Evaluation Measures for Ontology Matchers in Supervised Matching Scenarios

Dominique Ritze, Heiko Paulheim, and Kai Eckert

University of Mannheim, Mannheim, Germany
Research Group Data and Web Science
{dominique,heiko,kai}@informatik.uni-mannheim.de

Abstract. Precision and Recall, as well as their combination in terms of F-Measure, are widely used measures in computer science and generally applied to evaluate the overall performance of ontology matchers in fully automatic, unsupervised scenarios. In this paper, we investigate the case of supervised matching, where automatically created ontology alignments are verified by an expert. We motivate and describe this use case and its characteristics and discuss why traditional, F-measure based evaluation measures are not suitable for this use case. Therefore, we investigate several alternative evaluation measures and propose the use of Precision@N curves as a mean to assess different matching systems for supervised matching. We compare the ranking of several state of the art matchers using Precision@N curves to the traditional F-measure based ranking, and discuss means to combine matchers in a way that optimizes the user support in supervised ontology matching.

Keywords: Supervised Ontology Matching, Evaluation, Recall, Precision, F-Measure, Precision@N-Curves, ROC-Curves, Precision-Recall-Curves

1 Supervised Ontology Matching

An *ontology* provides a formal representation of domain knowledge by defining entities, i.e., instances, classes, and their relationships. This broad definition of an ontology also encompasses more restricted knowledge organization systems such as thesauri, classifications, or taxonomies. *Ontology matching* is the process of creating alignments between ontologies. An *alignment* A contains correspondences between the entities of two ontologies O_1 and O_2. A *correspondence* c relates an entity of ontology O_1 to an entity of ontology O_2. The semantics of the relationship depends on the application, the matching approach, and the formal language in which the ontology is described. Each of the prominent Semantic Web languages OWL, RDFS, and SKOS provide relationships to denote equivalent entities and to relate entities hierarchically. Additionally, each correspondence can have a confidence value to indicate how likely the relation holds.

The manual creation of ontology alignments is very time-consuming and often not feasible, for example, if various ontologies are used in a dynamic environment, where ontologies have to be matched ad-hoc, or for alignments between large-scale ontologies. Therefore, *automatic ontology matching* approaches have been developed. The term

ontology matching today is used mostly in the context of automatic matching. On the one hand, they have the advantage to not involve any human expert but on the other hand, the quality that can be achieved by fully automatic matching systems is limited [26].

There are, however, use cases in which the quality of the resulting mapping is essential, and where wrong correspondences cannot be tolerated. Such use cases demand for different approaches. In this paper, we investigate the approach of *supervised ontology matching*[1], where a machine-created alignment is double-checked by a human expert.

2 Problem Statement

The use case for supervised ontology matching that motivates our research is the manual creation of an alignment between medium-size to large ontologies, e.g., thesauri that are used in libraries, such as the Standard Thesaurus for Economics (STW) and the Thesaurus for the Social Sciences (TheSoz). For these thesauri, a partial alignment has been manually created in 2006 within the KoMoHe-project [20]. This was a significant effort, but it was not even feasible to maintain the resulting alignment as both thesauri have been further developed: several hundred concepts have been added to both thesauri, other concepts got changed or deleted.

Since 2012, STW and TheSoz are used in the *Library Track*[2] of the Ontology Alignment Evaluation Initiative (OAEI) as test-bed for automatic ontology matchers. While the matchers are evaluated based on the existing partial alignment, it would be appealing to use the results of the matchers to improve and maintain this alignment. The time of domain experts who could check the results and maintain the alignment, however, is precious. This leads to our research question: *Given a fixed amount of time that a human expert can invest for double checking the results of a matcher, which matcher should be used, so that the number of new correct correspondencies?*

A general approach to evaluate automatic ontology matchers is pursued by the OAEI, where matchers meet several challenges (including the Library Track) in an annual competition. As no special use case is presumed, the matchers resp. their candidate alignments are evaluated using the common measures precision, recall, and F-measure [30], i.e., it is calculated how precise (returned correct correspondences vs. returned correspondences) and how complete (returned correct correspondences vs. existing correct correspondences) an alignment is. As an improvement of precision generally leads to a decrease of recall and vice versa, F-measure, i.e., the harmonic mean of precision and recall, is used as an overall evaluation measure. To compute these measures, a (partial) reference alignment containing (almost) all correct correspondences needs to be available.

For our use case, however, we require a measure that takes into account the time a human expert can invest for double-checking matcher results. Thus, precision, recall, and F-measure are not sufficient to rate matchers. Generally, a high precision of the candidate alignment is desired, as every incorrect correspondence costs time, but does

[1] This is not to be confused with the term *supervised learning* used in machine learning, where examples are given to a learner for training a model.

[2] http://web.informatik.uni-mannheim.de/oaei-library/2012/

not increase the size of the manual alignment. On the other hand, the more time the expert invests, the more correspondences can be checked. Thus, we require a dynamic measure that takes the number of correspondences which can be checked as a parameter, and which optimizes the outcome (i.e., number of correct correspondences) among those.

Another important characteristic of this use case is that the correspondences that are included in the candidate alignment, but not checked by the human expert, do not affect the performance of the matcher for this use case. For example, if a matcher returns a candidate alignment with 100 correspondences but the human expert only has the time to check 10, this matcher has the same performance with respect to our use case as a matcher creating a candidate alignment containing only these 10 correspondences. As long as the expert is willing to check more correspondences, further correspondences should be offered. On the other hand, it can be assumed that the expert stops checking when the yield drops below a certain rate. Intuitively, a matcher therefore performs best if it sorts the correspondences so that correct correspondences are sorted before incorrect ones. Most matchers provide the means for such a ranking in the form of confidence values [10], but those values are not taken into account for computing the "classic" measures, such as recall, precision, and F-measure.

