From Surface to Intensive Matching of Semantic Web Ontologies *

S. Castano, A. Ferrara, S. Montanelli, G. Racca
Università degli Studi di Milano
DICO - Via Comelico, 39, 20135 Milano - Italy
{castano,ferrara,montanelli,racca}@dico.unimi.it

Abstract

In the Semantic Web context, the Web is enriched with several domain ontologies specifying formal semantics of data for various intelligent services for information search, retrieval, and transformation. The high number of different ontologies over the Web requires automatic and effective techniques for ontology matching in order to find mappings between them for discovery and sharing of knowledge. In this paper, we describe H-MATCH*, an algorithm for dynamically perform ontology matching at different levels of depth, with different degrees of flexibility and accuracy.*

1. Introduction

Ontologies are an enabling technology for Semantic Web [1]. The advent of the Semantic Web has dramatically increased the need for efficient and flexible mechanisms to provide semantic mappings among ontologies for effective discovery and sharing of knowledge. Moreover, the high number of different ontologies over the Web requires automatic and effective techniques for ontology matching in order to find such mappings [2, 7]. In this paper, we describe H-MATCH, an algorithm for dynamically performing ontology matching at different levels of depth, with different degrees of flexibility and accuracy. H-MATCH performs the matching of two ontologies, and provides, for each concept of one ontology, a similarity ranking of concepts in the other ontology. H-MATCH is based on affinity metrics, takes into account different levels of richness in ontology descriptions, and allows one to consider various metadata elements of ontology descriptions separately or in combination for matching result evaluation. The paper is organized as follows. In Section 2, we discuss the requirements for ontology matching. In Section 3, we address the problem of exploiting linguistic features for matching. In Section 4, we focus on exploiting contextual features for matching. In Section 5, we present our matching functions and models, together with experimental results. In Section 6, we discuss how to exploit matching results for discovering mappings between ontologies. In Section 7, we discuss related work. Finally, in Section 8, we give our concluding remarks and we present our future work.

* This paper has been partially funded by "Wide-scalE, Broadband, MIddleware for Network Distributed Services (WEB-MINDS)" FIRB Project funded by the Italian Ministry of Education, University, and Research, and by NoE INTEROP, IST Project n. 508011 - 6th EU Framework Programme.

2. Requirements for ontology matching

The matching of Semantic Web ontologies introduces a number of challenging issues to be addressed. In this paper, we focus on three requirements:

- In the Semantic Web ontologies, the same real world resource can be described according to specification mechanisms, due to the syntactical freedom of RDF. For instance, the OWL language provides three increasingly expressive sublanguages (i.e., OWL Lite, OWL DL, and OWL Full), providing different constructs for resource description [12]. An important requirement for matching of Semantic Web ontologies is to capture the elements that are relevant for matching purposes in resource descriptions in spite of the formalism chosen for their representation. With respect to this requirement, H-MATCH is based on a reference ontology model (called H-MODEL [4]) capable of abstracting ontology description features that are relevant for matching purposes in a language independent way, in terms of *concepts*, *properties*, and *semantic relations*. In particular in OWL ontologies, OWL class declarations are abstracted into concepts, OWL datatype and object property restrictions are abstracted into properties, while OWL class relations and operators are abstracted into semantic relations. Semantic relations provided by H-MODEL are same-as, a kind-of, and a part-of. In particular, the equivalent-

Class and the subClassOf relations in OWL are abstracted into the same-as and the kind-of relations, respectively. Moreover, the the intersectionOf, and the unionOf OWL operators are abstracted by means of the kind-of relation. For an intersection clause of the form $A \equiv B \sqcap C$ we set two kind-of relations of the form A kind $-$ of B and A kind $-$ of C. For an union clause of the form $A \equiv B \sqcup C$ we set a two kind-of relations of the form B kind $-$ of A and C kind $-$ of A. Finally, the oneOf OWL clause used for representing an enumerated class defined as a collection is abstracted by means of the part-of relation. In particular, we set a part-of relation between each H-MODEL element representing a component of the collection and the H-MODEL concept representing the enumerated class.

- The meaning of ontology concepts depends basically on the names chosen for their definition and on their contexts, namely on their properties and on the relations they have with other concepts in the ontology. These two features can have a different impact in different ontology structures and can play a different relevance in the matching process. H-MATCH addresses this requirement by computing a comprehensive value of matching of two concepts, by combining both their linguistic and their contextual features. Furthermore, H-MATCH allows one to set the relevance of the linguistic and contextual features in the matching process.
- Different ontologies can describe the same domain using different modeling choices. A key requirement of the matching process is the capability of coping with different levels of detail and structuring in modeling the resources of interest, by considering various ontology elements separately or in combination. H-MATCH addresses this requirement by providing four different matching models, that are used for dynamically suiting the matching process to different levels of richness in ontology descriptions.

