Results of the HMatch Ontology Matchmaker in OAEI 2006 *

Silvana Castano[1], Alfio Ferrara[1], and Gianpaolo Messa[1]

DICo Università degli Studi di Milano, via Comelico 39, 20135, Milano, Italy
{castano,ferrara,messa}@dico.unimi.it

Abstract. In this paper, we discuss our experience in testing the HMatch matchmaking system by means of the tracks proposed in the ontology alignment evaluation initiative of 2006 [1]. HMatch is a system conceived for the goal of ontology matching in open and distributed systems. It is based on linguistic and structural matching techniques for the evaluation of affinity considering concept names and concept contexts. The paper discusses the results that have been obtained and the possible improvements of the matching techniques in ongoing and future work.

1 Presentation of the system

HMatch is a system for dynamically matching distributed ontologies. It takes two ontologies as input and returns mappings that identify corresponding concepts in the two ontologies, namely the concepts with the same or the closest intended meaning. Mappings are established after an analysis of the similarity of the concepts in the compared ontologies. The similarity analysis is performed through affinity metrics to determine a measure of concept semantic affinity in the range [0, 1]. A threshold-based mechanism is enforced to set the minimum level of semantic affinity required to consider two concepts as matching concepts. HMatch is part of the Helios framework [4], conceived for supporting knowledge sharing and ontology-addressable content retrieval in peer-based systems.

A more detailed description of HMatch can be found in [3].

1.1 State, purpose, general statement

With respect to the different purposes of matching, the state of HMatch is the following:

- *Ontology matching* is the original purpose of HMatch which has been designed with the goal of working with the different languages of OWL (i.e., OWL Lite, OWL DL, and OWL Full) [11].

* This paper has been partially funded by BOEMIE, FP6-027538 - 6th EU Framework Programme and by ESTEEM MIUR PRIN project funded by the Italian Ministry of Education, University, and Research.

[1] http://oaei.ontologymatching.org/2006/

- *Schema matching*. In developing HMatch, we started from the schema matching functionalities of Artemis integration system [2]. From Artemis we borrowed the thesaurus-based approach for name affinity management, but we made a number of extensions for matching linguistic features of ontology elements in order to rely only on the WordNet lexical system, to provide a fully-automated matching process. Furthermore, we have moved from the notion of structural affinity, typical of schema elements based on attributes, to the notion of *contextual affinity*, typical of ontology elements, based on semantic relations with explicit semantics, with consequent development of suitable contextual affinity evaluation techniques.
- *Version matching*. Currently, we are extending HMatch towards version matching in the context of the BOEMIE European Project [1] Specifically, we are extending the tool to perform instance matching and to evaluate the differences between different versions of the same ontology to support the evolution of multimedia ontologies.
- *Directory matching*. HMatch can perform directory matching in the deep matching model configuration, by considering taxonomic knowledge in the directory as is-a relations in all cases. However, directory taxonomic relations have a different semantics (e.g., *part-of*, *contain*), and a manual pre-processing is required in order to distinguish them in the matching process.

1.2 Specific techniques used

Given two concepts, HMatch calculates their semantic affinity value as the linear combination of a *linguistic affinity* value and a *contextual affinity* value. The basic techniques used in HMatch are linguistic and structure-based techniques that are applied to concept names and contexts. For a more detailed classification of these and other techniques the reader can refer to [6].

