WikiMatch – Using Wikipedia for Ontology Matching

Sven Hertling, Heiko Paulheim

Technische Universität Darmstadt
Knowledge Engineering Group
hertling@ke.tu-darmstadt.de, paulheim@ke.tu-darmstadt.de

Abstract. Finding correspondences between different ontologies is a crucial task in the Semantic Web. Ontology matching tools are capable of solving that task in an automated manner, some even dealing with ontologies in different natural languages. Most state of the art matching tools use internal element and structure based techniques, while the use of large-scale external knowledge resources, especially internet resources, is still rare. In this paper, we introduce *WikiMatch*, a matching tool that exploits Wikipedia as an external knowledge source. We show that using Wikipedia is a feasible way of performing ontology matching, especially if different natural languages are involved.

Keywords: Ontology Matching, Ontology Alignment, Wikipedia, External Resources

1 Introduction

Ontologies are an essential building block of the Semantic Web. They formally describe the vocabulary used in a domain in a machine-interpretable way. Thus, ontologies can be used to unambiguously exchange information between machines. If multiple ontologies are used in parallel in a domain, e.g., when integrating two data sets, mappings between those ontologies are required.

Approaches for finding such mappings or alignments automatically are called *ontology matching* approaches [8]. Possible applications the integration of different data sets, the discovery of heterogeneously described web services, or the exchange of business data between business partners.

Many state of the art matchers are based on *internal* techniques, i.e., they only use the knowledge contained in the ontologies to match, but no external knowledge sources. Such matchers compare local names and labels of elements contained in the ontologies, and use structural features. In contrast, *external* techniques make use of external resources, such as synonym lists, dictionaries, or linguistic resources such as wordnet. In many cases, e.g., for recognizing synonyms, they produce useful results.

On the other hand, such matchers are restricted to the domain of the resources they use, and most of the resources are limited and outdated if not maintained by the tool developer.

In this paper, we present *WikiMatch*, an ontology matching approach based on Wikipedia as an external resource. The knowledge in Wikipedia is based on 23 million articles written by volunteers around the world, covering almost every possible domain at least to a certain depth. Wikipedia pages exist in 285 languages with links between articles in different languages[1], hence, it is also usable for matching ontologies in different languages.

The goal of our approach is to make this large knowledge source usable for ontology matching. To that end, we present an approach that exploits Wikipedia's search functionality and inter-language links for finding mappings between ontologies.

The rest of this paper is structured as follows. Section 2 introduces the state of the art and discussed related matching approaches. Section 3 describes our approach and two different variants, and section 4 shows the evaluation results for both variants, using ontologies and reference alignments from the Ontology Alignment Evaluation Initiative (OAEI)[2]. This paper is then summarized by a conclusion and an outlook on future work.

2 Related Work

Discovering and using relevant sources of background knowledge has been named as one of the ten main challenges to ontology matching [22]. One possible approach is the use of *upper ontologies*, i.e., general purpose ontologies, as background knowledge. Such approaches try to find mappings between two ontologies by relating their elements to a comprehensive upper level ontology and then computing similarities within that ontology between the mapped terms [15]. As such, upper ontologies may be detailed and contain rich information, e.g., about alternative names or spellings for concepts, this approach can help increasing the result quality of ontology matching.

Most approaches employing upper ontologies only use a small set of fixed comprehensive ontologies, such as *Proton* in *BLOOMS+* [11], or *SUMO* in *LOM* [13]. Some authors have also discussed the potential of using domain-specific upper level ontologies for matching tasks in certain domains [1]. In contrast, Sabou et al. [21] have discussed a generic approach using dynamic discovery of suitable external ontologies by employing the ontology search engine Swoogle[3]. This search engine is employed to find suitable ontologies to be used as upper ontologies in the matching process.

Apart from upper-level ontologies, another widely used external knowledge source are linguistic resources such as thesauri, e.g. *WordNet* [17]. Those resources contain synonym definitions, typical relations between words, or multilingual translations. State of the art tools using such resources comprise, e.g., *AgreementMaker* [7], *LogMap* [12], and *YAM++* [19]. In [5], the use of domain-specific, semi-structured corpora of documents is discussed as a means to ontol-

[1] `http://stats.wikimedia.org`
[2] `http://oaei.ontologymatching.org/`
[3] `http://swoogle.umbc.edu/`

ogy matching in specific domains, but it requires the availability and pre-selection of such documents.

