A Large Scale Taxonomy Mapping Evaluation

Paolo Avesani[1], Fausto Giunchiglia[2], and Mikalai Yatskevich[2]

[1] ITC-IRST,
38050 Povo, Trento, Italy
avesani@itc.it
[2] Dept. of Information and Communication Technology,
University of Trento,
38050 Povo, Trento, Italy
{fausto,yatskevi}@dit.unitn.it

Abstract. Matching hierarchical structures, like taxonomies or web directories, is the premise for enabling interoperability among heterogenous data organizations. While the number of new matching solutions is increasing the evaluation issue is still open. This work addresses the problem of comparison for pairwise matching solutions. A methodology is proposed to overcome the issue of scalability. A large scale dataset is developed based on real world case study namely, the web directories of Google, Looksmart and Yahoo!. Finally, an empirical evaluation is performed which compares the most representative solutions for taxonomy matching. We argue that the proposed dataset can play a key role in supporting the empirical analysis for the research effort in the area of taxonomy matching.

1 Introduction

Taxonomic structures are commonly used in file systems, market place catalogs, and the directories of Web portals. They are now widespread as knowledge repositories (in this case they can be viewed as shallow ontologies [23]) and the problem of their integration and interoperability is acquiring a high relevance from a scientific and commercial perspective. A typical application of hierarchical classification interoperability occurs when a set of companies wants to exchange products without sharing a common product catalog. The typical solution to the interoperability problem amounts to performing matching between taxonomies. The Match operator takes two graph-like structures as input and produces a mapping between the nodes of the graphs that correspond semantically to each other.

Many diverse solutions to the matching problem have been proposed so far, see for example surveys in [20, 21] and concrete solutions [13, 15, 6, 18, 24, 3, 19, 17, 8], etc. Unfortunately nearly all of them suffer from the lack of evaluation. Until very recently there were no comparative evaluations and it was quite difficult to find two systems which were evaluated on the same dataset. At the same time the evaluation efforts were mostly concentrated either on datasets artificially synthesized under questionable assumptions or on the "toy" examples.

In this paper we introduce a large scale dataset for evaluating matching solutions. The dataset is constructed from the mappings extracted from real web directories and contains thousands of mappings. We have evaluated the dataset using the most representative state of the art solutions to the matching problem. The evaluation highlighted that the dataset has four key properties namely *Complexity*, *Discrimination capability*, *Incrementality* and *Correctness*. The first means that the dataset is "hard" for state of the art matching systems, the second that it discriminates among the various matching solutions, the third that it is effective in recognizing weaknesses in the state of the art matching systems and the fourth that it can be considered as a correct tool to support the improvement and research on the matching solutions. At the same time the current version of dataset contains only "true positive" mappings. This fact limits the evaluations on the dataset to measuring only Recall. This is a weakness of the dataset that we plan to improve. However, as highlighted in [16], the biggest problem in nowadays matching systems is recall, while completeness is much less of an issue.

The rest of the paper is organized as follows. Section 2 summarizes the definition of the matching problem and recalls the state of the art. Section 3 expands more on the notion of mapping evaluation problem. Section 4 illustrates how the large scale dataset has been arranged. Section 5 is devoted to a large scale empirical evaluation on two leading matching systems. Section 6 presents the results of our experiments and argues why the proposed dataset is of interest. Section 7 concludes the paper.

2 The Matching Problem

In order to motivate the matching problem and illustrate one of the possible situations which can arise in the data integration task let us use the two taxonomies A and B depicted on Figure 1. They are taken from Yahoo! and Standard business catalogues. Suppose that the task is to integrate these two taxonomies.

We assume that all the data and conceptual models (e.g., classifications, database schemas, taxonomies and ontologies) can be represented as graphs (see [9] for a detailed discussion). Therefore, the matching problem can be represented as extraction of graph-like structures from the data or conceptual models and matching the obtained graphs. This allows for the statement and solution of a generic matching problem, very much along the lines of what done in [15, 13].

The first step in the integration process is to identify candidates to be merged or to have relationships under an integrated taxonomy. For example, $Computer\_Hardware_A$ can be assumed equivalent to $Computer\_Hardware_B$ and more general than $Personal\_Computers_B$. Hereafter the subscripts designate the schema (either A or B) from which the node is derived.