Linguistic-based affinity techniques. Linguistic techniques consider names of ontology elements and their meaning. To capture the meaning of names for ontology matching, a thesaurus of terms and weighted terminological relationships is exploited. In HMatch, the thesaurus is automatically derived from the lexical system WordNet [8]. The thesaurus is structured as a graph, where the nodes represent terms and the edges represent terminological relationships. Terminological relationships represented in the thesaurus are SYN, BT, NT, and RT. SYN (synonymy) denotes that two terms have the same meaning. BT (broader term) (resp., NT (narrower term)) denotes that a term has a more (resp., less) general meaning than another term. Finally, RT (related terms) denotes that two terms have a generic positive relationship. A weight W_{tr} is associated with each terminological relationship $tr \in \{\mathsf{SYN}, \mathsf{BT/NT}, \mathsf{RT}\}$ in the thesaurus. Such a weight expresses the implication of the terminological relationship for semantic affinity. Different types of relationships have different implications for semantic affinity, with $W_{\mathsf{SYN}} \geq W_{\mathsf{BT/NT}} \geq W_{\mathsf{RT}}$. Given the thesaurus of weighted terminological relationships, the linguistic affinity is evaluated by means of a term affinity function $\mathcal{A}(t, t') \rightarrow [0, 1]$ which evaluates the affinity between two terms t and t'. $\mathcal{A}(t, t')$ of two terms t and t' is equal to the value of the highest-strength path of terminological relationships between them 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 $n \geq 1$; $W_{t\rightarrow_i^n t'} = W_{1_{tr}} \cdot W_{2_{tr}} \cdot \ldots \cdot W_{n_{tr}}$ is the weight associated with the ith path, and $W_{j_{tr}}, j = 1, 2, \ldots, n$ denotes the weight associated with the jth terminological relationship in the path.

Structure-based affinity techniques. Structure-based techniques consider properties and concepts directly related to a concept c through a semantic relation in an ontology. Given a concept c, we denote by $P(c)$ the set of properties of c, and by $C(c)$ the set of concepts that participate in a semantic relation with c (namely, its *adjacents*). The context of a concept in HMatch is defined as the union of the properties and of the adjacents of c, that is, $Ctx(c) = P(c) \cup C(c)$. Also contextual features are weighted in HMatch. In particular, we associate a weight W_{sp} to strong properties, and a weight W_{wp} to weak properties, with $W_{\mathsf{sp}} \geq W_{\mathsf{wp}}$ to capture the different importance they have in describing the concept. In fact, strong properties are mandatory properties related to a concept and they are considered more relevant in contributing to concept description. Weak properties are optional for the concept in describing its structure, and, as such, are given less importance. Each semantic relation has associated a weight W_{sr} which expresses the strength of the connection expressed by the relation on the involved concepts. Considering the semantic relations of OWL, we have the weights $W_{\mathsf{equivalence}} \geq W_{\mathsf{subClassOf}}$. The greater the weight associated with a semantic relation, the higher the strength of the semantic connection between concepts. Given two elements e and e' in the contexts of c and c', respectively, their affinity is calculated according to the following function $\mathcal{C}(e,e') \rightarrow [0,1]$:

$$\mathcal{C}(e,e') = \mathcal{A}(n_e, n_{e'}) \cdot (1- \mid W_e, W_{e'} \mid \quad (2)$$

where n_e and $n_{e'}$ denote the names of e and e', respectively, while W_e and $W_{e'}$ denotes the weights associated with e and e'. As an example, suppose that we compare two concept contexts $Ctx(c)$ and $Ctx(c')$ both containing the property $author$ that is a strong property (i.e., featured by a minimum cardinality greater than or equal to 1) in the first context and a weak property (i.e., an optional property) in the second context. Thus, by using a weight equal to 1.0 for strong properties and equal to 0.5 for weak properties, we obtain:

$$\mathcal{C}(author_{Ctx(c)}, author_{Ctx(c')}) = \mathcal{A}(author, author) \cdot (1- \mid 1.0 - 0.5 \mid) = 0.5$$

since $\mathcal{A}(author, author) = 1.0$.

Given two concepts c and c', the comprehensive semantic affinity $SA(c,c')$ is calculated as the weighted sum between their linguistic affinity $LA(c,c')$ and their contextual affinity $CA(c,c')$, as follows:

$$SA(c,c') = W_{la} \cdot LA(c,c') + (1 - W_{la}) \cdot CA(c,c') \quad (3)$$

where $W_{la} \in [0,1]$ weights the relevance of the linguistic affinity in matching evaluation. The two measures of linguistic affinity $LA(c, c')$ and $CA(c, c')$ are calculated in a different way depending on the matching model that is selected in the configuration of HMatch.

