Facilitating the exchange of explicit knowledge through ontology mappings

Martin S. Lacher
Technische Universität München
Arcisstr. 21
80290 München
Germany
lacher@in.tum.de

Georg Groh
Universität Kaiserslautern
Gottlieb-Daimler-Strasse
67663 Kaiserslautern
Germany
ggroh@rhrk.uni-kl.de

Abstract

In this paper, we give an overview of a system (*CAIMAN*) that can facilitate the exchange of relevant documents between geographically dispersed people in Communities of Interest. The nature of Communities of Interest prevents the creation and enforcement of a common organizational scheme for documents, to which all community members adhere. Each community member organizes her documents according to her own categorization scheme (ontology). CAIMAN exploits this personal ontology, which is essentially the perspective of a user on a domain, for information retrieval. Related documents are retrieved on a concept granularity level from a central community document repository. To find the related concepts in the queried ontology, CAIMAN performs an ontology mapping. The ontology mapping in CAIMAN is based on a novel approach, which considers the concepts in an ontology implicitly represented by the documents assigned to each concept. Using machine learning techniques for text classification, a concept in a personal ontology is mapped to a concept in a community ontology. The CAIMAN system uses this mapping to provide document publishing and retrieval services both for the community and the user. First results of the prototype system showed that this approach can be a valid alternative to existing techniques for information retrieval.

Introduction

Loosely coupled groups of people with similar interests, so-called *Communities*, have been shown to be a very valuable source of knowledge. The broad deployment of platforms for community interaction support has further increased the leverage of synergies of a knowledge exchange of experts from all over the world. Members of Communities of Interest mostly don't work on a common task and thus have no common goals related to a defined task. Moreover, the motivation for participation of community members is exclusively intrinsic. Community members can usually not be forced to adopt common standards for information exchange. As a consequence, communities cannot be supported by existing tools for team support for cooperative work. New tools have to be designed with special attention to the nature of communities.

In the context of increasingly large collections of documents on the Internet, ontologies for information organization have become an active area of research (Noy & Hafner 1997), (Clark 1999), (Chandrasekaran, Josephson, & Benjamins 1999). There is little agreement on a common definition of an ontology, most works cite (Gruber 1993) as the common denominator of all ontology definitions. Ontologies are used in user profiles, for information retrieval (Pretschner & Gauch 1999), for databases (J.Bayardo *et al.* 1997) and agent communication (Huhns & Singh 1997). As proposed in (Welty 2000), talking about ontologies only makes sense regarding a certain application context. In this paper we look at ontologies for categorizing documents by their contents. The *YAHOO!* directory is one example for such a simple ontology (Labrou & Finin 1999), in which the ontology concepts serve as document content classes. Using ontologies for document repositories provides for efficient retrieval of stored documents. Moreover, the structure of the ontology provides a context for the stored documents (Chakrabarti *et al.* 1998) for user browsing as well as automated retrieval.

Most community websites already have a collection of documents organized according to an implicit or explicit ontology. The construction of such ontology-based community websites has been discussed in (Staab *et al.* 2000). These systems work fairly well, except for a problem that has been discovered in (Bonifacio, Bouquet, & Manzardo 2000). In order to be used effectively for storage and retrieval of documents, categorization schemes have to be understood and accepted by all community members in the same way. All community members should ideally have the same perspective on the knowledge domain that is represented by the community document repository. This is hardly the case in real life communities. Usually, the community members would like to keep their own perspective on a community repository. Moreover, users are mostly interested in more than one community, which inherently renders a global perspective impossible. As shown in (Grudin 1994), a centralized mandatory ontology would not be trusted by community members and thus lead to a participation decrease. We thus propose to have distributed ontologies, which serve as organization schemes for the user's documents as well as

the user's perspective on the world. The commmunity also has a proper ontology. For large document collections, it is desirable to have support for automated exchange of documents and still let the user keep his perspective on the world. In this paper we will show how this is realized with ontology mappings in the CAIMAN system. Our ontology mapping technique also allows to improve the retrieval of related douments by considering classification information for each document as well.

