# Improving Ontology Matching Using Meta-level Learning

Kai Eckert, Christian Meilicke, and Heiner Stuckenschmidt

KR&KM Research Group
University of Mannheim, Germany
{kai,christian,heiner}@informatik.uni-mannheim.de

**Abstract.** Despite serious research efforts, automatic ontology matching still suffers from severe problems with respect to the quality of matching results. Existing matching systems trade-off precision and recall and have their specific strengths and weaknesses. This leads to problems when the right matcher for a given task has to be selected. In this paper, we present a method for improving matching results by not choosing a specific matcher but applying machine learning techniques on an ensemble of matchers. Hereby we learn rules for the correctness of a correspondence based on the output of different matchers and additional information about the nature of the elements to be matched, thus leveraging the weaknesses of an individual matcher. We show that our method always performs significantly better than the median of the matchers used and in most cases outperforms the best matcher with an optimal threshold for a given pair of ontologies. As a side product of our experiments, we discovered that the majority vote is a simple but powerful heuristic for combining matchers that almost reaches the quality of our learning results.

## 1 Motivation

Despite significant research efforts automatic ontology matching is one of the unfulfilled promises of semantic web technologies and might turn out to become the Achilles' heel for large scale applications of ontologies on the web. So far, a significant number of automatic matching systems have been developed that address the matching problem by applying different heuristics, most of which are based on the similarity of representations. Depending on the kind of heuristics used, these matchers show a varying quality on different matching problems. This problem is typically addressed by approaches for selecting the optimal matcher based on the nature of the matching task and the known characteristics of the different matching systems. Such an approach that has been based on extensive interviews and tests is described in [1]. Another typical approach for dealing with the problem of adapting to a given matching task is to apply machine learning techniques for learning the optimal configuration of a matcher for a given dataset [2]. This approach amounts to determining the right heuristics and the appropriate parameters to be used in order to achieve the best result.

The approach proposed in this paper differs from the above mentioned ones by not signing up to a specific matcher but trying to exploit the results of available matchers which are treated as a black-box. This has the advantage that the weaknesses of individual matchers are compensated. Further, the approach settles on the idea that by using

L. Aroyo et al. (Eds.): ESWC 2009, LNCS 5554, pp. 158–172, 2009.

multiple matchers we can benefit from the high degree of precision of some matchers and at the same time the broader coverage of other matchers to complete the picture where highly precise matchers did not produce results.

Our approach is verified in a realistic setting on different standardized and open available datasets - details are discussed in Section 3.2. In particular, we use the results of the past OAEI campaigns that provide us with a gold standard mapping as well as a large number of mappings created by different matching systems. Thus, we can train a classifier on the outcome of different matching systems and learn what combination of results from different matchers provides the best indication of a correct correspondence. This proves to outperform previous attempts of combining matchers which have often been based on ad hoc methods or had to be customized manually.

**Related Work.** An approach to solve the problem of selecting correspondences from a set of matcher-generated mappings within the context of argumentation frameworks is presented in [3]. While it is based on theoretically well founded principles, the authors describe first experimental results as inconclusive and point out the necessity of further research. The general idea of combining individual matchers into a combined matching system constitutes also a principle used inside many existing matchers (e.g. [4,5]). The difference, however, is in the way the results of the individual matchers are combined. In contrast to existing approaches that combine a number of specific predefined classifiers, our approach is a more general one, as we do not make any assumptions about the individual matchers to be combined apart from the fact that they provide their results in a standardized format.

An approach that is very similar to ours is implemented in the GLUE System [6] that applies a meta-learning approach for generating matching hypotheses on the basis of multiple local classifiers that are trained on different aspects of the models to be matched. This approach, however requires that the input for meta learning is generated by specific probabilistic learning methods, in that case naive Bayes classifiers, that are integrated using a linear combination at the meta level. In our case, we do not make any assumptions about the matchers used and apply different machine learning techniques at the meta level. This makes our approach more widely applicable. Further, the evaluation in [6] is performed on a rather limited set of ontologies without the existence of commonly agreed reference alignments. We use two widely used benchmark datasets each containing multiple ontologies that have to be aligned. Looking at the results of [6] reveals that the meta-learning approach is dominated by the so-called content learner, which always performs almost as good as the integrated learning approach making the meta-learning step less important. As reported in section 3.3 our approach is in many cases significantly better than any of the local matchers which underlines the usefulness of our approach to meta-level learning.

Three other approaches in line with our ideas are described in [7], [8] and [9]. In [7] support vector machines are used to learn a classifier for mapping correctness based on a set of simple similarity measures. The classifier is evaluated on the benchmark datasets of the ontology alignment evaluation initiative and outperforms existing matching systems. The setting of the experiments, however, is a rather unrealistic one as all existing reference mappings are thrown together in one large training set and 10-fold cross-validation is used for computing the accuracy of the classifier. The results are only a

very unreliable estimation of the behavior that can be expected from the classifier for a realistic matching problem. To avoid this problem, we carefully design the evaluation scenario and test our method in a setting that can be expected for a real integration task (compare section 3.2). Further, the approach uses only similarity values as input for learning.