2.1. Running example

As an example of the ontology matching problem, we consider two real OWL ontologies describing different domains. The first ontology (Ka) describes research projects, while the second ontology (Portal) describes the contents of a Web portal. These ontologies are heterogeneous in terms of language specification (i.e., OWL Lite and OWL Full, respectively) as well as in terms of contents, although both of them contain concepts describing publications. Two portions of Ka and Portal describing publications are shown in Figure 1(a) and Figure 1(b), respectively, using the H-MODEL graphical notation. The main features of Ka and portal are summarized in Table 1. We are interested in exploiting H-MATCH for matching Ka against Portal in order to automatically discover the affinity between the concepts that describe publications, in spite of concept heterogeneity in the two ontology descriptions. Moreover, we use H-MATCH also for providing, for each concept of Ka, a measure of semantic affinity with the concepts of Portal.

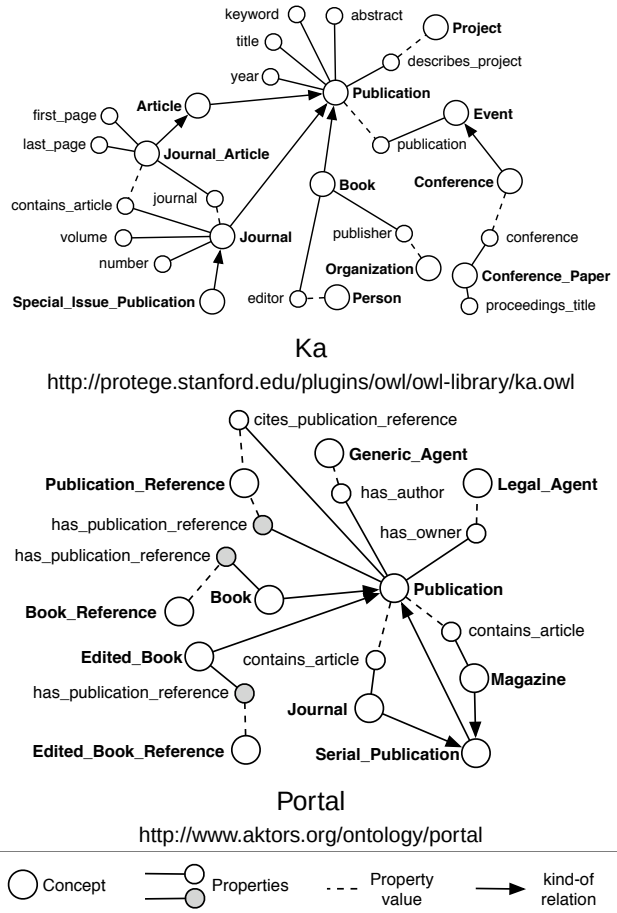

Figure 1. Two portions of Ka and Portal

	Ka	Portal
Language	OWL Lite	OWL Full
# of concepts	251	291
# of properties	154	204
Average # of properties per concept	13	4
Average # of relations per concept	2	2

Table 1. Features of the Ka and Portal ontologies

3. Exploiting linguistic features for matching using WordNet

Linguistic features refer to names of ontology elements and their meaning. To capture the meaning of names in an ontology, H-MATCH refers to a thesaurus Th of terms and terminological relationships among them. Th is automatically derived from the lexical system WordNet [11]. WordNet is an electronic lexical database where the different senses of English words are grouped by synonymy. The sets of synonyms (*synsets*) are organized hierarchically (i.e. each synset is connected to more general and more specific concepts by hypernymy and hyponymy relationships) and other semantic relations (e.g. meronymy) are available so as to build a semantic net. The thesaurus construction process is composed by three steps: i) extraction of the ontology element names; ii) thesaurus entries definition; iii) terminological relationships definition.

Extraction of the ontology element names. Given an ontology $\mathcal{O}$, we extract the name n_{c_i} of each concept $c_i \in \mathcal{O}$, together with the name n_{p_j} of each property $p_j \in \mathcal{O}$. For OWL ontologies, concept/property names correspond to the argument of the rdf:ID attributes associated with the owl:Class, the owl:ObjectProperty, and the owl:DatatypeProperty constructs, respectively. As an example, in Table 2, we report the names extracted from the Ka and from the Portal ontology portions of Figure 1.

Term definition. In the second step of the thesaurus construction, we define the entries of Th for the ontology element names extracted in the previous step. Given the set of concept names N_c and the set of property names N_p, we denote by T the set of terms used as names of ontology elements, that is $T = N_c \cup N_p$. We note that in real ontologies, like Semantic Web ontologies, ontology element names can be either single or compound. Term definition in the thesaurus is performed as follows.