The rest of the paper is organized as follows: First, we briefly describe how the CAIMAN system fits into a larger architecture for support of Communities of Interest. We describe the Knowledge Management approach that this architecture has been designed with and present the services that the CAIMAN system offers. Thereafter, we introduce the algorithm for the ontology mapping, which is based on a similarity measure for single ontology concept nodes. The technical realization of the mapping of the single concept nodes based on text classification algorithms is presented in the next section. We also present some first evaluation results and contrast them with related work. The paper ends with a conclusion and an outlook on future work,

The CAIMAN system

Each member in a Community of Interest organizes her documents according to her own categorization scheme (ontology). CAIMAN exploits this personal ontology, which is essentially the perspective of a user on a domain, to retrieve and publish relevant documents. For each concept node in the personal ontology, a corresponding node in the community ontology is identified. Documents that are only assigned to a node in the community ontology can then be proposed to the user and vice versa. Our novel approach to mapping concept nodes in two ontologies onto each other is based on an implicit extensional representation of the ontologies. Each concept node in an ontology is implicitly represented by the documents assigned to each concept. Using machine learning techniques for text classification, a measure for the probability that two concepts are corresponding is calculated for each concept node in the personal ontology. In the current prototype implementation, the CAIMAN system uses a user's bookmark folder hierarchy as the user side ontology as well as the directory structure of RESEARCHINDEX[1] as the community ontology. For the exchange of explicit knowledge in the form of documents, CAIMAN maps the personal ontology (bookmark folder hierarchy) to the community ontology (directory structure of RESEARCHINDEX).

The CAIMAN system is part of a larger architecture for Knowledge Management support in Communities of Interest. We will first briefly describe how this architecture has been designed and how CAIMAN fits into the framework.

Design with a Knowledge Management perspective

For designing collaborative IT systems for Communities of Interest, a lot of issues besides the technical problems have to be considered. Since the main purpose of Communities of Interest is to exchange knowledge, a Knowledge Management perspective on the problem can help to structure the problem space. For this purpose, we used a commonly used Knowledge Management process model (Probst & Büchel 1998), which is similar to other process models presented in (Liebowitz 1999). The process model helped us to identify the different problem fields that a community support architecture will have to offer support for. This resulted in a general framework for Knowledge Management support in Communities of Interest (Lacher & Koch 2000) (Koch & Lacher 2000). In (Probst & Büchel 1998), six distinct process classes have been identified for Knowledge Management: knowledge awareness, knowledge acquisition, knowledge creation and development, knowledge sharing and distribution, knowledge application and knowledge conservation. We considered for each of these process classes how respective processes in communities can be supported by information technology. We used detailed characteristics of communities as constraints for the design. The CAIMAN system is the result of a focus on the knowledge creation and knowledge distribution processes in communities.

To put the results of the requirements for the CAIMAN system in a nutshell, knowledge creation is best supported by delivering **personalized** information to the user. The information can be delivered through push or pull, with a proactive push being most effective. Knowledge distribution is greatly increased if access and retrieval of available relevant information is easy and straightforward for the user. This also means that information has to be indexed or categorized in a way that the user can understand and accepts. Both goals could be achieved by enforcing a standard community ontology, by which all knowledge in the community is organized. However, due to the loose coupling of members in a Community of Interest, this will not be possible and has been shown to be an unsuccessful practice in a case study presented in (Bonifacio, Bouquet, & Manzardo 2000). Instead, we propose to mediate between the user's ontology and the community ontology and build the respective services for knowledge exchange on top of the mediation infrastructure.

Services in CAIMAN

We assume that each community member organizes her collection of explicit knowledge (documents) according to her personal categorization scheme. An example for this is a bookmark collection. We also generally assume that the repositories both on the user as well as on the community side may store the actual documents as well as links to the physical locations of the documents. CAIMAN offers the two services *document publication* as well as *retrieval of related documents*. Both services can be configured as either push or pull services.