In [8] the authors focus on the use of different bayesian classifiers. Again their feature set is based on string distance measures typically used in matching systems to derive a syntactic similarity. The experimental results show that there is a strong correlation between different measures and the machine learning approach cannot significantly improve the results of the best individual measure. Contrary to this, we implemented a rich set of features describing different aspects of ontologies and correspondences. We show that using the additional feature set can significantly improve the classification result. In fact, we even show that confidence values of individual matchers, which are normally the result of a similarity estimation are not significant and do not contribute to the learning result.

The approach reported in [9] is probably the most similar to our work. Here decision trees and rule learners are used to learn rules for integrating the results of different matching systems, which is more or less the same we do. Similar to [7], the difference to our work is the restriction to confidence values respectively measured similarities as the basis for learning. The approach has been evaluated on a subset of the OAEI benchmark dataset, but no detailed results of the evaluation are provided making it hard to judge the quality of the proposed method. Our results suggest that basing the learning step just on confidence values is not a good choice (compare section 3.3).

**Contributions.** The contributions of this paper are the following:

- We present a new approach for combining different matching systems using machine learning techniques.
- We evaluate the approach in a realistic setting using well established benchmark datasets.
- We show that our approach systematically outperforms existing matching tools in the sense that it not only produces better results than the median of the matchers but also outperforms or measures up to the best matching system for every matching task we investigate.
- We identify a simple but very powerful heuristic for combining matching results that outperforms the best matching system and almost reaches the performance of the machine learning approach.

The paper is structured as follows: we first briefly discuss the problem of combining the result of different matching systems and present our approach for solving this problem in more detail. The major part of the paper is concerned with describing the setting and the results of the matching experiments we performed. We motivate the choice of the datasets and the setting of the experiments and compare our results with three different baselines. We conclude with a discussion of the results and their implication for ontology matching on the semantic web.

## 2 A Meta-level Learning Approach

As indicated in the introduction, we approach the problem of combining different matching systems by applying machine learning techniques on top of the results produced by different state of the art matching systems. Our approach is presented in more details in this section.

### 2.1 The Problem of Combining Matchers

The ontology matching can be defined as follows [10]: Given two ontologies $O_1$ and $O_2$, establish semantic relations - also referred to as correspondences - between certain matchable elements. In this work, we restrict ourselves to the detection of equivalence correspondences between named concepts and properties. This restriction is motivated by the state of the art in ontology matching as it is documented in the annual benchmarking activities of the ontology alignment evaluation initiative (cf. [11] and [12]). Most of the systems that have successfully participated in the benchmarking activities are limited to one-to-one mappings between named concepts and properties; only very few systems produce relations other than equivalence. Another reason for only considering equivalence is the absence of commonly accepted reference mappings that contain non-equivalence correspondences.

Besides a set of possible equivalence correspondences, matching systems often provide information about a level of confidence in the correctness of the correspondence. The meaning of this level of confidence has been subject of many discussions. According to Bouquet et al. [13] the confidence is

> ... a degree of trust (confidence) in that mapping (notice, this degree does not refer to the relation R, it is rather a measure of the trust in the fact that the mapping is appropriate ("I trust 70% the fact that the mapping is correct/reliable/..."). The trust degree can be computed in many ways, including users' feedback or log analysis.

The only formal requirement on the level of confidence mentioned in [13] is the existence of some partial order over the confidence values that allows the matcher to order its results according to its own belief in the correctness of the relation represented. According to the definition of Euzenat and Shvaiko, the result of a matching system can be described in terms of a set of correspondences of the form $O_1 : E = O_2 : E' \ (n)$ where $E$ and $E'$ are either both named concepts or properties and $n$ is a degree of confidence, normally a real number from the interval $[0, 1]$. An example could be $O_1 : \textit{Hotel} = O_2 : \textit{Accomodation} \ (0.6)$. This means that the matcher has a confidence of $0.6$ that concept *Hotel* in ontology $O_1$ describes the same objects as concept *Accomodation* in ontology $O_2$. In this definition, the rather vague notion of confidence causes serious problems when trying to combine the result of different matching systems as it cannot be guaranteed that they use the same notion of confidence, and in many cases the confidence values are not comparable because they have been computed based on fundamentally different principles. This means that if there are two other matchers that produce the correspondences $O_1 : \textit{Hotel} = O_2 : \textit{Accomodation} \ (0.4)$ and $O_1 : \textit{Hotel} = O_2 : \textit{Hotel} \ (0.7)$ it is not clear how to combine the results. On

the one hand, one might argue as follows: two matchers think that *Hotel* and *Accomodation* are equivalent concepts while only one does not agree. This means that *Hotel* = *Accomodation* should be accepted. On the other hand, one might raise an objection: the confidence in the truth of this statement is rather low compared to the confidence of the matcher which thinks that *Hotel* should be matched on *Hotel*. As different matchers can use very different methods for computing the confidence, there is no standard way of combining respectively comparing confidences to come to a conclusion.[1]

### 2.2 Combining Matchers Using Machine Learning

Our solution to the problem of combining different matchers is not to try to directly combine their output but to use machine learning techniques to train a classifier that decides whether two elements from different ontologies should be linked by an equivalence relation based on the output of different matching systems. As described above, this idea is similar to existing work in the area of ontology mapping. A major difference of our approach is that we do not restrict the learning approach to the output of the matchers, but add additional features as input for the learning approach.