1. *Basic terms.* A term $t \in T$ is a basic term, denoted as bt, if a noun entry for bt exists in WordNet[1]. Given the set $BT \subseteq T$ of the basic terms used as names of an ontology $\mathcal{O}$, an entry is defined in Th for each $bt_i \in BT$.
2. *Compound terms.* A term $t \in T$ is a compound term, denoted as ct, if an entry for ct does not exists in WordNet and ct is composed by more than one basic term bt_i. For managing compound terms, we exploit considerations discussed in [9]. In a typical compound term ct, one of its constituent basic terms denotes the central concept represented by ct, while the remaining basic terms denote a specification of such a central concept. In particular in English, the basic terms appearing on the left side of ct denote the specification of the meaning of term appearing on the right side. Our thesaurus organization makes explicit these considerations for compound terms, by introducing appropriate terms entries in Th necessary to correctly capture the meaning of a given compound term. Given a compound term ct, we derive the constituent basic terms that exist in WordNet for ct, that is, $ct = \langle bt_1, bt_2, \ldots, bt_k$. For each compound term $ct_i \in CT$, we define a term entry in Th for: i) ct_i and ii) for each constituent basic term $bt_j, j = 1, 2, \ldots, k$.

1 Terms different from nouns (e.g., verbs, adjectives, adverbs) are mostly used into compound terms, and are addressed by considering the whole compound term in Th.

Ka	Term	Thesaurus entry
Basic terms	Article, Abstract, Book, Conference, Event, Editor, Journal, Keyword, Number, Person, Project, Publication, Publisher, Organization, Title, Volume, Year	Article, Abstract, Book, Conference, Event, Editor, Journal, Keyword, Number, Person, Project, Publication, Publisher, Organization, Title, Volume, Year
Compound terms	describes_project	describes_project
	proceedings_title	proceedings_title, proceedings
	conference_paper	conference_paper, paper
	special_issue_-publication	special_issue_publication, issue, special
	contains_article	contains_article
	first_page	first_page, page, first
	last_page	last_page, last
	journal_article	journal_article

Portal	Term	Thesaurus entry
Basic terms	Book, Journal, Magazine, Publication, Serial_Publication	Book, Journal, Magazine, Publication, Serial_Publication
Compound terms	cites_publication_-reference	cites_publication_reference, reference
	generic_agent	generic_agent, agent
	has_author	has_author, author
	has_owner	has_owner, owner
	legal_agent	legal_agent
	contains_article	contains_article, article
	has_publication_-reference	has_publication_reference
	edited_book	edited_book
	edited_book_reference	edited_book_reference
	book_reference	book_reference
	publication_reference	publication_reference

Table 3. Thesaurus entries defined for terms of Ka and Portal

As an example of thesaurus entry definition, in Table 3, we report the basic and the compound terms used as names of Ka and Portal, together with their corresponding entries in Th.

Terminological relationships definition. Terminological relationships in Th are defined by considering the synsets and the relationships provided by WordNet. Terminological relationships considered in Th are SYN, BT/NT, and RT.

	Ka	Portal
Concept names	Article, Book, Conference, Conference_Paper, Event, Journal, Journal_Article, Person, Project, Publication, Organization, Special_Issue_Publication	Book, Book_Reference, Edited_Book, Edited_Book_Reference, Generic_Agent, Journal, Legal_Agent, Magazine, Publication, Publication_Reference, Serial_Publication
Property names	abstract, conference, contains_article, describes_project, editor, first_page, journal, keyword, last_page, number, proceedings_title, publication, publisher, title, volume, year	cites_publication_reference, contains_article, has_author, has_owner, has_publication_reference

Table 2. Names of the elements of Ka and Portal

- SYN: it is defined between two terms t_i and t_j that can be indifferently used to denote a certain concept. The SYN relationship is derived from synsets in WordNet.
- BT/NT: BT relationship is defined between two terms t_i and t_j such has t_i has a broader, more general meaning than t_j. The opposite of BT is NT. BT/NT relationships correspond to hypernymy and hyponymy relationships in WordNet, respectively.
- RT: it is defined between two terms t_i and t_j that are generally used together in the same context, both because t_i denotes a part-of t_j or because t_i and t_j are specifications of a common term t_k. RT corresponds to meronymy relationship and coordinate terms in WordNet, respectively.

Given two basic terms $bt_i, bt_j \in Th$, a terminological relationship tr between them is defined in Th as follows:

- $tr = \mathsf{SYN}$, if bt_i and bt_j belong to the same synset in WordNet;
- $tr = \mathsf{BT/NT}$, if a hypernymy/hyponymy relationships is retrieved in WordNet between the synsets of bt_i and bt_j, respectively;
- $tr = \mathsf{RT}$, if a meronymy relationship is retrieved in WordNet between the synsets of bt_i and bt_j, or if bt_i and bt_j are coordinate terms in WordNet;

Given a compound term $ct = \langle bt_1, bt_2, \ldots, bt_k \rangle$, the following terminological relationships are defined in Th:

- BT/NT between bt_k and ct, to capture that ct is a specification of bt_k;
- RT between each $bt_j, j = 1, 2, \ldots, k-1$ and ct, for addressing the fact that bt_j is used for specifying the central concept of ct;

Following this process, we define Th as a matrix of terms and terminological relationships between them. As an example, in Table 4, we show a portion of the thesaurus matrix referring terms and terminological relationships of the Publication concept in Ka.

