

A Semi-Automatic Semantic Annotation and Authoring Tool for a Library Help Desk Service

Antti Vehviläinen
Helsinki University of
Technology (TKK)
Laboratory of Media
Technology
Semantic Computing
Research Group
<http://www.seco.tkk.fi>
Antti.Vehvilainen@tkk.fi

Eero Hyvönen
Helsinki University of
Technology (TKK)
Laboratory of Media
Technology and
University of Helsinki
Semantic Computing
Research Group
<http://www.seco.tkk.fi>
Eero.Hyvonen@tkk.fi

Olli Alm
University of Helsinki and
Helsinki University of
Technology (TKK)
Semantic Computing
Research Group
<http://www.seco.tkk.fi>
Olli.Alm@tkk.fi

ABSTRACT

This paper discusses how knowledge technologies can be utilized in creating help desk services on the semantic web. To ease the content indexer's work, we propose semi-automatic semantic annotation of natural language text for annotating question-answer (QA) pairs, and case-based reasoning techniques for finding similar questions. To provide answers matching with the indexer's and end-user's information needs, methods for combining case-based reasoning with semantic search and browsing are proposed. We integrate different data sources by using large ontologies of upper common concepts, places, and agents. Techniques to utilize these sources in authoring answers are suggested. A prototype implementation of a real life ontology-based help desk application is presented as a proof of concept. This system is based on the data set of over 20,000 QA pairs and the operational principles of an existing national library help desk service in Finland.

1. INTRODUCTION

Companies and public organizations widely use help desk services in order to solve problems for their customers. The classic example of a help desk service is a call center, where support persons answer questions by phone or by email. As help desk services are being transferred to the Web, it's more and more common that the customers have also the possibility to solve their problems by themselves by using the knowledge and content accumulated at the service, without contacting a support person directly [5]. A simple approach, for example, is to publish Frequently Asked Questions (FAQ) lists on the web. The option to use a simple and fast question-answer (QA) self-service is appreciated not only by the customers, but by the authors of the answers, too. Their time is saved, if the QA service can automatically provide an answer to the customer. Furthermore, the author can use the accumulated QA knowledge of the service by herself, which helps in authoring the answers and improves the quality of the answers.

This paper discusses applications of semantic web technologies to help desk services. We focus on QA help desk ser-

vices, where the database of the service is composed of previously answered questions, i.e., QA pairs. In such a service the user has a question in mind, and the service has two major tasks:

1. Finding relevant previous answers. A search method is needed to find the already answered relevant QA-pairs from the repository.
2. Authoring a new answer. An existing QA pair may satisfy the customers information need, but usually some kind of adaptation of the old answer case is needed. Usually answers are created and modified manually by a human editor.

The research problem of this paper is to investigate how to support semi-automatic answer authoring in a QA help desk service. Our methodology is to use semantic web technologies in content annotation, in utilizing the QA repository, and in integrating information available online on the web with the authoring process and the answers.

In this paper, when we use the term *indexing* we refer to the old, existing way of doing indexing where *index terms* are just strings without an ontological reference. We use the term *annotation* to refer to the new way of using *annotation concepts* that have an ontological reference.

1.1 The Existing Service

The research is based on a real life case study: we use the data set of the operational *Ask a librarian* service¹ offered nationally in Finland by the editors of the Libraries.fi² portal. In this service the clients can send questions to a virtual librarian via email, and a librarian of the service provides an answer within three working days. Some of the questions that the clients send are simple and the librarian can answer them straight away. These include questions about the opening times of a library, how to make an inter-library loan etc. However, most of the questions require that the

¹<http://www.kirjastot.fi/tietopalvelu>

²Libraries.fi provides access to Finnish Library Net Services under one user interface, see <http://www.libraries.fi>.

