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

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

The advantages of Peer-to-Peer architectures over centralized approaches have been well advertised, and to some extent realized in existing applications: no centralized server (thus avoiding a bottleneck for both computational performance and information update), robustness against failure of any single component, scalability both in data volumes and the number of connected parties.

The research community has recently turned to the use of semantics in Peer-to-Peer networks to facilitate new ways of querying, sharing, and organizing knowledge within communities [B^+03, N^+02, C^+03a]. The use of semantic descriptions of data sources stored by peers and indeed of semantic descriptions of peers themselves helps in formulating queries such that they can be understood by other peers, in merging the answers received from other peers, and in routing queries across the network. In particular, the use of ontologies and of Semantic Web technologies has been identified as promising for Peer-to-Peer systems.

Bibster[1] [H^+04] 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.

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

However, often a user cannot represent his information need in a "perfect" query, which means that he needs some support to reformulate/refine his query. This is specially true for the distribute information repositories, like in the presented system. In our previous work we have developed a comprehensive approach for ontology-based query refinement, called Librarian Agent Refinement Process, which enables a user to navigate through the information content incrementally and interactively. In each refinement step a user is provided with a complete but minimal set of refinements, which enables him to develop/express his information need in a step-by-step fashion.

The effectiveness of the method depends on the quality of the underlying background information. In an ideal case the ontology should be highly specified to the underlying information repository and the user.

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.

The rest of this paper is organized as follows: We will first present use cases for the Bibster system in Section 2. We describe the role of ontologies in Bibster in Section 3. The methods for ontology learning and query refinement are elaborated in Sections 4 and 5, respectively. We discuss related work in Section 6 before we conclude in Section 7.

2 Major Bibster Use Cases

Bibster is aimed at researchers that share bibliographic metadata. Requirements for Bibster must include capabilities that support their daily work. Researchers may 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 an 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 of peers,

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. However, the bibliographic metadata in BibTeX files is semi-structured, single attributes may be missing or may not be interpreted correctly, and there is no agreed way of classification. Here, ontologies provide the means to establish a globally-agreed and formal representation of the shared metadata.

However, a globally shared and static ontology does not meet the requirements of the second use case, because of the diversing 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 ontology

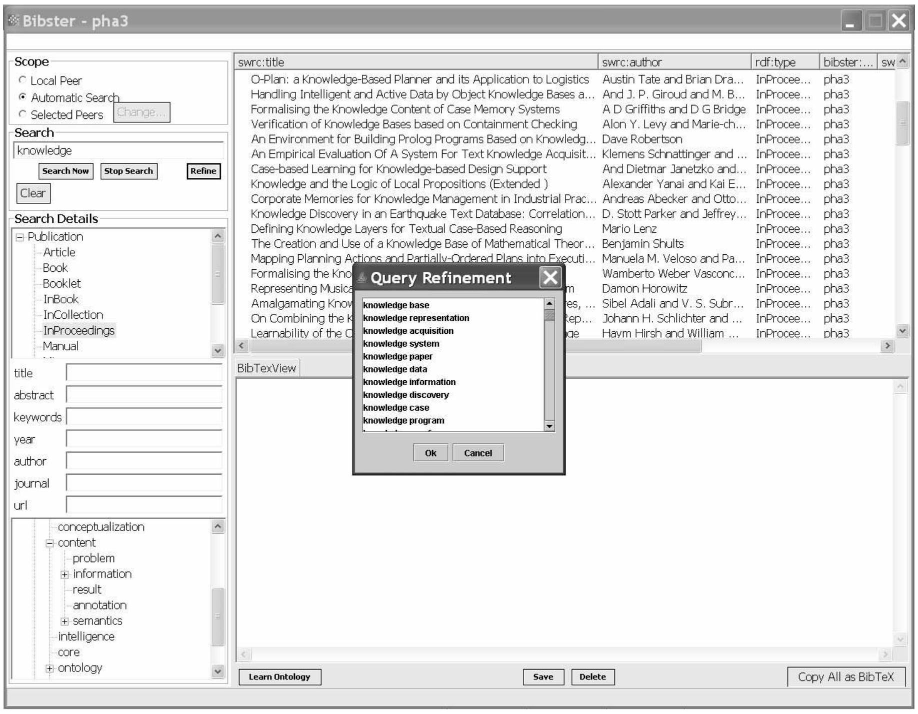

Abbildung 1: Interactive Information Retrieval in the Bibster User Interface

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. We mention just two reasons: (i) In a peer-to-peer environment a user cannot be familiar with the content of the other's peers information repositories. In order to avoid making an over-specified query that retrieves zero results (i.e. a failing query), a user makes a short query that should return some results for sure (e.g. he is not aware which type of optimization are treated in the underlying repository). Therefore he needs support not just to make a query, but also to explore the resulting list in an efficient manner. (ii) In searching for information researchers often have ill-defined needs. They start searching by assuming what can be the right information, but often, by exploring the repositories of other researchers, they redefine what they are actually searching for. Both cases should be treated in an incremental manner - based on the information in the previous retrieval step a user tries to redefine his query and to search further.

