Evaluating Instance-based Matching of Web Directories

Sabine Massmann
Department of Computer Sciences
University of Leipzig
Germany
massmann@informatik.uni-leipzig.de

Erhard Rahm
Department of Computer Sciences
University of Leipzig
Germany
rahm@informatik.uni-leipzig.de

ABSTRACT

Web directories such as Yahoo or Google Directory semantically categorize many websites and are heavily used to find relevant websites in a particular domain of interest. Mappings between different web directories can be useful to integrate the information of different directories and to improve query and search results. The creation of such mappings is a challenging match task due to the large size and heterogeneity of web directories. Our study evaluates to what degree current match technology can be used to automatically determine directory mappings. We further propose specific instance-based match techniques utilizing the URL, name and description of the categorized websites. We evaluate the instance-based approaches for different similarity measures and study their combination with metadata-based approaches.

Keywords

Instance-based Matching, Mapping Discovery, Web Directories.

1. INTRODUCTION

Web directories are ontologies to semantically categorize websites. The categories are typically hierarchically organized so that websites of a subcategory also belong to the respective supercategories. Web directories such as Yahoo or Google Directory are heavily used to find relevant websites in a particular domain of interest, typically by navigating the directory structure or by using search queries.

While previous studies focused on automatically classifying websites, i.e. the assignment of websites to categories, we are interested in finding equivalence mappings between different web directories. Such match mappings identify additional related websites for categories. They can thus be used to integrate the information of different directories, to improve query results, or to generate website recommendations.

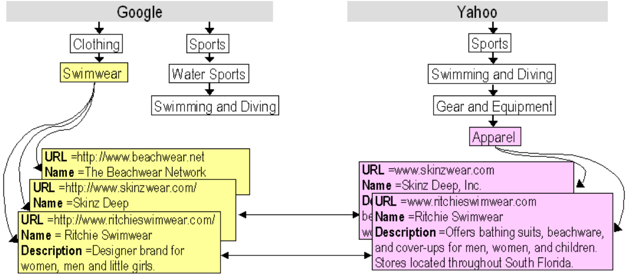

Google
Clothing
Sports
Swimwear
Water Sports
Swimming and Diving
Yahoo
Sports
Swimming and Diving
Gear and Equipment
Apparel
URL =http://www.beachwear.net
Name =The Beachwear Network
URL =http://www.skinzwear.com/
Name =Skinz Deep
URL =http://www.ritchieswimwear.com/
Name = Ritchie Swimwear
Description =Designer brand for women, men and little girls.
URL =www.skinzwear.com
Name =Skinz Deep, Inc.
URL =www.ritchieswimwear.com
Name =Ritchie Swimwear
Description =Offers bathing suits, beachware, and cover-ups for men, women, and children. Stores located throughout South Florida.

Figure 1. Portions of two web directories with associated instances

Proceedings of the 11th International Workshop on Web and Databases (WebDB 2008), June 13, 2008, Vancouver, Canada

The example in Figure 1 shows portions of two web directories for online shops: Google Directory[1] (left side) and Yahoo[2] (right side). Shop websites are classified into categories such as `Clothing` and `Sports`. The example assumes that the Google category `Swimwear` is matching to the Yahoo category `Apparel` (under `Swimming and Diving`). These categories not only have different names but are also differently placed in the directory structure (`Swimwear` is a subcategory of `Clothing` for Google and not under `Sports` as `Apparel` in the Yahoo directory).

The example thus illustrates some of the semantic heterogeneity problems that make automatic matching so challenging, especially for large directories with many categories and thus potentially many differences to other directories. A discussion of different kinds of heterogeneity can be found in [7], especially the distinction between *terminological heterogeneity* (e.g. use of synonyms, homonyms, abbreviations, different languages) and *conceptual heterogeneity* (differences in coverage, structure, granularity and perspective). A specific challenge is that categories within a web directory often overlap, i.e. a website (or even a subcategory) may explicitly be assigned to several categories. Such a redundancy increases the likelihood that a website will be found during navigation. A consequence of such redundancy (and of conceptual heterogeneity) is that a match mapping between directories is typically not 1:1, but n:m, i.e. a category of the first directory may be similar to several categories in the second directory.