3 Precision@N Curves

Given an ordered candidate alignment $\mathcal{C}$ with $|\mathcal{C}|$ correspondences. In a fixed time t, the human expert can verify the first n of the correspondences in $\mathcal{C}$, a subset denoted as $\mathcal{C}_n$. Since n is not known, a matcher performs best, if $\mathcal{C}_n$ contains a maximum amount of correct correspondences for every n, i.e., the precision of $\mathcal{C}_n$ is maximized for every n. We therefore define:

Definition 1 (Precision@N). *The Precision@N (Pr_n) is the precision of $\mathcal{C}_n$ for a given $\mathcal{C}$ and n, with $1 \leq n \leq |\mathcal{C}|$.*

While the definition of Precision@N is derived straight from the problem statement, there are some details to tackle. First of all, different matchers provide candidate alignments of different sizes – correlated to precision and recall (high precision $\rightarrow$ small alignment, high recall $\rightarrow$ large alignment). To be comparable, Precision@N must be defined for all matchers for all n. Therefore, the sizes of the candidate alignments have to be equalized by padding, i.e., smaller candidate alignments are simply filled up with placeholders from an artificial placeholder alignment $\mathcal{P}$ that contains artificial incorrect correspondences with a confidence of 0:

$$\widehat{\mathcal{C}} = \cup\{\mathcal{C}, \mathcal{P}\}; \qquad |\mathcal{P}| = |\widehat{\mathcal{C}}| - |\mathcal{C}| \tag{1}$$

There are different possibilities to define the size of $\widehat{\mathcal{C}}$. If there are several candidate alignments available, a human expert would all of them in an ideal case. Thus, we define the union of all candidate alignments $\mathcal{C}_{all}$ (without the artificial placeholders), which forms the largest candidate alignment, thus, the following equations holds: $|\widehat{\mathcal{C}}| = |\mathcal{C}_{all}|$.

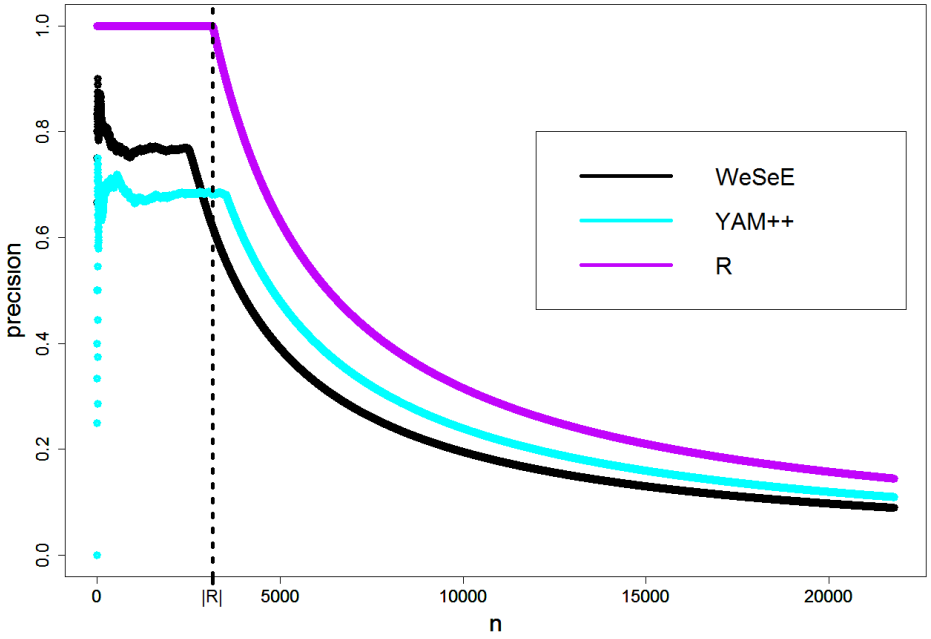


Fig. 1: Optimal and exemplary Precision@N curves

Figure 1 shows two exemplary curves for the Precision@N values of two candidate alignments, obtained with the matching systems *WeSeE* [25] and *YAM++* [21], together with the optimal curve based on the reference alignment $\mathcal{R}$. The reference alignment $\mathcal{R}$ contains all correct correspondences which can be found between the ontologies and serves as gold standard. Its size in our use case is $3,161$, the union of all matching results considered in this paper has the size $21,787 = |\mathcal{C}_{all}|$ (see Sect. 4. For the optimal curve, Precision@N is defined as

$$Pr_n(\mathcal{R}) = \begin{cases} 1 & 1 \leq n \leq |\mathcal{R}| \\ \frac{|\mathcal{R}|}{n} & n > |\mathcal{R}| \end{cases} \tag{2}$$

From Fig. 1, we can see that the number n of correspondences to be checked by an expert is required to decide on an optimal matcher: matcher WeSeE dominates matcher YAM++ for smaller values of n ($n < 2848$), while YAM++ dominates for higher values of n.

