WeSeE-Match Results for OAEI 2013

Heiko Paulheim[1] and Sven Hertling[2]

[1] University of Mannheim
Data and Web Science Group
heiko@informatik.uni-mannheim.de
[2] Technische Universitt Darmstadt
Knowledge Engineering Group
hertling@ke.tu-darmstadt.de

Abstract. *WeSeE-Match* is a simple, element-based ontology matching tool. Its basic technique is invoking a web search engine request for each concept and determining element similarity based on the similarity of the search results obtained. Multi-lingual ontologies are translated using a standard web based translation service. Furthermore, it implements a simple strategy for selecting candidate mappings interactively.

1 Presentation of the system

1.1 State, purpose, general statement

The idea of *WeSeE-Match* is to use information on the web for matching ontologies. When developing the algorithm, we were guided by the way a human would possibly solve a matching task. Consider the following example from the OAEI anatomy track[1]: one element in the reference alignment are the two classes with labels *eyelid tarsus* and *tarsal plate*, respectively. As a person not trained in anatomy, one might assume that they have something in common, but one could not tell without doubt.

For a human, the most straight forward strategy in the internet age would be to search for both terms with a search engine, look at the results, and try to figure out whether the websites returned by both searches talk about the same thing. Implicitly, what a human does is identifying relevant sources of information on the web, and analyzing their contents for similarity with respect to the search term given. This naive algorithm is implemented in *WeSeE-Match*.

Furthermore, *WeSeE-Match* uses a basic method for interactive matching, which tries to adaptively set the threshold for selecting the final alignment from the set of candidates.

1.2 Specific techniques used

The core idea of our approach is to use a web search engine for retrieving web documents that a relevant for concepts in the ontologies to match. For getting search terms

[1] http://oaei.ontologymatching.org/2013/anatomy/

from ontology concepts (i.e., classes and properties), we use the labels, comments, and URI fragments of those concepts as search terms. The search results of all concepts are then compared to each other. The more similar the search results are, the higher the concepts' similarity score.

To search for websites, we use URI fragments, labels, and comments of each concept as search strings, and perform some pre-processing, i.e., splitting camel case and underscore separated words into single words, and omitting stop words. For every search result, all the titles and summaries of web pages provided by the search engine are put together into one *describing document*. This approach allows us to parse only the search engine's answer, while avoiding the computational burden of retrieving and parsing all websites in the result sets. The answer provided by the search engine contains titles and excerpts from the website (i.e., some sentences surrounding the occurance of the search term in the website). Therefore, we do not use *whole* websites, but ideally only *relevant parts* of those web sites, i.e., we exploit the search engine both for information retrieval and for information extraction.

For each concept c, we perform a single search each for the fragment, the label, and the comment (if present), thus, we generate up to three documents $doc_{fragment}(c)$, $doc_{label}(c)$, and $doc_{comment}(c)$. The similarity score for each pair of concepts is then computed as the maximum similarity over all of the documents generated for those concepts:

$$sim(c_1, c_2) := max_{i,j \in \{fragment, label, comment\}} sim^*(doc_i(c_1), doc_j(c_2)) \quad (1)$$

For computing the similarity sim^* of two documents, we compute a TF-IDF score, based on the complete set of documents retrieved for all concepts in both ontologies.

Using the TF-IDF measure for computing the similarity of the documents has several advantages. First, stop words like *and*, *or*, and so on are inherently filtered, because they occur in the majority of documents. Second, terms that are common in the domain and thus have little value for disambiguating mappings are also weighted lower. For example, the word *anatomy* will occur quite frequently in the anatomy track, thus, it has only little value for determining mappings there. On the other hand, in the library track, it will be a useful topic identifier and thus be helpful to identify mappings. The TF-IDF measure guarantees that the word *anatomy* gets weighted accordingly in each track.

The result is a score matrix with elements between 0 and 1 for each pair of concepts from both ontologies. For each row and each column where there is a score exceeding τ, we return that pair of concepts with the highest score as a mapping. Furthermore, the filter chain explained in [2] is used, which removes mappings for datatype properties with different ranges, as well as mappings that refer to any imported ontologies.

For multi-lingual ontologies, we first translate the fragments, labels, and comments to English as a pivot language [4]. The translated concepts are then processed as described above.

While the threshold is set to fixed values for the standard tracks in the evaluation, we use an interactive method of selecting the threshold in the interactive track. We use a binary search for selecting the threshold, as discussed in [6], using the following algorithm:

1. Set τ to the average threshold of all candidates. Set $\tau_{min} = 0$, $\tau_{max} = 1$.
2. Present a candidate that has a confidence of τ (or the candidate whose confidence is closest to τ) to the oracle.
3. If the candidate is correct, set τ to $\tau_{min} + (\tau_{max} - \tau_{min}/2)$, τ_{min} to the previous value of τ, Otherwise set set τ to $\tau_{max} - (\tau_{max} - \tau_{min}/2)$, τ_{max} to the previous value of τ
4. If $\tau_{max} > \tau_{min}$, go to 2.
5. Select the final candidates: all candidates with a threshold above τ plus all candidates that are rated positive by the oracle minus all candidates that are rated negative by the oracle.