In this study we want to evaluate to what degree current approaches for schema and ontology matching can successfully be applied for matching web directories. In particular, we consider previous approaches for metadata-based matching using information such as category names and the structure of the directories. We further propose specific instance-based match techniques using information about the instances, i.e. websites, assigned to the categories. This is motivated by the observation that the real semantics of a category may be better expressed by the actual instances assigned to the category than by metadata such as the category name. While instance-based matching has been studied before [7, 8] we focus on websites as a specific and complex kind of instances. We use several methods to consider website properties such as URL, name and description. We further utilize for matching when the URLs of different directories overlap. For example, the correspondence between the `Swimwear` and `Apparel` categories in Figure 1 could be derived from the fact that these categories share two (of three) websites. We evaluate the instance-based matching using different similarity measures and preprocessing. Moreover we combine the

[1] http://www.google.com/Top/Shopping/

[2] http://dir.yahoo.com/Business_and_Economy/Shopping_and_Services/

instance-based algorithms with metadata-based approaches and recommend a default strategy for matching directories.

The rest of this paper is organized as follows. In the next section we briefly discuss some additional related work. Section 3 describes the instance-based match algorithms. In section 4 we describe the evaluation setup and analyze the evaluation results for the different match algorithms and their combination. Section 5 concludes.

2. RELATED WORK

There is a huge literature on algorithms for schema and ontology matching [7, 8]. The approaches exploit a wide range of information, such as metadata (e.g. element names, schema structure), instance data and background knowledge, e.g., from dictionaries.

Matching web directories has received very little attention so far. The problem has been posed as a match task within the OAEI (Ontology Alignment Evaluation Initiative), which organizes yearly contests for ontology matching. Since 2005, one of the match tasks is the directory test [6] with a few thousand small match tasks from Yahoo, Google and Looksmart. These directories do not contain instance data so only metadata-based matching is possible. This is in contrast to our focus on instance-based matching of directories. The results of the OAEI directory test confirm that matching directories is very challenging. In 2005 the 7 participants found on average only 22% of all correct correspondences, in 2006 24%. In 2007 the recall was still less than 50%, at a precision of 57%. So over the years the participants were able to get better results but there is still much room for improvement.

Matching product catalogs is similarly difficult than matching web directories. In [9] an instance-based approach was studied to match the product catalogs of two online shops. Like in our study, the approach utilizes instance overlap between the ontologies (product catalogs). This was facilitated by the existence of unique instance ids (product EANs, European Article Number) which are not generally available. We will use the URLs to identify which websites are overlapping between directories but consider several variations of URL usage. We also use additional instance attributes for matching and perform a much more comprehensive evaluation (six match tasks not only one, precision and recall results not only estimates thereof).

3. MATCH ALGORITHMS

Techniques and prototypes that semiautomatically solve the match problem [7, 8] can be roughly classified as metadata-based, instance-based or mixed forms. For our study we use and extend the generic match system COMA++ [1, 5] that supports many metadata-based algorithms, e.g. using element names and structural information. It is based on the COMA architecture [3] and supports matching of both schemas and ontologies. In the evaluation, we will use several metadata-based matchers of COMA++ (see next section) to determine the similarity between web directory categories based on category names and the directory structure (super-/subcategories). In the following we will focus on the instance-based match algorithms we devise for matching web directories.

Instance-based matchers use the instances assigned to the categories of an ontology. These instances can be simple (one value) or complex (attributes and attribute values). In a web directory every category – the leaves as well as the inner ones – can have assigned website instances. We utilize the following website attributes which are commonly available:

- URL, e.g. “http://www.johns-books.com”,
- Name, e.g. “Johns book shop”,
- Description, e.g. “A crime and mystery bookstore”

Further examples are shown in Figure 1. For matching, we use theses website attributes to determine three instance sets per category: the URL set, name set, and description set. We propose three kinds of instance-based matchers (URL-based matchers, website name matcher, description matcher), each focusing on one of the three kinds of instance sets. We first describe the matching using the URL sets.