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, as explained in Section 5.

3 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 [S$^+$03], 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 C, R, and I called *concept identifiers*, *relation identifiers*, and *instance identifiers*, 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*.

In Bibster, two ontologies are used to describe properties of bibliographic entries in Bibster, an application ontology and a domain ontology [Gua98]. 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 SWRC ontology has been used already in various projects, e.g. also in the semantic portal of the Institute AIFB[3].

The domain ontology is used for classification of metadata entries, enabling advanced querying and browsing. It describes topic hierarchies using relations such as SubTopic, RelatedTopic, etc. In Bibster, we initially used the ACM Topic Hierarchy[4] as the domain ontology. This topic hierarchy describes specific categories of literature for the Computer Science domain. It covers large areas of computer science, containing over 1287 topics ordered using taxonomic relations, e.g.:
$SubTopic(Artificial\_Intelligence, Knowledge\_Representation\_Formalisms)$.
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 indivual users, as described in Section 4.

[2] `http://ontoware.org/projects/swrc/`
[3] `http://www.aifb.uni-karlsruhe.de/about.html`
[4] `http://www.acm.org/class/1998/`

Use of Ontologies The ontologies are then used in the system in the following ways: Firstly, the system enables users to import their own bibliographic metadata into a local repository. Bibliographic entries made available to Bibster by a user are automatically aligned to the application and domain ontology. Secondly, queries are formulated in terms of the two ontologies: Queries may concern fields like author, publication type, etc. (using terms from the SWRC ontology) or queries may concern specific terms of the domain ontology. Thirdly, queries are routed through the network depending on the expertise models of the peers describing which concepts from the ACM ontology a peer can answer queries on. A matching function determines how closely the semantic content of a query matches the expertise model of a peer. Routing is then done on the basis of this semantic ranking [HSvH04]. Finally, answers are returned for a query. Due to the distributed nature and potentially large size of the Peer-to-Peer network, this answer set might be very large, and contain many duplicate answers. Because of the semistructured nature of bibliographic metadata, such duplicates are often not exactly identical copies. Ontologies help to measure the semantic similarity between the different answers and to remove apparent duplicates as identified by the similarity function. As described in the previous section, there is a need to realize the search in the network as an interactive process. Here, ontologies are used to improve the query refinement as described in Section 5.

4 Ontology Learning

In order to reduce efforts for engineering large ontologies, recent years have seen a surge of interests for learning ontologies from text in general and learning of taxonomies, i.e. concept hierarchies, in particular from textual data [MS01]. The principle paradigm exploited in many of these approaches is to derive knowledge from texts by analyzing how certain terms are used. The *distributional hypothesis* assumes that terms are similar to the extent to which they share similar linguistic contexts and thus gives rise to various methods that cluster terms based on their linguistic context and form corresponding taxonomies. 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 meta data. Wherever possible we also extract full text from the 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 TextToOnto [MV01], a tool suite for ontology learning by text mining techniques which is built upon the ontology management infrastructure KAON [KK04]. From the collection of independent tools for both ontology extraction and maintenance provided by TextToOnto we chose a small subset which we considered useful for our purposes.

TaxoBuilder is used to construct an initial taxonomy from the most frequent terms in the repository. 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. TaxoBuilder can be configured to employ one of two approaches: (i) The FCA-based approach described by [C^+03b] rests upon the assumption that a verb poses strong selectional restrictions on their arguments, so that a hierarchy of concepts can be derived from the inclusion relations between the extensions of the selectional restrictions of all the verbs, while the verbs themselves provide intensional descriptions for each concept. (ii) The second approach is based on a combination of Hearst-Patterns [Hea92], WordNet [Fel98] and various heuristics.

InstanceExtraction is applied for populating the ontology with instances. InstanceExtraction supports both semi-automatic and fully automatic learning of instances by applying a combination of various patterns from [Hea92] and [HS98]. Typical 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*.

