An information retrieval approach to ontology mapping

Xiaomeng Su [*,1], Jon Atle Gulla

Department of Computer and Information Science, IDI, Norwegian University of Science and Technology (NTNU), N-7491, Trondheim, Norway

Abstract

In this paper, we present a heuristic mapping method and a prototype mapping system that support the process of semi-automatic ontology mapping for the purpose of improving semantic interoperability in heterogeneous systems. The approach is based on the idea of semantic enrichment, i.e. using instance information of the ontology to enrich the original ontologies and calculate similarities between elements in two ontologies. The functional settings for the mapping system are discussed and the evaluation of the prototype implementation of the approach is reported.

Key words: Ontology Mapping, Semantic Heterogeneity, Linguistic Analysis

1 Introduction

System interoperability is an important issue, widely recognized in information technology intensive enterprises and in the research community of information systems (IS). The widely adoption of the World Wide Web to access and distribute informations further stress the need for systems interoperability. The current World Wide Web has well over 4.2 billion pages [15], but the vast majority of them are in human readable format only. In order to allow software agents to understand and process the web information in a more intelligent way, researchers have created the Semantic Web vision [7], where data has structure and ontologies describe the semantics of the data.

[*] Corresponding author. Tel.: +47 7359 3677
Email addresses: xiaomeng@idi.ntnu.no (Xiaomeng Su), jag@idi.ntnu.no (Jon Atle Gulla).

[1] Part of the research has been supported by Accenture, Norway.

The Semantic Web offers a compelling vision, yet it also raises many difficult challenges. One of the key challenges is to find semantic correspondences between ontologies. Ontology is a key factor for enabling interoperability in the Semantic Web [7]. However, it has long been argued that there is no one single universal shared ontology, which will be applauded by all players. It seems clear that ontologies face the same or even harder problems with respect to heterogeneity as any other piece of information [29]. The attempts to improve system interoperability will therefore rely on the reconciliation of different ontologies used in different systems. The reconciliation is often approached by manual or semi-automated integration of ontologies. The technical issue here is to help resolving ontology mismatches that evidently appear in semantic integration.

One of the fundamental elements of the ontology integration process is to establish mappings between ontologies. Mapping processes typically involve analyzing the ontologies and comparing them to determine the correspondences among concepts and detect possible conflicts. Sets of mapping assertions are the main output of mapping processes. The mapping assertions can be used directly in a translator component, which translates statements that are formulated by different ontologies. Alternatively, a follow-up integration process can use the mappings to detect merging points.

So, interoperability among applications in heterogeneous systems depends critically on the ability to map between their corresponding ontologies. Today, mapping between ontologies is still largely done by hand, in a labor-intensive and error-prone process [14]. In this paper, we present a heuristic mapping method and a prototype mapping system that support the process of semi-automatic ontology mapping for the purpose of improving semantic interoperability in heterogeneous systems. The approach is based on the idea of semantic enrichment, i.e. using instance information of the ontologies to enrich the original ontologies and calculate similarities between elements in two ontologies.

2 Overview of the approach

The word ontology has been used to describe artifacts with different degrees of structure. These range from simple taxonomies (such as the Yahoo! hierarchy), to metadata schemes (such as the Dublin Core [12]), to logical theories. In our context, the scope and assumption of our work are the following:

(1) An *ontology* specifies a conceptualization of a domain in terms of concepts, attributes and relations. Concepts are typically organized into a tree structure based on subsumption relationship among concepts. Ad hoc

relations further connect concepts and formulate a semantic net structure in the end.

(2) In this paper, we focus on finding mappings between concepts and between relations. This is because they are central components of ontologies and matching them successfully would aid in matching the rest of the ontologies.

(3) Ontologies can be expressed in different representational languages [28]. Here, we assume that it is possible to translate between different formats. In practice, a particular representation must be chosen for the input ontologies. Our approach is based on Referent Modeling Language (RML)[24][26], which is an Extended ER-like (Entity Relationship) graphic language with strong abstraction mechanism and sound formal basis.

The overall process of ontology mapping can then be defined as: given two ontologies O_a and O_b, mapping one ontology with another means that for each element in ontology O_a , find corresponding element(s), which has same or similar semantics, in ontology O_b and vice verse. We first define exactly what our mapping model is:

Definition 1 (ontology mapping model) *An ontology mapping model is a 5-tuple [S, T, $\mathcal{F}$, R(s_i, t_j), $\mathcal{A}$)] where*

- *S is a set composed of logical views (representation) for the elements in source ontology.*
- *T is a set composed of logical views (representation) for the elements in target ontology.*
- *$\mathcal{F}$ is a framework for representing ontology elements and calculating relationships between elements in the two ontologies.*
- *R(s_i, t_j) is a ranking function which associate a real number with an element $s_i \in S$ and an element $t_j \in T$. Such ranking defines an order among the elements in source ontology with regard to one element t_j in the target ontology.*
- *A is a set composed of mapping assertions. A mapping assertions is a formal description of the mapping result, which supports further description of the exact nature of the derived mappings. It has the following components:*
  - *· a pair of ontology elements,*
  - *· a type of correspondence,*
  - *· a degree of correspondence, and*
  - *· a set of sources of assertion.*

In other words, we can define the mapping process as : $S, T \xrightarrow{\mathcal{F},R} A$, i.e. given the ontologies S, T as input, using the framework and the ranking function to produce A as output. The rest of the paper will discuss the process and each of the individual component in greater detail.

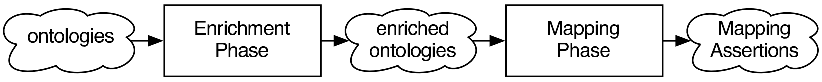


Fig. 1. Two phases of the whole mapping process.

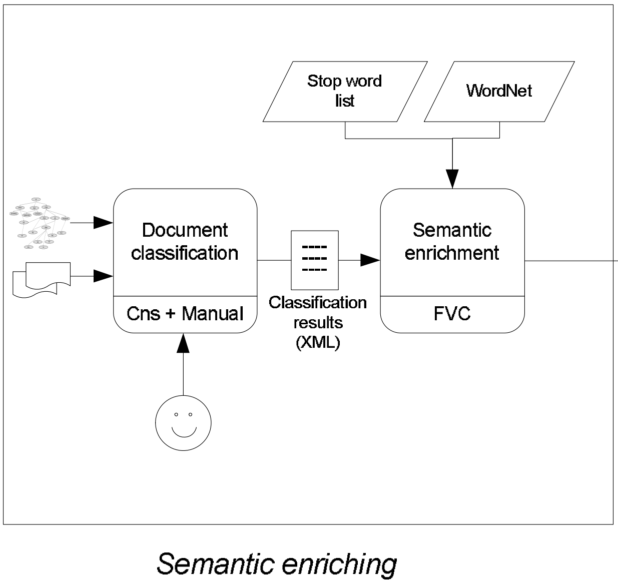


Fig. 2. Overview of the semantic enrichment process.