Instance-based Matching using URLs

The motivation behind URL-based matching is that website URLs of different directories overlap. We assume that two categories are the more similar the higher the overlap of their URL sets is. The degree of the overlap can thus be used for similarity calculation. URL comparison is based on equality and not just similarity because small differences can lead to a totally different meaning, e.g. book vs. look vs. cook.

The algorithms using the URL sets depend on overlap. However, for some sources the overlap may be low. In such cases, the overlap and thus the match recall may be increased by using only parts of the URL, e.g. to deal with URL variations of the same website. To find the most effective URL usage we start using the original (full) URLs recorded in the web directories and apply different preprocessing steps to reduce the URLs. The simplified URLs will likely increase the overlap and may help to find more category correspondences.

We will evaluate the use of the following URL variations and simplifications:

- **Original**: Complete URLs as found in the web directories, *Example*: http://www.Test.com/Shop/
- **Simpl1**: Remove “/” at the end and parameters after “?”, lower case, *Example*: http://www.test.com/shop
- **Simpl2**: Simpl1 + Remove “http://” and “www.” at the beginning, *Example*: test.com/shop
- **Simpl3**: Simpl2 + Remove everything after the first “/”, *Example*: test.com
- **Simpl4**: Simpl3 + Remove domain (everything after the last “.”), *Example*: test

The URL-based matchers determine the similarity of categories by comparing their URL sets. In this paper we study four different measures for determining the URL-based similarity between categories. Base-k similarity, $Sim_{Base\text{-}k,}$ is the simplest similarity measure. It matches two categories if they share at least k URLs. Using k=1 is the most optimistic case because one common URL suffices for two categories to match and reach a similarity value of 1.

The other three measures are the dice, minimum and maximum similarity measures. They relate the number of shared URLs to the sizes of the URL sets. The dice similarity measure Sim_{Dice} [10] between two categories c_1 and c_2 of the category sets C_{D1} and C_{D2} of the directories D_1 and D_2 is defined as follows:

$$Sim_{Dice}(c_1,c_2) = \frac{2 \cdot |I_{c1} \cap I_{c2}|}{|I_{c1}| + |I_{c2}|} \in [0 \ldots 1], \forall c_1 \in C_{D1}, c_2 \in C_{D2}$$

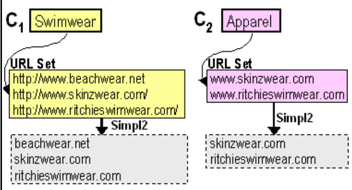
In the formula, $|I_{ci}|$ denotes the number of URLs that are associated to the category c_i. $|I_{c1} \cap I_{c2}|$ is the number of matched URLs that are associated to both categories, c_1 and c_2. The dice similarity is the relative overlap of the associated URLs.

In the case of larger cardinality differences between categories, the dice similarity values can become quite small, even if all URLs of the smaller category match to another category. We therefore consider the minimal similarity measure Sim_{Min}, which determines the URL overlap with respect to the smaller-sized category:

$$Sim_{Min}(c_1,c_2) = \frac{|I_{c1} \cap I_{c2}|}{\min(|I_{c1}|,|I_{c2}|)} \in [0 \ldots 1], \forall c_1 \in C_{D1}, c_2 \in C_{D2}$$

Additionally, we test the maximal similarity measure Sim_{Max}. It determines the URL overlap with respect to the larger-sized category.

Figure 2 illustrates the four similarity measures for a simple example (from Figure 1). Category c_1 has three assigned websites and category c_2 two. After a preprocessing of the URLs (e.g., Simpl2) we have two shared URLs between the categories. Depending on the used measure, the resulting similarity values range between 0.67 and 1.



$$Sim_{Base-1}(c_1,c_2) = 1$$

$$Sim_{Dice}(c_1,c_2) = \frac{2 \cdot 2}{3+2} = 0.8$$

$$Sim_{Min}(c_1,c_2) = \frac{2}{2} = 1$$

$$Sim_{Max}(c_1,c_2) = \frac{2}{3} = 0.67$$

Figure 2. Example for similarity measures $Sim_{Base\text{-}1}$, Sim_{Dice}, Sim_{Min} and Sim_{Max} (preprocessing Simpl2)

From the definition of the similarity measures it generally follows for the similarity between categories c_1 and c_2:

$$Sim_{Max}(c_1,c_2) \leq Sim_{Dice}(c_1,c_2) \leq Sim_{Min}(c_1,c_2) \leq Sim_{Base-1}(c_1,c_2) .$$

That is, Max and Dice are the most restrictive similarity measures while Min and Base-1 more easily achieve high similarity values.