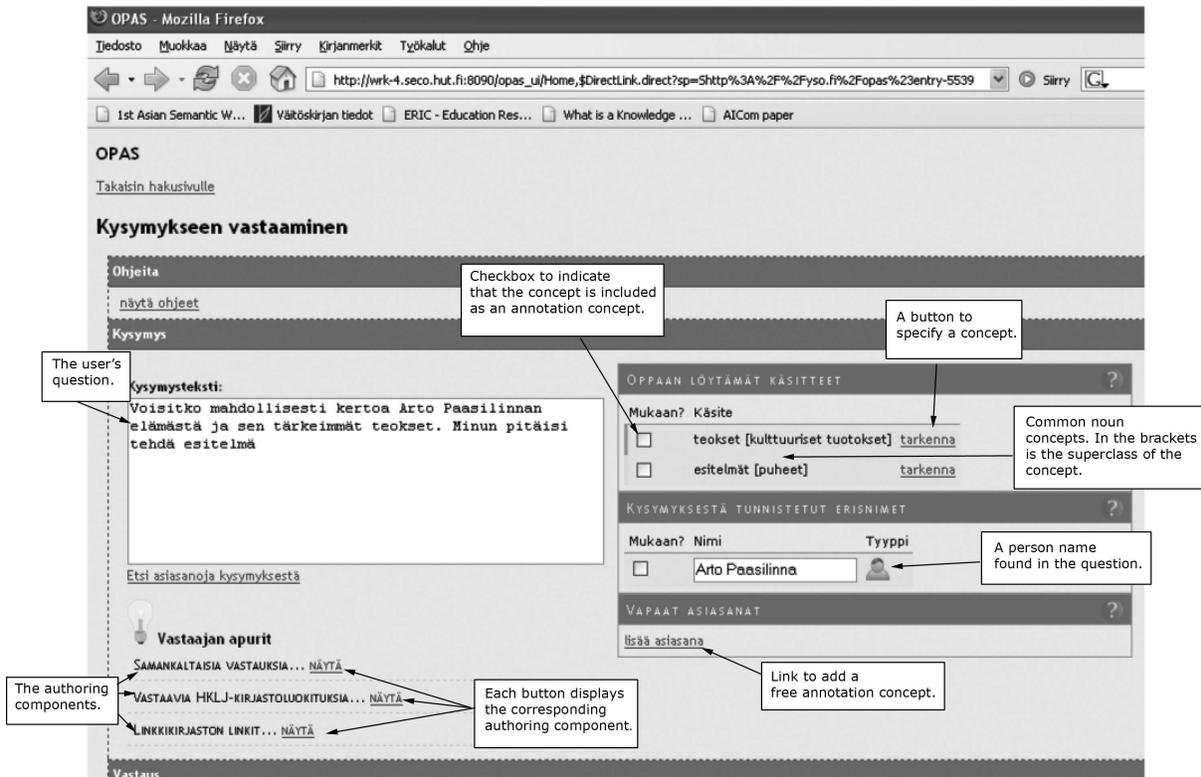


Figure 1: Question text, concepts found by Poka and similar questions in Opas UI.

librarian uses more time to investigate the subject of the question. These include questions like *I'm wondering where I could find information about studies of the library and information science?* or *I'm giving a presentation of Nokia. Where I could find helpful information?* Answers to these questions span typically a few paragraphs of text and contain some links to useful web sites. The librarians report that on average they use from half an hour to an hour to compose such an answer.

Each QA pair has been indexed using the YSA thesaurus³ of some 23,000 common Finnish terms. At the moment the data set consists of over 20,000 QA pairs. A keyword-based search service is available on the web for both end-users and answering librarians to use.

In the service, several problems were identified by enquiring the librarians employed by the service:

1. Accessing accumulated knowledge. For a new submitted question, the first thing to do is often to find out if there already exists a similar or at least related answer in the knowledge base.
2. Exploiting external resources in authoring. How to integrate different data sources and services, such as library systems on the web, and then use these sources in authoring a new answer?
3. Semantic annotation. How to help the librarian in choosing the appropriate annotation concepts for a

³<http://vesa.lib.helsinki.fi>

new QA pair? This problem was considered especially crucial by the practitioner.

1.2 The Proposed Solution

The problems described above are approached by describing a prototype of a semantic annotation and authoring tool OPAS⁴ [20]. The system is intended to be used by the librarians in authoring answers in the Ask the librarian service.

In the following, we first show how semi-automatic semantic annotation can be used to help in choosing concepts for the semantic annotation of QA pairs, based on ontologies. Then the problem of finding relevant answers for a new incoming question is approached by using ideas of case-based reasoning (CBR) [1]. It is also shown present how a common upper ontology can be used to integrate different data sources to help in authoring answers. We then present the results of the early evaluations conducted with the prototype. In conclusion, contributions of the work are summarized, related work discussed, and directions of further research outlined.

2. SEMI-AUTOMATIC SEMANTIC ANNOTATION

When interviewing the librarians, two problems related to the indexing the QA pairs were brought up: 1) Choosing the appropriate indexing terms for annotating a question-answer pair is often consuming and difficult. 2) There are different conventions used in indexing by different people, which makes the content unbalanced. For example, one li-

