Ontology Mapping via Structural and Instance-Based Similarity Measures

Konstantin Todorov[1] and Peter Geibel[2]

[1]IKW, Universtity of Osnabrück, Albrechstr. 28, 49076 Osnabrück, Germany
[2]TU Berlin, Fakultät IV, Franklinstr. 28/29, 10587 Berlin, Germany

Abstract. The paper presents an overview of a novel procedure for mapping hierarchical ontologies, populated with properly classified text documents. It combines structural and instance-based approaches to reduce the terminological and conceptual ontology heterogeneity. It yields important granularity and instantiation judgments about the inputs and is to be applied to mapping web-directories.

1 Introduction and Initial Setting

Heterogeneity between ontolgies can occur in many forms, not in isolation from one another [5]. We describe our approach to map two hierarchical, tree-structured ontologies designed to categorize text documents (web pages) with respect to their content, by reducing their terminological and conceptual heterogeneity. The paper extends previous work by one of the co-authors [10]. We make use of both intentional and extensional information contained in the input ontologies and combine them in order to establish correspondences between the ontologies concepts. In addition, the proposed procedure yields assertions on the granularity and the extensional richness of one ontology compared to another which will be helpful at assisting the eventual stage of ontology merging.

Definition 1. *A hierarchical ontology is a pair* $O := (C_O, \texttt{is\_a})$*, where* C_O *is a finite set whose elements are called concepts and* `is_a` *is a partial order on* C_O *with the following property:*

- there exists exactly one element $A_0 \in C_O$ *such that* $\{B \in C_O | (A_0, B) \in \texttt{is\_a}\} = \emptyset$*,*

- for every element $A \in C_O$*,* $A \neq A_0$*, there exists an unique element* $A' \in C_O$ *such that* $(A, A') \in \texttt{is\_a}$*.*

We will use the documents assigned to a given concept as instances of that concept in order to model it. A given class is assigned the union of the sets of documents assigned to all nodes subsumed by this class. In Figure 1(a), the node $c2$ contains the documents set $\{d1, d2, d3, d4, d5, d6\}$.

Our inputs are two ontologies O_1 and O_2 together with their corresponding sets of documents $D_{O_1} = \{d_1^{O_1}, ..., d_{n_{O_1}}^{O_1}\}$ and $D_{O_2} = \{d_1^{O_2}, ..., d_{n_{O_2}}^{O_2}\}$, where each document is represented as a TF/IDF vector [7]. We assume that O_1 and O_2

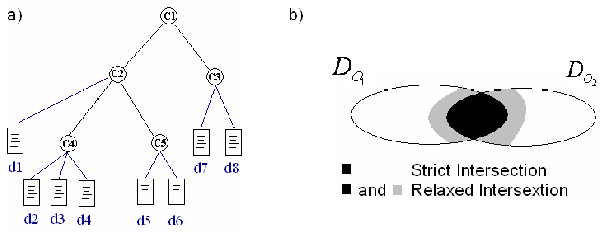

Fig. 1. a: A document populated taxonomy. b: Strict and relaxed intersections.

share a significant extensional overlap and all the documents are in the same natural language.

An entity which plays a key role in our approach is the intersection of D_{O_1} and D_{O_2}. However, when the sets elements are vectors of text documents, it is very likely that the sets contain documents which are similar, but not identical, and therefore not part of the intersection. In order to make use of such documents, we introduce the notion of relaxed intersection (RI) which integrates both identical *and* similar documents from both sets as opposed to the standard strict set intersection (Figure 1(b)): $RI(D_{O_1}, D_{O_2}) = \{d_i^{O_1}, d_j^{O_2} | dist(d_i^{O_1}, d_j^{O_2}) \leq c_d, d_i^{O_1} \in D_{O_1}, d_j^{O_2} \in D_{O_2}\}$, where $dist$ is a properly chosen distance measure on the set of TF/IDF documents [8] and c_d is a similarity parameter to be empirically set. In the sequel, by document set intersection we will mean their relaxed intersection.

2 Structural and Instance-based Mapping Strategies

In the following, we will describe the **structural approach** which forms the first part of our mapping strategy. A hierarchical ontology as described in definition 1 is directly translated to a directed rooted tree $G(V, E)$. Since only hyponimic relations are allowed at this stage, we will assume that the ontology graphs are unlabeled. Bunke et *al.* [1] introduced a graph distance, which accounts for the structural closeness of two taxonomies, represented as non-empty graphs G_1 and G_2:

$$d(G_1, G_2) = 1 - \frac{|mcs(G_1, G_2)|}{max(|G_1|, |G_2|)}.$$

The abbreviation mcs stands for maximal common subgraph of G_1 and G_2 defined as a maximal graph isomorphic to sub-graphs of both G_1 and G_2 and $|G|$ denotes the number of vertices in a graph G. The problem of finding a mcs is solved in polynomial time for trees. Various algorithms are discussed in [11].