Given that the ordering of candidates is optimal, i.e., all wrong candidates have a lower confidence score than all correct candidates, that algorithm will yield a threshold τ that separates correct from incorrect candidates.

1.3 Adaptations made for the evaluation

The 2012 version of *WeSeE-Match* [5] used Microsoft Bing as a search engine, as well as the Microsoft Translator API. In order to create a matching system that does not use any services which require payment, we decided to make the following changes for the 2013 evaluation campaign:

- The Bing web search was replaced by JFreeWebSearch[2], which encapsulates the free FAROO web search API[3].
- The Microsoft Translator API was replaced by the Web Translator API[4].

The other new feature for this year's evaluation was the inclusion of the interactive post-processing method. The same code for handling interactive matching was also used in the *Hertuda* matching system [3] for the OAEI 2013 evaluation campaign.

The parameter τ was set to 0.42 for multi-lingual and to 0.51 for mono-lingual matching problems in the non-interactive tracks.

1.4 Link to the system and parameters file

The system is available from `http://www.ke.tu-darmstadt.de/resources/ontology-matching/wesee-match/`.

2 Results

2.1 benchmark

The results from the benchmark track are not surprising. For those problems where labels, URI fragments and/or comments are present and contain actual terms, i.e., they

[2] `http://www.ke.tu-darmstadt.de/resources/jfreewebsearch`

[3] `http://www.faroo.com/`

[4] `http://sourceforge.net/projects/webtranslator/`

are not replaced by random strings, *WeSeE-Match* provides reasonable results. As soon as those strings are removed, the F-measure of *WeSeE-Match* drops, since no other evaluation (e.g., ontology structure) is used by *WeSeE-Match*.

The evaluation of runtime also reveals that *WeSeE-Match* is one of the slowest matching systems participating in the campaign. This is due to the fact that the search engine used restricts the usage to one request per second. Thus, *WeSeE-Match* spends a lot of idle time to fulfill that requirement.

2.2 anatomy and conference

On anatomy, *WeSeE-Match* is one of the worst performing matchers, suffering both in precision in recall. It is in particular interesting to see the drop from last year's performance (F-measure 0.829) to this year's (F-measure 0.47), where the only significant change was the use of a different search engine. This shows that the choice of the web search engine in search-engine based matching can make a huge difference in the results.

In the conference track, the differences to last year are not that significant, ending at a comparable F-measure of 0.55, which makes *WeSeE-Match* an average matcher in the conference track.

2.3 multifarm

For multifarm, the F-measure has dropped from 0.41 in OAEI 2012 to 0.15 in OAEI 2013. As discussed above, the performance on the English-only conference track, which underlies multifarm, has remained stable, so this is mainly an effect of the use of a different translation engine. Again, it becomes obvious that the choice of a different translation engine clearly influences the results.

In detail, the 2012 version of *WeSeE-Match* was capable of matching Chinese and Russian ontologies, because these languages were supported by the Microsoft Translator API, but not the Web Translator API. Furthermore, the results for Czech are significantly below the 2012 results, which shows a suboptimal support for that language in the Web Translator API. For the other languages, the results have remained on a similar level.

2.4 library and large biomedical ontologies

Since *WeSeE-Match* is not optimized for scalability (in particular, necessary waiting times to avoid blocking from search engine providers slow down the matcher), it could not be run on those larger tracks due to time limits. However, the approach does in principle scale up to larger ontologies as well. In the OAEI 2012 campaign, for example, *WeSeE-Match* was capable of completing the library track, when the time limit was set to one week instead of one day [1].

Table 1. Thresholds and results in the interactive track. The table depicts the best possible threshold and the F-measure that is achieved with that threshold, as well as the threshold chosen and the F-measure achieved by *WeSeE-Match*. In cases where an interval of thresholds leads to the same result, we report the average of that interval.