RelationLearning is used to add relations to the ontology. As many relations as possible are to be extracted to support the query refinement process (see section 5). Basically, the approach being applied by RelationLearning employs shallow text parsing in order to extract subcategorization frames, which can be restricted by using the information about selectional preferences [Res97], 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, concept hierarchy relationships, concept instantiations and relations. Whereas the relations can be used for query refinement, the taxonomy including concept hierarchy relationships and concept instantiations serves as a suitable basis for constructing a topic hierarchy which can be used for the classification of the documents contained in the user's local repository. Since in our experiments the whole ontology learning process was done in a fully automatic fashion the results cannot be compared to manually constructed ontologies. Nevertheless, one could also imagine a semi-automatic and interactive way of building the ontology. In this case the system will suggest new concepts, instances and relations to the user each time new BibTeX entries are added to his local repository.

5 Query Refinement Process

The goal of the Librarian Agent Query Refinement process 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. 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 decrease the number of irrelevant results or/and 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. Such an ontology can be predefined for a domain, independently on the underlying information repository. However, 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. In this system we assume that a personalized ontology will be learned for each peer, using the ontology learning methods described in Section 4. In the next three subsections we explain the query refinement method in details.

Phase 1 – Ambiguity Discovery: We define query ambiguity as an indicator of 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 define two types of the 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 term.

SenseAmbiguity. The sense of a query term represents the ontology context, which the term appear in. It can be clarified by analysing the relations with the senses of other query terms. For a query $Q = t_1, ..., t_n$ the SenseAmbiguity is defined as follows: $SenseAmbiguity(Q) = \frac{(\sum_{t_i,t_j \in Q} NumberOfSensesInContext(t_i,t_j)}{NumberOfSenses(Q)}$

$NumberOfSensesInContext(t_i, t_j) = |i_p \in Sense(t_i), i_k \in Sense(t_j) : Relation(i_p, i_k))|$

$NumberOfSenses(Q) = (| \sum_{t_i \in Q} |$

Sense(t_i) is the set of the senses of the term t_i in the ontology. Relation(i_p, i_k) is the function that returns 1 if there is an ontology relation between the concepts that correspond to these terms.

ContextClarity. This parameter models the existence of an incomplete information in a query, regarding the used concepts/relations. It means that the query can be automatically expanded in order to clear the meaning of the query. For measuring the context clarity of a query we use the following formulas:

$ContextualClarity(Q) = \prod_{i=1,n,j=1,n} Contextuality(Q, C_i, C_j)$, where $C_i, C_j \in Q$ and

$Contextuality(Q, C_i, C_j) = \frac{1}{Properties(C_1,C_2)} and Properties(C_1, C_2)$ is the function that returns the set of all relations between C_1 and C_2.

Clarity. The clarity factor represents the uncertainty to determine the user's interest in the given query

$Clarity(Q) = \frac{\sum_{t_i \in Q, i_p \in Sense(t_i)} TermClarity(i_p)}{NumberOfSenses(Q)}$, where TermClarity($i_p$) represents the reciprocity value of the number of subconcepts of the term i_p

Content-related ambiguity: An ontology defines just a model how the entities from a real domain should be structured. If there is a part of that model that is not instantiated in the given domain, then that part of the model cannot be used for calculating ambiguity. Therefore, we should use the content of the information repository to prune the results from the ontology-related analysis of a user's query. 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 an ambiguous entity - it can be (mis)interpreted as the result of two different queries. However, since a user posts a query and wants to refine a query and not to change directly the list of results, we will interpret all content-related ambiguities on the level of the user's query. Regarding the previous example, two queries that return the same list of results are treated as equivalent queries. In that case, after posting a query, a list of equivalent queries is presented to the user as an indicator of the content-related ambiguity of his query. 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 (see [NSS04] for definition of extensional equivalence and structural subsumption).

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 get 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 [C^+03b].

Phase 3 – Ranking: In order to determine the relevance of a refinement for a user's need, we use two sources of information: (a) user's preferences for such a refinement and (b) 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 are captured by observing user's behavior, so-called implicit relevance feedback. In the query refinement a user interacts subsequently with the system so that, by discovering 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) *Actuality* which reflects the phenomena that a user may change the criteria about the relevance of a query term, when encountering newly retrieved results. In other words, the constraints 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 some extent, to the user's information need. It means, that by analyzing the commonalities in the attributes of results a user selected for viewing, we can infer more information about the intension of the user in the current query session and (iii) *ImplicitIrrelevance* that is opposite to the previous type of relevance.

b) *Informativeness* defines the capability of a refinement regarding the underlying information repository. It uses information theory (i.e. entropy) to define the information content of a refinement. Finally, the total relevance of the constraint c for the refinement of the query Q_i is a function of all these four parameters.

