Similarity for Ontologies - a Comprehensive Framework

Marc Ehrig[1], Peter Haase[1],
Mark Hefke[2], Nenad Stojanovic[1]

[1]Institute AIFB, University of Karlsruhe
[2]FZI Research Center for Information Technologies at the University of Karlsruhe
{ehrig, haase, stojanovic}@aifb.uni-karlsruhe.de
hefke@fzi.de

Abstract. In this paper we present a comprehensive framework for measuring similarity within and between ontologies as a basis for the collaboration across various application fields. In order to define such a framework, we base our work on an abstract ontology model that allows to adhere to various existing and evolving ontology standards. The main characteristics of the framework is its layered structure: We have defined three levels on which the similarity between two entities (concepts or instances) can be measured: data layer, ontology layer, and context layer, that cope with the data representation, ontology meaning and the usage of these entities, respectively. In addition, in each of the layers corresponding background information is used in order to define the similarity more precisely. The framework is complete in the sense of covering the similarity between all elements defined in the abstract ontology model by comprising similarity measures for all above-named layers as well as relations between them. Moreover, we have validated our framework with several practical case studies in order to prove benefits of applying our approach compared to traditional similarity measures. One of these case studies is described in detail within the paper.

1 Introduction

The importance of ontologies as ”an explicit specification of a shared conceptualization” [1] increased drastically in the last years, especially for the applications that require integration of heterogenous data, like knowledge management. Indeed, ontology becomes a very important technology for the improvement of the inter/intra-organizational exchange of knowledge and services. Moreover, ontology-based knowledge management enables a variety of new retrieval services, like personalization and cooperative answering. The key issue is that an ontology supports more granular views on the knowledge items that should be exchanged, enabling in that way the more context-sensitive retrieval process than in the traditional knowledge management systems. For example, if a user makes the query ”fast Jaguars”, an ontology-based system, by using the conceptual model of a domain, can distinguish between the various interpretations of the term Jaguar (e.g. car and animal) and enable the user to find only relevant items. However, due to the ambiguity in translating a user's information need in a query, an efficient knowledge management system should enable the user to find some items that do not match perfectly his query, but that are relevant for his need. For example, if a user wants to find information about the safety issues of Jaguar X, the relevant information could be documents about safety issues of the Ferrari Y, which has showed the same characteristics as Jaguar X in several crash tests. It is clear this calculation of the similarity between two items has to be performed very carefully in order to ensure that all relevant items and no irrelevant items will be retrieved. Moreover, the similarity computation depends on personal rules and preferences as well as on personal interpretation. To get a grip on arising problems from the above-described examples, it is in our opinion necessary to combine ontology-based technologies with novel approaches for similarity computation. Therefore we have developed a general framework for calculating the similarity within and between ontologies by also focusing on domain knowledge that has direct influence on the similarity. The main characteristics of the developed comprehensive framework is its layered structure: We have defined three layers on which the similarity between two entities, can be measured: data layer, ontology layer, and context layer, that cope with the data representation, ontological meaning and the usage of these entities, respectively. In that way, different layers consider different aspects of the nature of entities which are combined in the final judging about their similarity. Moreover, in each layer corresponding background information is used in order to define the similarity more precisely. Our intention is not only to develop a collection of existing methods for measuring similarity, but rather to define a framework that will enable their effective systematization regarding the task of comparing ontologies. The framework is complete in the sense of covering the similarity between all elements defined in the abstract ontology model and comprises several new methods in order to achieve such a completeness. Moreover, it provides some inter-/intralayer relations between methods, that enable the methods to be applied more efficiently. The framework has been validated with several practical case studies which are directly focusing on every single layer as well as their interaction in order to deliver a proof of concept.

The paper is organized as follows: In Section 2 we first introduce a formal model for ontologies and similarity in ontologies and thendescribe our general framework comprising three different layers for similarity computation. In Section 4 we validate our framework with a selected case study. After a discussion of related work in Section 5 we conclude with remarks outlining some future work.