	best possible selection				actual selection			
Test case	Threshold	Precision	Recall	F-measure	Threshold	Precision	Recall	F-Measure
cmt-conference	0.14	0.438	0.467	0.452	0.03	0.304	0.467	0.368
cmt-confOf	0.15	0.385	0.313	0.345	0.26	0.714	0.313	0.435
cmt-edas	0.94	0.889	0.615	0.727	1.00	0.778	0.538	0.636
cmt-ekaw	0.90	0.625	0.455	0.526	0.19	0.500	0.455	0.476
cmt-iasted	0.66	0.800	1.000	0.889	1.00	0.800	1.000	0.889
cmt-sigkdd	0.14	0.786	0.917	0.846	0.23	0.769	0.833	0.800
conf-confOf	0.97	0.583	0.467	0.519	1.00	0.714	0.333	0.455
conf-edas	0.95	0.583	0.412	0.483	1.00	1.000	0.118	0.211
conf-ekaw	0.12	0.333	0.480	0.393	0.18	0.538	0.280	0.368
conf-iasted	0.15	0.500	0.357	0.417	0.79	1.000	0.214	0.353
conf-sigkdd	0.28	0.818	0.600	0.692	0.78	0.750	0.600	0.667
confOf-edas	1.00	0.643	0.474	0.545	1.00	0.778	0.368	0.500
confOf-ekaw	0.11	0.619	0.650	0.634	1.00	0.667	0.600	0.632
confOf-iasted	0.90	0.571	0.444	0.500	1.00	1.000	0.333	0.500
confOf-sigkdd	0.69	0.667	0.571	0.615	0.19	0.500	0.429	0.462
edas-ekaw	0.60	0.526	0.435	0.476	1.00	1.000	0.130	0.231
edas-iasted	0.28	0.500	0.421	0.457	0.19	0.500	0.105	0.174
edas-sigkdd	0.95	0.875	0.467	0.609	1.00	0.875	0.467	0.609
ekaw-iasted	0.57	0.429	0.600	0.500	1.00	0.556	0.500	0.526
ekaw-sigkdd	0.81	0.875	0.636	0.737	1.00	1.000	0.273	0.429
iasted-sigkdd	0.94	0.647	0.733	0.688	1.00	0.667	0.133	0.222

2.5 interactive

The interactive matching component in *WeSeE-Match* – which is the same as in Hertuda – tries to find an optimal threshold via binary search for selecting the final mapping, as discussed above. Table 1 depicts the results. For comparison, we also show the optimal threshold that could have been selected in theory.[5]

It can be observed that the thresholds chosen by our algorithm are most often far from the optimum, which is also reflected in the F-measure achieved. In most cases, the selected threshold is higher than the optimal, i.e., the selection is biased towards precision. In fact, 1.0 is often chosen as a threshold. The reason is that that mapping elements such as `conference#Conference = eads#Conference`, which naturally re-

[5] There are cases where the F-measure achieved by the actual selection is higher. This is due to the fact that in a post-processing step, all information obtained from the oracle is exploited by removing the wrong mappings and including the correct ones, even if they do not fall into the interval defined by the final threshold selection. Furthermore, in some cases, the precision is lower despite a higher threshold. This is due to the fact that the interactive track uses a slightly different reference alignment than the original conference track, on which the optimal thresholds have been determined.

ceive a confidence score of 1.0, are rated wrong by the oracle, so that our binary search approach sets the threshold τ to the maximum possible, i.e., 1.0.

In general, while our interactive approach would work well for ideally sorted candidates, its performance with real confidence scores provided by *WeSeE-Match* (and also Hertuda) are not satisfying. Thus, using a greedy algorithm here is maybe not the best choice, and other means to determine an optimal threshold, such as estimating the F-measure based on user interaction, as discussed in [6], may be a better option for future versions.

3 Conclusion

In this paper, we have discussed the results of the 2013 version of *WeSeE-Match*. It can be observed that the choice for free services instead of commercial ones has changed the performance of *WeSeE-Match* for the worse. The trade-off between the use of commercial services and high-quality results is not easy to address.

Furthermore, it can be seen that a naive greedy approach for selecting a threshold parameter interactively does not provide satisfying results, which calls for more sophisticated methods.

References

1. Jos Luis Aguirre, Kai Eckert, Jrme Euzenat, Alfio Ferrara, Willem Robert van Hage, Laura Hollink, Christian Meilicke, Andriy Nikolov, Dominique Ritze, Francois Scharffe, Pavel Shvaiko, Ondrej Svab-Zamazal, Cssia Trojahn, Ernesto Jimnez-Ruiz, Bernardo Cuenca Grau, and Benjamin Zapilko. Results of the ontology alignment evaluation initiative 2012. In *Seventh International Workshop on Ontology Matching (OM 2012)*, 2012.
2. 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 *Seventh International Workshop on Ontology Matching (OM 2012)*, 2012.
3. Sven Hertling. Hertuda results for oeai 2012. In *Seventh International Workshop on Ontology Matching (OM 2012)*, 2012.
4. Michael Paul, Hirofumi Yamamoto, Eiichiro Sumita, and Satoshi Nakamura. On the importance of pivot language selection for statistical machine translation. In *2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 221–224, 2009.
5. Heiko Paulheim. Wesee-match results for oeai 2012. In *Seventh International Workshop on Ontology Matching (OM 2012)*, 2012.
6. Heiko Paulheim, Sven Hertling, and Dominique Ritze. Towards evaluating interactive ontology matching tools. In *Lecture Notes in Computer Science*, 2013.