Web data is rarely used in ontology matching. One of the few approaches is *COMS*, which uses the online source *Wiktionary*[4] as lexical background knowledge, and employs the Google Translation API[5] for addressing multi-lingual ontologies [14]. The use of the Google Translation API for multi-lingual ontologies has also been proposed by Fu et al. [9] and Trojahn et al. [23]. Furthermore, the use of Google for synonym detection has been announced for *MapSSS* [4], but not implemented and evaluated to date.

Gligorov et al. have discussed the use of the Google search engine for ontology matching [10]. They use the *Google similarity distance* [6] to compare the similarity of two terms is computed from the number of search results for each of the terms alone, and the terms in combination. Since that approach requires a quadratic number of search engine calls, it does not scale well to larger problems.

Wikipedia, despite being one of the largest cross-domain knowledge collections, and also one of the best-known, has been rarely explored as a source of background knowledge in ontology matching so far. *BLOOMS* uses only Wikipedia's category tree and employs it as an upper ontology (see above), rather than exploiting Wikipedia as a whole. In [3], the exploitation of Wikipedia's cross-language links has been discussed as a means for addressing cross-language ontology matching. In [2], the use of Wikipedia as a large-scale text corpus for ontology alignment has been proposed, but no implemented prototype and evaluation are provided.

3 Approach

The basic idea of our approach is to use Wikipedia's search engine to retrieve a result set of Wikipedia articles describing the term. To support multilingual scenarios, we retrieve all language links per article in a second translation step. Since the article titles are unique for every Wikipedia in one language, we compare the sets of retrieved titles to compute the similarity between two concepts.

Wikipedia is based on a software platform called MediaWiki, written in PHP. This framework is used to run Wikipedia, Wiktionary, Wikinews, and so on. MediaWiki offers an API[6] for all pages which run on this software. That API offers two different search engines which vary in their purpose. The traditional search engine performs a full text search, while OpenSearch is used to assist users with suggestions when typing their search.

The standard search engine[7] performs a full text search in all articles. If a term is misspelled, the response contains a suggestion with for the correctly spelled word. The goal of the search engine is to find Wikipedia articles that contain all words from the input.

[4] http://www.wiktionary.org/
[5] http://code.google.com/apis/language/translate/overview.html
[6] http://www.mediawiki.org/wiki/API:Main_page
[7] http://www.mediawiki.org/wiki/API:Search