2 Definitions

In this section we introduce basic definitions of ontologies and similarity. Upon these we build our similarity framework.

2.1 Ontology Model

In our framework we will adhere to the Karlsruhe Ontology Model[2], which we adapt to also accommodate datatypes:

Definition 1 (Ontology with Datatypes). *An ontology with datatypes is a structure*

$$O_T := (C, T, \leq_C, \leq_T, R, A, \sigma_A, \sigma_R, \leq_R, \leq_A)$$

consisting of a set of concepts C aligned in a hierarchy $\leq_C$, a set of relations R with $\leq_R$, the signature $\sigma_R\colon R \to C \times C$, a set of datatypes T with $\leq_T$, a set of attributes A with $\leq_A$, and the signature $\sigma_A : A \to C \times T$. For a relation $r \in R$, we define its domain *and its range by* $\mathrm{dom}(r) := \pi_1(\sigma_R(r))$ *and* $\mathrm{range}(r) := \pi_2(\sigma_R(r))$.

Ontologies formalize the intensional aspects of a domain. The extensional part is provided by a knowledge base, which contains assertions about instances of the concepts and relations.

Definition 2 (Knowledge Base with Datatypes). *A* knowledge base with datatypes *is a structure*

$$KB_T := (C_{KB}, T_{KB}, R_{KB}, A_{KB}, I, V, \iota_C, \iota_T, \iota_R, \iota_A)$$

consisting of fours sets C_{KB}, T_{KB}, R_{KB} and A_{KB} as defined above, a set of instances I, a set of data values V, the concept instantiation $\iota_C\colon C_{KB} \to \mathfrak{P}(I)$, the data value instantiation $\iota_T : T_{KB} \to \mathfrak{P}(V)$, the relation instantiation $\iota_R\colon R_{KB} \to \mathfrak{P}(I^2)$, and the attribute instantiation $\iota_A\colon A_{KB} \to \mathfrak{P}(I \times V)$.

2.2 Similarity

Common sense tells that two entities need common characteristics (or attributes) in order to be considered similar. Formalizing the concept of similarity, we refer to the definition of a similarity function introduced by [3]:

Definition 3 (Similarity Measure). *A similarity measure is a real-valued function $sim(x, y) : S^2 \to [0, 1]$ on a set S measuring the degree of similarity between x and y.*

Though there may be split opinions about the properties of sim, it is generally agreed that sim ought to be reflexive and symmetric, i.e.

$\forall x, y \in S$ it holds 1. $sim(x, x) = 1$ (reflexivity)
2. $sim(x, y) = sim(y, x)$ (symmetry)

Similarity for Ontologies

Definition 4. *Now given two knowledge bases with datatypes and the corresponding ontologies (KB_i, O_i) and (KB_j, O_j), we can compare elements of $KB_i^{O_i}$ as given by the following family of functions:*

$$sim : (E_i) \times (E_j) \to [0..1],$$

where $E_i \in \{O_i, C_i, T_i, R_i, A_i, I_i, V_i\}$ and $E_j \in \{O_j, C_j, T_j, R_j, A_j, I_j, V_j\}$ and E_i and E_j are of the same kind, i.e. both are instances or both are concepts, etc.

3 General Framework

Since an ontology represents a conceptualization of a domain, comparing two ontology entities goes far beyond the representation of these entities (syntax level). Rather, it should take into account their relation to the real world entities they are referencing, i.e. their meaning, as well as their purpose in the real world, i.e. their usage. In order to achieve such a comprehensive comparison, we use a semiotic view (theory of signs) on ontologies and define our framework for similarity in three layers, as shown in Figure 1: Data-, Ontology-, and Context Layer. We further enhance these by an additional orthogonal field representing specific domain knowledge. Initial blueprints for such a division in layers can be found in the semiotics (theory of signs) for example in [4], where they are called *symbolic*, *semantic* and *pragmatic* layer, respectively.