As shown in figure 1, the approach consists of two phases: enrichment phase and mapping phase. The enrichment phase is based on analysis of the extension information the ontologies have. The extension we make use of in this work is written documents that are associated with the concepts in the ontologies. The intuition is that given two to-be-compared ontologies, we construct representative feature vectors for each concept in the two ontologies. The documents are "building material" for the construction process, as they reflect the common understanding of the domain. Outputs of the enrichment phase are ontologies with feature vector as enrichment structure. The mapping phase takes the enriched ontology and computes similarity pair wise for the elements in the two ontologies. We will discuss the two phases in turn.

3 Semantic enrichment of ontology

The main task in the semantic enrichment phase is to generate the enrichment structures, namely, the representative feature vectors. Figure 2 shows the two steps performed in the semantic enrichment process. The algorithm takes the two to-be mapped ontologies in RML format, together with document sets, as input. There can be one or two document sets. In the former case, we assume the documents are relevant to both ontologies, while in the latter, it is assumed that the two document sets share the same vocabulary.

3.0.1 Document assignment

Document assignment step aims to automatically assign documents to one or more predefined categories based on their contents. We use a linguistically based classifier CnS (Classification and Search) [8] to associate documents with the ontologies. Multiple association is allowed. This is a semi-automatic process, where users need to manually adjust the assignment results to guarantee the correct assignments.

The assigning of documents to concepts is necessary when no instance knowledge of the ontology is available. However, if documents have already been assigned to specific concepts, we can skip the first step and construct feature vector for the concepts directly.

3.0.2 Feature vector construction

The second step concerns building up feature vectors for each concepts in the two ontologies. The intuition is that for each concept a feature vector can be calculated based on the documents assigned to it. Following a classic Rocchio algorithm [1], the feature vector for concept a_i is computed as the average vector over all document vectors that belong to concept a_i. The outputs of this step are two intermediate ontologies, $O_A^{'}$ and $O_B^{'}$, where each concept has been associated with a feature vector.

Three sub-steps constitute the process. The first two steps aim at building document vectors, while the third step use the document vectors to build feature vectors for concepts.

(1) *Pre-processing.*The first step is to transform documents, which typically are strings of characters, into a representation suitable for the task. The text transformation is of the following kind: remove HTML (or other) tags; remove stop words; perform word lemmatization. WordNet is used to perform the necessary linguistic transformation.
(2) *Document representation.*We use the vector space model [25] to construct the generalization of the documents. In vector space model, documents are represented by vectors of words. There are several ways to determining the weight of word i in a document d. We use the standard tf/idf weighting [25], which assigns the weight to word i in document d in proportion to the number of occurrences of word in the document, and inverse proportion to the number of documents in the collection for which the word occurs. Thus, for each document d in a document collection D, a weighted vector is constructed as follows:

$$\vec{d} = (w_1, \ldots, w_n) \tag{1}$$

where w_i is is the weight of word i in document d.

$$w_i = f_i * \log (N/n_i) \tag{2}$$

where f_i is the frequency of word i in document d, N is the number of documents in the collection D and n_i is the number of documents that contains word i.

(3) *Concept vector construction.*We differentiate here leaf concept and non-leaf concept in the ontology. Leaf concepts are those which have no sub-concepts.

- For each leaf concept, the feature vector is calculated as an average vector on the documents vectors that have already been assigned to this concept. Let C^k be the feature vector for concept concept K and let D_j be the collection of documents that have been assigned to that concept K. Then for each feature i of the concept concept vector, it is calculated as:

$$C_i^k = \frac{\sum\limits_{D_j \in K} w_{ij}}{|D_j|} \tag{3}$$

- When it comes to non-leaf concepts, the feature vector C^k for a non-leaf concept K is calculated by taking into consideration contributions from the documents that have been assigned to it, its direct sub concepts and related concepts[2]. Let D_j be the collection of documents that have been assigned to that concept K, let S_t be the collection of its direct sub concepts and let S_r be the collection of its related concepts. The ith element of C^k is defined as:

$$C_i^k = \alpha * \frac{\sum\limits_{D_j \in K} w_{ij}}{|D_j|} + \beta * \frac{\sum\limits_{S_t \in K} w_{it}}{|S_t|} + \gamma * \frac{\sum\limits_{S_r \in K} w_{ir}}{|S_r|} \tag{4}$$

where $\alpha + \beta + \gamma = 1$. α ,β and γ are used as tuning parameters to control the contributions from the concepts instances, sub concepts, and related concepts respectively.

4 Mapping

The algorithm takes as input semantically enriched elements of two ontologies and produces as output suggestions to the user for possible correspondences. As figure 3 illustrates, the algorithm has the following five main components.

[2] At this point, all ad hoc relations other than subsumption are treated as related-to.

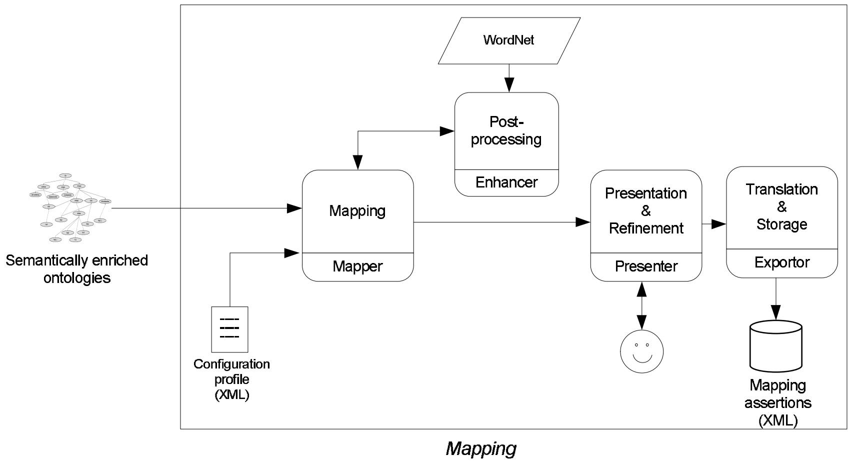


Fig. 3. Major steps in the mapping phase.

- The *mapper* performs a computation of correspondence measures for the pairs of compared ontology elements, based on the similarity of their enriched structures.
- The *enhancer* utilizes an electronic lexicon to adjust the similarity values that have been computed by the *mapper*, with the intention of re-ranking the mapping assertions in the result list.
- The *presenter* determines which recommendations to suggest to the user, based on the partial ordering of correspondence measures and the current configuration profile.
- The *exporter* translates and exports the mapping results to a desired format so that other follow-up applications can import and use the results in a loosely coupled way.
- The *configuration profile* is a user profile to assign individual variable values for different tuning parameters and a threshold value for exclusion of mappings with low similarity.