In order to create a ranking of matchers for this given task, we have to abstract from these curves: we propose to use the area under the curve (AUC) of the Precision@N curve as an evaluation measure that can be used to rank the matchers with respect to their overall performance in this use case, independent of a specific n. The AUC of the Precision@N can accurately be approximated by a Riemann sum [28]:

$$\text{AUC}(Pr_n) = \int_{n=1}^{|\widehat{\mathcal{C}}|} Pr_n = \sum_{n=1}^{|\widehat{\mathcal{C}}|} Pr_n \tag{3}$$

We further normalize AUC(Pr_n) using AUC($Pr_n(\mathcal{R})$)$^{-1}$ as normalization factor. The normalization factor can be determined by the following closed-form expression using Euler's approximation for the sum of a partial harmonic series [14]:

$$\text{AUC}(Pr_n(\mathcal{R})) = \sum_{n=1}^{|\mathcal{R}|} 1 + \sum_{n=|\mathcal{R}|+1}^{|\widehat{\mathcal{C}}|} \frac{|\mathcal{R}|}{n} \tag{4}$$

$$= |\mathcal{R}| + |\mathcal{R}| \cdot \left(\sum_{n=1}^{|\widehat{\mathcal{C}}|} \frac{1}{n} - \sum_{n=1}^{|\mathcal{R}|} \frac{1}{n} \right) \tag{5}$$

$$\approx |\mathcal{R}| + |\mathcal{R}| \cdot \left(ln(|\widehat{\mathcal{C}}|) - ln(|\mathcal{R}|) + \frac{1}{2|\widehat{\mathcal{C}}|} - \frac{1}{2|\mathcal{R}|} \right) \tag{6}$$

In the following, we use the normalized AUC of Precision@N, which has a range between 0 and 1, and compare it to other evaluation measures.

4 Evaluation

We use the OAEI Library Track data set and the results of all matchers participating in the OAEI 2012 campaign to evaluate Precision@N curves: AROMA [5], CODI [16], GOMMA [18], Hertuda [15], HotMatch [4], LogMapLt [17], LogMap [17], MapSSS [3], Optima [31], ServOMap-lt [2], ServOMap [2], WeSeE [25], and YAM++ [21]. The results of the latest OAEI evaluation can be found in the OAEI 2012 results overview paper [1] and include a comparison based on precision, recall and F-measure. All generated candidate alignments[3] as well as the reference alignment[4] are available.

4.1 Single Matcher Evaluation

We evaluate the Precision@N curves of the 13 matchers and compare them to related measures. Precision, recall and F-measure have already been mentioned in the beginning. These ones are commonly employed and provide an overview of the alignment quality, but, as discussed above, do not suit every use case. None of those measures takes the ordering of the correspondences according to their confidence values into account. For our use case, however, this ranking is essential. That is why we mainly focus on measures which exploit confidence scores, such as Precison-Recall curves.

Mean Absolute Error (MAE) measures how close predictions are to the actual values. It is the average of all differences between the prediction and the actual outcome. If we assume that a confidence value of 0 indicates an incorrect correspondence and a confidence value of 1 stands for a correct correspondence, we can apply this measure on the candidate alignments:

$$MAE(\mathcal{C}) = \tfrac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \begin{cases} 1.0 - conf(c) : \text{if c is correct} \\ conf(c) : \text{otherwise} \end{cases}$$

[3] http://web.informatik.uni-mannheim.de/oaei-library/results/2012/rawAlignments.zip
[4] http://web.informatik.uni-mannheim.de/oaei-library/results/2012/referenceAll.rdf

where $conf(c)$ is the confidence value of a correspondence c. MAE measures the ability of a matcher to predict good confidence values. An optimal candidate alignment with respect to MAE contains correct correspondences with a confidence value of 1 and incorrect ones with a confidence value of 0, and yields an MAE of 0.

Precision-Recall (PR) Curves illustrate which precision value is achieved at a certain recall value by plotting the precision values (y-axis) against the recall (x-axis) values [19]. Typically, the curve decreases from high precision values for low recall to lower precision values for higher recall. An optimal candidate alignment contains all correct correspondences without including any incorrect ones. Whenever incorrect correspondences occur in the candidate alignment, they need to be ranked low in the candidate alignment to keep the curve at an upper level before it drops down. The area under the PR curve (AUC(PR)) can be computed to receive a single value which can be used for simple comparisons.

Receiver Operator Characteristic (ROC) Curves have been introduced in the machine learning field to evaluate binary classifiers [13]. They show how the number of correct correspondences varies with the number of incorrect ones which represents the ability of the classifier to discriminate correct from incorrect correspondences. On the x-axis, the true positive rate is plotted against the false positive rate on the y-axis. ROC curves can be applied to see whether the matcher is able to assign distinguishable confidence values [23]. If all correct correspondences are ranked higher than the incorrect ones, the ROC curve is optimal. Again, the area under the ROC curve AUC(ROC) can be used for comparison, which is 1 for an optimal alignment. PR and ROC curves are strongly related and share similar characteristics, but they are not equivalent [6], since false positives influence the PR curve more strongly than the ROC curve.

4.2 Combining Matching Strategies

Due to the large amount of available matchers, we are not restricted to only concentrate on one system. Especially if the human expert has the time to check a huge amount of correspondences which exceeds the size of each single candidate alignment generated by the individual matchers, he or she should not be forced to stop only because the candidate alignment does not contain enough correspondences. By unifying candidate alignments, we can ensure that the human expert has enough correspondences to verify, at most all correspondences found by any of the matchers.

When combining matchers by unifying their candidate alignments, a strategy for ordering the correspondences in their union alignment $\mathcal{C}_{all}$ is required. For our use case, a particular challenge is to find a strategy to order the correspondences in $\mathcal{C}_{all}$ in a way that the Precision@N curve of the resulting sorted alignment is optimal. With this additional requirement, the ontology matching problem is reformulated as a sorting problem. A related approach is the combination of matchers to improve the quality of candidate alignments in terms of F-measure. Eckert et al. [8] have shown that a suitable combination of matchers can outperform the individual ones because the strengths of each matcher can be exploited, while weaknesses of single matchers can be cancelled out.