We think of a mapping element as a 4-tuple $\langle ID_{ij}, n1_i, n2_j, R\rangle$, $i = 1, ..., N_1$; $j = 1, ..., N_2$; where ID_{ij} is a unique identifier of the given mapping element; $n1_i$ is the i-th node of the first graph, N_1 is the number of nodes in the first graph; $n2_j$ is the j-th node of the second graph, N_2 is the number of nodes in

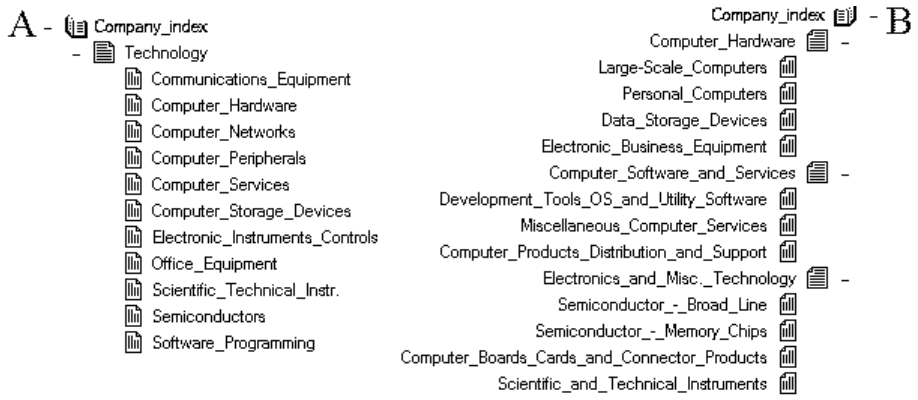

Fig. 1. Parts of Yahoo and Standard taxonomies

the second graph; and R specifies a similarity relation of the given nodes. A mapping is a set of mapping elements. We think of matching as the process of discovering mappings between two graph-like structures through the application of a matching algorithm.

Matching approaches can be classified into syntactic and semantic depending on how mapping elements are computed and on the kind of similarity relation R used (see [10] for in depth discussion):

- In *syntactic matching* the key intuition is to find the syntactic (very often string based) similarity between the labels of nodes. Similarity relation R in this case is typically represented as a [0, 1] coefficient, which is often considered as equivalence relation with certain level of plausibility or confidence (see [13, 7] for example). Similarity coefficients usually measure the closeness between two elements linguistically and structurally. For example, the similarity between $Computer\_Storage\_Devices_A$ and $Data\_Storage\_Devices_B$ based on linguistical and structural analysis could be 0,63.
- *Semantic matching* is an approach where semantic relations are computed between concepts (not between labels) at nodes. The possible semantic relations (R) are: equivalence ($=$); more general or generalization ($\supseteq$); less general or specification ($\subseteq$); mismatch ($\perp$); overlapping ($\cap$). They are ordered according to decreasing binding strength, i.e., from the strongest ($=$) to the weakest ($\cap$). For example, as from Figure 1 $Computer\_Hardware_A$ is more general than $Large\_Scale\_Com\text{-}puters_B$

In this paper we are focused on taxonomy matching. We think about taxonomy as a $\langle N, A, F_l \rangle$, where N is a set of nodes, A is a set of arcs, such that $\langle N, A \rangle$ is a rooted tree. F_l is a function from N to set of labels L (i.e., words in natural language). An example of taxonomy is presented on Figure 1. Notice that the distinguishing feature of taxonomies is the lack of formal encoding semantics.

3 The Evaluation Problem

Nearly all state of the art matching systems suffer from the lack of evaluation. Till very recently there was no comparative evaluation and it was quite difficult to find two systems evaluated on the same dataset. Often authors artificially synthesize datasets for empirical evaluation but rarely they explain their premises and assumptions. The last efforts [22] on matching evaluation concentrate rather on artificially produced and quite simple examples than real world matching tasks. Most of the current evaluation efforts were devoted to the schemas with tenth of nodes and only some recent works (see [6] for example) present the evaluation results for the graphs with hundreds of nodes. At the same time industrial size schemas contain up to tenth thousands of nodes.

The evaluation problem can be summarized as the problem of acquiring the reference relationship that holds between two nodes. Given such a reference relationship it would be straightforward to evaluate the result of a matching solution. Up to now the acquisition of the reference mappings that hold among the nodes of two taxonomies is performed manually. Similarly to the annotated corpora for information retrieval or information extraction, we need to annotate a corpus of pairwise relationships. Of course such an approach prevents the opportunity of having large corpora. The number of mappings between two taxonomies are quadratic with respect to taxonomy size, what makes hardly possible the manual mapping of real world size taxonomies. It is worthwhile to remember that web directories, for example, have tens thousands of nodes. Certain heuristics can help in reducing the search space but the human effort is still too demanding.