⁴<http://www.seco.tkk.fi/applications/opas/>

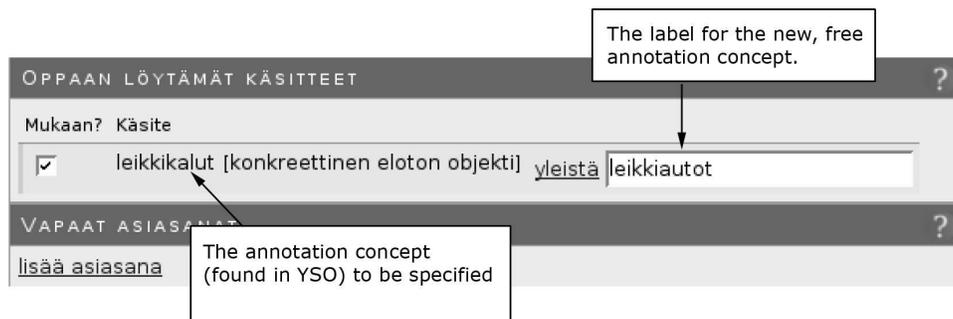


Figure 2: Specifying an annotation concept

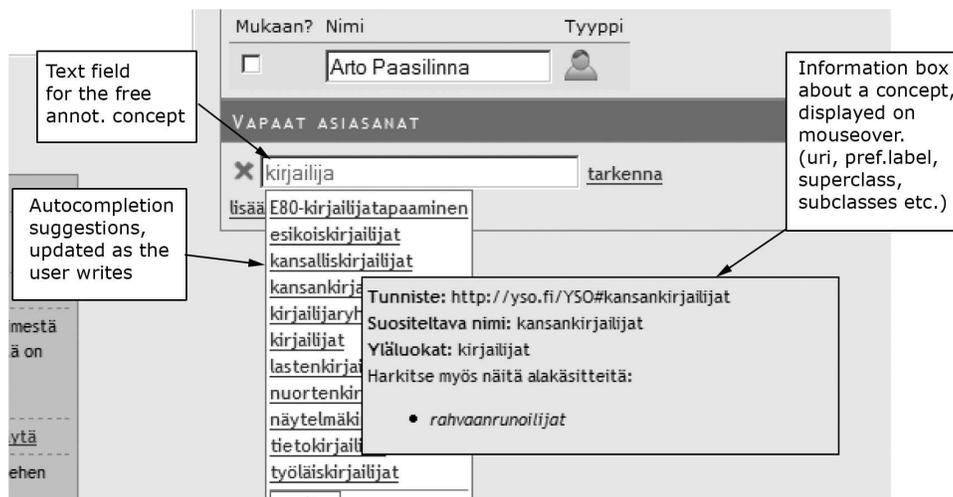


Figure 3: Annotating a question with an annotation concept that wasn't found in the question text.

brarian may use a few general terms to describe an answer, whereas another uses a large number of more detailed terms.

Our solution approach to these problems is to combine ontology-based semi-automatic annotation [13] and machine reasoning. The idea is to create a knowledge-based system that automatically provides the annotator with a *suggestion* of potential annotation concepts based on the textual material and other knowledge available, such as the QA database, earlier annotations, and common knowledge about indexing practices. The initial suggestion is then checked and edited by the human editor as she likes. This strategy not only helps the annotator in finding annotation terms (from tens of thousands of choices) but also enforces the annotators to use right terms based on the underlying annotation ontologies. Furthermore, content is likely to become more balanced because every annotator starts her job from a suggestion based on the same logic. By encoding indexers' knowledge and common indexing practices as rules, or by using automatic techniques such as collaborative filtering [7], it is possible to help especially novice indexers in their job even further.

As a first step towards such a knowledge-based semi-automatic annotation tool, we created an ontology-based information extraction tool POKA⁵ for textual data, and in-

⁵<http://www.seco.tkk.fi/applications/poka/>

tegrated it with OPAS. The following describes briefly how POKA works.

2.1 Extracting Annotation Concepts

POKA provides the QA indexer with a list of possible annotation concepts as ontological concepts (URIs), and the indexer chooses which concepts she wants to use. The selection of the concepts is based on the words and expressions found in the question and answer.

The librarians currently choose the indexing terms manually from the General Finnish Thesaurus YSA⁶. The terms in YSA are (with some exceptions) *common noun terms*, such as *dog*, *astronomy*, or *child*. In addition, the indexer may use *free indexing terms* that are not explicitly listed in the thesaurus. Free terms can be common nouns, such as names of flowers or animals, or proper nouns, such as person names (e.g., *John F. Kennedy*) or geographical places (*Finland*, *Beijing*). These categories of words, and free indexing terms not explicitly listed in the thesaurus, are treated by POKA in the following way.

⁶<http://www.vesa.lib.helsinki.fi>

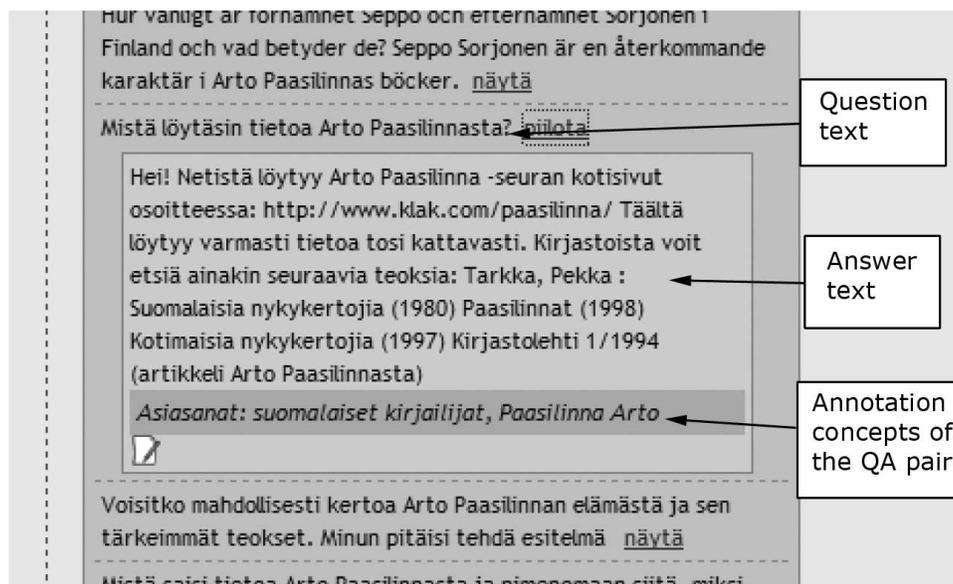


Figure 4: An example of an existing QA pair and its index terms

2.1.1 Common Nouns

In order to map common nouns in YSA with corresponding ontology concepts, YSA was transformed into the General Finnish Upper Ontology (YSO)⁷ [11]. YSO contains over 20,000 Finnish indexing concepts organized into 10 major subsumption hierarchies. Each concept is associated with one or more term labels, which allows mapping of words and terms onto YSO concepts (URIs).