4.1 The similarity calculation for concepts

To find concept pairs that are similar, we calculate a similarity value for concepts pairwise in the two ontologies. A threshold value is defined by the user to exclude pairs that have too low similarity values. The calculation of concept similarity is the foundation for similarity calculation of other more complex elements.

The similarity of two concepts in two ontologies is directly calculated as the cosine measure between the two representative feature vectors. Let two feature vectors for concept a and b respectively, both of length n, be given. The cosine similarity between concept a and concept b is defined as:

$$sim(a,b) = sim(C^a, C^b) = \frac{\vec{C^a} * \vec{C^b}}{|C^a| * |C^b|} = \frac{\sum_{i=1}^{n}(C_i^a * C_i^b)}{\sqrt{\sum_{i=1}^{n}(C_i^a)^2} * \sqrt{\sum_{i=1}^{n}(C_i^b)^2}} \quad (5)$$

where

- C^a and C^b are feature vectors for concept a and b, respectively
- n is the dimension of the feature vectors
- $|C^a|$ and $\left|C^b\right|$ are the lengths of the two vectors, respectively

For concept a in ontology A, to find the most related concept b in ontology B, the top k ranked concepts in ontology B are selected according to the initial similarity measure calculated above.

4.2 Adjust similarity value with WordNet

Given the central position of concept similarity calculation, it is desirable to make the suggestions as accurate as possible. This requires additional technique to adjust the similarity values. We use WordNet for that purpose [20]. We integrate the Java WordNet Library (JWNL) [9], a Java API, into the system for accessing the WordNet relational dictionary and calculate semantic relatedness based on the path length measurement described below. The computed relatedness will be amplified by a tuning parameter and then will be added to the similarity values computed in the previous step. The changing on similarity values will change the ranks of the involved mappings. The intention is to strengthen the prominent ones by the post processing procedure and rank them high in the results.

One way to measure the semantic similarity between two words a and b is to measure the distance between them in WordNet. This can be done by finding the paths from each sense of a to each sense of b and then selecting the shortest such path. Note that path length is measured in nodes rather than links. So the length between sister nodes is 3. The length of the path between member of the same synset is 1. We did not make any effort in joining together the 11 different top nodes of the noun taxonomy. As a consequence, a path cannot always be found between two nouns. When that happens, the program returns a "not related" message. We refer this measurement as the path length measurement. The path length measurement gives us a simple way to calculate relatedness between two words. However, there are still a few issues we need to address before we can use it.

- *Word form.* When looking up a word in WordNet, the word is first lemmatized. So the distance between *system* and *systems* is 0.
- *Multiple part-of-speech.* The path length measurement can only compare words that have the same part-of-speech. The words we compare in this context are concept names in the ontologies. Even though most of the names are made of single noun or noun phrase, verbs and adjectives do occasionally appear in a concept name label. In some cases, one word has more than one part-of-speech (for instance, ”backpacking” is both a noun and a verb in WordNet). For these words, we first check if it is a noun and if the answer is yes, we treat it as a noun. In the case of ”backpacking”, for instance, it will be treated as a noun and its verb sense will be disregarded. If it is not a noun, we check if the word is a verb and if the answer is yes, we treat it as a verb. Words that are neither nouns, verbs nor adjectives will be disregarded. This makes sense since the different part-of-speech of the same word are usually quite related and choosing one of them would be representative enough.
- *Compound nouns.* Compound nouns which have an entry in WordNet (for example, ”jet lag”, ” travel agent” and ”bed and breakfast”) are treated as single word. Others like ”railroad transportation” , which have no entry in WordNet, are split into tokens (”railroad” and ”transportation” in the example) and their relatedness to other word is calculated as an average over the relatedness between each token and the other word. For instance, the relatedness between ”railroad transportation” and ”train” will be the average relatedness of ”railroad” with ”train” and ”transportation” with ”train”.

4.3 The similarity calculation for complex elements

Based on the correspondences calculated for the concepts, we could further expand the correspondence discovery into other elements and structures in the ontologies. In this part, we introduce how similarity between relations and between clusters of concepts are defined.

Relations. The similarity of relations is calculated based on the corresponding domain concepts and range concepts of the relations. Precisely, the similarity between relation $R(X, Y)$ and $R'(X', Y')$ is defined as the arithmetic mean value of the similarity values their domain and range concepts have:

$$sim(R, R') = \frac{(sim(X, X') + sim(Y, Y'))}{2} \tag{6}$$

where

- X and X' are domain concepts of R and R', respectively
- Y and Y' are the range concepts of R and R' respectively
- $sim(X, X')$ and $sim(Y, Y')$ are calculated by equation 5 for concepts similarity.

Clusters. We define the concept of *cluster*. A cluster is a group of related concepts, which includes a center concept a and its k-nearest neighbors. A cluster of 1-nearest neighbor includes a center concept and its direct parent, its direct children. A cluster of 2-nearest neighbor includes the grandparent, the siblings and the grandchildren, in addition to the 1-nearest neighbor. The correspondences between clusters in two ontologies reveal "areas" that are likely to be similar. This helps knowledge workers to locate and concentrate on a bigger granularity level.

The similarity of clusters is calculated based on the weighted percentage of established mappings between member concepts in proportion to the number of all connections between the two clusters. We define the equation of calculating two cluster similarity as:

$$sim(X, Y) = \frac{\sum_{(a_i,b_j)\in M} sim(a_i, b_j)}{|X| * |Y|} \tag{7}$$

where

- X and Y are clusters of k-nearest neighbor. $X = \{a_1, a_2, a_3, \cdots, a_n\}$ and $Y = \{b_1, b_2, b_3, \cdots, b_m\}$.
- M is a subset of the cartesian product of X and Y, where $M \subseteq X * Y, M = \{(a_i, b_j)|(a_i \in X) \cap (b_j \in Y) \cap (sim(a_i, b_j) > 0)\}$
- $|X|$ and $|Y|$ are number of elements in the two sets, respectively.
- the $sim(a_i, b_j)$ is calculated by equation 5 for concepts similarity.

Ontologies. Extending the idea of cluster similarity one step further, we come to the point where the similarity between two ontologies can be quantified as the weighted percentage of established mappings in proportion to all the connections between concepts in the two ontologies, as defined in the following equation.

$$sim(O_1, O_2) = \frac{\sum_{(a_i,b_j)\in M} sim(a_i, b_j)}{|O_1| * |O_2|} \tag{8}$$

where

- O_1 and O_2 are two ontologies. $O_1 = \{a_1, a_2, a_3, \cdots, a_n\}$ and $O_2 = \{b_1, b_2, b_3, \cdots, b_m\}$. a_i (i=1...n) are the concepts in O_1 and b_j (j=1...m) are the concepts in O_2.