- The **document publication** service publishes documents that a user has newly assigned to one of the concept classes to the corresponding community concept class. Along with the document, information about how the user classified this document in his repository is sent to the community repository. The community repository can

[1] http://www.researchindex.com/

now classify the document (if not already existent) with respect to the community ontology. This classification can be based on either the document information itself, information about the concept node in the user ontology or a combination of both . This way the community as a whole always profits from the identification of new relevant documents by any community member without any effort by the user. This effortless publication of new information in crucial in Communities of Interest.

- The **retrieval of related documents** service delivers newly added documents from the community repository to the user. Documents can be retrieved at a class or document granularity level. On a document granularity level, CAIMAN can improve traditional retrieval of related documents by exploiting user side classification information. On a class granularity level, the user can choose a concept node in his ontology for which he would like to find related documents. Then, documents, which the user has not already assigned to the node in his personal ontology, can be delivered to the user. This can be a one-time query or a constant push of new documents, such that the user's document repository and the community repository are always balanced. A constant push service will propose documents to the user, but not automatically assign documents to a concept node.

The implementation of these services has been in a prototype stage at the publication time of this paper.

Ontology mapping in CAIMAN

In order to provide the above mentioned services, the CAIMAN system has to map one ontology onto another. This means that for each concept node in ontology graph A, a corresponding concept node in ontology graph B has to be found. We calculate a probability measure that for a node a_i in the personal ontology, node b_j in the community ontology is the corresponding node. The node b_j with the maximum probability measure wins. Currently we perform a mapping that does not consider the graph structure of the personal ontology. Since we expect the mapping to improve when the interconnection of nodes from the personal ontology is considered, we plan to include this in future work. However, we take some of the graph structure of the community ontology into account. We calculate two probability measures for the two most probable nodes b_j and b_k that could match a_i and if their difference is above a certain threshold, we say that b_j and a_i are corresponding nodes. However, if their difference is below the threshold, we calculate the probability difference of the parent nodes of b_j and b_k and so on until the difference is above the threshold. If we get to a common parent node or there is no parent node, the user has to decide, which of the pre-selected nodes are to be considered corresponding nodes.

The probability measure $p(a_i, b_j)$ we base our mapping on will be explained in the next section. To find corresponding nodes, we apply the following simple algorithm:

1. *for all concept nodes $a_i \in A$ (breadth first)*
   (a) *for all nodes $b_j \in B$ (breadth first) calculate $p(a_i, b_j)$*
   (b) *Find b_j and b_k such that $p(a_i, b_j)$ is maximized and then $p(a_i, b_j) - p(a_i, b_k)$ is minimized.*
   (c) *if $d := p(a_i, b_j) - p(a_i, b_k) \leq t$ mark a_i and b_j as corresponding*
   (d) *else repeat for the parent nodes of the current b nodes until a decision has been made*
      i. *calculate d for the current nodes*
      ii. *if $d > t$, pick the (grand*)child node of b_j*
      iii. *if there is no parent node for one of the b nodes or they have a common parent, let the user decide which node to pick*

What we need now to perform the actual mapping is an estimate of the probabilities $p(a_i, b_j)$. This calculation is described in the next section.

Ontology concept node mapping

We perform the calculation of the probability estimate for $p(a_i, b_j)$ by employing machine learning techniques for text categorization. We calculate a representative feature vector for each concept node in an ontology. We then measure similarity of two of those class vectors by a simple cosine measure. The representative feature vector for one concept node is calculated as a modified Rocchio centroid vector. Thus, one can say that the representative vector for a concept node represents an average of all documents assigned to that concept node. We use the word *class* as a synonym to *concept node* here, in accordance to most of the IR literature.

In (Goller *et al.* 2000) an overview of the state of the art of techniques for text categorization is given. More specifically, the comparison considers feature vector creation, stemming, stop-word removal, feature weighting, feature selection and finally learning and classification. We based our design decisions for the concept node mapping on the results presented in (Goller *et al.* 2000).