The rationale for using these additional features is the observation that matchers more or less heavily rely on the existence of certain structures and information in the ontologies to be matched. Examples are the existence of additional concept descriptions encoded in labels or the existence of a WordNet synset the description can be linked to. In the first case, a matcher that uses information retrieval techniques to compare concept descriptions will outperform a matcher that just uses string matching on the names of concepts or properties. In the latter case a matcher that uses WordNet as background knowledge will detect more correspondences than a matcher that does not. Approaches like [1] make this information explicit for a limited set of known matching systems. By including certain properties of the ontologies and the elements to be matched into the learning process, we are able to take the strengths and weaknesses of matchers into account without having to know the concrete matchers. The fact that a certain matcher performs better if concepts can be linked to WordNet is detected in the learning step. This means that we do not need to know the characteristics of involved matchers as required precondition.

Figure 1 illustrates the generation of training data for the machine learning step with respect to the simple case of two matchers. In a first step, the matchers are run on the ontologies and generate two mappings. Then a training set is created that contains the union of both mappings. Correspondences contained in the reference mapping are positive examples, those that are not contained are negative examples. Besides the distinction between positive and negative examples indicated by the 'target' attribute, attributes include the results of the individual matchers in terms of an attribute indicating whether the correspondence was found by a certain matcher and what confidence was

[1] As described above, the GLUE approach solves this problem by only allowing individual matchers that use the same notion of confidence (in this case Bayesian probabilities). Their approach, however, does not solve the general problem of integrating arbitrary matching systems in an optimal way.

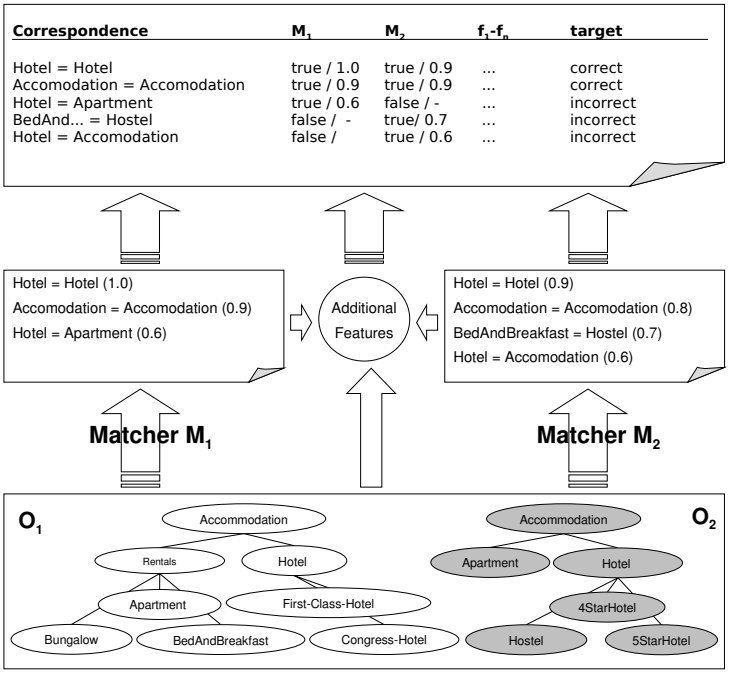


**Fig. 1.** Illustration of the Meta-level Approach to matcher combination

assigned by that matcher. A number of other features $f_1, \ldots, f_n$ that are extracted from the results of the matchers and the ontologies themselves are included. These features are motivated and described in detail in the following section.

The result of the learning step is a classifier modeling the training data set. Given a new matching problem, this classifier can be used to predict the correctness of correspondences. For this purpose, the same set of matchers used in the training phase is executed on the new matching problem. The resulting correspondences are used as input for the classifier together with the additional features that have to be computed for the new situation as well. The learned classifier can now be applied to the resulting descriptions of correspondences. In doing so, the correctness of a correspondence is decided based on matcher results and additional features.

### 2.3 Description of Features

The following set of additional features were used as input for our learning experiments. These features can roughly be separated into four groups:

1. **Matcher Features:** This group contains features that reflect the results of the involved matchers.
   *Matcher Found:* a boolean feature for each matcher that is *true*, if the corresponding matcher returned the correspondence in question as part of its mapping and *false* otherwise.

*Matcher Confidenc:* the confidence value specified by a particular matcher for a given correspondence.

*Matcher Vote:* a numerical feature with values between 0 and 1 reflecting the percentage of matchers that returned the correspondence as part of their mapping. Matcher Vote is a feature that follows the idea of a voting model to decide wether a given correspondence should be taken as correct. By modeling the feature this way we allow the classifier to learn how many individual votes are needed in combination with other feature values to decide the correctness of a correspondence.

2. **Ontology Features:** These features describe global characterstics of the two ontologies.

   *Ontology Ratios:* three features representing the ratio of the number of concepts, object properties and datatype properties between the two ontologies. We use the reciprocal value of the ratio, if the original value is above 1, thus the ratios are always between 0 and 1 with smaller values reflecting bigger differences in the number of matchable elements.

3. **Lexical Features:** As the lexical comparison of the texts belonging to the matching candidates are a common source of information for some matchers, we try to characterize these texts in this group of features.

   *String Equivalence:* a boolean feature that is *true*, if the normalized id of both elements are identical.

   *Number of Token Ratios:* These features are calculated in the same way as the Ontology Ratios and compare the number of tokens in id, label and comment. The rationale behind these features is the idea that lexical comparisons on these strings can only be effective if there is a similar amount of text available for both concepts. We decided for the number of tokens as a measure for the length of a text, as every token is a potential source for further analysis techniques.