We investigate the following seven strategies for ordering the union of alignments $\mathcal{C}_{all}$:

Random. The random strategy is a lower baseline. Each correspondence is assigned to a random position in the unified alignment no matter which matcher produced this correspondence or how high the confidence value is.

Confidence Sorted. Since each correspondence has a confidence value, we can take them to sort $\mathcal{C}_{all}$. Assuming that all matchers provide confidence values normalized to a $[0; 1]$ interval (in an extreme case, a matcher will deliver only correspondences with a confidence of 1, with all other possible correspondences implicitly having a confidence value of 0), we sort the correspondences by the maximum of confidence values provided by any matcher.

F-measure on Partial Reference Alignment. Assuming that a partial reference alignment is already given, all matchers can be evaluated against it, which yields an approximation of the actual matcher performance. As shown in [27], precision, recall, and F-measure on partial reference alignments strongly correlate with the respective measures of the full reference alignment. We use the F-measure on the partial reference alignment to sort $\mathcal{C}_{all}$, adding all correspondences found by the best performing matcher first, then adding all correspondences found by the second best matcher, and so on. For our experiment, we use a subset of our full reference alignment[5] as a partial reference alignment. All precision, recall and F-Measure values on this partial reference alignment can be found on the results page of the Library Track[6].

In cases where a partial reference alignment is not available, evaluation results of similar data sets can also be used for these kinds of strategies.

Precision on Partial Reference Alignment. This strategy is similar to the *F-measure on Partial Reference Alignment* strategy, but orders the correspondences by the precision of the matcher that found them on the partial reference alignment.

Recall on Partial Reference Alignment. This strategy is similar to the *F-measure on Partial Reference Alignment* strategy, but orders the correspondences by the recall of the matcher that found them on the partial reference alignment.

Majority Voting. In this strategy, the correspondences of the unified candidate alignment are sorted by the number of matchers which found these correspondences. The correspondences found by all matchers are ranked highest, followed by the correspondences found by all but one matcher and so on. If several correspondences have been detected by the same number of matchers, they are sorted in descending order of their confidence values. In our data set, 71 correspondences have been found by 13 matchers, 209 by 12 matchers, etc., but the majority of correspondences (16662) are only found by a single matcher. This approach has the advantage that no partial reference alignment or other evaluation results are required.

Current Leadership. As already indicated when having a look at the Precision@N curves, a certain matcher may perform best for a specific n. The Current Leadership strategy picks up that idea and takes the sorted correspondences of a candidate alignment as long as the corresponding matcher has the currently highest precision value. As soon as another matcher takes over the leadership, the correspondences contained in the other candidate alignment are now taken. This strategy serves as

[5] http://web.informatik.uni-mannheim.de/oaei-library/results/2012/reference.rdf

[6] http://web.informatik.uni-mannheim.de/oaei-library/results/2012/

a baseline and can only be applied when we already computed the Precision@N curves for all matchers.

With the evaluation, we want to answer the question which strategy can sort the correspondences in the unified candidate alignment best such that the Precision@N is maximized.

5 Evaluation Results

All data and scripts (source code as well as executable programs) we used for our evaluation can be found in the research data repository of our library[7] to ensure the reproducibility and traceability of our results. To compute the area under curves, we used the the *trapz*[8] function of the R package *pracma* which performs a trapezoidal integration.

5.1 Single Matcher Evaluation Results

Figure 2 shows the Precision@N curves for the individual candidate mappings. As discussed above, Precision@N curves provide an intuitive visualization to assess which matcher contains the most correct correspondences for a given n. For small n, candidate alignments containing exclusively correct correspondences are preferable. The more correspondences a domain expert can verify, the completeness of the candidate alignment becomes more important. To decide which matcher performs best for our use case, matchers with Precision@N curves which are always dominated by other curves, e.g. MapSSS or Optima, can be discarded at fist glance.

Several observations can be made from the curves. For example, the curves of GOMMA and Hertuda, which provide high recall values, stay at a rather low level of precision but over a large number of n. In contrast, the curves for ServOMap and LogMap, which have high precision values, start at higher precision values, but drop quicklier.

Comparing Precision@N curves to PR and ROC curves emphasizes the individual characteristics of each visualization. We selected two matchers to illustrate their curves in detail. Figure 3 shows the three plots for the matcher GOMMA. The PR curve represents, apart from the beginning, an almost horizontal line. Since the PR curve plots the precision values again the recall values, it indicates that correct and incorrect correspondences occur mostly alternately in the sorted candidate alignment. This can be observed from the ROC curve, which is close to the diagonal in that case. The Precision@N curve of GOMMA is similar to its PR curve. In both visualizations, the curves remain stable on certain level of precision (around 0.6).

Figure 4 depicts the same three plots for the matcher ServOMap. The PR curve shows a different behavior than the PR curve of GOMMA, starting at a higher precision value, but not reaching the recall of GOMMA. This characteristic predicts that most of the correct correspondences are ranked higher than the incorrect ones but the candidate alignment also only includes a few incorrect correspondences, which is indicated by a