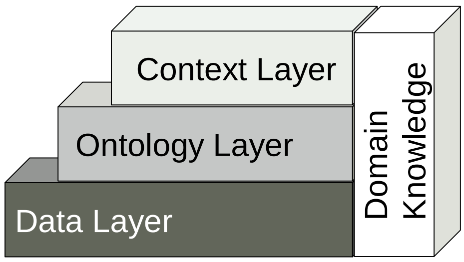

Fig. 1. Layer Model

Data Layer On this first layer we compare entities by only considering data values of simple or complex data types, such as integers and strings. To compare data values, we may use generic similarity functions such as the edit distance for strings. For integers we can simply determine a relative distance between them. The complex data types made up from simple data types would also require more complex measures, but which are effectively completely based on simple measures.

Definition 5 (Similarity based on Data).

$$sim_{data} : (E_i) \times (E_j) \rightarrow [0..1]$$

computes the similarity of entities based on the corresponding data values V_i *and* V_j *occurring in* KB_i *and* KB_j.

Ontology Layer In the second layer, the ontology layer, we consider semantic relations between the entities. In fact, we use the graph structure of the ontology to determine similarity. For specific predefined relations such as taxonomies or restrictions we can use specific heuristics. For example, certain edges could be interpreted as a subsumption hierarchy. It is therefore possible to determine the taxonomic similarity based on the number of *is-a* edges separating two concepts. Besides intensional features we can also rely on the extensional dimension i.e. assess concepts to be the same, if their instances are similar. The similarity measures of the ontology layer can include similarity measures of the data layer to determine the basic similarities.

Definition 6 (Similarity based on Ontology Structures).

$$sim_{ontology} : (E_i) \times (E_j) \rightarrow [0..1]$$

computes the similarity of entities based on the ontological structures of (KB_i, O_i) *and* (KB_j, O_j).

Context Layer On this layer we consider how the entities of the ontology are used in some external context. This implies that we use information external to the ontology itself. Although there are many contexts in which an ontology can be considered (for example the context in which an ontology is developed, or in which it has been changed), from the point of view of determining the similarity, the most important one is the application context, e.g. how an entity of an ontology has been used in the context of a given portal. An example for this is the *amazon.com* portal in which, given information about which people buy which books, we can decide if two books are similar or not in a given context. We assume that an ontology can be used for annotating content/documents in an information portal. Therefore, the similarity between two ontology entities can be easily determined by comparing their usage in an ontology-based application. A naive explanation is that similar entities have similar patterns of usage. However, the main problem is how to define these usage patterns in order to discover the similarity in the most efficient way. In order to generalize the description of such patterns we reuse the similarity principle from CBR (*similar problems have similar solutions*) in the terms of the usage: *similar entities are used in similar context*. We use both directions of the implication in discovering similarity: if two entities are used in the same (related) context then these entities are similar and vice versa: if in two contexts the same (related) entities are used then these contexts are similar.

Definition 7 (Similarity based on Context).

$$sim_{context} : (E_i) \times (E_j) \rightarrow [0..1]$$

computes the similarity of entities based on some context external to the ontologies.

Domain Knowledge Special shared ontology domains e.g. the bibliographic domain, have their own additional vocabulary. The right part of Figure 1 therefore covers domain-specific aspects. As this domain-specific knowledge can be situated at any level of ontological complexity, it is presented as a box across all of them. Just like we use general similarity features to compare ontologies we can also do so with domain specific features.

Amalgamation For the computation of the overall similarity between two entities we use an amalgamation function that combines the results of the individual similarity functions of the layers described above, i.e.

Definition 8 (Amalgamation of similarity functions).

$$sim(e_a, e_b) = \mathcal{A}(sim_1(e_a, e_b), \cdots, sim_n(e_a, e_b))$$

where sim denotes the overall similarity, and $\mathcal{A}$ the amalgamation function composing individual similarities sim_i ($i \in \{1, \cdots, n\}$).