First, the input question is analysed by a morphological analyser and a syntactic parser FDG⁸[18]. It produces tokenized output of the text in XML-form. FDG produces a lemmatized form of the word(s), morphological information, syntactical information, and type and reference of functional dependency to another token within a sentence, if there exist one.

For concept matching, also the labels of YSO-concepts are lemmatized. Lemmatized concepts are indexed in a prefix trie for efficient extraction. Lemmatization of text and concept names helps to achieve better recall in the extraction process; syntactical forms of words vary greatly in languages with heavy morphological affixation[17]. The architecture can be extended to support other languages with different language-dependent syntactic parsers.

2.1.2 Place Names

Place name recognition in POKA is based on the same method as common noun recognition. In this case, the place ontology of the MuseumFinland portal [10] extended in the CultureSampo-project⁹ is used instead of YSO.

2.1.3 Person Names

POKA's name recognition tool is a rule-based information extraction tool without initial gazetteers. The main idea

⁷<http://www.seco.tkk.fi/ontologies/yso/>

⁸<http://www.connexor.com>, Machine Syntax

⁹<http://www.seco.tkk.fi/projects/kulttuurisampo/>

of the recognizer is first to search for full names within the text at hand. After that, occurrences of the first and last names are mapped to full names. Simple coreference resolution within a document is implemented by mapping the individual name occurrences to corresponding unambiguous full name if there exist one. Individual first names and surnames without corresponding full names are discarded.

A strength of POKA's extraction process is that it recognizes also untypical names, unlike the tools based on gazetteers, such as tools that use the initial named entity recognition of the Gate framework[3]. Searching potential names is started from the uppercase words of the document. With morphosyntactic clues some hits can be discarded. For example, first names in Finnish rarely have certain morphological affixation like *-ssa* (similar to English preposition *in*) or *-lla* (preposition *on*). Also the FDG-parser's surface-syntactic analysis is used as clues for revealing the proper names.

Person name recognition may produce false hits. One wrong hit of full name may cause the corresponding wrong first and last name occurrences to be mapped to a full name. The good thing is that all the occurrences of the false name can be corrected by discarding the full name.

2.2 Free Annotation Concepts

POKA doesn't always suggest all annotation concepts that the librarian wants to use, even if the corresponding word can be found in the text to be annotated, and the word is considered a legal annotation concept. This happens always with free annotation concepts that by definition are not included in the ontology explicitly. Obviously, human intervention is necessary in such cases.

Our approach to the problem of extracting free annotation concepts is to provide a mechanism by which the end-users can define new free annotation entries in the ontology and share them with other annotators. A new annotation concept is defined by simply telling the system its class, label,

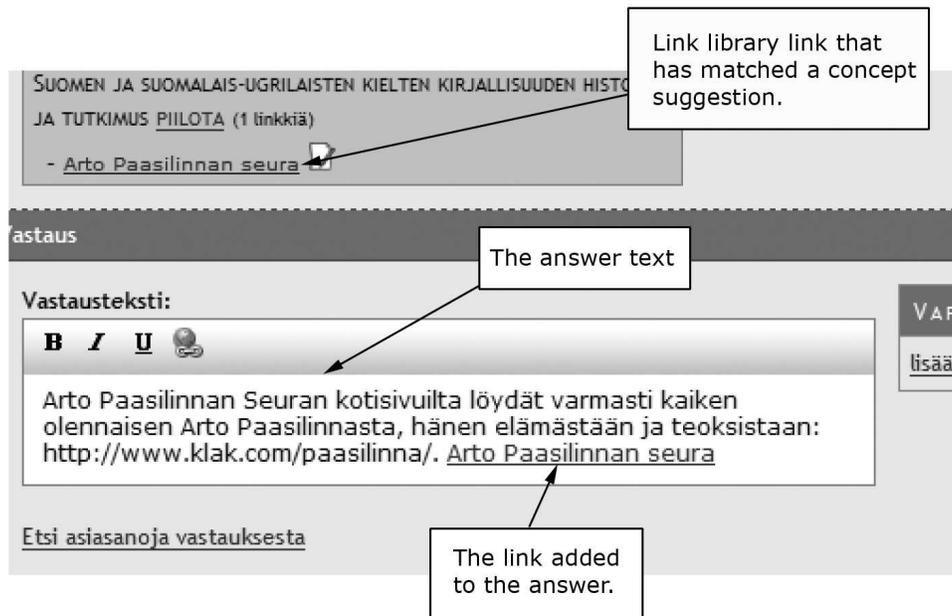


Figure 5: An example using an existing QA pair and a link from the link library in authoring an answer.

and an optional comment. For example, the term ”leikki-auto” (toy car) is not present in YSO ontology because lots of things can be used as toys, and it does not make much sense to list them all in the system. On the other hand, the concept toy car is useful from the indexing and information retrieval view points. In this case, the user can interactively create a new concept as a subclass of an existing ontological concept, here *toy* (”lelu”), label it, here ”leikkiauto” (toy car), and use it in the annotation. When searching for content later on by using the concept *toy* (”lelu”), also QA pairs annotated with *toy car* (”leikkiauto”) can be retrieved with the additional information that in this case the QA pair is about toy cars in particular. The new concept of toy car also be utilized in various ways in the user interface, e.g., as a search category in view-based semantic search [10]. Free indexing terms with the same name can be distinguished with different URIs and with an additional comment.