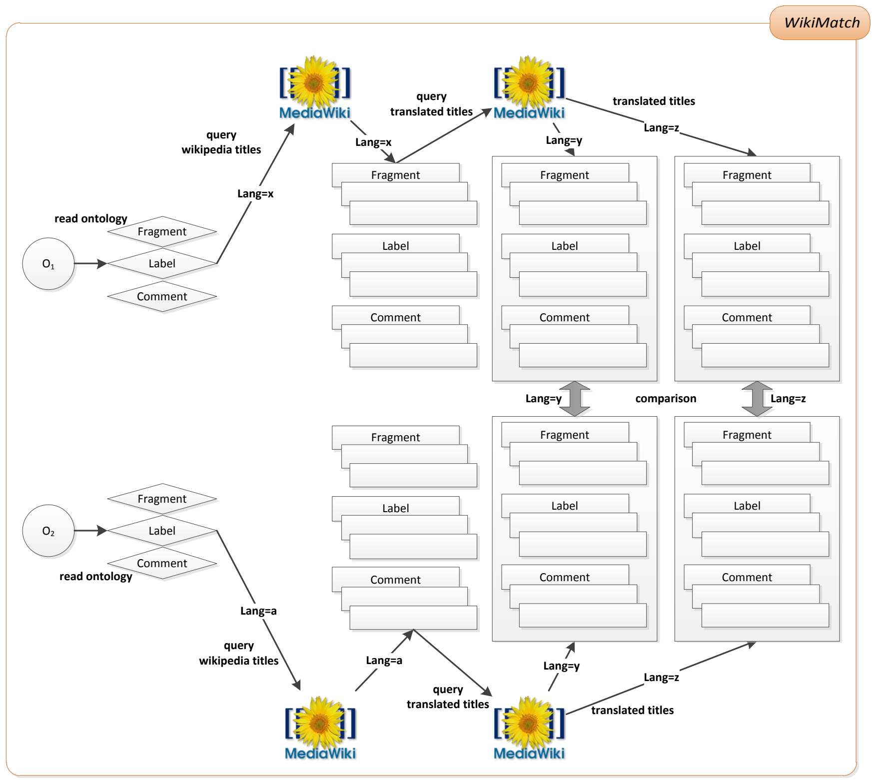

Fig. 1. Illustration of the matching process. For every fragment, label and comment we query wikipedia titles and retrieve all language links as a second step. We compare the requested titles with the same language and returns the maximum of the cross product from fragment, label and comment.

The other search engine is *Opensearch*[8]. It is used for suggesting some terms while the user is typing in the search box on Wikipedia. This search is not applicable for our task, because we already have the full term and hence do not need any hint on possible completions. In our preliminary experiments, that search engine did not work well. Often, labels and comments are composed of many words (tokens), and in that case, the reply of Opensearch is empty[9]. Thus, we use Wikipedia's standard search engine.

For performing searches, we use the URI fragments, labels, and comments of each concept as input strings to the search input. Since the search engine tries to

[8] `http://www.mediawiki.org/wiki/API:Opensearch`

[9] See this example taken from the OAEI conference track: `http://en.wikipedia.org/w/api.php?action=opensearch&search=Subject%20Area&limit=10&namespace=0&format=jsonfm`

find *all* the words in the input query in a Wikipedia article and does not ignore stop words, we remove stop words in a preprocessing step. After preprocessing, our approach uses two different variants for performing the search: taking the preprocessed strings as search input, or searching for each individual token in the input string.

For example, from the label *member of the program committee*, the stop words *of* and *the* would be removed in a first step. Our standard search approach then searches for *member program committee*, while the individual token search approach would trigger three searches for *member*, *program*, and *committee*.

In both search variants, we divide the ontology elements to match into three sets: classes, datatype properties, and object properties. We match only concepts of the same part. This yields on the one hand in better performance and higher precision and on the other hand all mappings are consistent with OWL Lite/DL. Additionally, we can adjust individual threshold values for every type of mapping, like class-class or property-property mappings.

For every ontology concept we extract the fragment, labels and comments, and compare each combination of concepts from both ontologies. Each fragment, label, and comment (or in the individual token search approach, each token thereof) is sought via the Wikipedia search engine. The result is a set of documents per fragment, label, and comment. From those sets, the similarity of two concepts is computed. We compare only titles with the same language. Given that the search for a term t – which can be a label, URI fragment, or comment – returns a set $S(t)$ of Wikipedia article titles (which can be in any language), the similarity between two concepts (i.e., classes and properties) c_1 and c_2 from two ontologies O_1 and O_2 is defined as

$$max_{t_i \in \{label(c_i), fragment(c_i), comment(c_i)\}, i \in \{1,2\}} \frac{\#(S(t_1) \cap S(t_2))}{\#(S(t_1) \cup S(t_2))} \tag{1}$$

If the similarity exceeds a certain threshold, we return a mapping element for the two concepts.

The sets of Wikipedia articles are retrieved by first searching for the Wikipedia article, and then retrieving all translations to other articles in a second step. Thus, our approach treats both single-language and multi-language ontology matching problems the same.

To address the correct search engine, we extract the ontologies' language tags and create a URL like *http://(lang-tag).wikipedia.org/w/api.php*. To this URL we send a request for n titles[10]. The results are all in the language we extracted from the ontology. To compare titles from other languages, we add all *language-links* appear on the requested wikipedia pages. If the answer contains a suggestion for a spelling correction, we make another query in order to get better result for misspelled words. Figure 1 depicts the matching process in a schematic way.