Instance-based Matching using Names and Descriptions

These two instance matchers determine the category similarity by comparing the set of website names (name set) and the set of website descriptions (description set), respectively. The name and description strings are tokenized and all tokens (words, terms) are added to the name set and description set, respectively. These sets are actually multisets since all duplicates are retained to consider the relative frequency of terms. We use the document similarity measure TFIDF [2] to determine the similarity for the name sets or description sets. This measure considers both the term frequency and the inverse document frequency. It thus correctly reduces the weight of frequent (stop) words such as “and” or “online” while significant keywords achieve a high weight for determining the category similarity. The advantage of the website name and description matchers is that they do not depend on the existence of website overlap between directories. They can also find category correspondences for different websites with similar names or descriptions.

4. EVALUATION

This section contains the evaluation of the instance-based and metadata-based match approaches on several web directories. We first describe the used data sets and analyze the influence of URL preprocessing. Then we discuss the achieved results of the different match approaches based on the standard quality measures recall, precision and fmeasure. Finally, we examine different combinations of match algorithms.

4.1 Evaluation Data Sets

For our evaluation we use four web directories - Dmoz[3], Google[4], Yahoo[5] and Web[6]. The directories have been limited to online shops and categories. For categories we consider directly or both directly and indirectly associated website instances. Indirectly associated instances are websites assigned to the subcategories of a category. We use the German versions of the mentioned directories to simplify the manual creation of the perfect match result which is needed for the evaluation of recall and precision. Of course, the match approaches are not bound to a certain language but work with other languages as well.

Table 1. Statistical information about the web directories

		Dmoz	Google	Web	Yahoo
#Categories		746	728	418	3,234
#Inner / #Leaves		207 / 539	202 / 526	56 / 362	959 / 2,275
#Categories having direct instances		738	720	380	3,143
#Direct instances		15,304	15,082	13,673	34,949
#Direct inst. per categ.		21	21	36	11
Avg. char. length of a URL / name / descr.		28 / 21 / 119	28 / 20 / 119	28 / 26 / 92	28 / 21 / 70
Avg. size of set	URL	21	21	36	11
	name	55	54	110	29
	descr.	272	275	374	86

Table 1 provides statistical information about the four web directories. Yahoo is by far the largest directory (more than 3,200 categories) and classifies the highest number of websites (35,000). Google is based on Dmoz so that these directories are similar in size and structure. The “Web” directory is comparatively small but contains almost as many instances as Dmoz and Google.

Most of the categories – even the inner ones – have directly associated instances. Yahoo categories have on average 11 instances, Dmoz and Google categories 21 and Web categories the most – 36 instances. Figure 3 provides more details on the size distribution of categories. Most Dmoz and Google categories have at least 5 instances whereas more than half of the Yahoo categories have only 4 and fewer instances. The categories of the Web directory differ the most in their size distribution – a quarter has fewer than 5 instances and almost another quarter has more than 50 instances. Some categories are very large. Dmoz and

[3] http://www.dmoz.org/World/Deutsch/

[4] http://directory.google.de/Top/World/Deutsch/

[5] http://de.dir.yahoo.com/

[6] http://dir.web.de/

Google have a category with more than 200 associated instances, Web has one with almost 400 and Yahoo one with more than 800 websites.

Table 1 also contains information about the average size of the URL sets, name sets, and description sets used for instance-based category matching (as described in the previous section). The sizes of the URL sets correspond to the number of associated instances. The name sets contain 2-3 terms per website while the description size varies more. Larger size differences between the instance sets (URL, names, descriptions) tend to reduce the similarity of categories and thus complicate instance-based matching.