Unknown but relevant annotation concepts without a corresponding concept in the ontologies are frequently encountered also in name recognition because new names (e.g., names of pop stars) are constantly introduced as time goes by. The same approach used with free annotation concepts can be employed here, too.

In some cases where a word does not have an exact match with an ontological concept, POKA is able to suggest related annotation concepts based on the ontology. Such reasoning can be based, for example, on the morphological structure of a compound word or the functional dependencies produced by the FDG-parser.

2.3 Ranking Annotation Concepts

Previous sections analyzed situations where a semantic annotator produces too few relevant annotation concepts. A reverse problem with automatic semantic annotation is that often too many irrelevant concepts are suggested. Espe-

cially, if the input text is long, a considerable number of possible annotation concepts are usually found. In such cases it is useful to rank the concepts according to their likely relevance, and provide the end-user with a simple mechanism for evaluating and deleting the irrelevant annotations.

OPAS uses the idea [16] of searching for *semantic cluster(s)* from the term set for determining the relevance of indexing concepts: terms in semantic clusters are ranked more relevant than semantically isolated terms. For example terms *doctor*, *sickness* and *medication* form a semantic cluster. For common noun terms we use the concept relations defined in the YSO ontology to identify these clusters.

In [8], an ontological extension of the classic tf-idf (term frequency – inverse document frequency) method is developed, which enables us to identify synonyms and to utilize the concept hierarchies of the ontology. We apply this work so that more weight is given to concepts that appear frequently in the text but haven’t been used often as annotation concepts in previous questions. In addition, OPAS can suggest annotation concepts that are usually used together. For example, if a question has the concept *aviation* extracted, and there are lots of questions annotated with both *aviation* and *airplane*, the concept *airplane* can be suggested for annotation concept, even though it is not explicitly present in the question text.

Our preliminary experiments with annotation concept weighting seem to suggest that relatively more weight should be given to terms that have a high term frequency, and the effect of inverse document frequency should be relatively smaller. The reasoning behind this is that if, say, the concept *poetry* appears in a question many times, it seems that the concept is relevant to the question even though it has been used frequently as an annotation concept in previous questions. So, in OPAS the main weight is determined by the

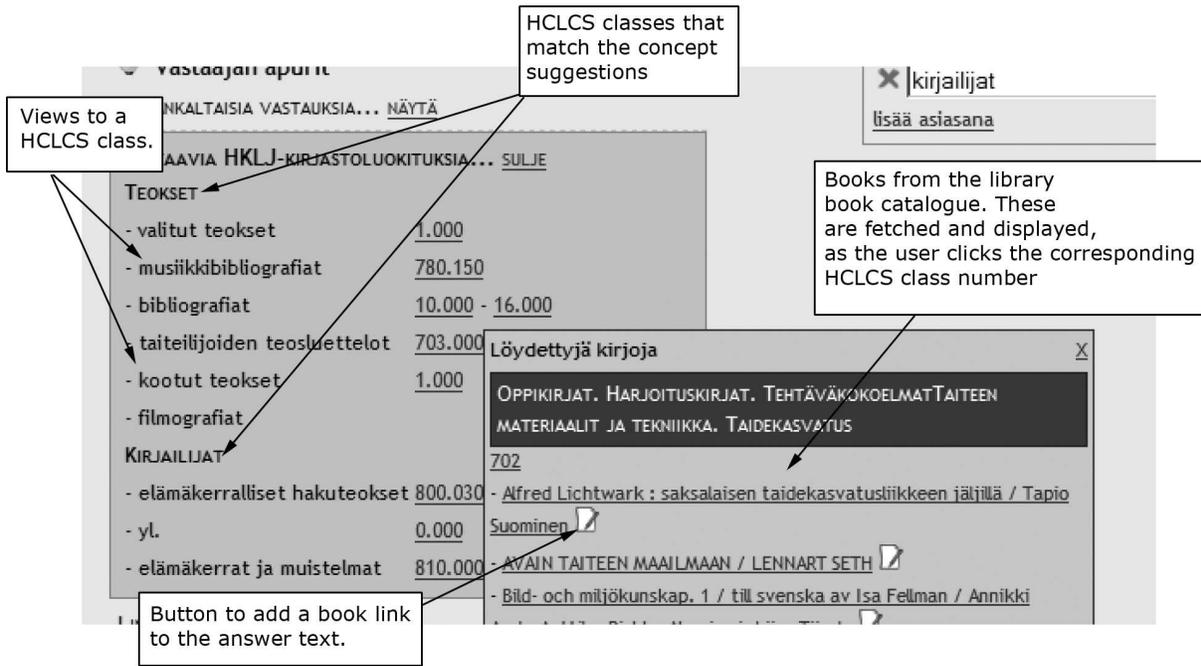


Figure 6: A book search based on the index terms and their views that were found in Helsinki City Library Classification System.

term frequency, whereas inverse document frequency and semantic clusters have a smaller impact on the weight.

2.4 An Example

Figure 1 depicts the first screen that the librarian sees when she has decided to answer a question. The end-user has submitted a question about Arto Paasilinna’s (a Finnish author) life and his books (on the left, in the box “Kysymysteksti” (Question Text). On the right, in the box “Oppaan löytämät käsitteet” (Indexing Concepts Found) there are two common noun concepts “teokset” (writings) and “esitelmät” (plays). POKA has also identified the person name “Arto Paasilinna”. Below the question text, there is the authoring component (“Vastaaajan apurit”) (Authoring Tools) to be discussed in detail in section 4.