   *Mean Significance:* We use the well known TF/IDF measure to calculate the mean significance $S$ of tokens in a given text $T$ against a text base $D$ containing some other texts. With $tf_t$ for the number of occurrences of token $t$ in $T$ and $df_t$ for the number of texts containing token $t$ in $D$, we get

$$S = \frac{1}{|T|} \cdot \sum_{t \in T} \left( (1 + \log(tf_t)) \cdot \log \left( |D| \cdot df_t^{-1} \right) \right). \tag{1}$$

   We use Equation 1 to calculate the mean significance of the comment against the id and the label, as well as the mean significance of the label against the id. Both features are calculated for each element in the correspondence in question. The mean significance is a measure for the additional information that can be obtained from a text (like the comment), if other texts (like id and label) are already known.

   *Wordnet Coverage.* As Wordnet is commonly used to get further information about common terms that might occur in the id or label of an element, we introduced the Wordnet Coverage as a feature that returns a nominal value that says whether *none*, *one* or *both* of the elements contain terms that can be found in Wordnet. The Wordnet Coverage is calculated separately for id and label.

4. **Structural Features:** These features reflect the structural environment of the two elements in question.
   *Type of Correspondence:* a nominal feature that characterizes the type of elements involved in a given correspondence. The result may be one of four values: *Concept - Concept*, *Object Property - Object Property*, *Datatype Property - Datatype Property* and *Object Property - Datatype Property*. This feature has been introduced to enable the classifier to learn in how far different matching systems might perform better with respect to matching certain types of ontological elements.
   *Node position.* The Node Position is used to get information about two special cases, namely, if *none*, *one* or *both* of the elements linked by a correspondence are root or leaf nodes in their respective ontologies. Obviously, techniques like e.g. similarity flooding will perform differently depending on the position of an element within the ontological hierarchy.

## 3 Experiments

We tested our approach using systematic experiments on the benchmark datasets provided by the ontology alignment evaluation initiative. The experiments, which are described in more detail in the following, show that our approach systematically outperforms state of the art matching systems.

### 3.1 Datasets

The ontologies used in our experiments have been part of the Ontology Alignment Evaluation Initiative (OAEI) over the previous years. The OAEI offers several tracks and sub tracks concerned with different types of matching problems.[2] We can distinguish between those tracks where automatically generated mappings are compared against a reference mapping (also referred to as gold standard) and tracks using other evaluation techniques. Only the matching problems of the former tracks can be used as both input and basis for evaluating our approach and are described in the following.

The test set of the *benchmark track* is based on one particular ontology #101 dedicated to the very narrow domain of bibliography and a number of alternative ontologies of the same domain. It consists of a series of synthetic ontologies (#1xx and #2xx series) and four real world ontologies #301 to #304 which have to be matched on ontology #101. In our experiments we only consider test cases #301 to #304 because we are in particular interested in how far our approach can successfully be applied to realistic matching problems. The test set of the *conference track* consists of 15 ontologies where each pair of ontologies constitutes a matching problem. These ontologies have been developed as part of the OntoFarm project [14] and describe the domain of conference organization covering the structure of a conference, involved actors, as well as issues related to submission and review process. Since october 2008 reference mappings for all possible combinations between five of these ontologies are available which we used in

[2] Detailed information available at `http://oaei.ontologymatching.org/2008/`

our experiments. These ontologies are CMT, ConfTool, Ekaw, Iasted and Sigkdd. The conference dataset can be seen as a much harder testcase compared to the benchmark dataset that is less heterogeneous and has been extensively studied over the past years. This claim is also supported by our experimental results reported in Section 3.3.

Since we base our experiments on relevant OAEI datasets, we can make use of a rich set of diverse matching systems. In 2008 thirteen matching systems submitted mappings to the OAEI. We only included those matching systems generating a non boolean confidence value in the range $[0, 1]$ because we expected the confidence value to be an important feature for our learning approach. Matchers considered are *Aroma* [15], *ASMOV* [5], *DSSim* [16], *Lily* [17], *RiMOM* [18], *SAMBOdtf* and *SAMBO* [19][3]. While all of these systems submitted mappings for the benchmark track, only *ASMOV*, *DSSim* and *Lily* participated in the conference track. Thus, for each matching problem of the benchmark track there are mappings available generated by seven state of the art matching systems implementing diverse matching approaches, while for the conference track we can access the mappings of three matching systems.

### 3.2 Experimental Setup

Figure 2 shows the structure of the data sets and the derived experimental setting. On the left side the structure of the benchmark dataset is depicted as tree. Ontology #101 can be seen as a central knowledge base with several knowledge sources linked to it via mappings. Contrary to this, the conference data set on the right side is a full mesh where no ontology plays an outstanding role. We assume a setting, where a system of linked ontologies, e.g. #101, #301, #302 and #303 (colored white) exists. These ontologies have been linked in the depicted way and the mappings have been set up and verified by domain experts. We further assume that another knowledge source has to be added to this structure, in this case ontology #304. Given a set of matching systems respectively automatically generated alignments, we can profit from the reference mappings available in the way described above and learn a classifier $C$.

