Personalized Information Retrieval in Bibster, a Semantics-Based Bibliographic Peer-to-Peer System

Peter Haase, Nenad Stojanovic, Johanna Völker and York Sure
Institute AIFB, University of Karlsruhe, D-76128 Karlsruhe, Germany
{haase,stojanovic,voelker,sure}@aifb.uni-karlsruhe.de

Abstract: Bibster is a semantics-based Peer-to-Peer system for exchanging bibliographic data among researchers. Bibster exploits ontologies in data storage, query formulation, query routing and answer presentation. While the original Bibster system assumed a globally shared domain ontology, we here describe extensions to the Bibster system, that allow to learn personalized ontologies from the local bibliographic metadata. These personal ontologies can not only be used for subsequently classifying the bibliographic metadata, but also for supporting an improved query refinement process.

1 Introduction

Bibster[1] [9] is an award-winning semantics-based Peer-to-Peer application aiming at researchers who want to benefit from sharing bibliographic metadata. Many researchers in computer science keep lists of bibliographic metadata, preferably in BibTeX format, that they must laboriously maintain manually. At the same time, many researchers are willing to share these resources, assuming they do not have to invest work in doing so. Bibster enables the management of bibliographic metadata in a Peer-to-Peer fashion: it allows to import bibliographic metadata, e.g. from BibTeX files, into a local knowledge repository, to share and search the knowledge in the Peer-to-Peer system, as well as to edit and export the bibliographic metadata.

In typical retrieval use cases, researchers want to: (1) search for bibliographic entries using simple keyword searches, but also more advanced, semantic searches, e.g. for publications of a special type, with specific attribute values, or about a certain topic, (2) organize, manage and query their bibliography using metadata descriptions that best reflect their personal interests and expertise, (3) explore the knowledge available in the peer network, either by directing queries to a specific set of peers (e.g. all colleagues at an institute) or the entire network.

To support the first use case of supporting semantic searches in a Peer-to-Peer network, the bibliographic metadata has to be represented in a structured and formal way. Here, ontologies provide the means to establish a globally-agreed and formal representation of the shared metadata. Yet, a globally shared and static ontology does not meet the requirements of the second use case, because of the diverse interests of the users in the peer network. On the other hand, the bibliographic content in the local repositories of the individual users already provide an implicit conceptualization of their domain of interest. By applying

[1] http://bibster.semanticweb.org/

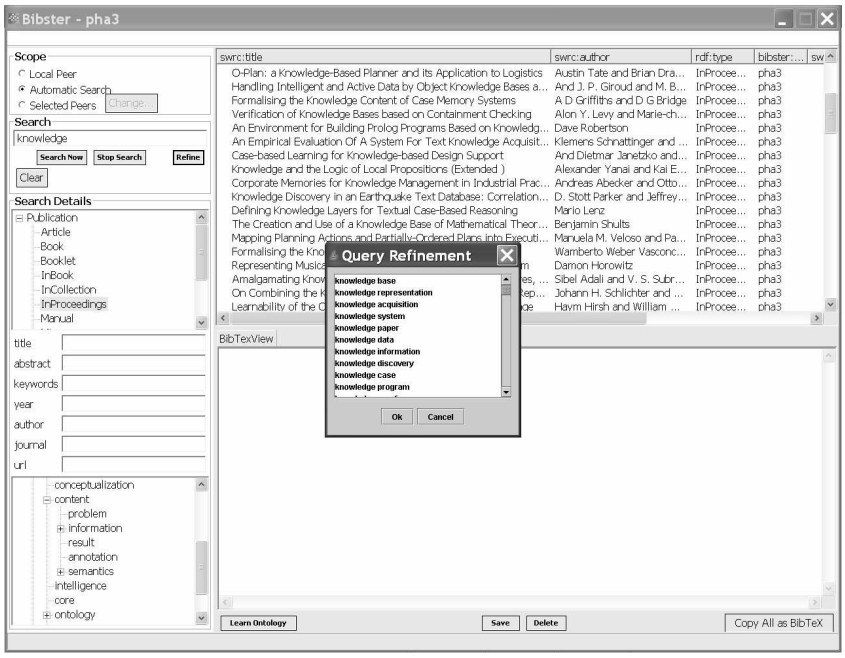

Figure 1: Interactive Information Retrieval in the Bibster User Interface

ontology learning techniques on this content, we can make these conceptualizations explicit and support personalized ontologies for the organization of the metadata. Especially from the third use case we see that a very important characteristic of the information retrieval task is that it is an exploratory process, as in a Peer-to-Peer environment users cannot be familiar with the content of the information repositories of the other peers. Further, in searching for information researchers often start with ill-defined needs and later redefine what they are actually searching for.