Figure 2 depicts the case where the free annotation concept “leikkiauto” (toy car) is encountered. In this case, POKA analyses the compound term into pieces and suggests the concept “leikkikalu” *toy* because it is found in the YSO ontology as a potentially related concept based on the first part of the compound. The librarian can then define the narrower concept *toy car* with the label “leikkiautot” *toy cars* by clicking on the link in the middle.

Figure 3 depicts the case where POKA is unable to make any suggestions, and the librarian wants to add the new annotation concept *writer* (“kirjailijat”) in the ontology. As she is typing in the word, OPAS uses *semantic autocompletion* [9] to suggest matching annotation concepts in YSO. The floating box on the bottom right displays information about a concept, its preferred and alternative labels, related concepts, subconcepts, and superconcepts. This information is displayed when the librarian points the concepts with the mouse. The purpose of the autocompletion component is to

1) ensure that the indexer uses a concept found in the ontology and 2) suggest semantically related indexing concepts that the librarian perhaps didn’t consider.

3. UTILIZING CASE-BASED REASONING TO FIND SIMILAR QUESTIONS

Case-based reasoning (CBR) [1] is a problem solving paradigm in artificial intelligence where new problems are solved based on previously experienced similar problems, cases. The CBR cycle consists of four phases: 1) Retrieve the most similar case or cases, 2) Reuse the retrieved case(s) to solve the problem, 3) Revise the proposed solution and 4) Retain the solution as a new case in the case base.

Since similar QA pairs recur in QA services, we decided to investigate the usefulness of CBR in QA indexing and information retrieval. CBR has been used in help desk applications previously. For example, Goker and Roth-Berghofer [6] argue that CBR can successfully be used in a help desk service and by using CBR in help desk service an organization can strengthen the common knowledge and reduce the time needed to answer a help request. Kai *et al.* [12] have found out that users of a CBR-based help desk system tend to remember solutions longer since they feel that they’ve solved the problem themselves, even though the solution was retrieved and possibly adapted from the case base.

What OPAS brings in to traditional CBR approach is that it integrates semantic annotation to the steps of the CBR cycle. For the first step, OPAS contains a CBR component that automatically searches for similar questions based on the concepts that POKA has extracted from the question text. The weighted annotation concept list discussed in section 2.3 is used as the basis for the search with the following

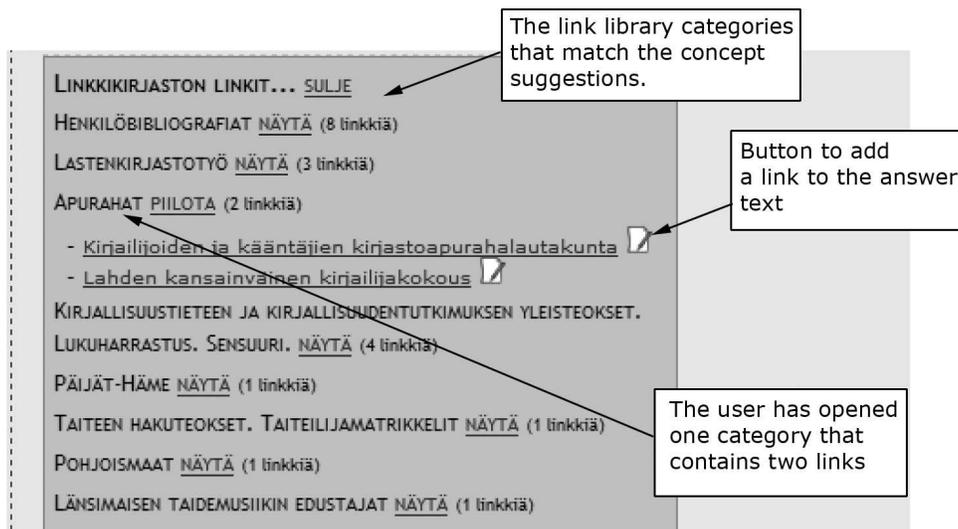


Figure 7: An example of link library links that are found based on Poka’s annotation concept suggestions.

modifications: 1) The concepts that the indexer has selected are given a substantially higher weight since their relevance has been confirmed by the indexer. 2) The extracted places, names and specified concepts are given a higher weight due to their specificity.

4. INTEGRATING DIFFERENT DATA SOURCES IN ANSWER AUTHORIZING

When discussed the current service with the librarians, a few things were remarkable about the information sources that the librarians use when answering a question. Firstly, nearly all of the librarians said that they use the reference library with real books to find useful resources. Secondly, even though nearly all the librarians agreed that the questions tend to repeat themselves, not many of them systematically use the question archive to find old similar questions. Besides that, it is remarkable that when the librarians aren’t able to answer a question in three working days, they nevertheless send an answer to the client. This answer usually contains pointers to different information resources, for example web sites, that might contain the answer to the question.

Based on the remarks described above, we decided to add an authoring component to OPAS. The purpose of this component is to help the librarian to compose the answer using different information sources. The authoring component can be seen in the figure 1 (“Vastaaajan apurit”). What is common to these authoring components is that each of them uses the annotation concept suggestions produced by POKA to query external resources. The common upper ontology YSO acts as a “glue” between different information resources. In the following the subcomponents of the authoring component are explained.