Before classification can be performed, the feature vectors have to be exctracted from the documents and weighted. In CAIMAN, we use the *Bow*[2] toolkit to perform a stop-word removal as well as a subsequent feature stemming. We generate a word-count feature vector and weight the features with a TF/IDF weighting scheme (Term Frequency/Inverse Document Frequency). A TF/IDF measure essentially aims to assign more weight to more important terms (see (Goller *et al.* 2000) for more details).

The comparison of classification techniques presented in (Goller *et al.* 2000) shows that most classification techniques perform sufficiently well for a lot of applications. Under certain circumstances more elaborate classifiers such as Support Vector Machines (Joachims 1998) perform (only slightly) better in terms of accuracy than the simpler techniques. For our purposes, we chose the Rocchio classifier for three reasons: 1) it is simple and cheap in runtime, 2) it performs well and 3) a class in an ontology is represented the same way as a document: by a feature vector. This makes a comparison of classes easy and fast and gives us the possibility to cmbine class and document vectors for retrieval performance improvement.

[2] http://www.cs.cmu.edu/m̃ccallum/bow/

Let c^k represent the centroid vector for class k, which holds the set of documents d_j. The individual components c_i^k of this vector for each feature i are then calculated as:

$$c_i^k = \frac{1}{||\{d_j \in k\}||} \sum_{\{d_j \in k\}} w_{ij} - \frac{1}{||\{d_j \notin k\}||} \sum_{\{d_j \notin k\}} w_{ij}. \quad (1)$$

where w_{ij} is the weight of the feature i in document d_j.

The term $\sum_{\{d_j \notin k\}} w_{ij}$ is introduced to improve discrimination between class-vectors and is dropped in our implementation for performance reasons. Experiments showed that the influence of this alteration is minimal. The similarity of two concept nodes in two ontologies is calculated by the cosine measure:

$$\begin{aligned} p(a_i, b_j) &= \cos(\sphericalangle(\mathbf{a}_i, \mathbf{b}_j)) & (2)\\ &= \frac{1}{||\mathbf{a}_i|| ||\mathbf{b}_j||} \mathbf{a}_i \, \mathbf{b}_j & (3)\\ &= \frac{1}{||\mathbf{a}_i|| ||\mathbf{b}_j||} \sum_k a_{i_k} \cdot b_{j_k}. & (4) \end{aligned}$$

where $\mathbf{a}_i$ and $\mathbf{b}_j$ are the vectors that represent the respective concept nodes. Our current implementation considers a flat list of classes for classification. We plan to implement hierarchical feature selection and classification as presented in (Chakrabarti *et al.* 1998) as future work.

The identical treatment of class and document representations allowed us to implement a simple but powerful mechanism to balance the retrieval and publication services in CAIMAN between traditional related-document scope and related-class scope. We calculate the concept node vector as a linear combination $\alpha \cdot c + (1-\alpha) \cdot d_j$ of the Rocchio vector and a document vector d_j. In case a document that the user has assigned to a certain concept node is not semantically linked to the rest of the documents in this class by its word statistics, the combination of feature vectors can greatly improve classification performance. The vector representation of concepts also allows us to calculate an incremental update of the concept vectors. Whenever new documents are added to a class, their feature vector just has to be added.

Evaluation

The CAIMAN system is currently in a prototype stage. We have conducted first tests of the quality of the mapping of concepts, however, the CAIMAN services have not been thoroughly evaluated yet.

We have conducted one test set with a collection of bookmarked webpages that have been mapped to the *Open Directory Project*[3]. For this test we used a stemming algorithm and a naive Bayes classifier to calculate the node mapping. The results were unsatisfactory. This was due to the fact that the webpages the bookmarks pointed to were mostly only title pages with links to a number of additional pages that held the actual document. We are planning to implement special web page classification algorithms such as the one presented in (Attardi, Gull'i, & Sebastiani 1999) as future work.