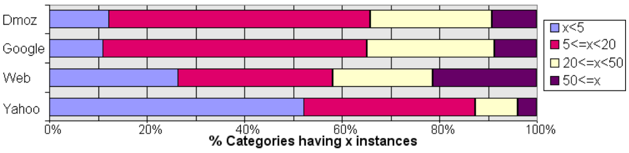


Figure 3. Instance distribution over categories

The four directories result into six match tasks which we use for evaluation and for which we determined a reference mapping. Table 2 shows the size of these reference mappings and indicates to which degrees the directories could be matched with each other. The mappings were manually created by looking into the names and descriptions of the categories and associated websites. Each mapping contains some hundred correspondences – in total 2,245 correspondences for all mappings. Only for the match task Dmoz↔Google almost all categories are covered because Google is based on Dmoz. For the other five match tasks the percentage of covered categories lies between 7% and 55%. So the majority of categories does not have a matching category in the other directories. This rather low similarity between directories can easily lead to the generation of wrong correspondences and makes matching challenging.

Table 2.Statistical information about the mappings

Match task	Dmoz↔ Google	Dmoz↔ Web	Dmoz↔ Yahoo	Google↔ Web	Google↔ Yahoo	Web↔ Yahoo
# Corresp.	729	218	436	211	416	235
Covered categories	98% ↔ 100%	29% ↔ 50%	55% ↔ 13%	29% ↔ 48%	55% ↔ 12%	52% ↔ 7%

4.2 Preprocessing and Overlap of Instances

To find out the applicability of URL-based matching we first analyze the URL overlap between the four directories and the impact of the different kinds of URL preprocessing. Figure 4 compares the five variants (Original, Simpl1 to Simpl4) with respect to the total number of URLs, over all match tasks, that are shared between two, three or all four directories. The main part of the about 60,000 distinct URLs appears in only one of the directories, the number of URLs shared between two directories lies between 10,800 and 14,600, the number of URLs shared between all directories after all preprocessing steps is limited to merely about 500. This illustrates that the overall applicability of URL-based matching is quite limited, albeit still relevant. Furthermore, some of the match tasks can benefit quite significantly as we will see.

We further observe that the more the URLs are simplified the higher the overlap becomes between the web directories. The first step, Simpl1, has the greatest influence. The number of found URLs shared by all directories increases from 53 to 474 while the number of URLs occurring in only one of the directories decreases by 16%. The further steps Simpl2 to Simpl4 only lead to small further increases in the number of shared URLs. As we will see in the next subsection, only Simpl2 will actually result in an improved match result while the further simplifications primarily lead to wrong category correspondences.

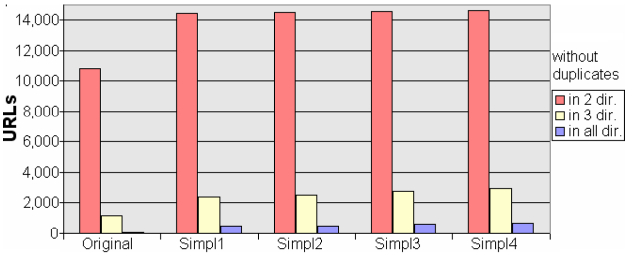


Figure 4. Influence of URL preprocessing on overlapping

Table 3 shows the URL overlap for each match task using Simpl2 preprocessing, e.g., 1,561 URLs of Google appear in Yahoo. The shown percentages indicate the resulting shares of a directory's URLs that are shared in both directories of the match task. For example the common Google↔Yahoo URLs represent 10% of Google's URLs and 5% of Yahoo's URLs. The small Yahoo values are influenced by its large relative size. The match task Dmoz↔Google has the highest overlap, about 85%. The overlap of the other match tasks is also rather low, between 5% and 13%, thereby limiting the applicability of URL-based instance matching.