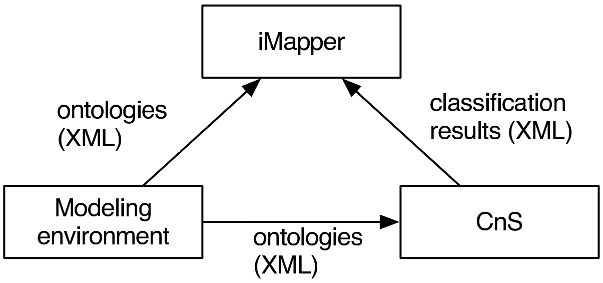


Fig. 4. Components of the system.

- M is a subset of the cartesian product of O_1 and O_2, where $M \subseteq O_1 * O_2, M = \{(a_i, b_j) | (a_i \in O_1) \cap (b_j \in O_2) \cap (sim(a_i, b_j) > 0)\}$
- $|O_1|$ and $|O_2|$ are number of concepts in the two ontologies, respectively.
- the $sim(a_i, b_j)$ is calculated by equation 5 for concepts similarity.

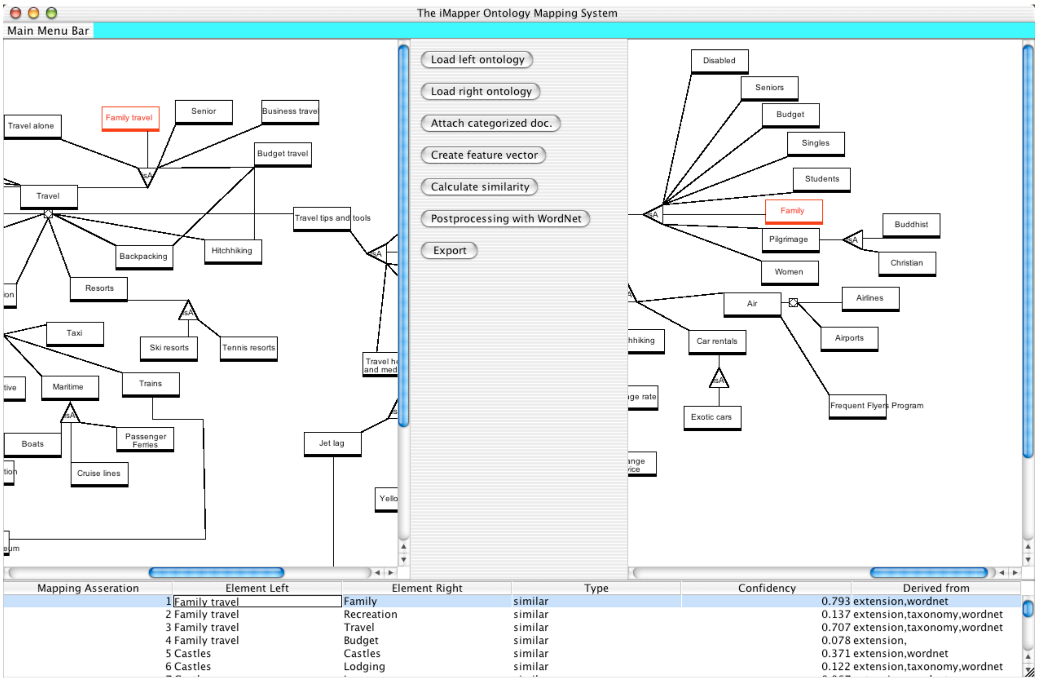
Such a value is useful when several ontologies in a domain need to be merged. This value can help to reveal the most similar two, which constitute good initial candidates for the merging process.

5 Implementation

A prototype has been implemented to verify that the proposed approach is an applicable solution. The system architecture is composed of three separately developed parts that communicate using XML. Figure 4 illustrates the three parts and how they interact with each other. The modeling environment is responsible for constructing/importing the ontologies in RML format. The ontologies are passed on to CnS for assigning documents to the relevant ontology elements. The ontologies, together with the classification results stored in XML, are delivered to the iMapper system for the mapping process.

Figure 5 shows the GUI of the iMapper system. The figure illustrates a mapping process and the obtained mapping assertions. Both ontologies are represented visually in RML (Referent Modelling Language). The CnS software performs the document assignment process and the result is being stored in an XML file. This information is then loaded into the mapping system. Next, the mapping system constructs the feature vector for each document and the concept consequently. Next, the mapper calculates the similarity measures pair wise for the elements in the two ontologies based on the vectors. The post processing step uses WordNet to re-rank the list. As a result, a list of top ranked mapping assertion will be generated and listed in the table at the lower part of the frame (see figure 5).

An ID uniquely identifies each mapping assertion. It involves two concepts, one from the ontology in the left, the other from the right. The fourth column



Mapping Asseration	Element Left	Element Right	Type	Confidency	Derived from
1	Family travel	Family	similar	0.793	extension,wordnet
2	Family travel	Recreation	similar	0.137	extension,taxonomy,wordnet
3	Family travel	Travel	similar	0.707	extension,taxonomy,wordnet
4	Family travel	Budget	similar	0.078	extension,
5	Castles	Castles	similar	0.371	extension,wordnet
6	Castles	Lodging	similar	0.122	extension,taxonomy,wordnet

Fig. 5. Snapshot of the iMapper system.

of the table describes what kind of mapping relation holds between these two. A confidence level is given to indicate the probability that this prediction is true. An explanation about the source of the mapping assertion is given in the last column. For example, in figure 5, mapping assertion 1 states that concept *family travel* in the left ontology is most *similar* to concept *family* in the right ontology with *a similarity value* of *0.793* and this mapping is derived mainly from both *extension analysis* and *WordNet relatedness calculation*.

When the user selects on one mapping assertion (by clicking the row of that assertion in the mapping table), the corresponding concepts in the two ontologies are highlighted, making it possible for the user to get a clear overview of the relevant locations of the concepts in the ontologies. It is also possible for the user to edit, delete or add mapping assertions. The approach discovers a number of useful mappings. Although the results still need to be manually approved or adjusted, it is easier for the user to approve than to search relevant pairs from scratch.

6 Evaluation

A comprehensive evaluation of the mapping approach supported by the iMapper system has been performed on two domains. The main goal was to evaluate the matching accuracy of iMapper, to measure the relative contributions from the different components of the system, and to verify that iMapper can contribute in helping the user in performing the labor intensive mapping task.

6.1 Experiment design

Performance criteria. To evaluate the quality of the match operations, we compare the match result returned by the automatic matching process (P) with manually determined match result (R). We determine the true positives, i.e. correctly identified matches (I). Based on the cardinalities of these sets, the following quality measures are computed.

Precision = $|I|/|P|$, is the faction of the automatic discovered mapping which is correct. It estimates the reliability for the match prediction.