Matching models. Four 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 ontologies, such as OWL ontologies. The main difference among the four matching models is the composition of the context. In the *surface model*, the context is not considered limiting to linguistic affinity. In the *shallow model*, only properties and property restrictions are considered for concept context. In the *deep model*, we consider both properties and semantic relations, such as taxonomic relations. Finally, in the *intensive model* we consider the whole context, by taking into account also the property ranges and values. For all the models the linguistic affinity $LA(c, c')$ between two concepts c and c' is calculated to be equal to the function $\mathcal{A}(n_c, n_{c'})$, where n_c and $n_{c'}$ denote the names of c and c', respectively. For the contextual affinity evaluation, we provide two main strategies, namely the *standard* strategy and the *Dice coefficient* strategy. The standard strategy produces a non-symmetric contextual affinity measure. For each element e in the source concept context $Ctx(c)$, we search for the best matching element e' in the target concept context $Ctx(c')$ by exploiting the function $\mathcal{C}(e, e')$ described above. Given the best matching value m_e found for e with respect to the elements in the context of c', the comprehensive contextual affinity is calculated as follows:

$$CA(c, c') = \frac{\sum_{e_i \in Ctx(c)} m_{e_i}}{\mid Ctx(c) \mid}$$

where $\mid Ctx(c) \mid$ denotes the number of elements in c.

According to the Dice coefficient strategy, the contextual affinity is calculated as follows:

$$CA(c, c') = \frac{\mid x \in Ctx(c) \cap Ctx(c') \mid}{\mid x \in Ctx(c) \cup Ctx(c') \mid}$$

where $\mid x \in Ctx(c) \cap Ctx(c') \mid$ denotes the number of matching elements in $Ctx(c)$ and in $Ctx(c')$, that is the number of elements having a value of $\mathcal{C}(e, e')$ higher than a given matching threshold.

1.3 Matching policies

Since HMatch has been developed with the goal of achieving a high level of flexibility and configurability of the matching process, a *matching policy* P has be set, which is defined as follows:

$$P = \langle W_{la}, T, M, C, I, S, E \rangle$$

where: W_{la} is the weight associated with the linguistic affinity; $T \in [0,1]$ denotes the threshold used for selecting matching results; $M \in$ {surface, shallow, deep, intensive} denotes the matching model; $C \in${one-to-one, one-to-many} denotes the matching cardinality; $I \in${true, false} denotes if the context elements inherited through the taxonomic relations are to be considered in the matching process; $S \in$ {standard_strategy, dice_strategy} denotes the metrics used for the contextual affinity evaluation; $E \in${empty-_pessimistic, empty_neutral, empty_optimistic} denotes the strategy to be enforced to handle empty contexts. Using the pessimistic strategy, the contextual affinity value is set to 0, to mean that no matching elements have been found in their contexts. In the neutral strategy the empty contexts are considered to have a semantics analogous to the one of the NULL value in relational databases; the contextual affinity is set to undefined to capture this semantics. In the optimistic strategy, the contextual affinity value is set to 1, to mean that two empty contexts are considered to fully match.

1.4 Adaptations made for the evaluation

For the purposes of the OAEI 2006 initiative, we adopted the standard implementation of HMatch as a Protégé[2] plugin. This version adopts the Protégé OWL API[3] and is fully integrated into the Protégé framework. We only introduce a command line version in order to use HMatch as an independent tool, especially for the benchmark. We have implemented two main extensions specifically conceived for the contest. The first extension is the support for the output Alignment format required by the organizers in addition to the native HMatch results format. The second extension regards the evaluation of the linguistic affinity. We introduced a new facility of HMatch that performs linguistic affinity evaluation using a n-gram algorithm [5]. This technique, being syntactic is faster than the thesaurus-based analysis, thus overcomes some scalability problems that we noticed with very large ontologies, such as in the case of *anatomy* and *directory-full*.

1.5 Link to the system and parameters file

The HMatch implementation used for the contest together with the policy used for configuration and the results can be find at:
http://islab.dico.unimi.it/OAEI2006/islab_results.html.