The screenshot in Figure 1 partially indicates how the above use cases are realized in Bibster. The *Scope* widget allows for defining the targeted peers (local search, entire network, etc.). The *Search* and *Search Details* widgets allow for keyword and semantic search; the tree in the lower left shows a fragment of the personal ontology learned from the local repository. The *Results Table* and *BibtexView* widgets allow for browsing and re-using query results. Finally, the *Query Refinement* dialog presents suggestions of how the query could be refined to improve search results. In particular, in the example the user posed a query for publications of type *InProceedings* with the search term *knowledge*, a term with ambigue senses. The query refinement process was able to discover the ambiguities and generate corresponding refinements, which are presented to the user in order of the obtained ranking.

The main contribution of this paper is the combination of three pillars: (i) the Bibster system itself, (ii) advanced query refinement and (iii) an extension for ontology learning from the information repository.

2 Ontologies in Bibster

Ontologies are crucial throughout the usage of Bibster, viz. for importing data, formulating queries, routing queries, and processing answers. Before we introduce the specific use of ontologies in Bibster, we will review the generic ontology model of [16], which we adhere to throughout this paper.

Ontology Model. An *ontology* is a structure $\mathcal{O} := (C, \leq_C, R, \sigma, \leq_R, I, \iota_C, \iota_R)$ consisting of three disjoint sets of entities C, R, and I called *concepts*, *relations*, and *instances*, a partial order $\leq_C$ on C *called concept hierarchy* or taxonomy, a function $\sigma_R \colon R \to C^2$ called *signature*, a partial order $\leq_R$ on R called *relation hierarchy*, a function $\iota_C \colon C \to \mathcal{P}(I)$ called *concept instantiation*, a function $\iota_R \colon R \to \mathcal{P}(I^2)$ called *relation instantiation*.

Two ontologies are used to describe properties of bibliographic entries in Bibster, an application ontology and a domain ontology [8]. Bibster uses the SWRC[2] ontology as application ontology, that describes different generic aspects of bibliographic metadata, including a concept hierarchy of types of publications, persons, etc.

The domain ontology conceptualizes the knowledge described in the shared documents, enabling advanced querying and browsing. Figure 2 shows a metamodel of the ontology and how documents are associated with it. The ontology elements basically reflect the ontology model described above. They are related with the documents via *document pointers* that index occurences of *ontology elements*. It is important to note the distinction between the *entities* of the ontology and their lexical references as *terms* in the document. For example, consider the term "library", which may be a lexical reference to either the concept library as a collection of books or the concept library as a software component.

In Bibster, we initially used the ACM Topic Hierarchy[3] as the domain ontology. This topic hierarchy describes specific categories of literature for the Computer Science domain. However, the ACM Topic Hierarchy does not always reflect the needs of the individual users. This is largely motivated by the sheer size of the ACM Topic Hierarchy which makes browsing, and therefore also querying and manual classification, difficult for users. As part of this work we therefore realized methods to learn personalized domain ontologies that reflect the actual content of the repositories of the indivual users.