Weighting terminological relationships. In order to express the implication of terminological relationships for semantic affinity, in H-MATCH a weight W_{tr} is associated with each terminological relationship $tr \in \{\mathsf{SYN}, \mathsf{BT/NT}, \mathsf{RT}\}$ in Th. Different types of relationships have different implications for semantic affinity. In particular, we have $W_{\mathsf{SYN}} \geq W_{\mathsf{BT/NT}} \geq W_{\mathsf{RT}}$. Synonymy is generally considered a more precise indicator of affinity than other relationships, consequently $W_{\mathsf{SYN}} \geq W_{\mathsf{BT/NT}}$. The lowest weight is associated with RT since it denotes a more generic relationship than BT/NT. These weights have been taken from the schema matching techniques of the ARTEMIS integration system [3], since they have been tested and tuned on several real integration cases.

4. Exploiting contextual features for matching

Contextual features refer to the context of an ontology concept, that is composed by properties and concepts directly related to a given concept in an ontology. Given a concept c, we denote by $P(c)$ the set of properties appearing into a property constraint associated with c (i.e., the properties of c), and by $C(c)$ the set of concepts having a semantic relation with c (in the following referred to as *adjacents*), respectively. The context $Ctx(c)$ of a concept c is defined as the union of the properties and of the adjacents of c, that is, $Ctx(c) = P(c) \cup C(c)$. In particular, a property $p \in P(c)$ represents an OWL property restriction for the concept c, and is characterized by a name n_p, a cardinality k_p, and a value v_p. $k_p \in \{0, 1\}$ is the minimal cardinality associated with p when applied to c, and v_p is the value associated with p when applied to c, and can be a datatype dt_p or a reference name (i.e., the name of a concept or an instance in the OWL property restriction). We call strong properties the properties with $k_p = 1$, and weak properties the ones with $k = 0$. Finally, for each concept $c_i \in C(c)$, a semantic relation $sr(c, c_i)$ denotes the semantic relations holding in OWL between c_i and c. As an example, in Figure 2, we show the contexts of the concept Publication in Ka and Portal, respectively.

Weighting contextual features. In H-MATCH, a weight W_{sr} is associated with each semantic relation to denote the strength of the connection expressed by the relation on the involved concepts for semantic affinity evaluation purposes. The greater the weight associated with a semantic relation, the higher the strength of the semantic connection between concepts. Furthermore, we associate a weight W_{sp} to strong

	Abstract	Describes_Project	Keyword	Publication	Title	Year
Abstract	SYN	-	-	-	-	-
Describes_Project	-	SYN	-	-	-	-
Project	-	BT	-	RT	-	-
Keyword	-	-	SYN	-	-	-
Publication	-	-	-	SYN	-	-
Issue	-	-	-	SYN	-	-
Page	-	-	-	RT	-	-
Book	-	-	-	BT	-	RT
Volume	-	-	-	BT	-	-
Title	-	-	-	-	SYN	-
Journal	-	-	-	-	RT	-
Number	-	-	-	-	RT	-
Year	-	-	-	-	-	SYN

Table 4. Example of the thesaurus matrix for the Publication concept and its properties