[7] http://dx.doi.org/10.7801/23

[8] http://www.inside-r.org/packages/cran/pracma/docs/trapz

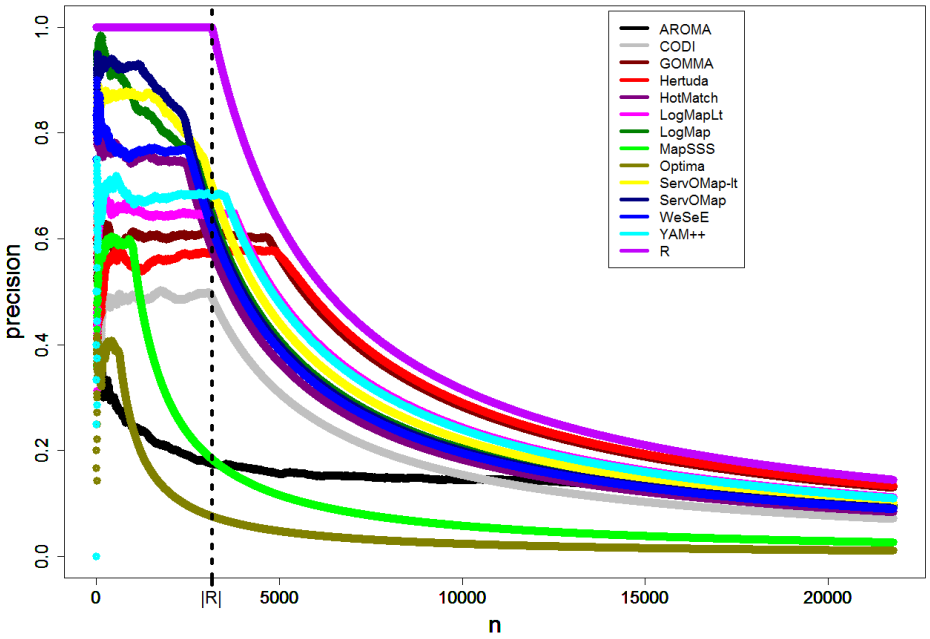


Fig. 2: Precision@N curves for all individual candidate alignments

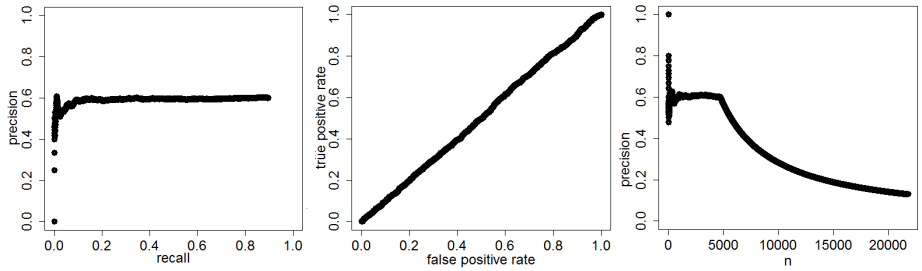
ROC curve which is significantly above the diagonal. Similar to the PC curve, the Precision@N curve starts at a high precision value and drops fast for larger n. If we compare both Precision@N curves, ServOMap outperforms GOMMA for smaller values of n, while GOMMA dominates ServOMap for larger values of n.

Precision@N curves thus provide a suitable means to compare two matchers w.r.t the number of correct correspondences among the first n results. In contrast, neither PR curves nor ROC curves indicate the size of the alignment or provide any expectation of a matcher's performance for a given n.

While the curves provide a visual means to compare different matchers, it is desireable to have a performance estimation reduced to one single real number such that matchers can be explicitly ranked. As already discussed, we can compute the AUCs for that purpose.

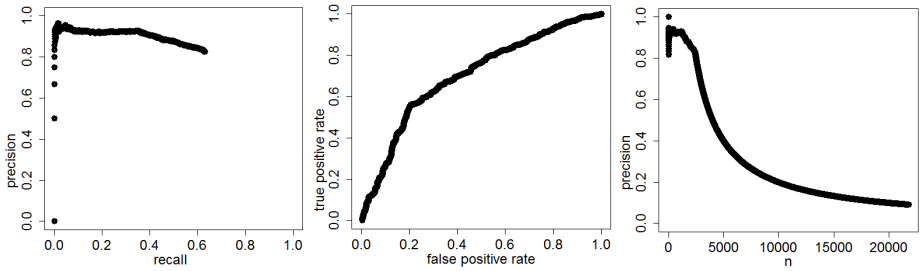
Table 1 shows the ranking of different matchers according to the measures introduced above, where rank 1 denotes the best system. It can be observed that the ranking according to different measures deviate. Some matchers always occupy a similar position, e.g. CODI, while others are ranked very differently, e.g. GOMMA which even has a standard deviation value of the positions of 3.4.

MAE has a correlation to precision which is easily explainable: if a matcher only adds correspondences to the candidate alignment which are very likely to be correct, and assigns high confidence values because the matcher is confident about their correctness, the MAE is close to 0 and the precision close to 1. In our experiments, the matcher LogMap shows an exception. LogMap is on position 2 according to precision



(a) GOMMA PR Curve (b) GOMMA ROC Curve (c) GOMMA Pr_n Curve

Fig. 3: Comparision of different curves for matcher GOMMA



(a) ServOMap PR Curve (b) ServOMap ROC Curve (c) ServOMap Pr_n Curve

Fig. 4: Comparision of different curves for matcher ServOMap

but only on position 9 when applying MAE. For all correspondences, LogMap assigns confidence values between 0.5 and 0.99. Since it has a high precision, most of these correspondences are correct, even though some of them have a confidence of only 0.5. Obviously, these high differences cause a high MAE value. In contrast, HotMatch, also a matcher with a high precision, only assigns confidence values between 0.88 and 1.0. Thus, the difference is rather small. MAE provides an assessment of the confidence values which are important for the ranking but it does not take into account how many of all correct correspondences have been found.

AUC(PR) is highly correlated (0.98) to F-measure, but the additional consideration of the ordering changes the positions of some matchers. The most significant changes concern GOMMA and Hertuda. While GOMMA drops from position 2 to position 5, Hertuda rises from position 7 to position 4. Thus, Hertuda assigns better confidence values to the correspondences found, which allow for a better ordering.