Recall = $|I|/|R|$, is the the fraction of the correct matches (the set R) which has been discovered by the mapping process. It specifies the share of real match that is found.

Precision and recall have been used extensively to evaluate the retrieval performance of retrieval algorithm in the information retrieval field [4] and have also been used in other studies [10][14]. For each mapping the system predicated, there is a similarity degree associated with it. The degree indicates the confidence of the predication. It also provides a practical way to rank the mappings. As the mappings are ranked in a descending order of the degree, we could calculate precision at different recall levels by gradually adding more mappings into consideration. We plot the precision versus recall curve at 11 standard recall levels[4]. Precision versus recall figures are useful because they allow us to evaluate quantitatively both the quality of the overall mapping collection and the breadth of the mapping algorithm.

For this version of the experiment, we made evaluation only on concept-concept mappings, whereas the more complex mappings between relations, clusters, etc. are not in focus. We did that because concept-concept mappings are the bases for any other more complex mappings and ensuring its high quality will form a sound base for the other type of mappings. Also when user manual work is involved, we have to carefully limited the scope and complexity of the task. Therefore, the more complex mappings are omitted in this version of the evaluation.

Domains and source ontologies. We evaluated iMapper on two domains. The product catalogue integration task was first introduced in [13], where consumers and vendors may use different classification schema (UNSPSC, UCEC, and eCl@ss, to name a few) to identify their requests or products. Links between different classification schema need to be defined in order to relate the corresponding concepts. In our experiment, the two relevant product catalogues are the United Nations Standard Products and Services Code (UNSPSC) [3] and the Standardized Material and Service Classificatione – eCl@ss [4] .

For our current experiment, two small segments of the relevant catalogues, both of which concern the domain of computer and telecommunication equipments, are selected. They contain 23 - 26 concepts (corresponding to the categories) and are organized in 3 - 4 levels by generalization relationship. Two datasets of product descriptions collected from online computer vendor websites are classified according to UNSPSC and eCl@ss. The classification is performed in two steps. First is the automatic classification by CnS client, then come human adjustments of the automatic results. The classified product description are viewed as the instances of the relevant concept.

The second domain we choose is the tourism section. The two ontologies are constructed based on vocabularies and structures from relevant travel sections of the Open Directory Project (ODP) [5] and Yahoo! Category [6] . In both ODP and Yahoo!, categories are organized in hierarchy augmented with related-to links.The Open Directory Project aims to build the largest human-edited directory of Internet resources and is maintained by community editors who evaluate sites for classify them in the right directory. Yahoo! category is maintained by the Yahoo! directory team for the inclusion of web sites into Yahoo! directory. As a result, in this domain, unlike the product catalogue example above, instances of each concept are already directly available without the need to classify them.

Experiment setup. For the manual part, we conducted a user study in the Information System Group at the Norwegian University of Science and Technology. 6 users have conducted the manual mapping independently. All of them have good knowledge of modeling in general. None of the users had addressed the problem of mapping ontologies prior to the experiment. For each of the two mapping tasks, each participant received a package containing: a diagrammatic representation of the two ontologies to be matched, a brief instruction of the mapping task, and a scoring sheet to fill in the user identified mappings. To help the user making decision, they can use numbers to indicate how confident they are towards each match (3 for fairly confident, 2 for likely and 1 for need to know more to suggest the match). The numbers also help to compare system performance when different confidence level are considered

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

[4] http://www.eclass-online.com/

[5] http://dmoz.org/

[6] htto://www.yahoo.com

6.2 Variables

The primary goal of our experiment is to evaluate the quality of iMapper's suggestions and examine the contribution from different component of the system. We also aim at testing the robustness of the approach by a series of sensitivity analysis. A number of variables affect the results. They are subjected to the sensitivity test.

- *Desired Mapping Results.* Both precision and recall are relative measures that depend on the *desired mapping results* [7] – the user identified mappings. For a meaningful assessment of mapping quality, the desired mapping result must be specified precisely. In this experiment, we have two versions of the desired mapping results. One is developed by 6 users independently, the other is based on group discussion. The intention is to test if different user efforts will lead to different mapping results and to what extend will the different desired mapping results affect the final precision and recall values. Another variable in the gold standard is related to the fact that we allow users to specify a confidence level to each mapping they suggest. 3 for fairly confident, 2 for likely and 1 for need to know more to suggest the match. Therefore, two variables are relevant here:
  · DESIRED_MAPPING_RESULT to indicate wether the gold standard is individual or group discussion based.
  · CONFIDENCE_LEVEL to specify whether only confident mappings are included into the gold standard (when set confidence level to 3) or less confident ones are included as well (when confidence level is 2 or 1) [8] .
- *Structural Information.* Recall in equation 4, for non-leaf concepts, contributions from the instances, the sub concepts and the related concepts are counted in when calculating feature vectors for such non-leaf concepts. α ,β and γ are used as tuning parameters to control the contributions from the concepts instances, sub concepts, and related concepts respectively. For instance, if we assign 1 to α, 0 to β and γ , it means that no structure information will be considered.
- *WordNet Contribution.* The contribution from WordNet postprocessing will be adjusted by a tuning parameter – $RELATEDNESS\_WEIGHT$. If $RELATEDNESS\_WEIGHT = 0$, it means WordNet contributions are not considered.

[7] Also being referred to as ”gold standard”.

[8] confidence level 2 includes all the mapping that have a confidence level equal or bigger than 2, and confidence level 1 includes those that are equal or bigger than 1.

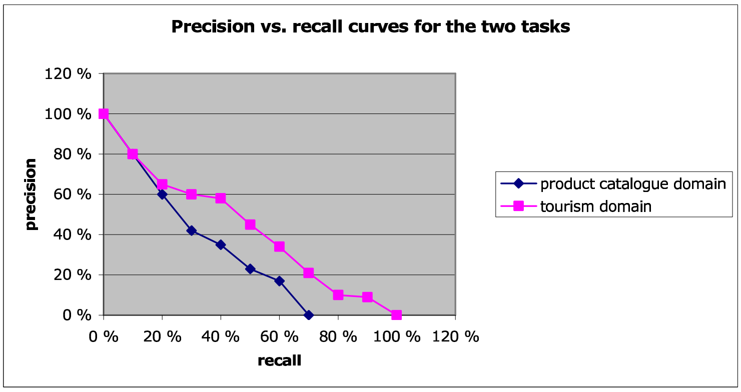


Fig. 6. Precision versus recall curve for the two tasks.

6.3 Quality of iMapper's predictions

6.3.1 Baseline analysis

To compare the situation in different configurations, we need a baseline configuration of the variables. For both domains, we set the different variable values as follows:

- α=0.5 β=0.25 γ=0.25
- $RELATEDNESS\_WEIGHT = 0$
- $CONFIDENCE\_LEVEL = 1$
- $DESIRED\_MAPPING\_RESULT = individual$

For the two tasks, a number of mappings were identified manually by the users. Overall, an average of 30 mappings are discovered by the users between the two product catalogues and an average of 62 in the tourism domain. The manual mappings are determined to be "correct" and are used as a gold standard to evaluate the quality of the automatically suggested mappings in the baseline version.