Our proposal is to build a reference interpretation for a node looking at its use. We argue that the semantics of nodes can be derived by their pragmatics, i.e., how they are used. In our context, the nodes of a taxonomy are used to classify documents. The set of documents classified under a given node implicitly defines its meaning. This approach has been followed by other researchers. For example in [5, 14] the interpretation of a node is approximated by a model computed through statistical learning. Of course the accuracy of the interpretation is affected by the error of the learning model. We follow a similar approach but without the statistical approximation. The working hypothesis is that the meaning of two nodes is equivalent if the sets of documents classified under those nodes have a meaningful overlap.

The basic idea is to compute the relationship hypotheses based on the cooccurence of documents. This document-driven interpretation can be used as a reference value for the evaluation of competing matching solutions. A simple definition of equivalence relationship based on documents can be derived by the F1 measure of information retrieval.

Figure 2 shows a simple example. In the graphical representation we have two taxonomies, for each of them we focus our attention on a reference node. Let be S and P two sets of documents classified under the reference nodes of the first and second taxonomies respectively. We will refer to A_S and A_P as the set of documents classified under the ancestor nodes of S and P. Conversely, we will refer to T_S and T_P as the set of documents classified under the subtrees of S

and P. The goal is to define a relationship hypothesis based on the overlapping of the set of documents, i.e. the pragmatic use of the nodes.

The first step, the *equivalence* relationship, can be easily formulated as the F1 measure of information retrieval [2]. The similarity of two sets of documents is defined as the ratio between the marginal sets and the shared documents:

$$Equivalence = \frac{|M_P^S| + |M_S^P|}{|O_P^S|}$$

where the set of shared documents is defined as $O_P^S = P \cap S$ and $M_P^S = S \setminus O_P^S$ is the marginal set of documents classified by S and not classified by P (similarly $M_S^P = P \setminus O_P^S$). The following equivalence applies $O_P^S = O_S^P$. Notice that "O" stands for "overlapping" and "M" stands for "Marginal set".

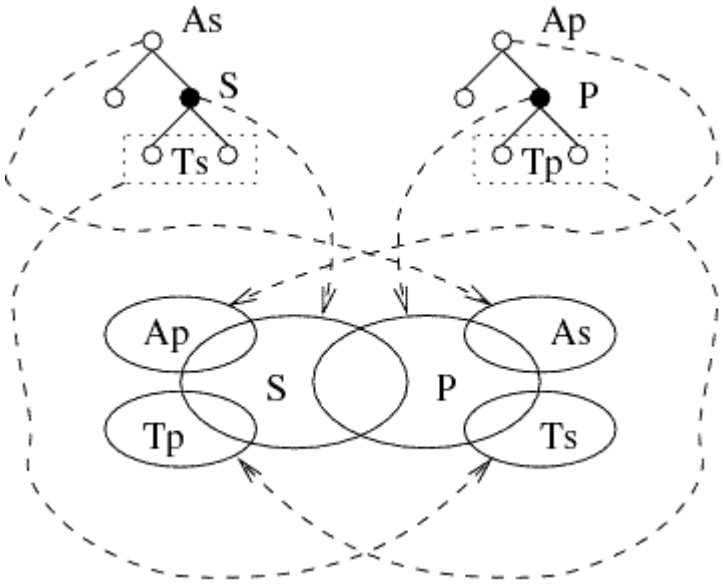

Fig. 2. The pairwise relationships between two taxonomies.

We do a step forward because we do not only compute the equivalence hypothesis based on the notion of F1 measure of information retrieval, but we extend such equation to define the formulation of generalization and specialization hypotheses. Generalization and specialization hypotheses can be formulated taking advantage of the contextual encoding of knowledge in terms of hierarchies of categories. The challenge is to formulate a generalization hypothesis (and conversely a specialization hypothesis) between two nodes looking at the overlapping of set of documents classified in the ancestor or subtree of the reference nodes [1].