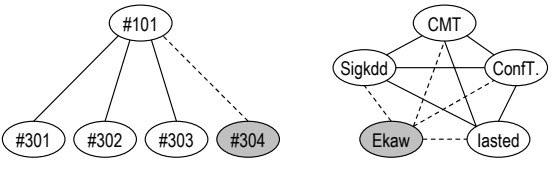


**Fig. 2.** Mapping structure of the benchmark (tree on the left) and conference data set (full mesh on the right) and experimental setting

Once we learned a classifier $C$, we apply it on the set of automatically generated mappings $M_{Aroma}, \ldots, M_{SAMBOdtf}$. We compare the mapping $M_C$, which consists of all those correspondences classified as correct by applying $C$, against reference mapping

[3] We did not include results of matching system SPIDER [20] due to its focus on generating complex non-equivalence correspondences not included in reference mappings.

$R_{\#304}$ to compute precision $p$, recall $r$ and f-measure $f$, the standard measures for evaluating mappings, defined as follows:

$$p(M,R) = \frac{|M \cap R|}{|M|} \qquad r(M,R) = \frac{|M \cap R|}{|R|} \qquad f(M,R) = \frac{2 \cdot p(M,R) \cdot r(M,R)}{p(M,R) + r(M,R)}$$

This way we evaluate the performance of our approach with respect to sub testcase $\#304$. We apply the same procedure for the other three sub testcases of the benchmark dataset.

For the conference dataset we apply a similar approach: We assume that a system of four pairwise mapped ontologies (Sigkdd, CMT, ConfTool and Iasted) exists and that another ontology (in this case EKAW) has to be linked with all of these ontologies. Here we use the pre-existing mappings between the four ontologies as training data and evaluate it on the mappings between the new ontology and the existing ones (dashed lines in Figure 2). We use each 4:1 split of the ontologies as a test case and aggregate the results.

**Baselines.** We used a number of baselines to compare our results to the state of the art in ontology matching. These baselines are described in the following. The first one is motivated by the usual approach to select one matcher from the set of available systems for a given task. This corresponds to selecting a mapping from $M_{Aroma}, \ldots, M_{SAMBOdtf}$. Without any additional knowledge this choice will be at random. We computed the f-measure for all mappings $M_{Aroma}, \ldots, M_{SAMBOdtf}$ and picked out the median of all measured values as first baseline to compare our results with. This baseline is referred to as *median matcher baseline* in the following.

In [1] the authors argue that it is possible to choose a well-suited matcher for a particular matching task given appropriate metadata describing the set of available matching systems. Although this metadata will often not be available and the approach does not ensure to choose the best system, we included a second baseline in our experiments by comparing our approach with the best mapping available for each particular matching task. This baseline will be referred to as *best matcher baseline*.

A more detailed analysis revealed that in many cases the threshold used for generating the mappings was not optimal with respect to the f-measure. Therefore, we included a third baseline where we successively increased the threshold to find the optimal threshold for each system with respect to each particular matching problem. This baseline is referred to as *optimal threshold baseline* and models the situation where the matcher is successfully adapted to the given matching task as suggested in [2].

Notice that it will not be easy to exceed any of these three baselines. In particular a comparison with the optimal threshold baseline will be a hard criterion for our approach. Another obvious baseline would be to compute the union of all mappings $M_1 \cup \ldots \cup M_n$. This baseline was used in [3]. Our experiments revealed that the union baseline cannot even reach the *median matcher baseline* for any of the described testcases. Therefore we resigned to include it in our experiments.

**Implementation.** Our experiments were conducted by means of the Weka Toolkit [21]. We used two different machine learning algorithms, on the one hand the decision tree

algorithm J48, the Weka implementation of the last publicly available version of C4.5 [22], and on the other hand the Weka implementation for a Bayesian network Learner, BayesNet, which, for our data turned out to be equivalent to the Naive Bayes approach in principle[4]. All experiments were conducted with the default settings (2005 book version) making our experiments easily reproducible.

### 3.3 Results

As described above, [8], [9] and [7] propose approaches where machine learning techniques have been applied to a feature set only consisting of confidence values respectively measured similarities. Therefore, in a first set of experiments we focussed on a comparable setting based on a reduced feature set which consist of a single group of features, namely the generated confidence values. Table 1 lists detailed results for each subtestcase as well as aggregated values for the benchmark and conference dataset.

With respect to the benchmark testcases we could beat the median baseline with the use of both decision trees and Naive Bayes by approximately 3%. In particular, we outperformed the median baseline for each individual testcase. Although this baseline seems to be the most realistic one, the improvements are only minor and thus the best matcher baseline could not be reached.

With respect to the conference testcase we observe a different behavior. While decision trees perform very well (we measured an improvement of 13.3% in average) no significant difference between Naive Bayes and the median baseline can be observed. A look at detailed results reveals, that this has been mainly caused by the bad performance for subtestcase ConfTool. A posteriori, this outlier can be explained by an unfortunate choice in the discretization of the numerical attributes. Overall, these results partially coincide with our expectations based on existing studies cited above. A feature set restricted to confidences (or similarities) slightly improves the results, but cannot optimize the results to a significant degree in general.

In a second series of experiments we used the complete set of features listed in section 2.3. The results are presented in Table 2. For the benchmark dataset we observe only minor changes compared to our first experiments. Again, we measure only slight improvements with respect to the median baseline. This might be caused by a ceiling effect, since most of the input mappings were highly precise contrary to the conference dataset. Since testcase #301 to #304 have been part of the OAEI evaluation and with reference mappings open available, an overfitting of some matching systems cannot be excluded. Thus, it is extremely hard to improve the input by a significant degree.