4 Application Scenario

We have validated our similarity framework in varoius different application scenarios, such as Case-based Reasoning [5] and Usage Mining [6]. To illustrate the application of our similarity framework, we present Bibster, a semantics-based bibliographic Peer-to-Peer system. It can be treated as a Peer-to-Peer-based knowledge management system, since it enables the efficient access to the information stored in a Peer-To-Peer network. Bibster addresses researchers in a community that share bibliographic metadata via a Peer-to-Peer system. Many researchers own hundreds of kilobytes of bibliographic information, in dozens of BibTeX files. At the same time, many researchers are willing to share these resources, provided 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 Bibster[1] we make use of two common ontologies for the representation of bibliographic metadata: The first ontology is the Semantic Web Research Community Ontology (SWRC)[2], which models among others a research community, its researchers, topics, publications, and properties between them [7]. The second ontology is the ACM topic hierarchy[3], according to which our publications are classified.

4.1 Usage of Similarity

Ontology-based similarity measures are used for a variety of functionalities in the Bibster system:

1 http://bibster.semanticweb.org/
2 http://www.semanticweb.org/ontologies/swrc-onto-2001-12-11-daml
3 http://www.acm.org/class/1998/

Duplicate detection Due to the distributed nature and potentially large size of the Peer-to-Peer network, the returned result set for a query might be large and contain duplicate answers. Furthermore, because of the heterogeneous and possibly even contradicting representation, such duplicates are often not exactly identical copies. The ontology based similarity function allows us to effectively determine the similarity between the different answers and to remove apparent duplicate results. Instead of confronting the user with a list of all individual results, we are able to present query results grouped by semantic duplicates.

Peer Selection with Semantic Topologies In the Bibster system, the user can specify the scope of a query: He can either query the local knowledge, direct the query to a selected set of peers, or can query the entire peer network. For the latter option, the scalability of the Peer-to-Peer network is essentially determined by the way how the queries are propagated in the network. Peer-to-Peer networks that broadcast all queries to all peers do not scale – intelligent query routing and network topologies are required to be able to route queries to a relevant subset of peers that are able to answer the queries. In the Bibster system we apply the model of expertise based peer selection as proposed in [8]. Based on this model, peers advertise semantic descriptions of their expertise specified in terms of the ACM topic hierarchy. The knowledge about the expertise of other peers forms a semantic topology, in which peers with a similar expertise are clustered. That means, a semantic link between two peers is established, if their expertise is similar according to the similiarity function. To determine an appropriate set of peers to forward a query to, a matching function determines how closely the semantic content of a query that references an ACM topic matches the expertise of a peer.

Recommendations Bibster features recommender functionality that allow personalized access to the bibliographic metadata available in the Peer-to-Peer network according to the particular needs of the users. In a nutshell, the recommender functions are based on the idea that if a publication is known to be relevant, a similar publication might also be relevant.

In more detail, the recommender functions build upon the semantic representation of the available metadata, including content and usage information: The bibliographic metadata is represented according to the two bibliographic ontologies (SWRC and ACM). These ontological structures are then exploited to help the user formulate semantic queries. Query results again are represented according to the ontology. These semantic representations of the knowledge available on the peers, the user queries and relevant results allow us to directly create a semantic user profile. The semantic similarity function determines how well a publication matches the user profile. Potentially interesting publication are then recommended to the user.

4.2 Methods

We will now describe the specific methods applied in the system grouped by the layers *Data Layer*, *Ontology Layer*, and *Context Layer*. We also show how we exploit *Domain Knowledge* in the individual layers.