Table 3. Statistical information about URL overlap (Simpl2)

Match task	Dmoz↔ Google	Dmoz↔ Web	Dmoz↔ Yahoo	Google↔ Web	Google↔ Yahoo	Web↔ Yahoo
URL Overlap (without Dupl.)	12,963	1,650	1,485	1,679	1,561	1,695
Covered URLs (with Dupl.)	85% ↔ 87%	11% ↔ 12%	10% ↔ 5%	11%↔ 13%	10% ↔ 5%	13% ↔ 5%

Due to space restrictions we omit details on the preprocessing of website names and descriptions. The word overlap in website names is lower than in website descriptions because the former contain more proper nouns and person names. The portion of name and description words occurring in all directories (7% and 10%) is much higher than for URLs (less than 1%). This indicates that name- and description-based instance matching may be able to find correspondences even between categories which have no (or few) shared instances.

4.3 Instance-based Matchers

First we study the effectiveness of URL-based instance matching. Figure 5 shows the average fmeasure results for the 5 variations of URL usage (Original – Simpl4) and six similarity measures (Base-1, Base-2, Base-3, Dice, Min, Max) averaged over the six match tasks. The graphs show that URL-based matching is surprisingly successful despite the comparatively small URL overlap between the directories. It achieves an average fmeasure value of up to 0.57 when only directly associated instances are considered (Figure 5, left) and an fmeasure of 0.6 for both directly and indirectly associated instances.

The most conservative similarity measures, Max and Dice, obtain the best and almost identical results with a slight advantage for Dice. This is because they minimize the number of wrong category correspondences thereby maximizing precision. Recall was still good since we do not prescribe a minimal similarity threshold but accepted a correspondence c_1 - c_2 if both c_2 resp. c_1 is the best matching category for c_1 resp. c_2 (selection strategy "both" of COMA [3], stable marriage). Furthermore, we do not restrict ourselves to 1:1 matches but consider all matching categories for which the similarity values are within a small delta difference from the best matching category. This approach proved to be quite robust for Dice and Max against differences in the sizes of URL sets. Furthermore, Max and Dice were the only measures benefitting from the use of indirectly associated instances. For the other similarity measures, the consideration of indirectly associated websites led to many wrong correspondences and much reduced fmeasure values.

Base-k matches two categories if they share at least k URLs.The most liberal similarity measure, Base-1, performs worst because it introduces too many wrong correspondences (poor precision). Base-2 performed better and similar to Min for Direct URLs. Min proved to be less stable than Max or Dice since it generates often correspondences even for very small URL overlaps.

Regarding the different variations of URL usage, we observe that URL preprocessing always pays off substantially. The first preprocessing step Simpl1 has the biggest influence on the results (as already observed from Figure 4). Simpl2 generally achieves another slight improvement while the further steps lead to decreasing fmeasure results. This is particularly pronounced for Base-1 where Simpl3 leads to a much reduced fmeasure compared to Simpl2. This is because Simpl3 often introduces false duplicate URLs by removing critical URL parts (e.g. "/johns-shop") so that different shops hosted on the same site (e.g. "stores.ebay.de", home.t-online.de) can no longer be correctly differentiated.

Based on the evaluation results we conclude that URL-based matching should be based on Simpl2 URL preprocessing and the Dice (or Max) similarity measure.

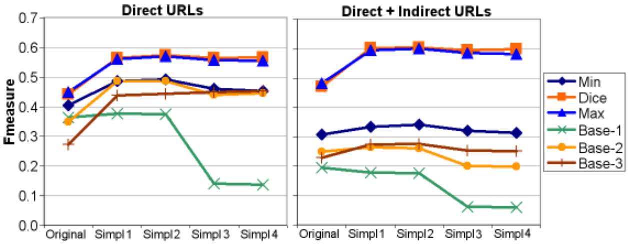


Figure 5. URL-based matching with different preprocessing and different similarity measures

Now we want to compare the URL-based results for Dice and Simpl2 with the results for name-based and description-based instance matching using TFIDF. Figure 6 presents the average precision, recall and fmeasure results for all match tasks. As already mentioned, we see that URL-based matching achieves not only good precision but also good recall. Name-based and description-based matching are also quite balanced in their precision and recall results. Description-based matching (fmeasure 0.58) is slightly better than name-based matching (fmeasure 0.54) because names are shorter than descriptions and often contain proper nouns and person names. Description-based matching is almost as good as URL-based matching with Dice, but not better. It suffered from highly diverse descriptions, partly influenced by the occurrence of many composite German words.