6 Related Work

There exists various systems that aim at applying semantics in Peer-to-Peer information systems: Edutella (*cf. eg.* [N+03]) is a Peer-to-Peer system based on the JXTA platform, which focuses on the exchange of learning material. P-Grid [A+03] is a structured, yet fully-decentralized Peer-to-Peer system based on a virtual distributed search tree. The DFN Science-to-Science (S2S) [Wer03] 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 [T+03], 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 [MS01], such as automatic annotation of web pages [CHS04]. The use of ontologies in information retrieval systems, especially focusing on query refinement, has been studied for example in [NSS04]. Approaches for Peer-to-Peer information retrieval systems have recently been proposed in [AKRW04] (concentrating on architectural issues) or [BNST05] (focusing on distributed ranking). To our knowledge, the Bibster system is the first running Peer-to-Peer that implements ontology-based information retrieval.

7 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.

Acknowledgments. Research reported in this paper has been partially financed by the EU in the IST-2003-506826 project SEKT (`http://www.sekt-project.com`). We would like to thank our colleagues for fruitful discussions, in particular to Max Völkel.

Literatur

[A+03] K. Aberer et al. P-Grid: a self-organizing structured P2P system. *ACM SIGMOD Record*, 32(3):29–33, 2003.

[AKRW04] Karl Aberer, Fabius Klemm, Martin Rajman und Jie Wu. An Architecture for Peer-to-Peer Information Retrieval. In *Workshop on Peer-to-Peer Information Retrieval*, 2004.

[B+03] J. Broekstra et al. A Metadata Model for Semantics-Based Peer-to-Peer Systems. In *Proc. of the WWW'03 WS on Semantics in Peer-to-Peer and Grid Comp.*, 2003.

[BNST05] Wolf-Tilo Balke, Wolfgang Nejdl, Wolf Siberski und Uwe Thaden. Progressive Distributed Top k Retrieval in Peer-to-Peer Networks. In *ICDE*, 2005.

[C+03a] A. Castano et al. Ontology-Addressable Contents in P2P Networks. In *Proc. of the WWW'03 WS on Semantics in Peer-to-Peer and Grid Computing*, 2003.

[C+03b] 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.

[CHS04] Philipp Cimiano, Siegfried Handschuh und Steffen Staab. Towards the self-annotating web. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, Seiten 462–471. ACM Press, 2004.

[Fel98] C. Fellbaum. *WordNet, an electronic lexical database*. MIT Press, 1998.

[Gua98] N. Guarino. Formal Ontology and Information Systems. In *Proceedings of the 1st Int. Conf. on Formal Ontologies in Information Systems (FOIS)*, 1998.

[H+04] 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.

[Hea92] M.A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics*, 1992.

[HS98] U. Hahn und K. Schnattinger. Towards text knowledge engineering. In *Proceedings of the AAAI'98/IAAI'98*, 1998.

[HSvH04] P. Haase, R. Siebes und F. van Harmelen. Peer Selection in Peer-to-Peer Networks with Semantic Topologies. In *Int. Conf. on Semantics of a Networked World: Semantics for Grid Databases, June 2004, Paris*, 2004.

[KK04] FZI Karlsruhe und AIFB Karlsruhe. KAON The Karlsruhe Ontology and Semantic Web Framework — Developer's Guide for KAON 1.2.7, 2004.

[MS01] Alexander Maedche und Steffen Staab. Ontology Learning for the Semantic Web. *IEEE Intelligent Systems*, 16(2):72–79, 2001.

[MV01] A. Maedche und R. Volz. The Ontology Extraction and Maintenance Framework Text-To-Onto. In *Proceedings of the ICDM'01 Workshop on Integrating Data Mining and Knowledge Management*, 2001.

[N+02] W. Nejdl et al. EDUTELLA: A P2P Networking Infrastructure based on RDF. In *Proc. of the Eleventh Int. World Wide Web Conference*, May 2002.

[N+03] W. Nejdl et al. Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks. In *Proceedings of the Twelfth International World Wide Web Conference (WWW 2003)*, Budapest, Hungary, May 2003.

[NSS04] Stojanovic N, R. Studer und L. Stojanovic. An Approach for Step-By-Step Query Refinement in the Ontology-based Information Retrieval. In *WI 2004*, SEP 2004.

[Res97] 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.

[S+03] G. Stumme et al. The Karlsruhe View on Ontologies, 2003. U of Karlsruhe, Inst. AIFB.

[T+03] I. Tatarinov et al. The Piazza Peer Data Mgt Project. *SIGMOD Rec*, 32(3), 2003.

[Wer03] Roland Wertlen. DFN Science-to-Science: Peer-to-Peer Scientific Research. In *Proceedings of the Terena Networking Conference (TNC 2003)*, Zagreb, Croatia, 2003.