As the simple search approach uses the Wikimedia search interface in a trivial way, requesting articles for whole strings such as *Member of the Program*

[10] For the evaluations in this paper, we have set $n = 50$

```
float getsimilarity(term1, term2) {
  titlesForTerm1 = getAllTitles(term1);
  titlesForTerm2 = getAllTitles(term2);

  commonTitles = intersectionOf(titlesForTerm1, titlesForTerm2);
  allTitles = unionOf(titlesForTerm1, titlesForTerm2);

  return #(commonTitles) / #(allTitles);
}

List<WikipediaPage> getAllTitles(searchTerm) {
  removeStopwords(searchTerm);
  removePunctuation(searchTerm);

  if(simpleSearch) {
    resultList = searchWikipedia(searchTerm);
  }

  if(individualTokenSearch) {
    tokens = tokenize(searchTerm);
    for each token in tokens
      resultList = resultList + searchWikipedia(searchTerm);
  }

  for each page in results
    resultList = resultList + getLanguageLinks(page);

  return resultList;
}
```

Fig. 2. Algorithm for exploiting Wikipedia in Ontology Matching

Committee. Since especially comments can be fairly long, we have also implemented an alternative variant searching for *individual tokens* in the names, such as *member*, *program*, *committee*. This approach is expected to increase the recall, but maybe yield lower precision. Figure 2 shows the algorithm for both search approaches in pseudo code.

4 Evaluation

We have evaluated our tool with benchmarks from the OAEI matching campaign[11]. In this paper, we will focus on the real-world use case from the conference domain (conference), as well as the multi-lingual dataset (multifarm).

[11] http://oaei.ontologymatching.org/

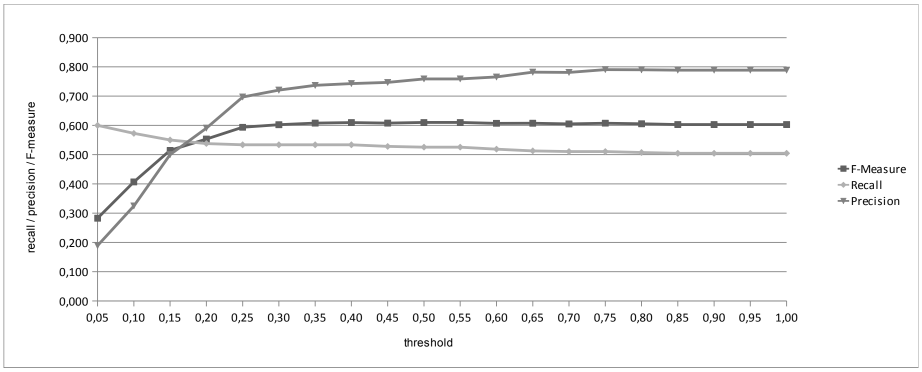

Fig. 3. Average F-measure, precision and recall for OEAI conference track with *Simple Search Approach*.

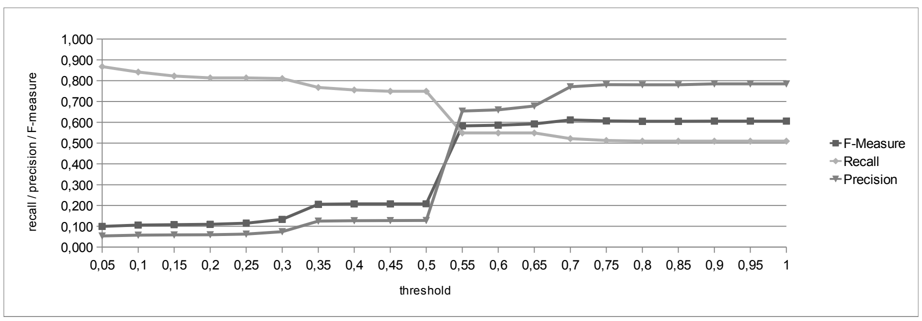

Fig. 4. Average F-measure, precision and recall for OEAI conference track with *Individual Token Search Approach*.

4.1 Evaluation on Conference Track

The conference track consists of 16 ontologies about the domain of conferences. Each of those ontologies are compared in a pairwise setting. For self-evaluation, a subset of seven ontologies is given with reference alignments, thus resulting in 21 possible test cases combinations. The following results are based on those 21 test cases and show average values over all 21 cases.