[3]http://www.dmoz.org/

For now, our testbed consists of user bookmarks which links to documents in *PDF* of *PS* format as the user repository. For the community repository, we have two testbeds: the RESEARCHINDEX[4] and a self-implemented community repository, managed by the *Community Items Tool* (Koch & Lacher 2000). We have conducted a number of experiments that showed that CAIMAN was able to identify corresponding nodes in two ontologies that had been assembled by the same person. We are currently evaluating CAIMAN for the case of more essentially different ontologies as well as for a larger number of users.

Related Work

There is very little research about the application of ontology mappings for community support. In (Takeda, Matsuzuka, & Taniguchi 2000), a system has been presented that performs collaborative document recommendations by finding related folders in users' bookmarks. The links from related folders of a peer user are recommended to each user. The quality of the recommended links seemed to be insufficient while the quality of the mapping of folders has been judged satisfactory. This may be due to the fact that the mapping of the folders is based on a naive keyword comparison. Moreover, we consider the direct exchange of documents between users problematic, as perspectives on a knowledge domain may vary too much to find suitable mappings. This is not as much the case for a mapping between the community and user perspective. The relatively stable community perspective makes it more feasible to find a suitable mapping over time. In (Takeda, Matsuzuka, & Taniguchi 2000), the structure of the bookmark hierarchies is nto considered at all.

In (Mitra, Wiederhold, & Kersten 2000), a graph-based model for expressing ontology interdependencies is presented. An algebra for set operations with ontologies is constructed. The interdedependencies of two source ontologies are expressed with a third ontology, the articulation ontology. The articulation ontology consists of concepts which subsume concepts from both source ontologies. The two source ontologies are connected to the articulation ontology by links which are termed *semantic implication*. In contrast to our work, an additional ontology is created. The mapping is performed manually by the user on the explicit representation of the ontology. In the CAIMAN system, the mapping is performed semi-automatically and based on the extensional representation of an ontology for document repositories: the document collection.

Other interoperability frameworks which employ ontology mappings like (Melnik, Garcia-Molina, & Paepcke 2000) and (J.Bayardo *et al.* 1997) focus on the mediation for sources of structured information. Here, we focus on unstructured information.

Future Work

The most important issue is the evaluation of the mapping as well as the CAIMAN services for a larger number of users. The services, for example their granularity for retrieval of related documents offered by CAIMAN, can still be improved.

[4]http://www.researchindex.com/

The process of ontology mapping should, even though the results are satisfactory, allow for more interaction with the user.

For the calculation of the measure for node correspondance, we plan to implement specific classification techniques for web pages as well as a real hierarchical classifier, to take advantage of the structure information that is stored in the ontology structure on both sides.

Conclusion

We presented a novel technique for mapping ontologies, which are used for categorization of documents. This is to our knowledge the first work that focusses on an implicit representation of ontologies for an ontology mapping. Especially for document repositories, it makes sense to represent content categories by the contained documents instead of some one-word label. We use this implicit representation to calculate a similarity measure between nodes of the ontologies that have to be mapped. First results showed that the quality of the mapping is good. The ontology mapping service is part of the CAIMAN system, which offers publishing and retrieval services for users who are members of a community. The users can keep their perspective on a document space and exclusively work on their ontology for document management. This facilitates the exchange of explicit knowledge in communities considerably.

References

Attardi, G.; Gull'i, A.; and Sebastiani, F. 1999. Automatic web page categorization by link and context analysis. In Hutchison, C., and Lanzarone, G., eds., *THAI-99, 1st European Symposium on Telematics, Hypermedia and Artificial Intelligence*, p. 105–119.

Bonifacio, M.; Bouquet, P.; and Manzardo, A. 2000. A distributed intelligence paradigm for knowledge management. In *2000 AAAI Spring Symposium, Bringing Knowledge to Business Processes*, pp. 69–76. AAAI Press.

Chakrabarti, S.; Dom, B.; Agrawal, R.; and Raghavan, P. 1998. Scalable feature selection, classification and signature generation for organizing large text databases into hierachical topic taxonomies. *The VLDB Journal* (7):pp. 163–178.