2 Results

All the results have been obtained by configuring HMatch with the following policy:

	Policy	Value
W_{la}	Linguistic affinity weight	0.5
T	Threshold	0.6
M	Matching model	Deep
C	Matching type	One-to-One
I	Inheritance	True
E	Empty context strategy	Neutral
S	Contextual affinity strategy	Dice coefficient

[2] http://protege.stanford.edu/

[3] http://protege.stanford.edu/plugins/owl/api/

In particular, the most relevant parameters are i) the matching model, since the deep model forces HMatch to consider both properties and semantic relations in the concept contexts, ii) the weight for linguistic affinity, since the value 0.5 determines a perfect balance between the linguistic affinity evaluation and the contextual affinity evaluation, and iii) the threshold, which is used for cutting off the results that are not considered relevant in the matching case. We have tested several possible configurations of HMatch on the benchmark. In some matching cases there are other configuration policies that produce better results in terms of precision and recall than the one we have chosen. The actual choice was motivated by the fact that we considered the various tasks proposed in the contest with the goal of configuring HMatch with a policy that could guarantee a generally satisfactory behavior of the system in the different matching cases. In particular, we have tested HMatch on all the test cases provided in the contest, with the goal of receiving a feedback about the application of the system to different and highly heterogeneous matching cases.

2.1 Benchmark

Obtained results on the proposed benchmark are strongly affected by the fundamental role that the ontology linguistic features play in the HMatch matching process. In fact, we obtained an average precision value of 0.84 and an average recall value of 0.55. These results are influenced by the fact that we obtained poor result for the ontology cases where the concept and property labels were substituted with randomly generated strings of characters. The difference between precision and recall values when we consider all the cases is due to the fact that, in some of the randomly generated ontologies (e.g., case 259), there is a property which maintains the original name (i.e., lastName). This matching is retrieved by HMatch and it increases the precision of the results. Another issue that affects the results quality, from the linguistic point of view, is the presence of matching cases where the concept and property labels are french terms. In these cases, since some of the properties match, we obtained precision values about 0.4 and recall values about 0.2. The benchmark results are also useful to suggest possible improvements of HMatch, with the goal of addressing also the anomalous cases where the linguistic information is completely missing due to the design choices.

2.2 Anatomy

With the anatomy track, obtained results suggest the following considerations. Due to the domain specific terminology used in the ontologies, either using the WordNet thesaurus or a string matching technique, the results are affected by the fact that the concepts are labeled with long strings describing specific terms. In the case of a domain specific terminology, the linguistic matching would benefit from the availability of specific thesauri. Given the large amount of data in the two compared ontologies, the string-matching procedure for linguistic affinity is more suitable, while affecting the capability of the system to capture the semantics of the terms used in the two ontologies that would instead be possible using the thesaurus. Moreover, the openGalen ontology has a anomalous OWL structure, since OWL classes are used as meta-classes, while individuals represent the domain concepts. For this reason we needed a wrapper to compare the FMA concepts with the concepts of openGalen, and only the linguistic comparison was possible.

2.3 Directory

The directory matching is a new task for HMatch, which was not originally designed for dealing with peculiar features of directory repositories. In particular, two main characteristics of directory taxonomies require specific support not directly provided by HMatch. The terminology used for labeling the directories is often affected by the structure of the taxonomy itself more than by the subject of the directory. Examples of this terminology is given by terms like A-H that is referred to the alphabetic order more than to the subject of the directory, or African_2 where the name of the directory is associated with information about the number of equivalent directories in the taxonomy. A second problem is given by the taxonomy itself. In fact, HMatch gives the is-a semantics to the OWL sub-class relations as in formal ontologies. Although, the sub-directory relations represented as OWL sub-class relations have in fact different meanings. For example, we have a sub-class relation between Animal_Webcams and Space_and_Science that denotes a generic positive relation between the two concepts rather than an is-a relation. Another example is given by the sub-class relation between California and United_States that denotes a geographical part-of relation.