AUC(ROC), unlike others, is neither negatively nor positively correlated to any of the other measures. The difference in the positions can vary a lot, e.g. GOMMA has position 2 according to F-measure but only position 10 when applying AUC(ROC). This shows that a matcher with a high F-measure does not necessarily assign confidence values to the correspondences which can indicate whether a correspondence is indeed correct or not.

Table 1: Positions of the matchers according to different measures

Matcher	AUC(Pr_n)	Precison	Recall	F-Measure	AUC(PR)	AUC(ROC)	MAE
GOMMA	1	8	2	2	5	10	7
Hertuda	2	10	1	7	4	5	10
ServoMap-lt	3	3	5	1	1	1	4
YAM++	4	6	4	5	6	9	5
LogMapLt	5	7	3	6	7	11	6
ServOMap	6	1	8	3	2	2	1
LogMap	7	2	7	4	3	4	9
WeSeE	8	5	9	8	8	3	3
HotMatch	9	4	10	9	9	8	2
CODI	10	11	11	10	10	12	11
AROMA	11	13	6	12	11	6	13
MapSSS	12	9	12	11	12	13	8
Optima	13	12	13	13	13	7	12

AUC(Pr_n) is significantly correlated (0.97) to AUC(PR) and acts in most cases similar to recall as well as F-measure, but puts a stronger focus on recall than AUC(PR) and F-measure since the candidate alignments need to contain as many correct correspondences as possible. AUC(PR) is tolerant against incorrect correspondences in the candidate alignment as long as they are sorted at the end of the candidate alignment, and the recall is high. If the expert has a lot of time, it is even better to find a few correct correspondences among the incorrect ones than to find none or stop the manual verification.

While the AUC values are a suitable means to provide a ranking between matchers, there is a loss of information from the original curves. For example, while we can observe from the curves that ServOMap outperforms GOMMA for small n, but is inferior for for larger n, but this is not reflected in the AUC values. Thus, the AUC values are suitable for a rough estimate of the performance of different matchers in supervised matching scenarios, but Precision@N curves are essential for fine-grained assessments for individual values of n.

Since even the best Precision@N curve is not very close to the optimal Precision@N curve, there is still room for improvement. Due to the large amount of available candidate alignments generated by the various matchers, we do not have to focus only on one candidate alignment but can combine them to obtain alignments that are even more valuable for the supervised matching use case.

5.2 Combining Matching Strategies Evaluation Results

Figure 5 shows the Precision@N curves for the seven strategies we implemented. Since it is always the same alignment $\mathcal{C}_{all}$, the precision, recall, and F-measure values are exactly the same for each strategy, but their Precision@N curves fundamentally differ from each other. Thus, the Precision@N curves reveal significant details between the approaches. The best strategies commonly start with precision values close to 1 and

even have a precision value of about 0.8 when n reaches the size of the reference alignment. Thus, they are close to the optimal Precision@N curve.

The random strategy is very close to the average precision value ($\frac{|R|}{|C_{all}|} = \frac{3161}{21787} = 0.145$) over all n. Sorting the correspondences by their confidence value is only slightly above the random baseline. Since maximum confidences across all matchers are used for sorting, this strategy is prone to adding up all false positives found by all matchers, i.e., single matchers assigning large values to single faulty correspondences.

Both recall and F-measure on the partial reference alignment approaches result in a similar Precision@N curve which is located in the midfield. They remain on a quite stable precision level for most n up to 5000 but never achieve very high precision values. The shape of the Precision@N curve highly depends on the matcher with the highest recall/F-measure value. For example, Hertuda has the highest recall value on the partial reference alignment and its candidate alignment already contains 5559 correspondences. Thus, it dominates the results of all other matchers. Similar phenomena can be noticed for the strategy which is based on F-measure values.

In contrast, the strategy which orders the candidate alignments according to the precision value of the partial reference alignment shows good results. Most of the correct correspondences are ranked high, which results in a high precision value for smaller n. Other promising strategies are the Majority Vote and the Current Leadership. In particular the Majority Vote outperforms all other strategies for most values of n. Moreover, in contrast to the Precision on Partial Reference Alignment or the Current Leadership approach, it does not even require any additional resources like a (partial) reference alignment. This makes Majority Vote the favorable strategy for combining matchers in supervised matching scenarios.

Altogether, the experiments show that combining matchers by unifying their candidate alignments and applying a proper sorting strategy helps to significantly increase performance. Thus, a human expert can get (almost) the maximal number of new correct correspondences within a particular time frame. Furthermore, it can be observed that, although all strategies yield the same recall, precision, and F-measure, the performance differs significantly w.r.t. Precision@N curves.

6 Related Work

Supervised ontology matching is related to the employment of automatic indexing systems as recommender for intellectual indexing, as described in [11]. In most cases, the terms assigned to publications need to be perfectly correct such that an automatic approach without any further manual verification of the indexed terms is not feasible. Semi-automatic indexing approaches automatically generate a list of index terms for each publication, e.g. with the best x index terms that have been detected [12]. Afterwards, a domain expert can manually check this list and take over the correctly assigned index terms. It is similar to our use case, whenever the domain expert likes to index as many publications as possible with a maximal amount of suitable terms.