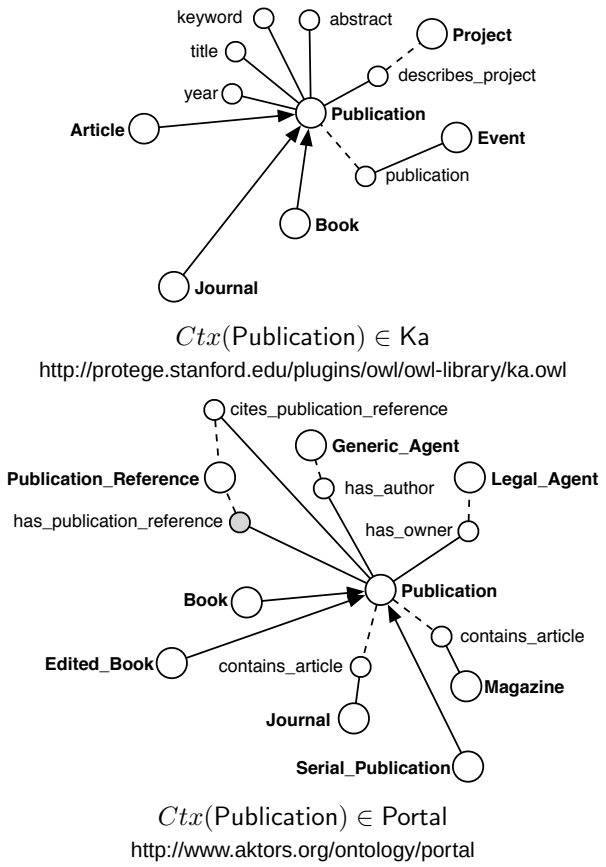

Figure 2. Example of Publication context in Ka and Portal

properties, and a weight W_{wp} to weak properties, respectively, with $W_{\mathsf{sp}} \geq W_{\mathsf{wp}}$ to capture the importance of the property in characterizing the concept for matching. In fact, strong properties are mandatory related to a concept and are relevant to give its structural description. Weak properties are optional for the concept in describing its structure, and, as such, are less important in featuring the concept than strong properties.

5. Matching functions and models

In this section, we present the basic functions and the matching models for performing ontology matching with H-MATCH.

5.1. Matching functions

Affinity function. The aim of the affinity function $\mathcal{A}(t, t') \rightarrow [0, 1]$ is to evaluate the affinity between two terms t and t' in the thesaurus Th. $\mathcal{A}(t, t')$ is equal to the value of the highest-strength path of terminological relationships between t and t' in Th if at least one path exists, and is zero otherwise. A path strength is computed by multiplying the weights associated with each terminological relationship involved in the path, that is:

$$\mathcal{A}(t, t') = \begin{cases} \max_{i=1\ldots k} \{W_{t \rightarrow_i^n t'}\} & \text{if } k \geq 1 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where: k is the number of paths between t and t' in Th; $t \rightarrow_i^n t'$ denotes the ith path of length $n \geq 1$; $W_{t \rightarrow_i^n t'} = W_{1_{tr}} \cdot W_{2_{tr}} \cdot \cdots \cdot W_{n_{tr}}$ is the weight associated with the ith path, where $W_{j_{tr}} \mid j = 1, 2, \ldots, n$ denotes the weight associated with the jth terminological relationship tr in the path.

Datatype compatibility function. A datatype compatibility function is defined to evaluate the compatibility of data types of two properties according to a pre-defined set CR of compatibility rules. The datatype compatibility function $\mathcal{T}(dt, dt') \rightarrow \{0, 1\}$ of two data types dt and dt' returns 1 if dt and dt' are compatible according to CR, and 0 otherwise, that is:

$$\mathcal{T}(dt, dt') = \begin{cases} 1 & \text{iff } \exists \text{ a compatibility rule for } dt, dt' \text{ in } CR \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

For instance, with reference to XML Schema datatypes, examples of compatibility rules that hold between datatypes

are: xsd:integer ⇔ xsd:int, xsd:integer ⇔ xsd:float, xsd:decimal ⇔ xsd:float, xsd:short ⇔ xsd:int.

Property and relation closeness function. The closeness function $\mathcal{C}(e, e') \rightarrow [0, 1]$ calculates a measure of the distance between two context elements of concepts (i.e., two properties, two semantic relations, or a semantic relation and a property, respectively). $\mathcal{C}(e, e')$ exploits the weights associated with context elements and returns a value in the range [0,1] proportional to the absolute value of the complement of the difference between the weights associated with the elements, that is:

$$\mathcal{C}(e, e') = 1 - \mid W_e - W_{e'} \mid \quad (3)$$

where W_e and $W_{e'}$ are the weights associated with e and e', respectively. For any pairs of elements e and e', the highest value (i.e., 1.0) is obtained when weights of e and e' coincide. The higher the difference between W_e and $W_{e'}$ the lower the closeness value of e and e'. We note that the closeness of a property and a semantic relation can be evaluated in order to capture the structural heterogeneity among different ontological descriptions.

5.2. Matching models

The general goal of ontology matching techniques is to find concepts that have a semantic affinity with a target concept, giving a measure of their affinity. To satisfy the main requirements for ontology matching described in the introduction, we have defined four matching models for H-MATCH. The matching models have been conceived to span from surface to intensive matching, with the goal of providing a wide spectrum of metrics suited for dealing with many different matching scenarios that can be encountered in comparing real Semantic Web ontologies. Each model calculates a semantic affinity value $SA_{c,c'}$ of two concepts c and c' which expresses their level of matching. $SA_{c,c'}$ is produced by considering linguistic and/or contextual features of concept descriptions, by applying the basic matching functions. In a matching model, the relevance of the linguistic and the contextual features of c and c' in the matching process can be properly set. $W_{la} \in [0, 1]$ is a weight used to set the impact of the linguistic affinity (and consequently of the contextual affinity) in the semantic affinity evaluation process. A description of the four matching models provided by H-MATCH is reported in Table 5.