With respect to the conference dataset we observe a dramatic change to the positive. Adding the complete feature set we measured an enhancement of the average f-value from 55.5% to 65.7% for decision trees and from 41.4% to 63.1% for Naive Bayes classifiers. This improvement topped our expectations. Obviously, our meta-level learning approach detects important interdependencies between those aspects described by the set of chosen features. Thus, we not only outperformed the median baseline but also the

[4] The results differ slightly to the default settings of the Weka NaiveBayes implementation, as the BayesNet implementation uses a supervised discretization step, which is not the default for NaiveBayes.

**Table 1.** Results based on features restricted to confidences

| | Baselines | | | Decision Tree | | | | Naive Bayes | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Ontology | M(edian) | B(est) | O(ptimal) | results | Δ-M | Δ-B | Δ-O | results | Δ-M | Δ-B | Δ-O |
| #301 | 0.825 | 0.877 | 0.877 | 0.855 | +0.030 | -0.022 | -0.022 | 0.863 | +0.038 | -0.014 | -0.014 |
| #302 | 0.709 | 0.753 | 0.753 | 0.753 | +0.044 | +0.000 | +0.000 | 0.753 | +0.044 | +0.000 | +0.000 |
| #303 | 0.804 | 0.860 | 0.891 | 0.816 | +0.012 | -0.044 | -0.075 | 0.860 | +0.056 | +0.000 | -0.031 |
| #304 | 0.940 | 0.961 | 0.961 | 0.967 | +0.027 | +0.006 | +0.006 | 0.954 | +0.014 | -0.007 | -0.007 |
| **Average** | **0.820** | **0.863** | **0.871** | **0.848** | **+0.028** | **-0.015** | **-0.023** | **0.857** | **+0.038** | **-0.005** | **-0.013** |
| CMT | 0.435 | 0.512 | 0.512 | 0.500 | +0.065 | -0.012 | -0.012 | 0.400 | -0.035 | -0.112 | -0.112 |
| ConfTool | 0.471 | 0.484 | 0.484 | 0.474 | +0.003 | -0.010 | -0.010 | 0.203 | -0.268 | -0.281 | -0.281 |
| Ekaw | 0.411 | 0.471 | 0.516 | 0.593 | +0.182 | +0.122 | +0.077 | 0.441 | +0.030 | -0.030 | -0.075 |
| Iasted | 0.403 | 0.478 | 0.489 | 0.649 | +0.246 | +0.171 | +0.160 | 0.453 | +0.050 | -0.025 | -0.036 |
| Sigkdd | 0.390 | 0.462 | 0.475 | 0.560 | +0.170 | +0.098 | +0.085 | 0.575 | +0.185 | +0.113 | +0.100 |
| **Average** | **0.422** | **0.481** | **0.495** | **0.555** | **+0.133** | **+0.074** | **+0.060** | **0.414** | **-0.008** | **-0.067** | **-0.081** |

**Table 2.** Results based on a complete feature set

| | Baselines | | | Decision Tree | | | | Naive Bayes | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Ontology | M(edian) | B(est) | O(ptimal) | results | Δ-M | Δ-B | Δ-O | results | Δ-M | Δ-B | Δ-O |
| #301 | 0.825 | 0.877 | 0.877 | 0.883 | +0.058 | +0.006 | +0.006 | 0.830 | +0.005 | -0.047 | -0.047 |
| #302 | 0.709 | 0.753 | 0.753 | 0.759 | +0.050 | +0.006 | +0.006 | 0.753 | +0.044 | +0.000 | +0.000 |
| #303 | 0.804 | 0.860 | 0.891 | 0.816 | +0.012 | -0.044 | -0.075 | 0.851 | +0.047 | -0.009 | -0.040 |
| #304 | 0.940 | 0.961 | 0.961 | 0.960 | +0.020 | -0.001 | -0.001 | 0.966 | +0.026 | +0.005 | +0.005 |
| **Average** | **0.820** | **0.863** | **0.871** | **0.855** | **+0.035** | **-0.008** | **-0.016** | **0.850** | **+0.031** | **-0.012** | **-0.020** |
| CMT | 0.435 | 0.512 | 0.512 | 0.580 | +0.145 | +0.068 | +0.068 | 0.546 | +0.111 | +0.034 | +0.034 |
| ConfTool | 0.471 | 0.484 | 0.484 | 0.572 | +0.101 | +0.088 | +0.088 | 0.480 | +0.009 | -0.004 | -0.004 |
| Ekaw | 0.411 | 0.471 | 0.516 | 0.621 | +0.210 | +0.150 | +0.105 | 0.659 | +0.248 | +0.188 | +0.143 |
| Iasted | 0.403 | 0.478 | 0.489 | 0.746 | +0.343 | +0.268 | +0.257 | 0.750 | +0.347 | +0.272 | +0.261 |
| Sigkdd | 0.390 | 0.462 | 0.475 | 0.766 | +0.376 | +0.304 | +0.291 | 0.723 | +0.333 | +0.261 | +0.248 |
| **Average** | **0.422** | **0.481** | **0.495** | **0.657** | **+0.235** | **+0.175** | **+0.162** | **0.631** | **+0.209** | **+0.150** | **+0.136** |

best matcher baseline. Even the best matcher with an optimal threshold obtains 16.2% respectively 13.6% worse results for the conference dataset.

