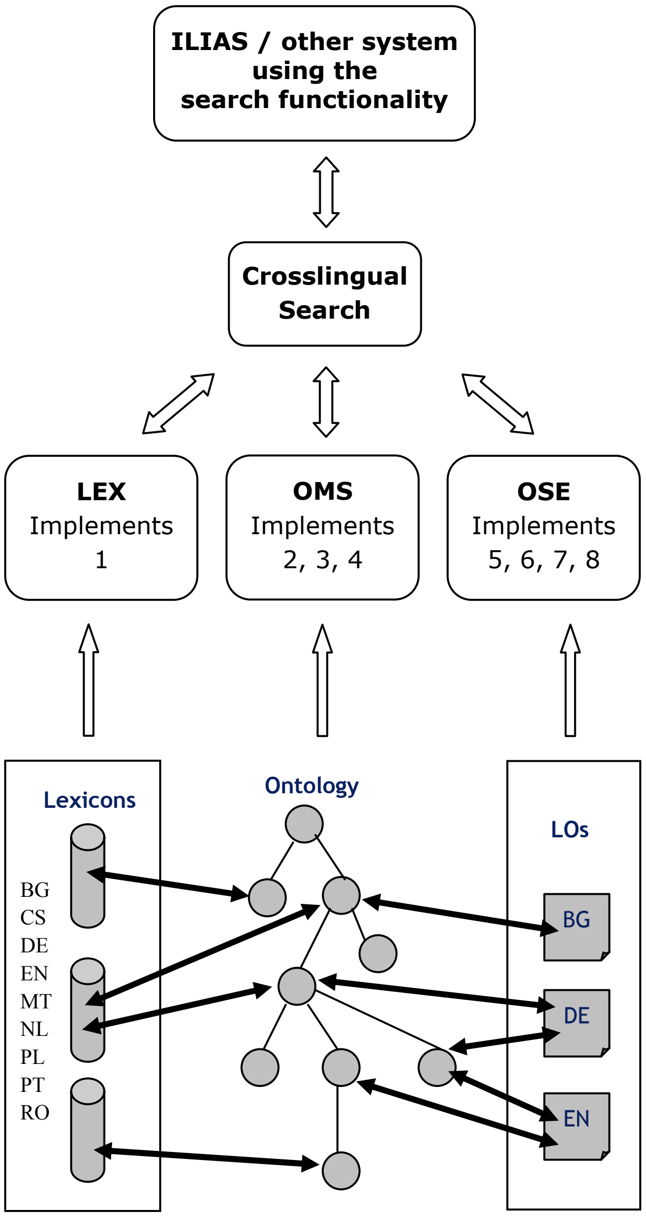


Figure 3 - The architecture in which the search functionality is integrated. The numbers in the upper part refer to the functions described in this section. The lower part of the diagram shows the relationships between terms, ontology and learning objects.

- Ontology-based Search Engine (OSE). It is used for the functions 5 through 8, but also provides more advanced functionality, such as finding documents in which several concepts occur within the same paragraph.

Figure 3 gives an overview of the architecture. The search functionality is being integrated into the Learning Management System ILIAS, in which the collected LOs are stored.

5. Conclusion and further work

An approach for crosslingual ontology-based document retrieval has been devised and is being implemented. It allows the user to enter a query in any language that is part of the system and retrieve documents in selected languages. A domain ontology and term-concept lexicons, containing synonymous terms where applicable, are used to overcome discrepancies between the search query and the words occurring in the documents, in a monolingual situation for the individual languages as well as in a crosslingual setting.

The ontology is used in two different ways. First, concepts relevant for a search query are found automatically and used to retrieve documents. Second, relevant parts of the ontology are displayed to the user, who can navigate further starting from the displayed part of the ontology, and explicitly select concepts to continue the search with.

Several decisions had to be taken about the chain from search query through terms and concepts to documents, about ranking and query expansion. The chosen solutions and some alternatives were discussed in this paper.

The search functionality is being integrated and tested in the Learning Management System ILIAS in the framework of the project LT4eL, which provides a multilingual collection of learning materials that are annotated with concepts, a domain ontology that covers the domain of a substantial part of the learning materials, and ontology, and a term lexicon with mappings to the concepts of the ontology.

Experiments, to be carried out in the near future, have to show whether the provided options are useful, whether the relevant documents are easily found, and whether the ranking mechanism is satisfying.

For the monolingual case, we will evaluate in terms of recall/precision, whether documents relevant to a certain query are better found by the semantic search than by full-text search.

For the multilingual case, scenarios involving bilingual or multilingual test persons will be set up, to show the added value of crosslingual search in a learning management system. Learning material that is useful for a certain task will have to be found from the available collection; a collection that covers the topic better can be obtained by selecting more than one retrieval language.

References

[1] S. Baumann, A. Dengel, M. Junker, T. Kieninger. "Combining Ontologies and Document Retrieval Techniques: A Case Study for an E-Learning Scenario". *Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA 2002)*, 133-137. Aix-en-Provence, France, 2002.

[2] L. Lemnitzer, P. Monachesi, K. Simov, A. Killing, D. Evans and C. Vertan. “Improving the search for learning objects with keywords and ontologies”. To appear in: *Proceedings of the Second European Conference on Technology Enhanced Learning (EC-TEL 2007)*. Crete, Greece, 2007.

[3] M. Song, I. Song, X. Hu and R. Allen. “Semantic Query Expansion Combining Association Rules with Ontologies and Information Retrieval Techniques.” In: A. Tjoa and J. Trujillo (eds.), *Proceedings of the 7th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2005)*, 326-335. Copenhagen, Denmark, 2005.

[4] D. Vallet, M. Fernández and P. Castells. “An Ontology-Based Information Retrieval Model.” In: Gómez-Pérez, A., Euzenat, J. (eds.), *The Semantic Web: Research and Applications: 2nd European Semantic Web Conference (ESWC 2005)*, 455-470.