5.3. Experimental results

In Table 6, we report the experimental results produced by H-MATCH matching the whole Ka ontology against Portal using terminological relations and contextual features weights shown in Table 7 and 8, respectively, and $W_{la} =$ 0.6. A ranking threshold of 0.5 has been used for filtering results.

Model	# of results	Average # of results per concept
Surface	3337	13.29
Shallow	1252	4.99
Deep	1489	5.93
Intensive	1194	4.76

Table 6. Number of results obtained matching Ka against Portal

Terminological relationship	Weight
SYN	1.0
BT/NT	0.8
RT	0.5

Table 7. Weights associated with terminological relationships

Context element	Weight
same_as relation	1.0
kind_of relation	0.8
part_of relation	0.5
strong_property	1.0
weak_property	0.5

Table 8. Weights associated with contextual features

In Table 9, the results specifically related to the Publication concept of the running example are reported. A main difference between Ka and Portal is that the relations among concepts are represented basically by means of semantic relations in Ka and by means of property values in Portal. This kind of heterogeneity can be better captured by using the Deep and Intensive models, because they are defined for considering semantic relations and property values, respectively. The Publication concept is identified by the same name in Ka and Portal, and thus the Surface model produces the same value. However, Publication is used in a different context with a different meaning in each ontology. In Ka, it refers to scientific publications describing research projects, while in Portal it refers to generical publications in a commercial context. This kind of difference is captured by the

Model	$\mathbf{SA_{c,c'}}$	Matching functions	Description
Surface	$\mathcal{A}(n_c, n_{c'})$		The surface matching is defined to consider only the linguistic features of concept descriptions. Surface matching addresses the requirement of dealing with high-level, poorly structured ontological descriptions
Shallow	$W_{la} \cdot \mathcal{A}(n_c, n_{c'}) + (1 - W_{la}) \cdot \cdot \frac{\sum_{i=1}^{\lvert P(c)\rvert} m(p_i)}{\lvert P(c)\rvert}$	$m(p_i) = \max\{\mathcal{A}(n_{p_i}, n_{p_j}) \cdot \mathcal{C}(p_i, p_j)\}$, $\forall p_j \in P(c')$, where $m(p_i)$ is the property matching value for the ith property of $P(c)$	The shallow matching is defined to consider both concept names and concept properties. With this model, we want a more accurate level of matching, by taking into account not only the linguistic features but also information about the presence of properties and about their cardinality constraints.
Deep	$W_{la} \cdot \mathcal{A}(n_c, n_{c'}) + (1 - W_{la}) \cdot \cdot \frac{\sum_{i=1}^{\lvert Ctx(c)\rvert} m(e_i)}{\lvert Ctx(c)\rvert}$	$m(e_i) = \max\{\mathcal{A}(n_{e_i}, n_{e_j}) \cdot \mathcal{C}(e_i, e_j)\}$, $\forall e_j \in Ctx(c')$, where $m(e_i)$ is the element matching value for the ith element of $Ctx(c)$	The deep matching model is defined to consider concept names and the whole context of concepts, in terms of properties and semantic relations.
Intensive	$W_{la} \cdot \mathcal{A}(n_c, n_{c'}) + (1 - W_{la}) \cdot \cdot \frac{\sum_{i=1}^{\lvert Ctx(c)\rvert} m(e_i) + \sum_{j=1}^{\lvert P(c)\rvert} v(p_j)}{\lvert Ctx(c)\rvert + \lvert P(c)\rvert}$	$v(p_i) = \begin{cases} \max\{\mathcal{A}(n_{p_i}, n_{p_j}) \cdot \mathcal{T}(v_{p_i}, v_{p_j})\}, & \text{iff } v_{p_i} \text{ is a datatype} \\ \max\{\mathcal{A}(n_{p_i}, n_{p_j}) \cdot \mathcal{A}(v_{p_i}, v_{p_j})\}, & \text{iff } v_{p_i} \text{ is a reference name} \end{cases}$ $\forall p_j \in P(c')$ $m(e_i) = \max\{\mathcal{A}(n_{e_i}, n_{e_j}) \cdot \mathcal{C}(e_i, e_j)\}$, $\forall e_j \in Ctx(c')$, where $v(p_i)$ is the measure of the property value matching for the value associated with the ith property of $P(c)$ and $m(e_i)$ is the element matching value for the ith element of $Ctx(c)$	The intensive matching model is defined to consider concept names, the whole context of concepts, and also property values, for the sake of a highest accuracy in semantic affinity evaluation. In fact, by adopting the intensive model not only the presence and cardinality of properties, but also their values have an impact on the resulting semantic affinity value.