The significant difference caused by taking into account the full feature set raises doubts about the importance of confidence values. Thus, in a third series of experiments we removed the confidence values from the complete feature set. Results presented in Table 3 confirm our scepticism about the validity of confidence values. Decision trees performed as good or even better for all of our nine sub testcases. With suppressed confidences a Naive Bayes classifier performs in six sub testcases better, in two cases we observed no changes, and only in one case we measured a slight decline. In average for all of our four combinations of dataset and classifier we nearly measure up or even clearly exceed the optimal threshold baseline. Notice again, that this baseline requires a perfect knowledge about the best matcher and its optimal threshold with respect to a certain matching task that is not available in nearly all realistic scenarios.

**Table 3.** Results based on a feature set where confidences have not been included

| | Baselines | | | Decision Tree | | | | Naive Bayes | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Ontology | M(edian) | B(est) | O(ptimal) | results | Δ-M | Δ-B | Δ-O | results | Δ-M | Δ-B | Δ-O |
| #301 | 0.825 | 0.877 | 0.877 | 0.883 | +0.058 | +0.006 | +0.006 | 0.841 | +0.016 | -0.036 | -0.036 |
| #302 | 0.709 | 0.753 | 0.753 | 0.759 | +0.050 | +0.006 | +0.006 | 0.753 | +0.044 | +0.000 | +0.000 |
| #303 | 0.804 | 0.860 | 0.891 | 0.816 | +0.012 | -0.044 | -0.075 | 0.860 | +0.056 | +0.000 | -0.031 |
| #304 | 0.940 | 0.961 | 0.961 | 0.960 | +0.020 | -0.001 | -0.001 | 0.966 | +0.026 | +0.005 | +0.005 |
| **Average** | **0.820** | **0.863** | **0.871** | **0.855** | **+0.035** | **-0.008** | **-0.016** | **0.855** | **+0.035** | **-0.008** | **-0.015** |
| CMT | 0.435 | 0.512 | 0.512 | 0.625 | +0.190 | +0.113 | +0.113 | 0.597 | +0.162 | +0.085 | +0.085 |
| ConfTool | 0.471 | 0.484 | 0.484 | 0.572 | +0.101 | +0.088 | +0.088 | 0.526 | +0.055 | +0.042 | +0.042 |
| Ekaw | 0.411 | 0.471 | 0.516 | 0.621 | +0.210 | +0.150 | +0.105 | 0.667 | +0.256 | +0.196 | +0.151 |
| Iasted | 0.403 | 0.478 | 0.489 | 0.746 | +0.343 | +0.268 | +0.257 | 0.740 | +0.337 | +0.262 | +0.251 |
| Sigkdd | 0.390 | 0.462 | 0.475 | 0.766 | +0.376 | +0.304 | +0.291 | 0.732 | +0.342 | +0.270 | +0.257 |
| **Average** | **0.422** | **0.481** | **0.495** | **0.666** | **+0.244** | **+0.184** | **+0.171** | **0.652** | **+0.230** | **+0.171** | **+0.157** |

One major result of our experiments is concerned with the role of confidence values. By comparing the results based on three feature configurations we conclude that confidence values without additional features cannot be exploited successfully by a meta-level learning approach. Moreover, given a well designed comprehensive set of additional features, confidence values have even negative effects on the performance of the overall approach. Notice that our feature set still contains matcher specific information. These are the features describing wether a correspondence has been found by a particular matcher and the aggregated feature reflecting how many systems found the correspondence. These features are obviously sufficient to model interdependencies between certain characteristics of a particular correspondence and its correctness.

## 4 Conclusions

There are a couple of implications resulting from our work. Some of them have already been discussed in connection with the motivation of this work and the contribution to the state of the art. Others where hidden in the results of the matching experiments. We now briefly recall the major implications and point to some of the hidden aspects.

First of all, we have shown that machine learning is an adequate - maybe the best - way of combining the results of different heterogeneous matching systems. At the same time, we have shown that it is not enough to base the learning step on the results of the matching systems alone, but that additional features representing and aggregating information about the mapping and the mapped ontology have to be taken into account. These features enable us to put matcher results into context and get a better basis for deciding when to trust a certain matcher. We have also shown, that the confidence measures produced by matching systems are almost meaningless if these contextual features are known as they do not improve the result. A hidden implication of this observation is, that we can also apply our method on the result of matchers that do not return a degree of confidence. This significantly broadens the applicability of our approach.

Another very interesting and important result of our work is hidden in the classifiers learned. Analyzing the decision trees generated for the two datasets we discovered that

the most significant feature was the fraction of matchers that found a correspondence. From this observation, we can derive a straightforward heuristic for combining matchers. In particular, it turned out that just following the majority of the matchers involved produces results that are almost as good as the results reported in the last section. After making this observation, we compared the result of the majority vote heuristic with our baselines. While on the benchmark dataset the result of the majority vote was comparable with the best matcher, it significantly outperformed all baselines on the conference data set: While the baselines ranged from 0.422 to 0.495 on average, the majority vote reached an average of 0.633, which is more than 25% better than the best baseline. For comparison the averages we reached on this dataset with the two machine learning methods where 0.666 and 0.652 which is only about 5% better than the result of the majority vote.