2.4 Food

The food track requires to match two XML thesauri. We developed a wrapper from the SKOS XML format to OWL in order to match the thesauri with HMatch. The track requires also to recognize different kind of mapping relations between the source and the target, i.e., exactMatch, broadMatch, and narrowMatch. Using HMatch, we provide a measure of the semantic affinity between two concept, that is a measure of the fact the the two terms have the same meaning. Because of this reason, we provided only an evaluation of the exactMatch mapping between the two ontologies. In order to evaluate the broad and narrow matching relations, the thesaurus component of HMatch could be exploited, but this has not been done due to the contest requirement of using the same techniques for all different cases.

3 General comments

One of the main issues in the field of ontology matching is the need of flexible algorithms and tools, capable to adapt to different domains and also to different interpretation of the notions of *alignment* and *similarity*. Some of these differences depends on the concept descriptions provided by the ontologies to be compared with their specific level of semantic complexity. The choice of the best approach or the best combination of approaches depends on the specific matching case we are dealing with and on the domain of the ontologies. For example, formal ontologies can benefit from a logic approach, while thesauri and dictionaries require a deep linguistic analysis; finally, structure affinity is suitable for directories and repositories. The domain affects also the kind of techniques that are used as well as the matchmaking utilities (e.g., thesauri, external sources, type of mapping relations) that are involved in the matching process. A good example is given by the *anatomy* track of the contest. In this domain, we work with a specific and domain dependent terminology that requires a specific linguistic analysis. A second example is given by the matching of directories or also by the matching of spatial or temporal ontologies. In this cases, in fact, some properties or relations should be matched by using specific matching operators. For example, the property *author* and the property *below* have a different role on concept definition when used in a spatial domain, even if they are represented by means of the same language construct. The matching should take into account all these specific requirements by adapting the matching process and the matching operations to the specific domain that is taken into account.

3.1 Comments on the results

The results obtained in the OAEI tasks show how HMatch can provide a good balance in the results between precision and recall with a fully automated matching that does not require any specific external source neither in terms of a training set of results nor in terms of domain specific thesauri. Although, if on a side this characteristic means that HMatch can be used in several different scenarios, on the other side, it shows a limitation of the system in working either with very specific domain ontologies or with ontologies in which the linguistic information is missing. Some other limitations regard the scalability of the linguistic techniques adopted by the system is the case of large ontologies. To overcome this limitation, we have implemented for the purpose of the contest a new string matching functionality. The main considerations that we can make based on matching cases and obtained results experienced are the following.

1. *Linguistic features*: the terminology used for naming and labeling concepts and properties is an important aspect of ontologies and provides information to conclude the similarity between the ontology elements. We are conscious that, In many cases, it is not sufficient alone, also because they embed a subjectivity element, deriving from who has been designed the ontology. However, the linguistic features are undoubtedly an important starting point also for deriving a first set of mappings to be refined by exploiting other kinds of matchings.
2. *Structural features*: concepts can be similar also in terms of their structure. The structure is seen in terms of the links that connect different concepts and also as the number and type of properties that characterize each concept. It is important to note that the structure evaluation does not refer to the semantics of the concept relations and properties. For example, in the directory taxonomies the semantics of the *sub-category* relation is not ever well defined and can denote many different real relations among categories, e.g., containment, is-a, part-of. In this case, the structure of the taxonomies that are considered is the key feature for detecting the similarity of the concepts, more than the relation semantics.
3. *Logical features*: from the logical point of view, the ontology matching should consider the formal semantics of the ontologies to be compared in order to i) evaluate the consistency between the mappings and the concept descriptions, ii) apply deductive reasoning to retrieve new mappings starting from an initial set of mappings

(e.g., manually provided or retrieved by means of other techniques), iii) provide an interpretation of the resulting mappings.

3.2 Discussions on the way to improve the proposed system

By analyzing the results obtained in the different tracks, together with the general comments discussed in the previous section, a first improvement that can be introduced in HMatch is to emphasize the distinction among the linguistic, structural, and logic approaches to ontology matching. HMatch is based mainly on linguistic features. We believe that linguistic matching is a fundamental component for a semantic matchmaker, but we noticed that, in some cases, structure and logics of the ontologies to be compared should be considered with no reference to the ontology element names. Another important direction for improving HMatch is to emphasize the need of different metrics in order to take into account the specific features of the different ontology domains. HMatch provides four different matching models to address the fact that different ontologies can vary with respect to their semantic complexity and with respect to their structure. A further improvement in this direction is to support specific relations in the matching process, such as spatial or temporal relations.