Chandrasekaran, B.; Josephson, J. R.; and Benjamins, V. R. 1999. What are ontologies, and why do we need them ? *IEEE Intelligent Systems* 14(1):pp. 20–26.

Clark, D. 1999. Mad cows, metathesauri and meaning. *IEEE Intelligent Systems* 14(1):pp. 75–77.

Goller, C.; Löning, J.; Will, T.; and Wolff, W. 2000. Automatic document classification: A thorough evaluation of various methods. In *Internationales Symposium für Informationswissenschaft (ISI 2000)*.

Gruber, T. R. 1993. Toward principles for the design of ontologies used for knowledge sharing. Technical report, Stanford University.

Grudin, J. 1994. Groupware and social dynamics: eight challenges for developers. *Communications of the ACM* 37(1):pp. 93–105.

Huhns, M. N., and Singh, M. P. 1997. Ontologies for agents. *IEEE Internet Computing* 1(6):p. 81–83.

J.Bayardo, R.; Bohrer, W.; Brice, R.; Cichocki, A.; Fowler, J.; Helal, A.; Kashyap, V.; Ksiezyk, T.; Martin, G.; Nodine, M.; Rashid, M.; Rusinkiewicz, M.; Shea, R.; Unnikrishnan, C.; Unruh, A.; and Woelk, D. 1997. Infosleuth: agent-based semantic integration of information in open and dynamic environments. In *ACM SIGMOD international conference on Management of data*, p. 195–206.

Joachims, T. 1998. Text categorization with support vector machines: learning with many relevant features. In *European Conference on Machine Learning (ECML 98)*.

Koch, M., and Lacher, M. S. 2000. Integrating community services - a common infrastructure proposal. In *Proc. Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, pp. 56–59.

Labrou, Y., and Finin, T. 1999. Yahoo! as an ontology - using yahoo! categories to describe documents. In Gauch, S., ed., *Proc. 8th Intl. Conf. on Information Knowledge Management (CIKM'99)*, pp. 180–187. ACM Press.

Lacher, M. S., and Koch, M. 2000. An agent-based knowledge management framework. In *Proc. AAAI Spring Symposium 2000*, pp. 145–147.

Liebowitz, J., ed. 1999. *Knowledge Management Handbook*. CRC Press.

Melnik, S.; Garcia-Molina, H.; and Paepcke, A. 2000. A mediation infrastructure for digital libraries. In *ACM Digital Libraries*.

Mitra, P.; Wiederhold, G.; and Kersten, M. 2000. A graph-oriented model for articulation of ontology interdependencies. In *VII. Conference on Extending Database Technology (EDBT 2000)*.

Noy, N. F., and Hafner, C. D. 1997. The state of the art in ontology design. *AI Magazine* (Fall 1997):pp. 53–74.

Pretschner, A., and Gauch, S. 1999. Ontology based personalized search. In *Proc. 11th IEEE Intl. Conf. on Tools with Artificial Intelligence*, pp. 391–398.

Probst, G. J., and Büchel, B. S. 1998. *Organisationales Lernen: Wettbewerbsvorteil der Zukunft*. Gabler.

Staab, S.; Angele, J.; Decker, S.; Erdmann, M.; Hotho, A.; Mädche, A.; Schnurr, H.-P.; Studer, R.; and Sure, Y. 2000. Ai for the web - ontology-based community web portals. In *AAAI2000/IAAI2000 - Proc. 17th National Conf. on Artificial Intelligence and 12th Innovative Applications of AI Conf.* AAAI Press/MIT Press.

Takeda, H.; Matsuzuka, T.; and Taniguchi, Y. 2000. Discovery of shared topics networks among people. In Riichiro, M., and John, S., eds., *International Conference on Artificial Intelligence (PRICAI 2000)*, pp. 668–678. Springer.

Welty, C. 2000. Towards a semantics for the web. In *Proc. Dagstuhl-Seminar: Semantics for the Web*.