Table 5. The H-MATCH matching models

SA	Surface	Shallow	Deep	Intensive
$SA_{\text{Publication,Book}}$	0.8	0.6184	0.66	0.6394
$SA_{\text{Publication,Book\_Reference}}$	0.64	0.5531	0.5733	0.5497
$SA_{\text{Publication,Edited\_Book}}$	0.64	0.5224	0.5641	0.5434
$SA_{\text{Publication,Edited\_Book\_Reference}}$	0.64	0.5531	0.5637	0.5420
$SA_{\text{Publication,Generic\_Agent}}$	-	-	-	-
$SA_{\text{Publication,Journal}}$	0.64	0.5224	0.5538	0.5381
$SA_{\text{Publication,Legal\_Agent}}$	-	-	-	-
$SA_{\text{Publication,Magazine}}$	0.8	0.6184	0.6498	0.6341
$SA_{\text{Publication,Publication}}$	1.0	0.7384	0.8074	0.7814
$SA_{\text{Publication,Publication\_Reference}}$	0.64	0.5531	0.5741	0.5503

Table 9. Example of matching results with H-MATCH

Shallow, Deep and Intensive models in spite of the concept name.

6. Ontology mapping

The H-MATCH algorithm can be exploited for discovering automatically inter-ontology mappings. Given two ontologies $\mathcal{O}$ and $\mathcal{O}'$, a mapping $m_{SA}(c, c')$ relates a concept c of $\mathcal{O}$ and a concept c' of $\mathcal{O}'$, and is associated with the semantic affinity $SA_{c,c'}$ between c and c'. Using the H-MATCH results, the mappings can be defined according to a one-to-one (1:1) or a one-to-many (1:n) strategy. In the 1:1 strategy, a mapping $m_{SA}(c_i, c'_j)$ is defined between a concept $c_i \in \mathcal{O}$ and its best matching concept $c_j \in \mathcal{O}'$, that is the concept c_j having the highest SA value with c_i. In the 1:n strategy, the ranking of affinity results produced by H-MATCH is exploited for defining a set of mappings for each concept $c_i \in \mathcal{O}$. A mapping $m_{SA}(c_i, c'_j)$ is defined as between the concept $c_i \in \mathcal{O}$ and each concept $c'_j \in \mathcal{O}'$ that belongs to the ranking of results obtained for c_i. In Figure 3 we show the 1:1 mappings defined between the main concepts of Ka and Portal portions of Figure 1 using the Intensive model of H-MATCH.

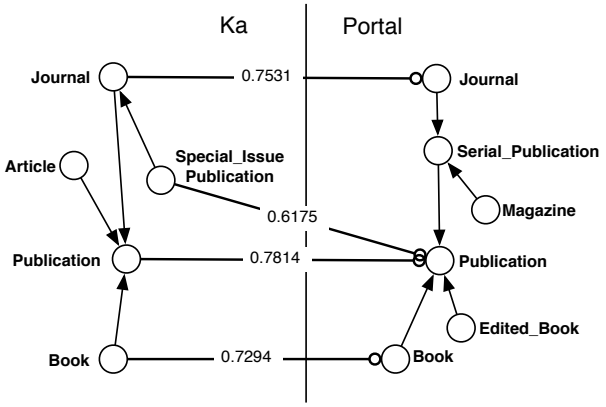

Figure 3. Example of mappings between Ka and Portal

6.1. Exploiting mappings in Peer-to-Peer systems

H-MATCH has been used in the HELIOS project, to support knowledge sharing and ontology-addressable content retrieval in peer-based systems [5, 6]. In HELIOS, each peer provides a formal representation of its knowledge by means of a *peer ontology*. H-MATCH is used by each peer to match incoming queries on target concepts against its peer ontology. Based on collected answers, each peer stores mappings to semantically related concepts of other peer ontologies in form of location relations within its peer ontology. Such mappings allow a peer to support a more effective routing of queries over the network, enhancing the routing strategies currently used in P2P systems.

7. Related work

Work on ontology matching techniques can be grouped into two main families, namely model-based and logic-based approaches. The model-based approach is based on the idea of exploiting the ontology metadata model for working on the ontology structure through a set of techniques of analysis, matching and learning. As an example, the Glue [7] approach exploits machine learning techniques to find semantic mappings between concepts stored in distinct and autonomous ontologies. Given two distinct ontologies, the mapping discovery process between their concepts is based on the measure of similarity which is defined through the joint probability distribution. The measure of similarity between two concepts is computed as the likelihood that an instance belongs to both the concepts. Another approach for model-based metadata matching in described in [10], where the choice of metadata for classifying data sources according to the requirements of a given application or task is discussed. In this approach, metadata information is organized as a set of categories and concepts, and the matching is enforced through fuzzy metrics.