3 Ontology Learning

The benefits of ontology learning for the personalization of information retrieval in Bibster are twofold: First, by extracting an ontology from the bibliographic

[2] `http://ontoware.org/projects/swrc/`
[3] `http://www.acm.org/class/1998/`

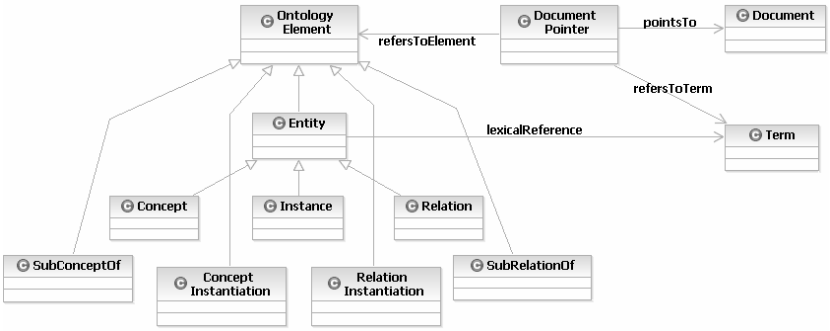

Figure 2: Ontology and Document Model

metadata stored in the user's local repository one obtains a domain ontology which is custom tailored to his personal interests. Such a learned ontology allows for creating a personalized classification scheme leading to an increased effectiveness of ontology-based browsing and searching. Second, a personalized domain ontology extracted from the user's repository can be used for improving the query refinement process described in Section 4 the effectiveness of which strongly depends on the quality of the underlying background information. Concept hierarchy relationships, concept instantiations as well as relations between concepts in the learned ontology can be used for personalized refinement of the user's query. If the underlying ontology is tailored to the repository, the refinement process is more reliable, since only the concepts and relations relevant for the repository will be taken into account.

To learn a personalized ontology from the user's local repository we have to extract sufficient amounts of textual data from the user's BibTeX entries. This is done primarily by considering the abstracts which are part of the bibliographic metadata. Wherever possible we also extract text from full text documents, e.g. if available via a specified URL included in some of the BibTeX entries.

For the ontology learning process we make use of Text2Onto [6], a framework for ontology learning from textual resources. From the collection of independent tools which is provided by Text2Onto for different ontology learning tasks we chose a subset which we considered useful for our purposes.

Concept extraction and **instance extraction** identify the most relevant concepts and instances in the repository by means of the `TFIDF` measure. Moreover, these algorithms associate each concept or instance with the set of terms representing its possible lexicalizations.

An algorithm for the extraction of **concept hierarchy relationships** is used to construct an initial taxonomy from the previously extracted concepts. This taxonomy can not only be used for classifying the documents in the user's local repository, but it also serves as a basis for the following extraction of instances and relations. The algorithm can be configured to employ one of two approaches: Whereas the first one makes use of Formal Concept Analysis (FCA)

as described by [4], the second approach is based on a combination of Hearst-Patterns [11], WordNet [7] and various heuristics.

Instance classification is applied to learn concept instantiations using a combination of various patterns from [11] and [10]. Examples for these patterns are: Hearst patterns such as *instance* `and other` *concept* and *concept* `such as` *instance*, definites like `the` *instance concept*, copulas such as *instance* `is a` *concept* and appositions like, for instance, *instance*`, a` *concept*.

Finally, **relation extraction** and **relation instance extraction** enrich the ontology with relations and relation instantiations. Basically, the approach being applied by RelationExtraction employs shallow text parsing in order to extract subcategorization frames, which can be restricted by using the information about selectional preferences [14], that is typical co-occurrences of predicates and conceptual classes, derived from the ontology.

The result of the ontology learning process is an ontology consisting of concepts, instances, concept hierarchy relationships, concept instantiations, relations and relation instantiations.

4 Query Refinement Process

The goal of the Librarian Agent Query Refinement process [15] is to enable a user to efficiently find results relevant for his information need in an ontology-based information repository, even if his query does not match ideally his information need, so that either a lot of irrelevant results and/or only a few relevant results are retrieved. As queries we consider conjunctions of terms from the metamodel presented in Section 2. In the Librarian Agent Query Refinement process, potential ambiguities (i.e. misinterpretations) of the initial query are firstly discovered and assessed (cf. the so-called *ambiguity discovery phase*). Next, the suitable query refinements are generated in order to decrease the accounted ambiguities (cf. the so-called *refinement generation phase*). Finally, the recommendations for refining the given query are ranked according to their relevance for fulfilling the user's information need and according to the possibility to disambiguate the meaning of the query (cf. the so-called *ranking phase*). In that way, the user is provided with a list of relevant query refinements ordered according to their capabilities to increase the number of relevant results.