3.3 Comments on the OAEI 2006 procedure

The OAEI 2006 procedure is well suited to give to matching researchers a complete feedback about their work. Although, we believe that the requirement of using only one set of parameters for the whole contest was a strong limitation, especially because some of the test cases (i.e., anatomy, food) have peculiar features that would benefit from a more flexible configuration. We believe that the capability of matching algorithms to be configured in order to deal with different scenarios is a key feature for ontology matching, but the flexibility cannot be appreciated using the same configuration. If the goal is to test generic-purpose algorithms, the test cases should be more homogeneous with respect to the ontology type and domain. Otherwise, it should be possible to modify the algorithms configuration for the different cases.

3.4 Comments on the OAEI 2006 test cases

The only comment we have is that, at the end of the evaluation phase, would be useful to have the expected results also for the blind tests, in order to improve the algorithms used where required.

3.5 Comments on the OAEI 2006 measures

The traditional precision and recall measures seem to be the most suitable for the matching result evaluation. Although, these measures should be calculated in a flexible way. For example, we should allow the algorithms to provide mappings also among external elements that are imported in the ontologies.

3.6 Proposed new measures

A simple suggestion for new measures is referred to the need of taking into account the time of computation in the matching evaluation. The idea is to combine the computation time with precision and recall, in order to measure the trade-off between time performances of the algorithms and quality of the results.

4 Conclusion

The experience of the OAEI 2006 contest was extremely useful as a feedback about the design and implementation of the current version of HMatch. We had some confirmation of the results obtained in the previous tests, but we had also some new helpful tip about possible improvements of the approach and related techniques. In particular, our future work will be devoted to: i) study new matching techniques that could be used in combination with the linguistic techniques of HMatch, in order to improve the flexibility of the system with respect to different matching scenarios; ii) address new purposes of the matching, such as directory of ontology version matching, by studying specific metrics and techniques for these cases; iii) implement and test a new version of HMatch in the context of the BOEMIE project, where our matchmaking system is used for the purpose of ontology evolution.

References

1. BOEMIE Project. Bootstrapping ontology evolution with multimedia information extraction. IST-2004-2.4.7 Semantic-based Knowledge and Content Systems Proposal, 2004. http://www.boemie.org.
2. 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.
3. S. Castano, A. Ferrara, and S. Montanelli. Matching ontologies in open networked systems: Techniques and applications. *Journal on Data Semantics (JoDS)*, V, 2006.
4. S. Castano, A. Ferrara, and S. Montanelli. *Web Semantics and Ontology*, chapter Dynamic Knowledge Discovery in Open, Distributed and Multi-Ontology Systems: Techniques and Applications. Idea Group Publishing, Hershey, Pennsylvania, USA, 2006.
5. Y. S. Ching. n-gram statistics for natural language understanding and text processing. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1:164–172, 1979.
6. Alfio Ferrara. *Matching of Independent Ontologies in Open Networked Systems: Methods and Techniques*. PhD thesis, Università degli Studi di Milano, 2005.
7. Y. Kalfoglou and M. Schorlemmer. Ontology mapping: the state of the art. *The Knowledge Engineering Review Journal*, 18(1), 2003.
8. G. A. Miller. WordNet: A lexical database for english. *Communications of the ACM (CACM)*, 38(11):39–41, 1995.
9. N. Noy. Semantic integration: a survey of ontology-based approaches. *SIGMOD Record Special Issue on Semantic Integration*, December 2004.
10. P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. *Journal on Data Semantics (JoDS)*, 1, 2005.
11. M. K. Smith, C. Welty, D. L. McGuinness, and (eds.). Owl web ontology language guide, 2004. World Wide Web Consortium (W3C), http://www.w3.org/TR/owl-guide/.