4.1 Authoring Using Existing QA Pairs

Existing QA pairs can be used as a basis for composing the new answer. In figure 4 the librarian has opened one of the questions in order to see whether it provides useful information for answering the question. The answer can

be used as basis for the new answer by clicking the link (the white paper sheet with a pen). Figure 5 depicts how the librarian has used an existing answer as a basis for the answer.

As the retrieval of similar QA pairs can be seen as the first step in the CBR cycle, using them in authoring component can be seen as a part of the second step: Reuse the retrieved case(s) to solve the problem.

4.2 Authoring Using a Library Classification System

An ontology for a library classification system was created for OPAS, and then the Helsinki City Library Classification System (HCLCS)¹⁰ was converted into this ontologized form. The basis for the classification ontology is Simple Knowledge Organisation System (SKOS)¹¹ and the conversion was made following the guidelines given in [19]. In addition to class hierarchies the HCLCS contains index terms, and each of these terms has got a relation to a library class. For example the term *Treatment of alcoholics* has got a relation to the library class 371.71 *Alcohol policy*.

Index terms in the HCLCS contain also *views*, as can be seen in the figure 6. For example the term *pieces of art* (“Teokset”) embodies different viewpoints such as bibliographies and art collections. Each of these viewpoint is related to a library class. These relations between index terms and library classes are used to search for books that could be relevant to the answer. These books are searched based on the library class, as depicted in the figure 6. The librarian can use the results of the book search 1) for searching an answer for the question and 2) by enhancing the answer with links to interesting books.

¹⁰<http://hklj.kirjastot.fi/>

¹¹<http://www.w3.org/2004/02/skos/>

4.3 Authoring Using a Link Library

The editors of the Libraries.fi maintain a collection of links to interesting web sites. This link library is categorized using the same classification system that is used in the HCLCS. An ontology was created and then the data was converted into an ontologized form in a similar manner than described in the previous section. The figure 7 depicts a screenshot of this link library. The links are categorized by the HCLCS ("Henkilöbibliografiat", "Lastenkirjastotyö", etc.), and the librarian has opened one category to see whether there are interesting links. These links can be added to the answer text as can be seen in the figure 5.

5. EVALUATION

To evaluate the current version of the prototype and to find out librarians' initial attitudes towards the new version of the system, a few user tests were run with real users of the service. The tests were conducted so that the librarian was first introduced with the prototype and its features. Then, she was asked to answer a question using the prototype. The questions were real questions of the existing version of the service. Finally, the librarian was interviewed about the answering process.

The results of the evaluation were encouraging. All librarians found the features of the prototype useful and said that they would take the prototype into use, if it were possible. The most impressive and useful feature for the librarians seemed to be the authoring features of the prototype, especially the component that searches for existing similar questions automatically. All librarians were also pleased with the authoring features that enable to add resources (old answers, links, book references) to the answer by clicking a button.

The annotation concept suggestions were welcomed, but not as eagerly as the authoring components. Some of the librarians said that the concept suggestions were entirely irrelevant. The semantic autocompletion component that searches for concepts in YSO was considered useful. Based on the tests, nothing can yet be said about how good the ranking of the concept suggestions was.

When a librarian hasn't selected and confirmed any of the suggested annotation concepts, the authoring component fetches resources based on all of the concepts in the list. However, when the librarian had selected one or more suggestions to be used, it was confusing that still the authoring component fetched resources related to unselected concepts. Although these resources were given a smaller weight and thus they were lower in the result list, it seems that when the librarian has selected one or more concept suggestion or inserted a free annotation concept, the other, unselected concepts should be ignored totally in the result lists of the authoring components.

6. DISCUSSION

First experiments with combining semi-automatic semantic annotation and authoring with the ideas of case-based reasoning seem promising. Even though the evaluation of the prototype wasn't extensive, it can be concluded that OPAS would be a valuable tool to librarians if taken into use. However, systematic empirical evaluations of the application are yet to be done.

Currently the book search component isn't using semantically annotated content, but instead fetches web pages and then parses the results from the HTML content. In consequence, one of the major benefits of the semantic web, disambiguation of terms (for example, "Nokia" as an enterprise and as a city) is not possible. OPAS would benefit more from a system with semantically annotated content.

The utilization of case-based reasoning in OPAS can be seen somewhat shallow. The ideas of CBR and the steps of the CBR-process fit well with OPAS, but the details of each step could be examined more carefully. For example a framework for similarity assessment presented in [4] could be utilized for the retrieval of similar QA pairs.

A result of the the evaluation was that the annotation concept suggestions weren't optimal. Sophisticated methods for ranking the suggestions and finding out which concepts really are relevant for a user query should be investigated and developed further.

6.1 Related Work

To search for similar questions some other approaches would have been possible as well. For example Kohonen *et al.* [15] demonstrate how Self Organizing Maps [14] (SOM) can be used to organize a vast collection of patent abstracts and then use the SOM to search if similar patents exist for a new patent application. A standard text search by using for example the Java search engine Lucene¹² would also probably yield sufficient results when searching for similar questions. However these methods don't take into account the semantics of the text, and we want to be able to utilize the semantic relations defined in the common upper ontology YSO.