Figure 6 summarizes average precision versus recall figures for the two mapping tasks respectively. For the tourism ontology mapping task, the precision is 93% at recall level 10% and drops gradually when more mappings are included. For the product catalogue task, the precision at levels of recall higher than 70% drops to 0 because in the baseline version, not all user identified mappings in this task can be discovered by the iMapper system automatically. In that particular case, it is 69%. For the tourism domain, 92% are discovered.

The tourism ontology mapping task achieved higher precision than the product catalogue task at all recall levels. There are several possible explanations for the difference. First, the number of instances of the product catalogues is smaller than that of the tourism ontologies. The feature vectors generated

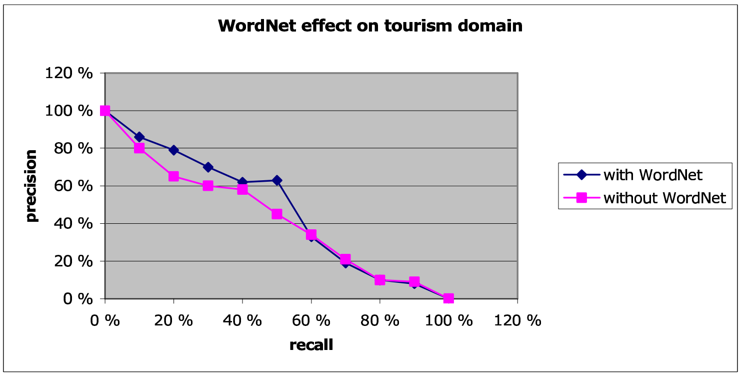


Fig. 7. Precision versus recall curves pre and after using WordNet for postprocessing in tourism domain.

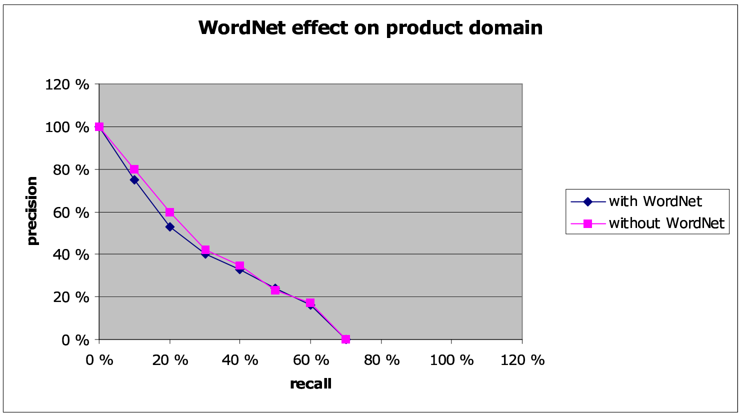


Fig. 8. Precision versus recall curves pre and after using WordNet for postprocessing in product catalogue domain.

by a larger instance set will have a better chance to capture and condense the terms that differentiate one concept from others. More accurate feature vectors will in turn boost the accuracy of the mappings. Second, the documents used in the two tasks have different characteristics. In the product domain, there exist a fair amount of technical terms, proper nouns and acronyms (for instance, ”15inch”, ”Thinkpad”, ”LCD”, etc.). Lacking of special means to treat these special terms hampers the system from generating high quality feature vectors. In contrast, in the tourism domain, the documents contain far much less specific technical terms or proper nouns.

6.4 Further experiment

6.4.1 Tuning with WordNet

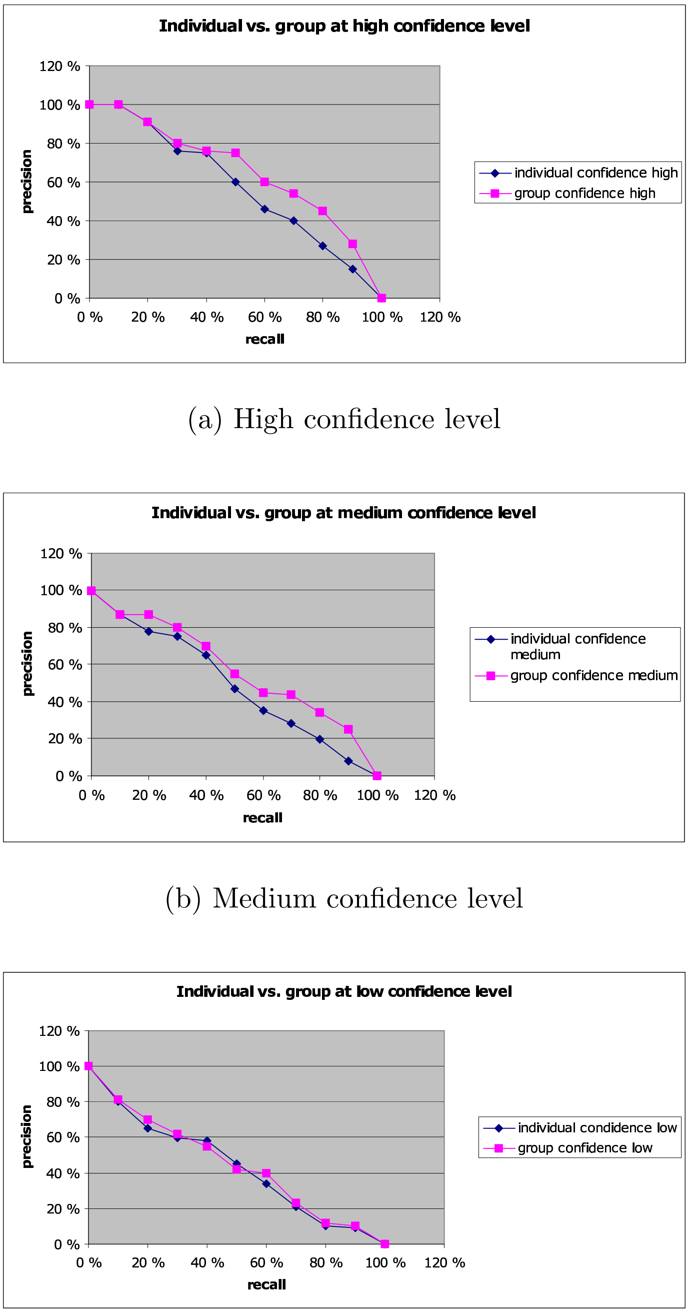
With both domains, we did further experiment on assessing the effect of using WordNet to post-process the system initially generated mappings. WordNet is used to strengthen the mappings whose concept names have a close relatedness in WordNet.