The approach requires rich background knowledge about the domain in order to provide as relevant as possible refinements. We here exploit that a personalized ontology is learned for each peer, using the ontology learning methods described in Section 3.

Phase 1 – Ambiguity discovery: Query ambiguity is a measure for the gap between the user's information need and the query that results from that need. If a query is more ambiguous, then it follows that there are more (mis)interpretations of that query. We distinguish two types of ambiguity that can arise in interpreting a query: (i) the semantic ambiguity, as the characteristic of the used ontology and (ii) the content-related ambiguity, as the characteristic of the repository.

Semantic ambiguity: Semantic ambiguity is defined using several levels of the ambiguity of a query. First, we consider the sense ambiguity of the query terms. The sense of a query term refers to the set of ontology entities that have the term as a lexical reference. We can then identify the ontology context by analyzing how these ontology entities are related. Further we consider the clarity of the context, a measure for the existence of incomplete information in the query. The meaning of a query can be clarified by generating refinements that complete this missing information. For a complete defintion of the measures we refer the reader to [15].

Content-related ambiguity: From the content point of view, the results of a query can be used for defining potential ambiguities which arise in the query process. For example, if two queries have the same result set, then that list of results can be treated as ambiguous - it can be (mis)interpreted as the result of two different queries. Therefore, the content-related ambiguity of a query can be measured by comparing the results of the given query with the results of other queries. More precisely, we defined two relations between queries, which are, thereafter, used for estimating the content-based ambiguity of a query: extensional equivalence and structural subsumption between queries.

Phase 2 – Refinement generation: The previous phase indicates what are problems in the interpretation of a query. The candidates that should help in resolving these problems are generated in this phase. In order to help a user to find the most appropriate refinements for his information need, we support so called step-by-step query refinement. This is the process in which only one query terms should be added to the user's query in a refinement step. Moreover all equivalent queries are added to that refinement, so that the user gets a whole picture about the effect of a refinement. This type of the refinement requires that in each step a complete and minimal set of refinements is generated. We achieve these properties by using formal concept analysis [4].

Phase 3 – Ranking: In order to determine the relevance of a refinement for a user's need, we use two sources of information: (a) the user's preferences for such a refinement and (b) the informativeness of a refinement. Due to lack of space we just sketch these approaches:

a) Since the users are reluctant to provide an explicit information about the relevance of a result, the ranking has to be based on the implicit information that is captured by observing the user's behavior, so-called implicit relevance feedback. In the query refinement a user interacts subsequently with the system so that, in order to discover the user's preferences, we have to take into account not only the last query a user made, but rather the whole process of creating a query. We define three types of such an implicit relevance feedback: (i) *Recency* which captures that the terms most recently introduced in a user's query are more indicative of what the user currently finds relevant for his need; (ii) *ImplicitRelevance* which postulates that if a user selects a resource from the list of retrieved results, then this resource corresponds to the user's information need; (iii) *ImplicitIrrelevance* that is opposite to the previous type of relevance.

b) *Informativeness* describes the value of a refinement regarding the under-

lying information repository. It uses information theory (i.e. entropy) to define the information content of a refinement. Finally, the total relevance for the refinement of the query is a function of all these four parameters.

5 Related Work

There exist various systems that aim at applying semantics in Peer-to-Peer information systems: Edutella [13] is a Peer-to-Peer system based on the JXTA platform, which focuses on the exchange of learning material. P-Grid [1] is a structured, yet fully-decentralized Peer-to-Peer system based on a virtual distributed search tree. The DFN Science-to-Science (S2S) [18] system enhances content based searching by using peer-to-peer technology to make locally generated indexes accessible in an ad hoc manner. Various systems address the issue of heterogeneity in Peer-to-Peer systems on the schema level, such as the Piazza peer data management system [17], which allows for information sharing with different schemas relying on local mappings between schemas. However, none of these systems address the issue of automatically creating ontologies from the local content available on the peers. On the other hand, the topic of ontology learning has received attention in various other contexts of the emerging semantic web [12], such as automatic annotation of web pages [5]. The use of ontologies in information retrieval systems, especially focusing on query refinement, has been studied for example in [15]. Approaches for Peer-to-Peer information retrieval systems have recently been proposed in [2] (concentrating on architecture) or [3] (focusing on distributed ranking). To our knowledge, the Bibster system is the first running Peer-to-Peer that implements ontology-based information retrieval.