The *generalization* relationship holds when the first node has to be considered more general of the second node. Intuitively, it happens when the documents classified under the first nodes occur in the ancestor of the second node, or the documents classified under the second node occur in the subtree of the first node. Following this intuition we can formalize the generalization hypothesis as

$$Generalization = \frac{|M_P^S| + |M_S^P|}{|O_P^S| + |O_{A_S}^P| + |O_{T_P}^S|}$$

where $O_{A_S}^P$ represents the set of documents resulting from the intersection between M_S^P and the set of documents classified under the concepts in the hierarchy above S (i.e. the ancestors); similarly $O_{T_P}^S$ represents the set of documents resulting from the intersection between M_P^S and the set of documents classified under the concepts in the hierarchy below P (i.e. the children).

In a similar way we can conceive the *specialization* relationship. The first node is more specific than second node when the meaning associated to the first node can be subsumed by the meaning of the second node. Intuitively, it happens when the documents classified under the first nodes occur in the subtree of the second node, or the documents classified under the second node occur in the ancestor of the first node.

$$Specialization = \frac{|M_P^S| + |M_S^P|}{|O_P^S| + |O_{T_S}^P| + |O_{A_P}^S|}$$

where $O_{T_S}^P$ represents the set of documents resulting from the intersection between M_S^P and the set of documents classified under the concepts in the hierarchy below S (i.e. the children); similarly $O_{A_P}^S$ represents the set of documents resulting from the intersection between M_P^S and the set of documents classified under the concepts in the hierarchy above P (i.e. the ancestors).

The three definitions above allow us to compute a relationship hypothesis between two nodes of two different taxonomies. Such an hypothesis relies on the assumption that if two nodes classify the same set of documents, the meaning associated to the nodes is reasonably the same. Of course this assumption is true for a virtually infinite set of documents. In a real world case study we face with finite set of documents, and therefore, this way of proceeding is prone to error. Nevertheless, our claim is that the approximation introduced by our assumption is balanced by the benefit of scaling with the annotation of large taxonomies.

4 Building a Large Scale Mapping Dataset

Let us try to apply the notion of document-driven interpretation to a real world case study. We focus our attention to web directories for many reasons. Web directories are widely used and known; moreover they are homogeneous, that is they cover general topics. The meaning of a node in a web directory is not defined with formal semantics but by pragmatics. Furthermore the web directories

address the same space of documents, therefore the working hypothesis of co-occurence of documents can be sustainable. Of course different web directories don't cover the same portion of the web but the overlapping is meaningful.

The case study of web directories meets two requirements of the matching problem: to have heterogeneous representations of the same topics and to have taxonomies of large dimensions.

We address three main web directories: Google, Yahoo! and Looksmart. Nodes have been considered as categories denoted by the lexical labels, the tree structures have been considered as hierarchical relations, and the URL classified under a given node as documents. The following table summarizes the total amount of processed data.

Web Directories	**Google**	**Looksmart**	**Yahoo!**
number of nodes	335.902	884.406	321.585
number of urls	2.425.215	8.498.157	872.410

Let us briefly describe the process by which we have arranged an annotated corpus of pairwise relations between web directories.

Step 1. We crawled all three web directories, both the hierarchical structure and the web contents, then we computed the subset of URLs classified by all of them.

Step 2. We pruned the downloaded web directories by removing all the URLs that were not referred by all the three web directories.

Step 3. We performed an additional pruning by removing all the nodes with a number of URLs under a given threshold. In our case study we fixed such a threshold at 10.

Step 4. We manually recognized potential overlapping between two branches of two different web directories like

```
   Google:/Top/Science/Biology
Looksmart:/Top/Science-and-Health/Biology

    Yahoo:/Top/Computers-and-Internet/Internet
Looksmart:/Top/Computing/Internet

   Google:/Top/Reference/Education
    Yahoo:/Top/Education
```

We recognized 50 potential overlapping and for each of them we run an exhaustive assessment on all the possible pairs between the two related subtrees. Such an heuristic allowed us to reduce the quadratic explosion of cartesian product of two web directories. We focussed the analysis on smaller subtrees where the overlaps were more likely.

Step 5. We computed the three document-driven hypothesis for *equivalence*, *generalization* and *specialization* relationships as described above. Hypotheses of equivalence, generalization and specialization are normalized and estimated by a number in the range [0,1]. Since the cumulative hypothesis of

all three relationships for the same pair of nodes can not be higher than 1, we introduce a threshold to select the winning hypothesis. We fixed such a threshold to 0.5.