Figure 7 shows the precision and recall curves pre and after using WordNet for post-processing in the tourism ontology mapping task. The figure shows that WordNet marginally improves the precision at levels of recall lower than 60%. This suggests that WordNet is useful in strengthening the similarity value of the correct mappings and boost their ranks in the result set. The changing of ranks makes the predication more accurate at lower recall levels. At recall level 20% and 50%, WordNet makes an apparent improvement for the precision.

Figure 8 demonstrates the precision and recall curves pre and after using WordNet in the product catalogue domain. On the contrary to the tourism domain, at high recall levels, the effect of WordNet here is not apparent and indeed the precision gets worse after using WordNet than that before using it. One possible reason to explain it is that lots of technical terms are used in the product catalogue domain. This technical terms are not documented and classified specifically in accordance to their usage in technical domains. For instance, "*IT*" has no entry in WordNet, in the case of "*CD writable*", *writable* has not entry in WordNet either, and in the case of "*portable*", its only noun sense is related to *"a small light typewriter"* which has not much to do with *"portable computer"*. This plus a relatively small set of concepts lead to the effect that WordNet strengthened the pairs in a more or less random way, which in turn results in slightly worsening the situation.

We also noticed the limitations for using WordNet to calculate concept relatedness in both domains. In WordNet, nouns are grouped into several hierarchies by hyponymy/hypernymy relations, each with a different unique beginner. Topic related semantic relations are absent in WordNet. So *travel agent* and *travel* have no relation between them. In fact, they are in two different taxonomies, since *travel agent* is in the taxonomy which has *entity* as top node and *travel* is in the taxonomy where *act* is the top node. This results in a not-related result being returned when applying the path length measure for measuring the relatedness of the two terms. This result however does not mirror the human judgment. A possible way to overcome this limitation is to augment WordNet with domain specific term relations or use other domain specific lexicon.

In this experiment, we used the path length measurement to estimate the semantic relatedness of terms in WordNet. There are other measures being proposed in the literature, for instance, the most informative class measures[23] or the conceptual density [2]. It would be interesting to see the performance of other alternative measurements.



(a) High confidence level

(b) Medium confidence level

(c) Low confidence level

Fig. 9. Precision recall curves at three confidence levels in tourism domain

6.4.2 Desired mapping results

Figure 9 presents precision recall curves of individual vs. group discussion based gold standard at three confidence levels respectively. At both high and medium confidence levels, the precision is generally higher under group gold standard than that under individual gold standard. It is especially true when the recall level increases. We observed that in the group discussion session, users tend to read the ontologies more carefully. Discussions took place when users had different understanding or interpretations of the concepts. It is reasonable to conclude that users put more efforts in group discussion based sessions than they do in individual based sessions. Some of the suggestions a user made in the individual cases were identified to be false in the group discussion session. One typical scenario is that when one user proposed a mapping based on the fact that the two concepts are synonyms, another user argued that the mapping is not valid, since the concepts actually have different meanings if taking their structures into consideration. The two argues a little bit, while others added their opinions as well. In the end, all agreed that the mapping was not a valid one (or at least one that should not have high confidence). As a result, some of the very obvious mistakes one user made in individual bases vanished through the group discussion phase. It seems that the group discussion based gold standard is a more accurate one.

At low confidence level, we observed no significant difference between the individual and group ones. This may relate to the fact that when users had dispute over a mapping, they quite often made compromises in the end, so that instead of completely delete a proposed mapping or assign a high confidence level to it, they would agree as a middle way to assign a low confidence level to it. Since the low confidence level gold standard includes all the mappings, we end up with more or less similar set of mappings in the individual case and the group case.

6.4.3 Structural information

The last experiment we did was to test whether taking into account structural information makes any differences for the mapping accuracy. A rather straight forward test was conducted on the tourism domain. We tuned the structural parameter β and γ to 0, and compared the precision recall curves in that setting with that in the baseline version. Recall that in the baseline version, $\alpha = 0.5$, $\beta = 0.25$, and $\gamma = 0.25$. In figure 10, the β,$\gamma = 0$ version is referred to as structure-off version, while the baseline is referred to as the structure-on version. As shown in the figure, the structure-off version has a decrease in precision at recall level 20%, recall level 50% and above. This indicates that to disregard structural information completely makes the mapping accuracy worse at high recall levels.

6.5 Discussion

To summarize, the main results of our study were the following:

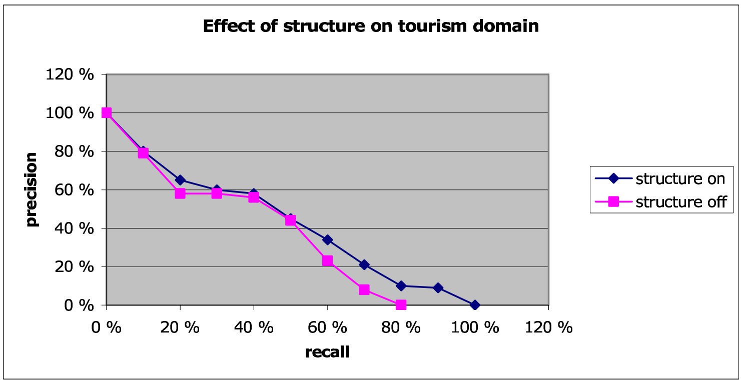


Fig. 10. Precision recall curves when structure information is turned on/off in tourism domain.

- The system discovered most of the mappings and ranked them in a correct way.
- The number of documents, the nature of the terms used in the documents and the overlapping of the ontologies account for difference in mapping accuracy in the two tasks.
- The effect WordNet has on the mapping accuracy through re-ranking varies in terms of the domain and document characteristics.
- The gold standards significantly influence the results. It seems that a group discussion based gold standard has less errors.
- Take into consideration structural information helps improve the mapping accuracy.

7 Related work

There has been a number of works on semantic reconciliation developed in the context of schema translation and integration, knowledge representation, machine learning and information retrieval. In the multi database and information systems area, there exist approaches dealing with database schema integration. In [5], a variety of database schema integration methods were studied and the schema integration process is divided into three major phases: schema comparison, schema conforming and schema merging. DIKE [21], MOMIS [6], OBSERVER [19], and Cupid [16] are systems, which focus on schema matching. In [22], schema matching approaches were classified into schema-level matchers and instance-level matchers . It pointed out that instance-level approaches can be used to enhance schema-level matchers in that evaluating instances reveals a precise characterization of the actual meaning of the schema elements. And in general more attention should be given to the utilization of instance-level information to perform match.