Data Layer On this layer we compare the literal values of specific attributes of the publication instances. For example, to detect typical differences in the representation of a publication title for the duplicate detection, we use the *Syntactic Similarity* function and are thus able to handle spelling errors or mismatches in capitalization, which is important for example for duplicate detection:

Syntactic Similarity [9] introduced a measure to compare two strings, the so called edit distance. For our purposes of similarity we rely on the syntactic similarity of [4] which is inverse to the edit distance measure:

$$sim_{syntactic}(v_1, v_2) := max(0, \frac{min(|v_1|, |v_2|) - ed(v_1, v_2)}{min(|v_1|, |v_2|)}) \quad (1)$$

The idea behind this measure is to take two strings and determine how many atomic actions are required to transform one string into the other one. Atomic actions would be addition, deletion, and replacement of characters, but also moving their position.

Further, we are using domain specific background knowledge to define more meaningful similarity measures. For example, for bibliographic metadata we know that attributes such as first and middle names are often abbreviated: In these cases we compare only the characters in front of the abbreviation dot. For other attributes expansion of the abbreviations makes sense before comparing them.

Ontology Layer To compare the classifications of two publications according to the ACM topic hierarchy, we use the *Taxonomic Similarity for Concepts*. We can then build the semantic topology of the Peer-to-Peer network according to the taxonomic similarity of the peers' expertise, i.e. the classified topics of the publications shared by the peer. This is necessary for efficient peer selection and routing of queries.

Taxonomic Similarity One possible generic function to determine the semantic similarity of concepts C in a concept hierarchy – or taxonomy – $\leq_C$ has been presented by Rada et al. in [10]:

$$sim_{taxonomic}(c_1, c_2) := \begin{cases} e^{-\alpha l} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}, & \text{if } c_1 \neq c_2, \\ 1, & \text{otherwise} \end{cases} \quad (2)$$

$\alpha \geq 0$ and $\beta \geq 0$ are parameters scaling the contribution of shortest path length l and depth h in the concept hierarchy, respectively. The shortest path length is a metric for measuring the conceptual distance of c_1 and c_2. The intuition behind using the depth of the direct common subsumer in the calculation is that concepts at upper layers of the concept hierarchy are more general and are semantically less similar than concepts at lower levels. This measure can be easily used analogously for relation R comparisons through $\leq_R$.

SWRC Concept Similarity For our specific scenario with the SWRC ontology as domain ontology, we have further background knowledge that allows us to define a simpler, but also more appropriate similarity function. There are many subconcepts of publications:

articles, books, and technical reports to just name a few. We know that if the type of a publication is not known, it is often provided as Misc (e.g. in Citeseer). We therefore use the following function:

$$S(c_1, c_2) = \begin{cases} 1, & \text{if } c_1 = c_2, \\ 0.75, & \text{if } (c_1 = \text{Misc} \vee c_2 = \text{Misc}) \wedge c_1 \neq c_2 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Furthermore, we analyze the graph structure of the metadata, specifically we check how publication instances are structurally linked with person instances, e.g. authors. Thus we can compare two publications on the basis of the similarity of the sets of authors using the function for *Set Similarity*, which is useful for example to detect duplicates or to recommend publications that are similar based on co-authorship:

Set Similarity Often it is necessary to compare not only two entities but two sets of entities. As the individual entities have various and very different features, it is difficult to create a vector representing whole sets of individuals. Therefore we use a technique known from statistics as multidimensional scaling [11]. We describe each entity through a vector representing the similarity to any other entity contained in the two sets. This can easily be done, as we rely on other measures which already did the calculation of similarity values $[0..1]$ between single entities. For both sets a representative vector can now be created by calculating an average vector over all individuals. Finally we determine the cosine between the two set vectors through the scalar product as the similarity value.

$$sim_{set}(E, F) := \frac{\sum_{e \in E} \boldsymbol{e}}{|E|} \cdot \frac{\sum_{f \in F} \boldsymbol{f}}{|F|} \quad (4)$$

with entity set $E = \{e_1, e_2, \ldots\}$, $\boldsymbol{e} = (sim(e, e_1), sim(e, e_2), \ldots sim(e, f_1), \ldots)$; F and f are defined analogously.