All the following diagrams are organized as follows: On the x-axis, we use different values for the threshold t, and depict recall, precision, and f-measure for those different values.

Figure 3 shows the results for the *Simple Search Approach*. The maximum f-measure of 0.610 is reached when using a threshold of 0.5. In general, for threshold values above 0.25, there are only small variations in f-measure.

Figure 4 shows the results achieved with *Individual Token Search*. There is a significant leap between 0.5 and 0.55. With a threshold of 0.5 we only get a f-measure of 0.208, while ith a threshold of 0.55, the f-measure rises to 0.582. The maximum f-measure that can be reached with this approach is 0.611, using a threshold of 0.7.

These results show that for the conference track, both approaches converge to about the same maximum f-measure when setting an appropriate threshold. If we compare our result to the OAEI 2011.5 results[12], *WikiMatch* is on the fourth rank, between *CODI* and *Hertuda*, and in particular performs significantly better than the baseline comparing concepts based on string similarity and stop word filtering.

Our matching time for the *Simple Search Approach* is 1340 seconds, which is 22 minutes and 20 seconds. *Individual Token Search Approach* takes a little bit longer. It was about 1454 seconds (24 minutes and 14 seconds).

4.2 Evaluation on Multifarm Track

The multifarm dataset is designed for multilingual ontology matching. It is based on the conference dataset described above, which is translated into eight different languages, i.e., Chinese, Czech, Dutch, French, German, Portuguese, Russian, and Spanish [16].

The evaluation of *Simple Search Approach* is depicted in figure 5. It shows that the maximum f-measure of 0.210 is reached for a threshold of 0.06, with recall decreasing at a great pace. The maximum recall is only 0.35.

Figure 6 shows the results that can be achieved on the multifarm track with individual token search. The maximum f-measure that can be achieved is 0.179 at a threshold of 0.06, with recall and precision behaving similarly to simple search. Thus, for multi-lingual problems, simple search yields slightly better results. The results are competitive with the top 5 tools at the recent OAEI 2011.5 evaluation[13].

Table 1 depicts results on the multifarm track, showing those language pairs that were part of the OAEI evaluation 2011.5 (i.e., excluding Chinese and Russian).

4.3 Performance Evaluation and Scalability

Requesting web resources at run-time usually generates run times that are not competitive internal matching approaches or external matching approaches using only local resources. On the conference and multifarm datasets, a pair of ontologies takes about 360 seconds with simple search and 450 seconds[14] with individual token search to process.

However, in contrast to approaches using co-occurence analysis on Wikipedia [18] or Google Distance [6], which require a quadratic number of search requests, our approach only issues a linear number of search requests, since it only searches for concepts in the ontologies, not for combinations of such concepts. Thus, despite being slower than other approaches, it is scalable to larger matching problems.

---

[12] `http://oaei.ontologymatching.org/2011.5/results/conference/index.html`

[13] `http://oaei.ontologymatching.org/2011.5/results/multifarm/index.html`

[14] On a Windows 7 64bit PC with an Intel i7(3.4 GHz) processor and 8 GB RAM

	Different ontologies (type i)			Same ontologies (type ii)		
language-pair	F-measure	precision	recall	F-measure	precision	recall
cz-de	0.250	0.295	0.247	0.140	0.488	0.083
cz-en	0.245	0.289	0.233	0.179	0.495	0.110
cz-es	0.269	0.292	0.269	0.147	0.452	0.089
cz-fr	0.211	0.264	0.191	0.143	0.463	0.085
cz-nl	0.219	0.259	0.208	0.152	0.508	0.091
cz-pt	0.157	0.189	0.163	0.106	0.308	0.066
de-en	0.290	0.280	0.345	0.252	0.475	0.173
de-es	0.256	0.259	0.301	0.198	0.423	0.134
de-fr	0.275	0.278	0.307	0.200	0.516	0.126
de-nl	0.277	0.310	0.283	0.224	0.587	0.141
de-pt	0.230	0.218	0.276	0.154	0.345	0.100
en-es	0.281	0.265	0.350	0.279	0.489	0.198
en-fr	0.283	0.290	0.315	0.257	0.550	0.171
en-nl	0.304	0.303	0.344	0.237	0.526	0.155
en-pt	0.263	0.250	0.340	0.257	0.431	0.185
es-fr	0.248	0.217	0.312	0.260	0.485	0.179
es-nl	0.224	0.224	0.242	0.224	0.516	0.143
es-pt	0.272	0.207	0.472	0.299	0.453	0.231
fr-nl	0.282	0.252	0.348	0.233	0.529	0.150
fr-pt	0.203	0.159	0.311	0.228	0.382	0.164
nl-pt	0.185	0.163	0.254	0.173	0.315	0.120
average	0.249	0.251	0.291	0.207	0.464	0.138