In addition to the structural approach, we employ **two extensional methods** for deriving concepts similarity assertions. Even though independent from one another, they can be combined yielding an improved similarity learner. In both approaches, we make use of Support Vector Machines (SVMs) [3], operating on the sets of TF/IDF documents assigned to the input ontologies. SVMs are machine learning classifiers which can be trained on a given data set and learn to

discriminate between positive and negative instances. For two concepts $A \in C_{O_1}$ and $B \in C_{O_2}$ we define the data sets $S_{O_1} = \{(d_i^{O_1}, y_i^A)\}$ and $S_{O_2} = \{(d_j^{O_2}, y_j^B)\}$, where $d_i^{O_1}, d_j^{O_2} \in \mathbb{R}^d$, $i = 1, ..., n_{O_1}$, $j = 1, ..., n_{O_2}$ with d - the dimension of the TF/IDF vectors. y^A and y^B are labels taking values $+1$ when the corresponding document is assigned to A or B, respectively, and -1 otherwise. The labels separate the documents in each ontology into such that belong to a given concept A or B, respectively (positive instances) and such that do not (negative instances).

One convenient way of making use of extensional information is to model ***concepts as "bags" of instances*** and measure their similarity on set theoretic accounts considering A and B very similar when $A \cap B \approx A$. A standard instance-based similarity measure is the Jaccard coefficient [6], defined as: $Jacc(A, B) = \frac{P(A \cap B)}{P(A \cup B)}$, where $P(X)$ is the probability of a random instance to be an element of X. Note that $P(A \cap B) = P(A, B)$ and $P(A \cup B) = P(A, B) + P(A, \overline{B}) + P(\overline{A}, B)$, where the entity $P(A, B)$ denotes the *joint probability* of A and B. Each of the three joint probabilities is estimated by the fraction of documents that belong to both A and B: $P(A, B) = \frac{|A \cap_{O_1} B| + |A \cap_{O_2} B|}{|D_{O_1}| + |D_{O_2}|}$, where $\cap_{O_1}$ denotes intersecting documents belonging to O_1 only. By training an SVM classifier on the data set S_{O_1} and applying it on the document set D_{O_2} we come up with an estimation of the quantity $|A \cap_{O_2} B|$. Repeating the procedure after inversing the roles of O_1 and O_2 yields $|A \cap_{O_1} B|$. The same algorithm is applied for the other joint probabilities until we have approximations of all of them, as described in [4].

The second extensional indicator for semantic closeness we propose is based on a ***variable selection procedure for SVMs***. Variable selection in descriptive statistics is about pointing out the input variables, which most strongly affect the response. For a given data set of the type S_{O_1} it indicates which of the TF/IDF vector dimensions are most important for the separation of the documents into such that belong to a given concept and such that do not. Our variable selection criterion is the sensitivity of the VC dimension [3] - an indicator of the classifying capacity of a set of classifiers (e.g. the set of hyperplanes in a multidimensional space). Our initial experiments have shown variations in the estimation of that parameter according to the presence or absence of a given variable (vector dimension) in the data set. The procedure yields for each ontology an ordered list of variables on top of which are found the variables which are most important for the class separation. If the orders of the variables in both sets are similar, or if a significant number of most pertinent variables from both sets coincide, then the two concepts A and B are identified as similar.

3 A Procedure for Ontology Mapping

The structural and extensional approaches described so far are our instruments used to build a combined procedure for ontology mapping. Another important criterion for concept similarity is the presence of similar concept names in both ontologies. Linguistic analysis approaches to this problem, relying on names and textual description of ontology elements, are used in [2] and [9]. Even though

not explicitly discussed in this section, we keep in mind that this name-based criterion is to be checked at any step before measuring the instance-based similarity of a pair of concepts and is to become an integral part of the structural similarity approach.

Let us take as input again the ontologies O_1 and O_2 together with their corresponding document sets D_{O_1} and D_{O_2}. In the following, we describe our method for combining the mapping approaches earlier.

Case 1: $|D_{O_1}| \approx |D_{O_2}|$

The first big case considers ontologies which contain similar number of documents. The ratio $r_\Delta = \frac{|D_{O_1} \cap D_{O_2}|}{|D_{O_1} \cup D_{O_2}|}$ is an indicator of the size of the intersection of both sets relative to the sets size. There are two further possibilities:

- **Case 1.1.** $r_\Delta > c_{r_\Delta}$, where $c_{r_\Delta} \in (0, 1)$ is a parameter to be fixed. In this case we have two different ontologies on (almost) the same documents sets. It is very likely that they share a conceptual similarity. We proceed to checking the graph distance between them.

- *Case 1.1.a.* $d(G_{O_1}, G_{O_2}) \approx 0$. The taxonomies have similar structures, describe the same domain and have the same extensions. It is left to establish the precise concept-to-concept mappings, done by the help of the *instance based* similarity check.