So far, we have shown that our approach works very well in a setting where an existing system of ontologies is to be extended with links to additional ontologies about the same domain. This is a very realistic setting for concrete applications as there is often a central ontology, i.e. the Gene Ontology many other ontologies are connected to. From a theoretical point of view, it would be interesting to see how far we can generalize this to arbitrary matching problems, i.e. it would be interesting to test whether the classifier learned for the benchmarking dataset can be applied to the conference dataset and vice versa. This would further enhance the usefulness of your method as there is no need to have reference mappings for new application domains, but classifiers could be trained on existing mapping sets like the ones we used in our experiments or the ones available in the area of medicine.[5] Investigating this question in detail will be subject of future work.

## References

1. Mochol, M., Jentzsch, A.: Towards a rule-based matcher selection. In: Proc. of the 16th international conference on Knowledge Engineering, Acitrezza, Italy (2008)
2. Ehrig, M., Staab, S., Sure, Y.: Bootstrapping ontology alignment methods with APFEL. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 186–200. Springer, Heidelberg (2005)
3. Isaac, A., Trojahn, C., Wang, S., Quaresma, P.: Using quantitative aspects of alignment generation for argumentation on mappings. In: Proceedings of the ISWC 2008 Workshop on Ontology Matching, Karlsruhe, Germany (2008)
4. Aumueller, D., Do, H.H., Massmann, S., Rahm, E.: Schema and ontology matching with COMA++. In: Proc. of the ACM SIGMOD International Conference on Management of Data, Baltimore, Maryland, USA (2005)
5. Jean-Mary, Y.R., Kabuka, M.R.: Asmov: results for OAEI 2008. In: Proc. of the ISWC 2008 Workshop on Ontology Matching, Karlsruhe, Germany (2008)
6. Doan, A., Madhavan, J., Dhamankar, R., Domingos, P., Halevy, A.: Learning to match ontologies on the semantic web. VLDB Journal (2003)
7. Ichise, R.: Machine learning approach for ontology mapping using multiple concept similarity measures. In: Seventh IEEE/ACIS International Conference on Computer and Information Science, ICIS 2008, Portland, Oregon, USA (2008)

[5] `http://www.obofoundry.org/index.cgi?sort=type&show=mappings`

8. Svab, O., Svatek, V.: Combining ontology mapping methods using bayesian networks. In: Proc. of the ISWC 2006 Workshop on Ontology Matching, Athens, USA (2006)
9. Maio, P., Bettencourt, N., Silva, N., Rocha, J.: Evaluating a confidence value for ontology alignment. In: Proc. of the ISWC 2007 Workshop on Ontology Matching, Busan, Korea (2007)
10. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, Heidelberg (2007)
11. Euzenat, J., Isaac, A., Meilicke, C., Shvaiko, P., Stuckenschmidt, H., Svab, O., Svatek, V., van Hage, W., Yatskevich, M.: Results of the ontology alignment evaluation initiative 2007. In: Proc. of the ISWC 2007 Workshop on Ontology Matching, Busan, Korea (2007)
12. Caracciolo, C., Euzenat, J., Hollink, L., Ichise, R., Isaac, A., Malaise, V., Meilicke, C., Pane, J., Shvaiko, P., Stuckenschmidt, H., Svab-Zamazal, O., Svatek, V.: First results of the ontology alignment evaluation initiative 2008. In: Proc. of the ISWC 2008 Workshop on Ontology Matching, Karlsruhe, Germany (2008)
13. Bouquet, P., Ehrig, M., Euzenat, J., Franconi, E., Hitzler, P., Krötzsch, M., Serafini, L., Stamou, G., Sure, Y., Tessaris, S.: Specification of a common framework for characterizing alignment. Deliverable D2.2.1 (version 2), KnowledgeWeb Network of Excellence (February 2005), `http://knowledgeweb.semanticweb.org/semanticportal/deliverables/D2.2.1v2.pdf`
14. Svab, O., Svatek, V., Berka, P., Rak, D., Tomasek, P.: Ontofarm: Towards an experimental collection of parallel ontologies. In: Poster Proceedings of the International Semantic Web Conference (2005)
15. David, J.: AROMA results for OAEI 2008. In: Proc. of the ISWC 2008 Workshop on Ontology Matching, Karlsruhe, Germany (2008)
16. Nagy, M., Vargas-Vera, M., Stolarski, P., Motta, E.: DSSim results for OAEI 2008. In: Proc. of the ISWC 2008 Workshop on Ontology Matching, Karlsruhe, Germany (2008)
17. Wang, P., Xu, B.: Lily: ontology alignment results for OAEI 2008. In: Proc. of the ISWC 2008 Workshop on Ontology Matching, Karlsruhe, Germany (2008)
18. Zhang, X., Zhong, Q., Li, J., Tang, J.: RiMOM results for OAEI 2008. In: Proc. of the ISWC 2008 Workshop on Ontology Matching, Karlsruhe, Germany (2008)
19. Lambrix, P., Tan, H., Liu, Q.: SAMBO and SAMBOdtf results for the ontology alignment evaluation initiative 2008. In: Proceedings of the ISWC 2008 Workshop on Ontology Matching, Karlsruhe, Germany (2008)
20. Sabou, M., Gracia, J.: Spider: Bringing non-equivalence mappings to OAEI. In: Proc. of the ISWC 2008 Workshop on Ontology Matching, Karlsruhe, Germany (2008)
21. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco (2005)
22. Quinlan, R.: C4.5: Programs for Machine Learning (1993)