The logic-based approach is based on the idea of exploiting the semantics associated with ontological descriptions for defining and analyzing mappings through automatic reasoning techniques. In particular, mapping discovering is reduced to the problem of checking a set of logical relations. For instance, in [2] the Ctx-Match algorithm is defined in order to point out semantic mapping between concepts stored in distinct peers of a Peer-to-Peer system. This algorithm compares the knowledge contained in different contexts looking for semantic mappings denoting peers interested in similar concepts. Ctx-Match is based on a semantic explication phase where concepts are associated with the correct meaning with respect to their context, and on a semantic comparison phase where concepts are translated in logical axioms and matched. As another example, in [8] the meaning of mappings is formally defined. The semantics provides a basis for reasoning about mappings (e.g., determining whether two mappings are equivalent or if a certain mapping formula is entailed by a mapping), combining evidence to propose likely mappings, and learning mappings. In particular, the reasoning is used for determining whether two mappings are equivalent, and whether a mapping is minimal (i.e., removing any formula from the mapping loses information).

With respect to these approaches, a main advantage of H-MATCH is the capability of dealing with different levels of accuracy in ontological descriptions by considering both the linguistic and the contextual features of ontology concepts. H-MATCH is suitable for dynamic scenarios like the Semantic Web, where ontologies evolves quickly and are characterized by different levels of accuracy in resource description.

8. Concluding remarks

In this paper, we have presented the H-MATCH approach to the problem of ontology matching in multi-ontology contexts such as the Semantic Web. H-MATCH has been implemented using the C++ programming language and has been tested both on Unix and on WinNT systems. Our future work will be devoted to the intensive experimentation on H-MATCH on real ontology matching cases. A final remark regards the fact that H-MATCH takes into account only direct hypernyms in the WordNet taxonomy for the thesaurus construction. This problem is an open issue, because it affects both the efficiency and the precision of H-MATCH. In fact, considering an overmuch number of indirect hypernyms for a given term would introduce in the thesaurus a large number of BT/NT relationships without distinction between the direct and the indirect ones. With respect to this

problem, we are studying how to take into account terminological relationships deriving from indirect ancestors without affecting the precision of the linguistic affinity evaluation.

References

[1] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, May 2001.

[2] P. Bouquet, B. Magnini, L. Serafini, and S. Zanobini. A SAT-based Algorithm for Context Matching. In *Proc. of the 4th Int. and Interdisciplinary Conference on Modeling and Using Context (CONTEXT 2003)*, pages 66–79, Stanford, CA, USA, June 2003. Springer Verlag.

[3] S. Castano, V. De Antonellis, and S. De Capitani Di Vimercati. Global Viewing of Heterogeneous Data Sources. *IEEE Transactions on Knowledge and Data Engineering*, 13(2):277–297, March/April 2001.

[4] S. Castano, A. Ferrara, and S. Montanelli. H-MATCH: an Algorithm for Dynamically Matching Ontologies in Peer-based Systems. In *Proc. of the 1st Int. Workshop on Semantic Web and Databases (SWDB) at VLDB 2003*, Berlin, Germany, September 2003.

[5] S. Castano, A. Ferrara, S. Montanelli, E. Pagani, and G. Rossi. Ontology-Addressable Contents in P2P Networks. In *Proc. of the 1st Int. Workshop on Semantics in Peer-to-Peer and Grid Computing (SemPGRID) at WWW 2003*, Budapest, Hungary, May 2003.

[6] S. Castano, A. Ferrara, S. Montanelli, and D. Zucchelli. HELIOS: a General Framework for Ontology-based Knowledge Sharing and Evolution in P2P Systems. In *IEEE Proc. of the 2nd Int. Workshop on Web Semantics (WEBS) at DEXA 2003*, Prague, Czech Republic, September 2003. IEEE Computer Society.

[7] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to Map between Ontologies on the Semantic Web. In *Proc. of the 11th Int. World Wide Web Conference (WWW 2002)*, pages 662–673, Honolulu, Hawaii, USA, May 2002.

[8] P. D. Jayant Madhavan, Philip A. Bernstein and A. Y. Halevy. Representing and Reasoning about Mappings between Domain Models. In *Proc. of the 18th National Conference on Artificial Intelligence and 14th Conference on Innovative Applications of Artificial Intelligence*, pages 80–86, Edmonton, Alberta, Canada, July/August 2002. AAAI Press.

[9] M. Lauer. *Designing Statistical Language Learners: Experiments on Noun Compounds*. PhD thesis, Macquaire University, Australia, 1995.

[10] J. Madhavan, P. A. Bernstein, and E. Rahm. Generic Schema Matching with Cupid. In *Proc. of the 27th Int. Conference on Very Large Data Bases (VLDB 2001)*, pages 49–58, Rome, Italy, September 2001.

[11] G. A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM (CACM)*, 38(11):39–41, 1995.

[12] M. K. Smith, C. Welty, and D. L. McGuinness (Eds.). OWL Web Ontology Language Guide, 2004, http://www.w3.org/TR/2004/REC-owl-guide-20040210/.