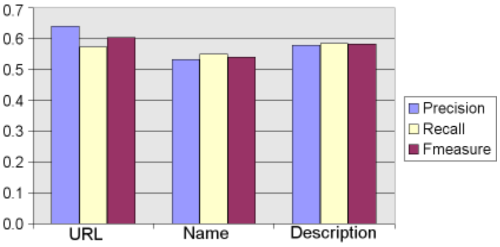


Figure 6. Average Results for instance-based matching

4.4 Metadata-based Matchers

Metadata-based matchers have the advantage that they do not depend on the existence of instance data. For our evaluation we apply six metadata-based matchers of COMA++ to all match tasks: Name, NameStat, Path, Children, Leaves and Parents [4]. The average results of the six web directories tasks are shown in Figure 7. The matchers use 31 synonym pairs that have been added manually. They led to a slight improvement, e.g. an fmeasure increase of 2.8% for the Path matcher. Overall, the best average fmeasure per matcher is 0.61 (Path) and thus similarly effective than the use of instance-based matching.

The *Name* matcher calculates the similarity of categories by comparing their names using Trigram and the given synonyms. This matcher finds the most correct correspondences (best recall) but also many wrong ones because it does not consider the context of the categories. *NameStat* combines the matcher Name with statistical information such as the number of subcategories. It finds less correct correspondences than Name alone because of the heterogeneity of the directories that lead to different statistical information and avoids finding the right correspondences.

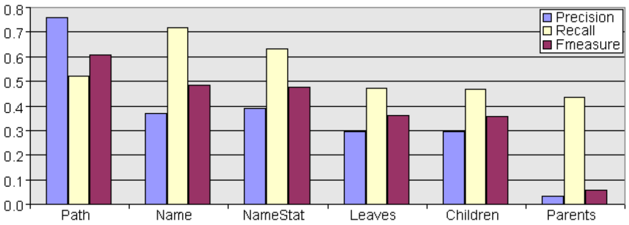


Figure 7. Average Results for metadata-based matching

Path uses all category names from the root to the specified category and is thus a simple approach to consider the context of categories. Compared to Name it achieves an excellent precision (0.76) albeit at a reduced recall. Overall Path achieves the highest fmeasure value (0.61) of all six metadata-based matchers. This is in line with previous schema matching evaluations of COMA / COMA++ [3, 4] where Path was the most effective single matcher.

The structural matchers Leaves, Children and Parents are rather ineffective due to the high heterogeneity of the web directories. *Leaves* and *Children* compare sets of elements, which are either the leaves or the children of two inner elements. Both miss the correspondences between inner categories and leaf categories which account for 22% of all correspondences. *Parents* derives the similarity between categories from the similarity between their parents. It has a very low precision because the structure of the web directories is very broad and flat leading to many wrong correspondences.

4.5 Matcher Combination

So far we only evaluated single instance-based and metadata-based matchers. We use the flexibility of COMA++ to combine an arbitrary set of matchers, i.e. we combine their similarity values for selecting category correspondences. Such combinations are often able to compensate weaknesses of individual matchers and thus improve results. In our evaluation we combined the three instance-based matchers of subsection 4.3 (URL matching using Dice/Simpl2 and using both directly and indirectly associated URLs, website name matching, description matching) and the six metadata-based matchers of subsection 4.4. We tested all possible combinations using different configurations for the similarity value calculation and correspondence selection. Due to space limitations we only present results for some combinations in Figure 8. One of them is a default strategy that we describe later. The shown precision, recall, and fmeasure results are averages for all match tasks. The vertical black lines indicate the minimum and maximum values for the six underlying match tasks.

The best single matchers – URL for instance-based and Path for metadata-based– achieve each about a fmeasure of 0.60. Their combination significantly improves both recall and precision and results in an average fmeasure of 0.72. This confirms the expectation that combined matchers are useful for mapping website directories. But does combining more matchers further improve results? Not always. For example combining all 9 matchers achieves a fmeasure of 0.68. It thus performs better than the best single matcher but cannot outperform the combination of only 2 matchers, e.g. Path and Name. Yet combining more than two matchers is helpful. One successful combination includes the best two metadata-based matchers (Path, Name) and the best two instance-based matchers (URL, Description). The average fmeasure improves to 0.76, the minimum recall value for a match task is 0.68.