We discarded all the pairs where none of the three relationship hypotheses was detected. This process allowed us to obtain 2265 pairwise relationships defined using the document-driven interpretation. Half are equivalence relationships and half are generalization relationships (notice that by definition generalization and specialization hypothesis are symmetric).

In the following we will refer to this dataset as TaxME, TAXonomy Mapping Evaluation.

5 The Empirical Evaluation

The evaluation was designed in order to assess the major dataset properties namely:

- *Complexity*, namely the fact that the dataset is "hard" for state of the art matching systems.
- *Discrimination ability*, namely the fact that the dataset can discriminate among various matching approaches.
- *Incrementality*, namely the fact that the dataset allows to incrementally discover the weaknesses of the tested systems.
- *Correctness*, namely the fact that the dataset can be a source of correct results.

We have evaluated two state of the art matching systems *COMA*[3] and *S − Match* and compared their results with *baseline solution*. Let us describe the matching systems in more detail.

The *COMA* system [13] is a generic syntactic schema matching tool. It exploits both element and structure level techniques and combines the results of their independent execution using several aggregation strategies. *COMA* provides an extensible library of matching algorithms and a framework for combining obtained results. Matching library contains 6 individual matchers, 5 hybrid matchers and 1 reuse-oriented matcher. One of the distinct features of the COMA tool is the possibility of performing iterations in the matching process. In the evaluation we used default combination of matchers and aggregation strategy (*NamePath+Leaves* and *Average* respectively).

S-Match is a generic semantic matching tool. It takes two tree-like structures and produces a set of mappings between their nodes. *S-Match* implements semantic matching algorithm in 4 steps. On the first step the labels of nodes are linguistically preprocessed and their meanings are obtained from the Oracle (in the current version WordNet 2.0 is used as an Oracle). On the second step the

[3] In the evaluation we use the version of *COMA* described in [13]. A newer version of the system *COMA++* exists but we do not have it. However as from the evaluation results presented in [10, 11], *COMA* is still best among the other syntactic matchers.

meaning of the nodes is refined with respect to the tree structure. On the third step the semantic relations between the labels at nodes and their meanings are computed by the library of element level semantic matchers. On the fourth step the matching results are produced by reduction of the node matching problem into propositional validity problem, which is efficiently solved by SAT solver or ad hoc algorithm (see [10, 11] for more details).

We have compared the performance of these two systems with *baseline solution*. The pseudo code of baseline node matching algorithm is given in Algorithm 1. It is executed for each pair of nodes in two trees. The algorithm considers a simple string comparison among the labels placed on the path spanning from a node to the root of the tree. Equivalence, more general and less general relations are computed as the corresponding logical operations on the sets of the labels.

Algorithm 1 Baseline node matching algorithm

```
1: String nodeMatch(Node sourceNode, Node targetNode)
2: Set sourceSetOfLabels=getLabelsInPathToRoot(sourceNode)
3: Set targetSetOfLabels=getLabelsInPathToRoot(targetNode)
4: if sourceSetOfLabels ≡ targetSetOfLabels then
5:    result="≡"
6: else if sourceSetOfLabels ⊆ targetSetOfLabels then
7:    result="⊆"
8: else if sourceSetOfLabels ⊇ targetSetOfLabels then
9:    result="⊇"
10: else
11:    result="Idk"
12: end if
13: return result
```

The systems have been evaluated on the dataset described in Section 4. We computed the number of matching tasks solved by each matching system. Notice that the matching task was considered to be solved in the case when the matching system produce specification, generalization or equivalence semantic relation for it. For example, TaxME suggests that specification relation holds in the following example:

```
Google:/Top/Sports/Basketball/Professional/NBDL
Looksmart:/Top/Sports/Basketball
```

COMA produced for this matching task 0.58 similarity coefficient, which can be considered as equivalence relation with probability 0.58. In the evaluation we consider this case as true positive for *COMA* (i.e., the mapping was considered as found by the system).

Notice that at present TaxME contains only true positive mappings. This fact allows to obtain the correct results for Recall measure, which is defined as a ratio of reference mappings found by the system to the number of reference mappings. At the same time Precision, which is defined as ratio of reference

mappings found by the system to the number of mappings in the result, can not be correctly estimated by the dataset since, as from Section 4, TaxME guarantee only the correctness but not completeness of the mappings it contains.

6 Discussion of Results

Evaluation results are presented on Table 1. It contains the total number of mappings found by the systems and the partitioning of the mappings on semantic relations. Let us discuss the results through the major dataset properties perspective.