As for semantic authoring, David Aumuller [2] presents a technique to semantically author Wiki pages. The technique is not just for adding annotations to the pages but also for editing the text. His ideas could be applied in authoring the answers.

6.2 Future Work

Currently OPAS is focused on the indexers' role in QA applications but OPAS will include the end-users' side, too. Here we work on questions such as: how to classify the QA pairs for semantic view-based search, how to do semantic recommending in order to show other interesting answers, and how to integrate the system with semantic content and services at other locations on the web related to the end-user's information needs. The CBR component that searches for similar questions can be used with little modifications at the end-users' side, too.

Acknowledgments

Our work is a part of the National Semantic Web Ontology Project in Finland (FinnONTO)¹³, funded by the National Funding Agency for Technology and Innovation (Tekes) and a consortium of 36 public organizations and companies.

¹²<http://lucene.apache.org>

¹³<http://www.seco.tkk.fi/projects/finnonto/>

7. REFERENCES

- [1] A. Aamodt and E. Plaza. Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI Commun.*, 7(1):39–59, 1994.
- [2] D. Aumueller. Semantic authoring and retrieval within a wiki, Aug 2005. Demo paper, 2nd European Semantic Web Conference 2005 (ESWC2005).
- [3] K. Bontcheva, V. Tablan, D. Maynard, and H. Cunningham. Evolving gate to meet new challenges in language engineering. *Natural Language Engineering*, 10(3/4):349–373, 2004.
- [4] G. Falkman. *Issues in Structured Knowledge Representation A Definitional Approach with Application to Case-Based Reasoning and Medical Informatics*. PhD thesis, Chalmers University of Technology, Göteborg University, 2003.
- [5] S. Foo, S. C. Hui, P. C. Leong, and S. Liu. An integrated help support for customer services over the world wide web: a case study. *Comput. Ind.*, 41(2):129–145, 2000.
- [6] M. Goker and T. Roth-Berghofer. The development and utilization of the case-based help-desk support system homer. *Engineering Applications of Artificial Intelligence*, 12(6):665–680, Dec 1999.
- [7] J. H. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Computer Supported Cooperative Work*, pages 241–250. ACM, 2000.
- [8] M. Holi, E. Hyvönen, and P. Lindgren. Integrating tf-idf weighting with fuzzy view-based search. In *Proceedings of the ECAI Workshop on Text-Based Information Retrieval (TIR-06)*, Aug 2006. To be published.
- [9] E. Hyvönen and E. Mäkelä. Semantic autocompletion. In *Proceedings of the 1st Asian Semantic Web Conference (ASWC-2006), Beijing, Sep 4-9, 2006*. forth-coming.
- [10] E. Hyvönen, E. Mäkelä, M. Salminen, A. Valo, K. Viljanen, S. Saarela, M. Junnila, and S. Kettula. MuseumFinland – Finnish museums on the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(2-3):224–241, Oct 2005.
- [11] E. Hyvönen, A. Valo, V. Komulainen, K. Seppälä, T. Kauppinen, T. Ruotsalo, M. Salminen, and A. Ylisalmi. Finnish national ontologies for the semantic web - towards a content and service infrastructure. In *Proceedings of International Conference on Dublin Core and Metadata Applications (DC 2005)*, Nov 2005.
- [12] H. Kai, P. Raman, W. Carlisle, and J. Cross. A self-improving helpdesk service system using case-based reasoning techniques. *Computers in Industry*, 30(2):113–125, September 1996.
- [13] N. Kiyavitskaya, N. Zeni, J. R. Cordy, L. Mich, and J. Mylopoulos. Semi-automatic semantic annotations for web documents. In *SWAP 2005, Semantic Web Applications and Perspectives, Proceedings of the 2nd Italian Semantic Web Workshop University of Trento, Trento, Italy, 14-15-16 December 2005*, 2005.
- [14] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, Sep 1990.
- [15] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, and A. Saarela. Self organization of a massive document collection. *Neural Networks, IEEE Transactions*, 11(3):574–585, May 2000.
- [16] G. S. Luit Gazendam, Veronique Malaisé and H. Brugman. Deriving semantic annotations of an audiovisual program from contextual texts. In *Semantic Web Annotation of Multimedia (SWAMM'06) workshop*, 2006. <http://www.cs.vu.nl/~guus/papers/Gazendam06a.pdf>.
- [17] L. Löfberg, D. Archer, S. Piao, P. Rayson, T. McEnery, K. Varantola, and J.-P. Juntunen. Porting an english semantic tagger to the finnish language. In *Proceedings of the Corpus Linguistics 2003 conference*, pages 457–464. UCREL, Lancaster University, 2003.
- [18] P. Tapanainen and T. Järvinen. A non-projective dependency parser. *Proceedings of the 5th Conference on Applied Natural Language Processing*, pages 64–71, 1997.
- [19] M. van Assem, M. R. Menken, G. Schreiber, J. Wielemaker, and B. Wielinga. A method for converting thesauri to rdf/owl. In *Third International Semantic Web Conference ISWC 2004*, volume 3298, 2004.
- [20] A. Vehviläinen, O. Alm, and E. Hyvönen. Combining case-based reasoning and semantic indexing in a question-answer service, June 20 2006. Poster paper, 1st Asian Semantic Web Conference (ASWC2006).