6 Conclusion

The use of ontologies in Peer-to-Peer systems is a promising approach to enable richer organization and searching of knowledge within communities. Bibster, a semantics-based Peer-to-Peer system for the exchange of bibliographic metadata between researchers, has proven to be a successful realization of this approach. In this paper we have presented extensions of the Bibster system by integrating Ontology Learning to support personalized ontologies and the Librarian Agent Refinement Process to support an interactive information search. By extracting natural language text from the bibliographic metadata stored in the user's local repository we acquired sufficient amounts of data for learning an ontology which reflects the user's personal interests.

Several evaluation studies are planned for the future work. We will primarly try to evaluate the benefits regarding the the quality of the retrieval process, namely the time spent in the searching and the precision of retrieved results.

Acknowledgments. Research reported in this paper has been partially financed by the EU project SEKT, IST-2003-506826 (`http://www.sekt-project.com/`).

References

1. K. Aberer et al. P-Grid: a self-organizing structured p2p system. *ACM SIGMOD Record*, 32(3):29–33, 2003.
2. K. Aberer, F. Klemm, M. Rajman, and J. Wu. An architecture for peer-to-peer information retrieval. In *Workshop on Peer-to-Peer Information Retrieval*, 2004.
3. W.-T. Balke, W. Nejdl, W. Siberski, and U. Thaden. Progressive distributed top k retrieval in peer-to-peer networks. In *ICDE*, 2005.
4. P. Cimiano et al. Automatic acquisition of taxonomies from text: Fca meets nlp. In *Proc. of the PKDD/ECML'03 Int. WS on Adaptive Text Extraction and Mining*, 2003.
5. P. Cimiano, S. Handschuh, and S. Staab. Towards the self-annotating web. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 462–471. ACM Press, 2004.
6. Philipp Cimiano and Johanna Voelker. Text2onto - a framework for ontology learning and data-driven change discovery. In *Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB'2005)*, JUN 2005. to appear.
7. C. Fellbaum. *WordNet, an electronic lexical database*. MIT Press, 1998.
8. N. Guarino. Formal ontology and information systems. In *Proceedings of the 1st Int. Conf. on Formal Ontologies in Information Systems (FOIS)*, 1998.
9. P. Haase et al. Bibster - a semantics-based bibliographic peer-to-peer system. In *Proc. of the Third Int. Semantic Web Conference, Hiroshima, Japan, 2004*, NOV 2004.
10. U. Hahn and K. Schnattinger. Towards text knowledge engineering. In *Proceedings of the AAAI'98/IAAI'98*, 1998.
11. M.A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics*, 1992.
12. A. Maedche and S. Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16(2):72–79, 2001.
13. W. Nejdl et al. Edutella: A P2P networking infrastructure based on rdf. In *Proc. of the Eleventh Int. World Wide Web Conference*, May 2002.
14. P. Resnik. Selectional preference and sense disambiguation. In *Proc. of the ACL SIGLEX WS on Tagging Text with Lexical Semantics: Why, What, and How?*, 1997.
15. N. Stojanovic, R. Studer, and L. Stojanovic. An approach for step-by-step query refinement in the ontology-based information retrieval. In *WI 2004*, SEP 2004.
16. G. Stumme et al. The Karlsruhe view on ontologies, 2003. U of Karlsruhe, Inst. AIFB.
17. I. Tatarinov et al. The piazza peer data mgt project. *SIGMOD Rec*, 32(3), 2003.
18. R. Wertlen. Dfn science-to-science: Peer-to-peer scientific research. In *Proceedings of the Terena Networking Conference (TNC 2003)*, Zagreb, Croatia, 2003.