Table 1. Evaluation Results

	Google vs. Looksmart	Google vs. Yahoo	Looksmart vs.Yahoo	Total
COMA	608	250	18	876 (38,68%)
$=$	608	250	18	876
$\subseteq$	not applicable	not applicable	not applicable	not applicable
$\supseteq$	not applicable	not applicable	not applicable	not applicable
S-Match	584	83	2	669 (29,54%)
$=$	2	5	0	7
$\subseteq$	46	19	2	67
$\supseteq$	536	59	0	595
Baseline	54	76	0	130 (5,39%)
$=$	52	0	0	52
$\subseteq$	0	76	0	76
$\supseteq$	2	0	0	2

6.1 Complexity

As from Table 1, the results of *baseline* are surprisingly low. It produced slightly more than 5% of mappings. This result is interesting since on the previously evaluated datasets (see [4] for example) the similar baseline algorithm performed quite well and found up to 70% of mappings. This lead us to conclusion that the dataset is not trivial (i.e., it is essentially hard for simple matching techniques).

As from Figure 3, *S-Match* found about 30% of the mappings in the biggest (Google-Yahoo) matching task. At the same time it produced slightly less than 30% of mappings in all the tasks. *COMA* found about 35% of mappings on Google-Looksmart and Yahoo-Looksmart matching tasks. At the same time it produced the best result on Google-Yahoo. *COMA* found slightly less than 40% of all the mappings. These results are interesting since, as from [13, 10], previously reported recall values for both systems were in 70-80% range. This fact turn us to conclusion that the dataset is hard for state of the art syntactic and semantic matching systems.

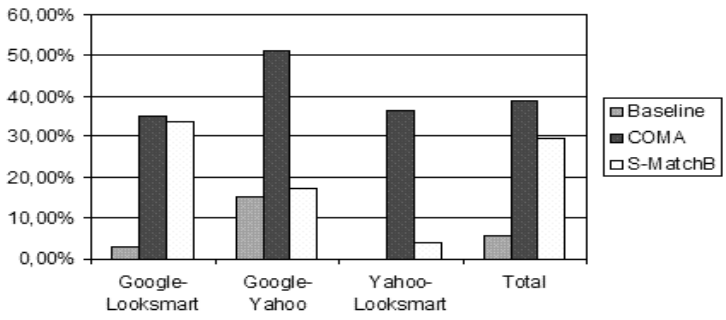

Fig. 3. Percentage of correctly determined mappings(Recall)

6.2 Discrimination ability

Consider Figure 4. It presents the partitioning of the mappings found by *S-Match* and *COMA*. As from the figure the sets of mappings produced by *COMA* and *S-Match* intersects only on 15% of the mappings. This fact turns us to an important conclusion: the dataset is discriminating (i.e., it contains a number of features which are essentially hard for various classes of matching systems and allow to discriminate between the major qualities of the systems).

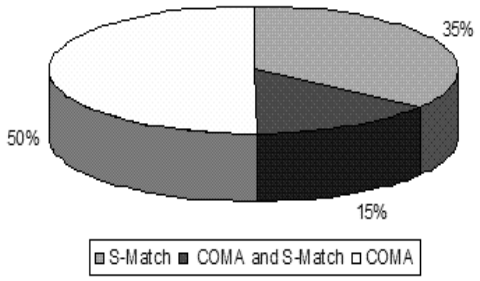

Fig. 4. Partitioning of the mappings found by COMA and S-Match

6.3 Incrementality

In order to evaluate incrementality we have chosen *S-Match* as a test system. In order to identify the shortcomings of *S-Match* we manually analyzed the mappings missed by *S-Match*. This analysis allowed us to clasterize the mismatches into several categories. In this paper we describe in detail one of the most important categories of mismatches namely *Meaningless labels*.

Consider the following example:

```
Google:/Top/Science/Social_Sciences/Archaeology/Alternative/
       South_America/Nazca_Lines
Looksmart:/Top/Science_&_Health/Social_Science/Archaeology/
       By_Region/Andes_South_America/Nazca
```

In this matching task some labels are meaningful in the sense they define the context of the concept. In our example these are *Social_Sciences*, *Archaeology*, *South_America*, *Nazca*. The other labels do not have a great influence on the meaning of concept. At the same time they can prevent *S-Match* from producing the correct semantic relation. In our example *S-Match* can not find any semantic relation connecting *Nazca_Lines* and *Nazca*. The reason for this is *By_Region* label, which is meaningless in the sense it is defined only for readability and taxonomy partitioning purposes. An other example of this kind is

```
Google:/Top/Arts/Celebrities/A/Affleck,_Ben
Looksmart:/Top/Entertainment/Celebrities/Actors/Actors_A/
        Actors_Aa-Af/Affleck,_Ben/Fan_Dedications
```

Here, *A* and *Actors_A/Actors_Aa-Af* do not influence on the meaning of the concept. At the same time they prevent *S-Match* to produce the correct semantic relation holding between the concepts.