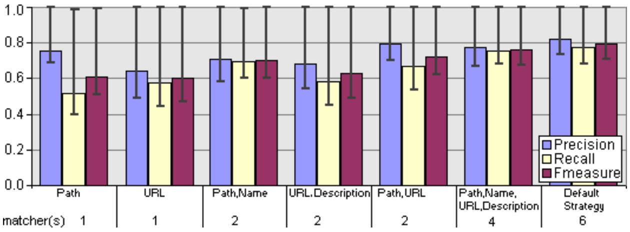


Figure 8. Average Results for single matchers and matcher combinations

Looking at the results for all match tasks we observe that some matcher combinations perform well for all tasks while other combinations achieve good results only for the simplest match task Dmoz↔Google. Overall we find the following combination to be effective for all tasks and thus recommend it as a promising default strategy for matching web directories:

- Combination of 3 instance-based matchers (URL, website name, description) and 3 metadata-based matchers (Path, Name, Parent) by applying an averaging of the individual similarity values
- Selection of correspondences by using the “both” and delta strategies (see 4.3, [3]).

This default strategy achieves an average precision, recall, and fmeasure of 0.82, 0.77, and 0.79, respectively. The best result per match task for any matcher combination was at most 2% better indicating that the default strategy is suitable for all tasks.

5. CONCLUSIONS

To address the challenging problem of mapping web directories we proposed and evaluated the use of instance-based methods and their combination with metadata-based matchers. Our instance-based methods utilize the URLs of classified websites as well as their names and descriptions. For URL matching we considered different preprocessing alternatives and different similarity measures to derive the similarity of categories from their overlap of assigned websites. Our evaluation used challenging match tasks on four real-life web directories of online shops. Despite a moderate similarity and website overlap of the directories the match problems could be solved to a large degree (average fmeasure for six match tasks of up to 0.79). URL matching alone achieved an average fmeasure of 0.6 after preprocessing URLs and using a dice similarity measure. We thus propose to use this method for mapping web directories in combination with other instance-based and metadata-based matchers. We identified a matcher combination of three instance-based and three metadata-based matchers that could successfully solve all match tasks.

6. REFERENCES

[1] Aumueller, D.; Do, H.- H.; Massmann, S. and Rahm, E.: Schema and ontology matching with COMA++. SIGMOD Conference. USA, 2005: 906-908.

[2] Cohen, W. ; Ravikumar, P. and Fienberg, S.: A comparison of string distance metrics for name-matching tasks. In Proceedings of the IJCAI-2003.

[3] Do, H.-H. and Rahm, E.: COMA - A System for Flexible Combination of Schema Matching Approaches, Proc. 28th Intl. Conference VLDB, Hongkong, Aug. 2002

[4] Do, H.-H. and Rahm, E.:Matching large schemas: Approaches and evaluation. Information Systems, Volume 32, Issue 6, September 2007, Pages 857-885

[5] Engmann, D. and Massmann, S.: Instance Matching with COMA++.BTW Workshops. Germany, 2007:28-37.

[6] Euzenat, J.; Isaac, A.; Meilicke, C.; Shvaiko, P.; Stuckenschmidt, H.; Šváb, O.; Svátek,V.; van Hage, W.R. and Yatskevich, M.: Results of the Ontology Alignment Evaluation Initiative 2007. In Proceedings of the Ontology Matching Workshop at ISWC'07.

[7] Euzenat, J.; Shvaiko, P.: Ontology Matching. Springer-Verlag, 2007.

[8] Rahm, E. and Bernstein, P.: A survey of approaches to automatic schema matching. The VLDB Journal 10, 4 (Dec. 2001), 334-350.

[9] Thor,A.; Kirsten, T.; Rahm, E.: Instance-based matching of hierarchical ontologies. BTW. Germany, 2007: 436-448

[10] van Rijsbergen, C. J.: Information retrieval. Butterworths, London, 2nd edition, 1979.