Also in the field ontology matching, approaches with user involvement have been developed. Most of these interactive matchers ask the user for the validation of correspondences during the matching process. With the knowledge of the correctness of

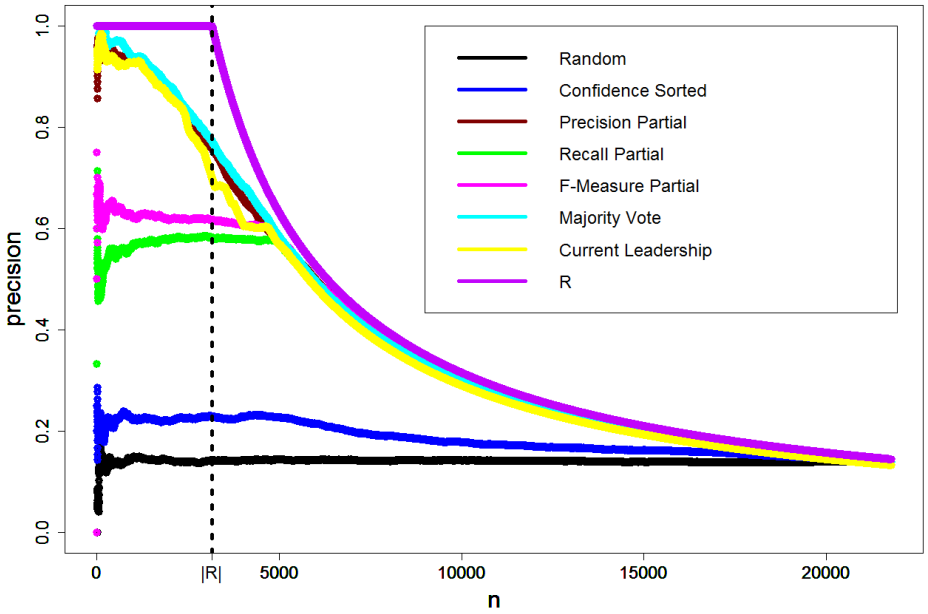


Fig. 5: Precision@N curves for the combined strategies

a correspondence, they try to generate a better candidate alignment. For example, the matchers PROMPT [24] and LogMap2 [17] use the verifications to detect incorrect correspondences and solve inconsistencies. To et al. describe a whole framework supporting supervised and semi-supervised learning approaches [29]. To reduce the amount of user interactions, active learning [22] can be applied. Other techniques use the manual input to learn suitable weights to merge different matching strategies [7, 9]. Combining several strategies or even matchers can significantly increase the quality of the resulting candidate alignment [8].

There are a lot of measures to evaluate matchers according to different criteria. Since we already compared Precision@N curves to the most related ones during the experiments, we renounce to list them again as related work.

7 Conclusion and Future Work

In this paper, we have introduced supervised ontology matching as a solution for a practical problem: the manual ontology alignment generation based on automatically generated correspondences. With Precision@N curves, we have developed an adequate visualization of the candidate alignments which is perfectly tailored to our use case. Precision@N curves show the different characteristics of the matchers in order to decide which matcher performs best as preprocessor for the human expert to generate an alignment with the maximal number of correct correspondences within a given time frame. In an optimal case, the expert first checks all correct correspondences followed by the incorrect ones to always get the maximal amount of correct correspondences.

Since most matchers assign confidence values to correspondences, we can use them to sort the candidate alignment.

Commonly used measures like precision, recall, and F-measure do not leverage confidence values provided by many matchers, do not provide an ordering of the candidate alignment, and are thus not suitable for the evaluation according to our use case. Measures such as PR or ROC curves consider the ordering, but they emphasize other facets of the evaluation, and are thus not suitable for supervised matching. For other use cases, other measures might be more suitable, and it is not feasible to only have one measure for all use cases. Reducing the evaluation to the comparison of one single value, e.g. compare AUC(Pr_n) values instead of considering the curves, allows for ranking matchers for our use case, but induces a certain loss of information compared to the original curves.

Using the candidate alignments of several matchers, we have performed experiments on finding the optimal sorting for the union of all alignments. Majority vote has been shown to be the best suitable strategy for combining matchers. Furthermore, the combination of all matchers – given a suitable strategy – outperforms even the best single matcher.

As future work, even more sophisticated strategies for matcher combination can be developed, e.g., by taking individual characteristics of the matchers into account. By now, the domain expert manually checks each correspondence. This enormous effort might get reduced whenever the interactions between human experts and matchers become more interlinked. For example, if the expert can be sure that some correspondences created by the matcher are correct, they can be just adopted without a manual verification. This is especially interesting for applications where a high quality of the alignment is required but it does not harm if a small amount of incorrect correspondences is contained.

Another promising line of research is the combination of supervised and interactive ontology matching. In this paper, we have strictly separated the matching step from the manual inspection. However, using the user action as feedback, the matching process can also be re-tuned according to the user's needs while it is running, hereby providing even better results.

In our experiment, we assume that the costs of verifying a correspondence is always the same. In reality, this is generally not the case because some correspondences are harder to check than others, for instance obviously incorrect ones. By presenting the expert related correspondences – containing the same classes or classes which are close to each other in the hierarchies – the time which is needed for the verification can be further reduced. This especially holds whenever correspondences are mutually exclusive or a strict 1:1 alignment is required.

Acknowledgments. We would like to thank Andreas Oskar Kempf and Benjamin Zapilko from GESIS - Leibniz Institute for the Social Sciences - for the manual evaluation of the matching results and the collaboration in the OAEI library track.