An optimized version of *S-Match* (*S-Match++*) has a list of meaningless labels. At the moment the list contains only about 30 words but it is automatically enriched in preprocessing phase. A general rule for considering natural language label as meaningless is to check whether it is used for taxonomy partitioning purposes. For example, *S-Match++* consider as meaningless the labels with the following structure *by* $\langle word \rangle$, where $\langle word \rangle$ stands for any word in natural language. However, this method is not effective in the case of labels composed from alphabet letters (such as *Actors_Aa-Af* from previous example). *S-Match++* deals with the latter case in the following way: the combination of letters are considered as meaningless if it is not recognized by WordNet, not in abbreviation or proper name list, and at the same time its length is less or equal to 3. The addition of these techniques allowed to improve significantly the *S-Match* matching capability. The number of mappings found by the system on TaxME dataset increased by 15%. This result gives us an evidence to incrementality of the dataset (i.e., the dataset allows to discover the weaknesses of the systems and gives the clues to the systems evolution).

Analysis of *S-Match* results on TaxME allowed to identify 10 major bottlenecks in the system implementation. At the moment we are developing ad hoc techniques allowing to improve *S-Match* results in this cases. The current version of *S-Match* (*S-Match++*) contains the techniques allowing to solve 5 out of 10 major categories of mismatches. Consider Figure 5. It contains the results of comparative evaluation *S-Match++* against the other systems. As from the figure *S-Match++* significantly outperforms all the other systems. It found about 60% of mappings in all the matching tasks, what is twice better than *S-Match* result. This significant improvement would hardly be possible without comprehensive evaluation on TaxME dataset.

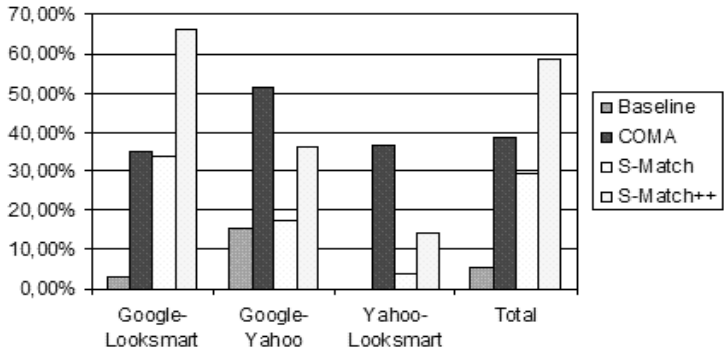

Fig. 5. Percentage of correctly determined mappings(Recall)

6.4 Correctness

We manually analyzed correctness of the mappings provided by TaxME. At the moment 60% of mappings are processed and only 2-3% of them are not correct. Taking into account the notion of idiosyncratic classification [12] (or the fact that human annotators on the sufficiently big and complex dataset tend to have resemblance up to 20% in comparison with their own results), such a mismatch can be considered as marginal.

7 Conclusions

In this paper we have presented a mapping dataset which carries the key important properties of *Complexity*, *Incrementality*, *Discrementality* and *Correctness*. We have evaluated the dataset on two state of the art matching systems representing different matching approaches. As from the evaluation, the dataset can be considered as a powerful tool to support the evaluation and research on the matching solutions.

The ultimate step which needs to be performed is to acquire the user mappings for TaxME dataset. We have already arranged such a kind of test and the results though preliminary are promising. Unfortunately at the moment more significant statistics needs to be collected in order to further improve TaxME.

Acknowledgment

We would like to thank Claudio Fontana and Christian Girardi for their helpful contribution in crawling and processing the web directories. We also would like to thank Pavel Shvaiko and Ilya Zaihrayev for their work on S-Match.

This work has been partially supported by the European Knowledge Web network of excellence (IST-2004-507482).