In the research area of knowledge engineering, a number of ontology integration methods and tools exist. Among them, Chimaera [18] and PROMPT [14] are the few, which have working prototypes. Both tools support the merging of ontological terms i.e. class and attribute names from various sources. The processes start by running a matching algorithm on class names in the pair of ontologies to suggest the merging points. The matching algorithm either looks for an exact match in class names or for a match on prefixes, suffixes, and word root of class names. A user can then choose from these matching points, or proceed on his/her own initiative. For each merging operation, PROMPT suggests the user to perform a sequence of actions on copying the classes and their attributes, creating necessary subclasses and putting them in the right places in the hierarchy. Some other systems use a composite approach to achieve better matching results. GLUE [11] is a system uses composite approach to combine different matching strategy. It uses machine learning techniques for individual matchers and an automatic combination of match results. Recently, there have emerged some preliminary studies trying to perform ontology mapping via analyzing an extensional description of concepts and deriving mappings by comparing extensional descriptions. [27] proposes a method called FCA-MERGE, based on the theory of formal concept analysis, for merging ontologies following a bottom up approach and the method is guided by application-specific instances of the given source ontologies that are to be merged. [3] uses techniques well known from the area of data mining (association rules) for the task of catalogue integration.

Our approach is in line with the latter group of endeavours. Yet it differs from them in the following ways. First information retrieval models are used to represent concept extension and calculate similarity between concepts. This gives us a practical way to rank the mapping results. When a large amount of mappings are predicated, ranking them is especially useful for the user. In addition, the leverage of vector space model allows us to take into account the hierarchical and contextual information of concepts in a unified manner. Second, we use graphical notations to represent the ontology and the mapping results. This makes it easier for the user to understand the model and get quick overviews. Furthermore, we have explored the possibility of incorporating WordNet into the framework to achieve better mapping results.

8 Conclusions

We have introduced a heuristic mapping method and prototype mapping system, iMapper, that supports the process of semi-automatic ontology mapping. The approach is based on the idea of semantic enrichment. The proposed functionality facilitates the mapping process in that the semantic heterogeneity and the risk for anomalies are reduced. Furthermore, the cognitive overload of the user is reduced in the manual part of the mapping process.

The approach can be applied in several other different contexts. One such context is documents retrieval and publication between different web portals. Users may conform to their local ontologies through which the web portals are organized. It is desirable to have support for automated exchange of documents between the portals and still let the users keep their perspectives. Service matching is yet another candidate to apply the method. Assume a situation where there are some service description hierarchies (the MIT process handbook for instance [17]) and the provider and the requester use different classification schema. By using the extension descriptions of the service hierarchies, we can compute a feature vector for each service concept. Then the matching can be conducted by calculating the distance between the representative feature vectors.

References

[1] K. Aas and L. Eikvil. Text categorisation: A survey. Technical report, Norwegian Computing Center, Oslo, 1999.

[2] E. Agirre and G. Rigau. Word sense disambiguation using conceptual density. In *Proceedings of the 16th conference on Computational linguistics - Volume 1*, 1996.

[3] R. Agrawal and R. Srikant. On integrating catalogs. In *Proceeding of the WWW-11*, Hong Kong, 2001.

[4] R. Baeze-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* Addison Wesley, 1999.

[5] C. Batini and M. Lenzerini. A comparative analysis of methodologies for database schema integration. *ACM Computer Surveys.*, 18(4), 1986.

[6] S. Bergamaschi, S. Castano, and M. Vincini. Semantic integration of semistructured and structured data sources. *SIGMOD Record*, 28(1):54–59, 1999.

[7] T. Berners-Lee. The semantic web. *Scientific american*, 284(5):35–43, 2001.

[8] T. Brasethvik and J. A. Gulla. Natural language analysis for semantic document modeling. *Data and Knowledge Engineering*, 38(1):45–62, 2001.

[9] J. Didion. Jwnl (java wordnet library), http://sourceforge.net/projects/jwordnet/, 2004.

[10] H. H. Do and E. Rahm. Coma-a system for flexible combination of schema matching approaches. In *Proceeding of the 28th VLDB conference.*, 2002.

[11] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to map between ontologies on the semantic web. In *Proceedings of WWW-2002, 11th International WWW Conference.*, Hawaii, 2002.

[12] DublinCore. http://www.dublincore.org.

[13] D. Fensel, Y. Ding, B. Omelayenko, E. Schulten, G. Botquin, M. Brown, and A. Flett. Product data integration in b2b e-commerce. *IEEE Intelligent Systems (Special Issue on Intelligent E-Business)*, 16(4):54–59, 2001.

[14] N. Fridman and M. A. Musen. Prompt: algorithm and tool for automated ontology merging and alignment. In *Proceeding of Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, 2000.

[15] Google. www.google.com, 2004.

[16] J. Madhavan, P. Bernstein, and E. Rahm. Generic schema matching using cupid. In *Proceeding of Very Large Database Conference (VLDB) 2001*, 2001.

[17] T. W. Malone, K. Crowston, J. Lee, and B. Pentland. Tools for inventing organizations: Toward a handbook of organizational processes. Technical Report 141, MIT, 1993.

[18] D. McGuinness, R. Fikes, J. Rice, and S. Wilder. An environment for merging and testing large ontologies. In *Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning*, Colorado, USA, 2000.

[19] E. Mena, A. Illarramendi, V. Kashyap, and A. P. Sheth. Observer: an approach for query processing in global infomation systems based on interoperation across pre-exist ontologies. *International journal on distributed and parallel databases*, 8(2):223–271, April 2000.

[20] G. A. Miller. Wizard of the new wordsmiths: His idea to link words rewrote the dictionary, the star ledger, January 2002.

[21] L. Palopoli, G. Terracina, and D. Ursino. The system dike: Towards the semi-automatic synthesis of cooperative information systems and data warehouses. In *ADBIS-DASFAA Symposium 2000*, pages 108–117. Matfyz Press, 2000.

[22] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10:334–350, 2001.

[23] P. Resnik. Wordnet and distributionaly analysis: A class-based approach to lexical discovery. In *Statistically-based Natural Language Processing Techniques: Papers from the 1992 AAAI Workshop.* AAAI Press, 1992.

[24] RML. Referent modelling language http://www.idi.ntnu.no/ ppp.

[25] G. Salton and M. J. McGill. *An Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

[26] A. Sølvberg. Data and what they refer to. In P. P. Chen, editor, *Concept Modeling: Historical Perspectives and Future Trends.* Springer Verlag, 1998.

[27] G. Stumme and A. Maedche. Fca-merge: Bottom-up merging of ontologies. In *Proceedings of the International Joint Conference on Artificial Intelligence IJCAI01.*, Seattle, USA, 2001.

[28] X. Su and L. Ilebrekke. A comparative study of ontology languages and tools. In *Proceeding of the Seventh IFIP-WG8.1 International Workshop on Evaluating of Modeling Methods in Systems Analysis and Design (EMMSAD'02)*, 2002.

[29] A. Valente, T. Russ, R. MacGrecor, and W. Swartout. Building and (re)using an ontology for air campaign planning. *IEEE Intelligent Systems*, 14(1):27–36, 1999.