- *Case 1.1.b.* $d(G_{O_1}, G_{O_2}) \approx 1$. The maximal common subgraph of both ontologies is quite small, i.e. one of the taxonomies contains significantly lower number of nodes compared to the other (Figure 2(a)). Let us assume that $|C_{O_1}| < |C_{O_2}|$. Since both ontologies are "built" on approximately the same sets of documents, this means that O_2 is more specific than O_1, and contains more concept nodes. O_1 can be directly injected into O_2. The concept-to-concept correspondences indicating the exact injection pattern are provided by instance-based concept similarity applied on the set of the nodes of the *mcs* of both taxonomies.

- **Case 1.2.** $r_\Delta \leq c_{r_\Delta}$. The ontologies are little likely to be similar since their extensions share very little (or no) overlap.

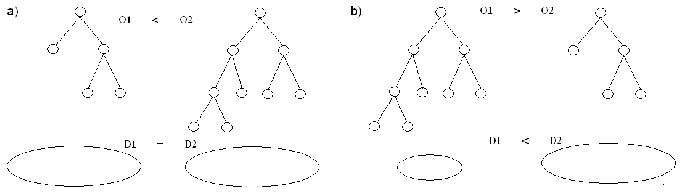

Fig. 2. a) Case 1.1.b. b) Case 2.3.b.2)

Case 2: $|D_{O_1}| < |D_{O_2}|$

In the second big case, the set D_{O_1} contains less documents than the set D_{O_2} (conventional choice). One can distinguish between three further sub-cases: Case

2.1. - the two sets do not intersect ($r_\Delta = 0$); Case 2.2. - the two sets intersect, but do not fully overlap; and Case 2.3. - the smaller set is a subset of the bigger one. Case 2.1. is in conflict with a major assumption introduced in the beginning and therefore does not provide mapping candidates. Case 2.2. conforms with either Case 2.1. or Case 2.3., depending on the size of the intersection $D_{O_1} \cap D_{O_2}$ relative to $|D_{O_1}|$. We will study in details Case 2.3. and proceed to measure the structural similarity between the inputs.

- *Case 2.3.a.* $d(G_{O_1}, G_{O_2}) \approx 0$. O_1 is structurally very similar to O_2. Hence, it is just as specific as O_2, but less populated with documents. This indicates that O_1 can be replaced entirely by O_2.

- *Case 2.3.b.* $d(G_{O_1}, G_{O_2}) \approx 1$. There are two different scenarios, depending on which of the two input ontologies contains more nodes.

1) $|C_{O_1}| < |C_{O_2}|$, i.e. there are less concepts in O_1 than in O_2. This is the case when O_1 is a sub-taxonomy of O_2 and can be entirely injected into it, as described in Case 1.1.

2) $|C_{O_2}| < |C_{O_1}|$. O_1 is more granular a hierarchy, but less populated than O_2 (Figure 2(b)). We will take instances from O_2 and assign them to O_1 by first aligning the nodes of both ontologies by the help of the *instance-based* mapping procedure. to another in terms of both conceptualization and instantiation.

References

1. H. Bunke, K. Shearer. A graph distance metric based on the maximal common subgraph, *Pattern Recogn. Lett.*, volume 19, number 3-4, 255–259, 1998.
2. P. Cimiano, A. Hotho, S. Staab. Learning Concept Hierarchies from Text Corpora Using Formal Concept Analysis, *JAIR Volume 24*, 305-339, 2005.
3. N. Cristianini, J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods.*, Cambridge University Press, ISBN 0-521-78019-5, 2000.
4. A. Doan, J. Madhavan, P. Domingos, A. Halevy. Learning to map between ontologies on the semantic web, *WWW '02: Proceedings of the 11th international conference on World Wide Web*, 662–673, 2002.
5. J. Euzenat, P. Shvaiko. *Ontology Matching*, Springer-Verlag New York, Inc., 2007.
6. A. Isaac, L. van der Meij, S. Schlobach, S. Wang. An empirical study of instance-based ontology matching. In *Proceedings of the 6th International Semantic Web Conference*, Busan, Korea, 2007.
7. T. Joachims. Text categorization with support vector machines: learning with many relevant features. *Proceedings of ECML-98, 10th European Conference on Machine Learning*, Number 1398, 137-142, 1998.
8. R. Korfhage. *Information Storage and Retrieval*, Section 5.7, Document Similarity, 125-133. Wiley and Sons, 1997.
9. G. Stumme, A. Maedche. FCA-MERGE: Bottom-Up Merging of Ontologies, *IJCAI*, 225-234, 2001.
10. K. Todorov. Combining Structural and Instance-Based Ontology Similarities for Mapping Web Directories, *ICIW, Third International Conference on Internet and Web Applications and Services, Athens*, 596-601, IEEE, 2008.
11. G. Valiente. *Algorithms on Trees and Graphs*, Springer-Verlag, 2002.