Context Layer On the context layer, we exploit information about the usage of the bibliographic metadata. The usage information includes recently relevant results (i.e. publications that have for example been stored by the user into his local knowledge base), and queries the user has performed. As the recently relevant results are bibliographic metadata themselves, they can be directly compared. For the recent queries, the situation is similar: A query can be represented as an underspecified publication with the attribute-value pairs that have been specified in the query (As known from *Query-by-Example*). With the similarity function one can then determine, how closely a publication matches a query, instead of only considering exact matches. The context layer is of big value for the recommender functionality.

Amalgamation Function To combine the local similarities to the global similarity Sim, we use a weighted average by assigning weights w_i to all involved local similarities:

$$\mathcal{A}(sim_1, \ldots, sim_n) = \frac{\sum_{i=1}^{n} w_i \cdot sim_i}{\sum_{i=1}^{n} w_i}$$

The weighted average allows a very flexible definition of what similar means in a certain context. For example, to detect duplicate publications, the similarity based on the title has a high weighted, and the global similarity needs to be close to 1. For the recommendation of potentially relevant publications on the other hand, one might set the weights to consider similarity based on the co-authorship or the topic classification. Additionally, one certainly does not want to recommend duplicate publications.

5 Related Work and Conclusion

5.1 Related Work

Similarity measures for ontological structures have been widely researched, e.g. in cognitive science, databases, software engineering and AI. Though this research covers many areas and application possibilities, most applications have restricted their attention to the determination of the similarity of the lexicon, concepts, and relations within one ontology.

The nearest to our comparison between two ontologies come [12] and [13]. In [12] the attention is restricted to the conceptual comparison level. In contrast to our work the new concept is described in terms of the existing ontology. Furthermore, he does not distinguish relations into taxonomic relations and other ones, thus ignoring the semantics of inheritance. [13] computes description compatibility in order to answer queries that are formulated with a conceptual structure that is different from the one of the information system. In contrast to our approach their measures depend to a very large extent on a shared ontology that mediates between locally extended ontologies. Their algorithm also seems less suited to evaluate similarities of sets of lexical entries, taxonomies, and other relations.

MAFRA[14] describes a framework for mapping ontologies. Detecting similarities among entities constitutes one module in the mapping framework. In this sense, our framework can be seen as a complementary to MAFRA.

Research in the area of database schema integration has been carried out since the beginning of the 1980s. Schema comparison analyzes and compares schema in order to determine correspondences and comes therefore near to our approach. The most relevant to our framework is the classification of schema matching approaches given in [15]. The authors distinguish three levels of abstraction. The highest level differs between schemata- and instance-based information. The second level distinguishes the similarity among elements and among structures. On the third level the calculation can be based on linguistic or information about a model's constraints. On the other hand our approach uses a conceptual decomposition: if the similarity of entities can be discovered on the data representation level (e.g. two strings are similar), then it can be expanded to the semantic level (e.g. if these strings are label for two concepts, then it can be an evidence that the concepts are similar) and finally this information can be propagated on the level of the usage of these concepts (e.g. if they are used similarly, then there is more evidence for their similarity). In that context our framework is more "compact" and goal-oriented, whereas all methods mentioned in [15] can be found in our framework. Moreover, we use background information about the given domain and not only “auxiliary” linguistic information (like synonyms, hypernyms) in all layers. Further, we base our framework on a formal ontology model, that enables us to define all methods formally. Finally, none of the related approaches, as known to the authors, had the intention to define a framework which will drive the comparison between ontologies, e.g. none of them consider the context layer as a source for discovering similarities.