References

1. José Luis Aguirre, Kai Eckert, Jérôme Euzenat, Alfio Ferrara, Willem Robert van Hage, Laura Hollink, Christian Meilicke, Andriy Nikolov, Dominique Ritze, François Scharffe, Pavel Shvaiko, Ondřej Šváb Zamazal, Cássia Trojahn, Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, and Benjamin Zapilko. Results of the Ontology Alignment Evaluation Initiative 2012. In *Proc. of the 7th Int. Workshop on Ontology Matching*, 2012.
2. Mouhamadou Ba and Gayo Diallo. ServOMap and ServOMap-lt Results for OAEI 2012 . In *Proc. of the 7th ISWC workshop on ontology matching (OM)*, 2012.
3. Michelle Cheatham. MapSSS Results for OAEI 2011. In *Proc. of the 6th ISWC workshop on ontology matching (OM)*, 2011.
4. Thanh Tung Dang, Alexander Gabriel, Sven Hertling, Philipp Roskosch, Marcel Wlotzka, Jan Ruben Zilke, Frederik Janssen, and Heiko Paulheim. HotMatch Results for OEAI 2012. In *Proc.of the 7th ISWC workshop on ontology matching (OM)*, 2012.
5. Jérôme David, Fabrice Guillet, and Henri Briand. Association rule ontology matching approach. *International Journal on Semantic Web and Information Systems*, 3(2):27–49, 2007.
6. Jesse Davis and Mark Goadrich. The relationship between Precision-Recall and ROC curves. In *Proc. of the 23rd int. conference on Machine learning (ICML)*, 2006.
7. S. Duan, A. Fokoue, and K. Srinivas. One Size Does Not Fit All: Customizing Ontology Alignment Using User Feedback. In *Proc. of the 9th Int. Semantic Web Conference*, pages 177–192, 2010.
8. Kai Eckert, Christian Meilicke, and Heiner Stuckenschmidt. Improving Ontology Matching using Meta-level Learning. In *Proc. of the 6th European Semantic Web Conference (ESWC)*, 2009.
9. Marc Ehrig, Steffen Staab, and York Sure. Bootstrapping Ontology Alignment Methods with APFEL. In *Proc. of the 4th Int. Semantic Web Conference (ISWC)*, 2005.
10. Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer, 2007.
11. Hermann Fangmeyer. Semi Automatic Indexing : State of the Art. Technical report, North Atlantic Treaty Organization. Advisory Group for Aerospace Research and Development, 1974.
12. Clifford W. Gay, Mehmet Kayaalp, and Alan R. Aronson. Semi-Automatic Indexing of Full Text Biomedical Articles. In *Proc. of the Fall Symposium of the American Medical Informatics Association*, 2005.
13. James A. Hanley and Barbara J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982.
14. Michiel Hazewinkel. *"Harmonic series", Encyclopedia of Mathematics*. Springer, 2001.
15. Sven Hertling. Hertuda Results for OEAI 2012. In *Proc. of the 7th ISWC workshop on ontology matching (OM)*, 2012.
16. Jakob Huber, Timo Sztyler, Jan Noessner, and Christian Meilicke. CODI: Combinatorial Optimization for Data Integration - Results for OAEI 2011. In *Proc. of the 6th ISWC workshop on ontology matching (OM)*, 2011.
17. Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau, Yujiao Zhou, and Ian Horrocks. Large-scale interactive ontology matching: Algorithms and implementation. In *Proc. of the 20th European Conference on Artificial Intelligence (ECAI)*, 2012.
18. Toralf Kirsten, Anika Gross, Michael Hartung, and Erhard Rahm. Gomma: a component-based infrastructure for managing and analyzing life science ontologies and their evolution 2, 6 (2011). *Journal of Biomedical Semantics*, 2(6), 2011.
19. Chris Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT Press, 1999.

20. Philipp Mayr and Vivien Petras. Building a Terminology Network for Search: The KoMoHe Project. In *Proc. of the Int. Conference on Dublin Core and Metadata Applications*, 2008.
21. DuyHoa Ngo and Zohra Bellahsene. YAM++ : A Multi-strategy Based Approach for Ontology Matching Task. In *Proc. of the 18th Int. Conference on Knowledge Engineering and Knowledge Management (EKAW)*, 2012.
22. Axel-Cyrille Ngonga Ngomo, Lehmann Jens, Sören Auer, and Konrad Höffner. RAVEN - active learning of link specifications. In *Proc. of the 6th ISWC workshop on ontology matching (OM)*, 2011.
23. Xing Niu, Haofen Wang, Gang Wu, Guilin Qi, and Yong Yu. Evaluating the Stability and Credibility of Ontology Matching Methods. In *Proc. of the 8th Extended Semantic Web Conference (ESWC)*, 2011.
24. N. F. Noy and M. A. Musen. The PROMPT suite: interactive tools for ontology merging and mapping. *Int. Journal of Human-Computer Studies*, 59(6):983–1024, 2003.
25. Heiko Paulheim. WeSeE-Match Results for OEAI 2012. In *Proc. of the 7th ISWC workshop on ontology matching (OM)*, 2012.
26. Heiko Paulheim, Sven Hertling, and Dominique Ritze. Towards evaluating interactive ontology matching tools. In *Proc. of the 10th Extended Semantic Web Conference (ESWC)*, 2013.
27. Dominique Ritze and Heiko Paulheim. Towards an Automatic Parameterization of Ontology Matching Tools based on Example Mappings. In *Proc. of the 6th ISWC workshop on ontology matching (OM)*, 2011.
28. George B. Jr. Thomas and Ross L. Finney. *Calculus and Analytic Geometry*. Addison Wesley, 9 edition, 1996.
29. Hoai-Viet To, Ryutaro Ichise, and Hoai-Bac Le. An Adaptive Machine Learning Framework with User Interaction for Ontology Matching. In *Proc. of the International Joint Conferences on Artifical Intelligence, Workshop on Information Integration on the Web*, pages 35–40, 2009.
30. Cornelis Jost van Rijsbergen. *Information retrieval*. Butterworths, London (UK), 1975.
31. Hong Zhou, Jian Kang, Feng Chen, and H. Yang. OPTIMA: An Ontology-Based PlaTform-specIfic software Migration Approach. In *Proc. of the 7th Int. Conference on Quality Software (QSIC)*, pages 143–152, 2007.