References

1. P. Avesani. Evaluation framework for local ontologies interoperability. In *AAAI Workshop on Meaning Negotiation*, 2002.
2. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.
3. S. Bergamaschi, S. Castano, and M. Vincini. Semantic integration of semistructured and structured data sources. *SIGMOD Record*, (28(1)):54–59, 1999.
4. P. Bouquet, L. Serafini, and S. Zanobini. Semantic coordination: a new approach and an application. In Fensel D., Sycara K. P., and Mylopoulos J., editors, *The Semantic Web*, volume 2870 of *LNCS*, Sanibel Island, Fla., 20-23 October 2003.
5. H. Doan, P. Domingos, and A. Halevy. Learning to match the schemas of data sources: A multistrategy approach. *Machine Learning*, 50:279–301, 2003.
6. M. Ehrig and Y. Sure. Ontology mapping - an integrated approach. In Christoph Bussler, John Davis, Dieter Fensel, and Rudi Studer, editors, *Proceedings of the First European Semantic Web Symposium*, volume 3053 of *Lecture Notes in Computer Science*, pages 76–91, Heraklion, Greece, MAY 2004. Springer Verlag.
7. J. Euzenat and P. Valtchev. An integrative proximity measure for ontology alignment. In *Proceedings of Semantic Integration workshop at International Semantic Web Conference (ISWC)*, 2003.
8. J. Euzenat and P. Valtchev. Similarity-based ontology alignment in OWL-lite. In *Proceedings of European Conference on Artificial Intelligence (ECAI)*, pages 333–337, 2004.
9. F. Giunchiglia and P. Shvaiko. Semantic matching. *The Knowledge Engineering Review Journal*, (18(3)):265–280, 2003.
10. F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-match: an algorithm and an implementation of semantic matching. In Bussler C., Davies J., Fensel D., and Studer R., editors, *The semantic web: research and applications*, volume 3053 of *LNCS*, Heraklion, 10-12 May 2004.
11. F. Giunchiglia, M. Yatskevich, and E. Giunchiglia. Efficient semantic matching. In *Proceedings of the 2nd european semantic web conference (ESWC'05)*, Heraklion, 29 May-1 June 2005.
12. D. Goren-Bar and T.Kuflik. Supporting user-subjective categorization with self-organizing maps and learning vector quantization. *Journal of the American Society for Information Science and Technology JASIST*, 56(4):345–355, 2005.
13. H.H.Do and E. Rahm. COMA - a system for flexible combination of schema matching approaches. In *Proceedings of Very Large Data Bases Conference (VLDB)*, pages 610–621, 2001.
14. R. Ichise, H. Takeda, and S. Honiden. Integrating multiple internet directories by instance-based learning. In *IJCAI*, pages 22–30, 2003.
15. J. Madhavan, P. A. Bernstein, and E. Rahm. Generic schema matching with cupid. *The Very Large Databases (VLDB) Journal*, pages 49–58, 2001.
16. B. Magnini, M. Speranza, and C. Girardi. A semantic-based approach to interoperability of classification hierarchies: Evaluation of linguistic techniques. In *Proceedings of COLING-2004*, August 23 - 27, 2004.
17. D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder. An environment for merging and testing large ontologies. In *Proceedings of International Conference on the Principles of Knowledge Representation and Reasoning (KR)*, pages 483–493, 2000.
18. S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: A versatile graph matching algorithm. In *Proceedings of International Conference on Data Engineering (ICDE)*, pages 117–128, 2002.

19. Noy N. and Musen M. A. Anchor-prompt: Using non-local context for semantic matching. In *Proceedings of workshop on Ontologies and Information Sharing at International Joint Conference on Artificial Intelligence (IJCAI)*, pages 63–70, 2001.
20. E. Rahm and P. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, (10(4)):334–350, 2001.
21. P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. *Journal on Data Semantics*, IV, 2005.
22. Y. Sure, O. Corcho, J. Euzenat, and T. Hughes. *Evaluation of Ontology-based Tools*. Proceedings of the 3rd International Workshop on Evaluation of Ontology-based Tools (EON), 2004. http://CEUR-WS.org/Vol-128/.
23. C. Welty and N. Guarino. Supporting ontological analysis of taxonomic relationships. *Data and Knowledge Engineering*, (39(1)):51–74, 2001.
24. L. Xu and D.W. Embley. Using domain ontologies to discover direct and indirect matches for schema elements. In *Proceedings of Semantic Integration workshop at International Semantic Web Conference (ISWC)*, 2003.