5.2 Conclusion

In this paper we present a general framework for calculating similarity among ontologies for various application fields. In order to define a general framework we base our work on an abstract ontology model that allows to adhere to various existing and evolving ontology standards. The main characteristics of the framework is its layered structure: we have defined three levels on which the similarity between two entities can be measured: data layer, ontology layer, and context layer, that cope with the data representation, ontology meaning and the usage of these entities, respectively. In that way, different layers consider different aspects of the nature of entities which are combined in the final judging about their similarity. Moreover, in each of the layers corresponding background information (like the list of synonyms of a term) is used in order to define the similarity more efficiently (precisely). Our intention was not only to develop a collection of existing methods for measuring similarity, but rather to define a framework that will enable their efficient systematization regarding the task of comparing ontologies. The framework is complete in the sense of covering the similarity between all elements defined in the abstract ontology model. We developed several new methods in order to achieve such a completeness. Moreover, the framework provides some inter-/intralayer relations between methods, that enables more efficient applications. In a case study regarding searching for bibliographic metadata in a Peer-to-Peer network we showed the advantages of using our approach in knowledge management applictions. Currently, we are evaluating our framework in several new application areas. The future work will be oriented to the more formal treatment of the context layer which will enable the reasoning about the similarity of the contexts as well.

5.3 Acknowledgements

Research reported in this paper has been partially financed by the EU in the IST projects SWAP (IST-2001-34103), SEKT (IST-2003-506826), and Dot.Kom (IST-2001-34038), and the BMBF project SemiPort (08105939). Many thanks also to our colleagues for the fruitful discussions, especially Philipp Cimiano and Thomas Gabel, who helped to create the first draft.

References

1. Gruber, T.R.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In Guarino, N., Poli, R., eds.: Formal Ontology in Conceptual Analysis and Knowledge Representation, Deventer, The Netherlands, Kluwer Academic Publishers (1993)

2. Stumme, G., Ehrig, M., Handschuh, S., Hotho, A., Maedche, A., Motik, B., Oberle, D., Schmitz, C., Staab, S., Stojanovic, L., Stojanovic, N., Studer, R., Sure, Y., Volz, R., Zacharias, V.: The Karlsruhe view on ontologies. Technical report, University of Karlsruhe, Institute AIFB (2003)
3. Richter, M.M.: Classification and learning of similarity measures. Technical Report SR-92-18 (1992)
4. Maedche, A., Staab, S.: Measuring similarity between ontologies. In: Proceedings of the European Conference on Knowledge Acquisition and Management (EKAW), Springer LNCS (2002)
5. Hefke, M.: A framework for the successful introduction of km using cbr and semantic web technologies. In: Proceedings of the 4th International Conference on Knowledge Management (I-KNOW'04), Graz, Austria (2004) 731–739
6. N. Stojanovic, L.S., Gonzalez, J.: More efficient searching in a knowledge portal - an approach based on the analysis of users queries. In: PAKM 2002, Springer Verlag (2002)
7. Handschuh, S., Staab, S., Maedche, A.: CREAM - creating relational metadata with a component-based. In: Proceedings of the First International Conference on Knowledge Capture K-CAP 2001. (2001)
8. Haase, P., Siebes, R., Harmelen, F.: Peer selection in peer-to-peer networks with semantic topologies. In: International Conference on Semantics of a Networked World: Semantics for Grid Databases, 2004, Paris. (2004)
9. Levenshtein, I.V.: Binary codes capable of correcting deletions, insertions, and reversals. Cybernetics and Control Theory (1966)
10. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. In: IEEE Transactions on Systems, Man and Cybernetics. (1989) 17–30
11. Cox, T., Cox, M.: Multidimensional Scaling. Chapman and Hall (1994)
12. Bisson, M.: Learning in FOL with a similarity measure. In: Proceedings of the Tenth National Conference on Artificial Intelligence. (1992) 8287
13. Weinstein, P., Birmingham, W.P.: Comparing concepts in differentiated ontologies. In: Proceedings of the Twelfth Workshop on Knowledge Acquistion, Modeling and Management (KAW'99), Banff, Alberta, Canada (1999)
14. Maedche, A., Motik, B., Silva, N., Volz, R.: Mafra - a mapping framework for distributed ontologies. In: Proceedings of the EKAW 2002. (2002)
15. Rahm, E., Bernstein, P.: survey of approaches to automatic schema matching. VLDB Journal **10** (2001) 334350