Table 1. Evaluation on Multifarm track with language and type specific results (*Simple Search Approach*). The threshold is 0.06. The bottom line shows the average of all language pairs.

4.4 Further Observations

There are certain cases that WikiMatch is capable of covering well, while there are others which are more problematic. In the single language scenario, as expected, class names that are equal or similar are matched without problems. For example *sponsorship* vs. *sponzorship* is matched based on the Wikipedia search engine suggesting an alternate spelling. Complex terms such as *member of the program committee* and *program committee member* can also be matched, since stopwords such as *of* and *the* are removed first, as described above. On the other hand, complex property names, such as *has written* and *is author of* are problematic for our approach, since the result lists for *written* and *author* are very dissimilar.

In the multi-lingual case, simple translations such as *Stadt(de)* and *city(en)* and close translations such as *Bankett(de)* and *dinner banquet(en)* can be handled by WikiMatch, as well as property names such as *hat E-Mailadresse(de)* and *has email(en)*. Cases where the translated terms are different, e.g., *Autor von(de)* and *has written(en)*, are equally problematic as in the single language case.

For the multi-lingual case, we have further analyzed the relation between the different language Wikipedia's sizes and the F-Measure achieved. F-Measure and recall are strongly correlated with the Wikipedia's sizes; the best results are

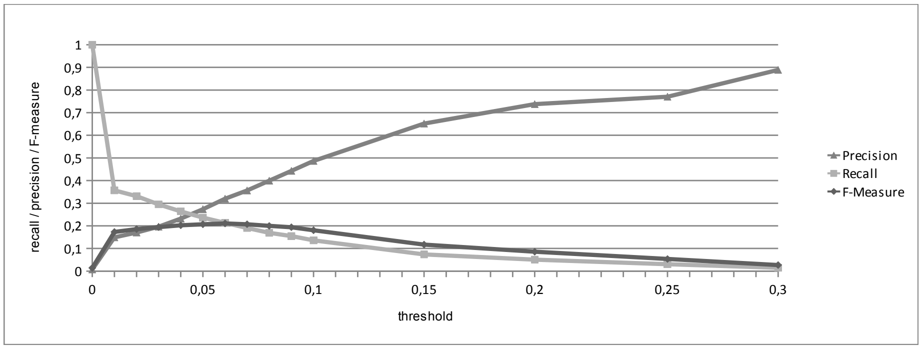

Fig. 5. Average F-measure, precision and recall for OEAI multifarm track with *Simple Search Approach*.

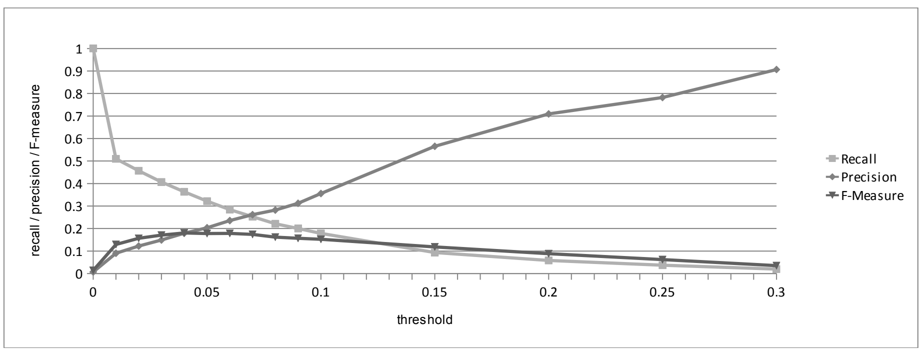

Fig. 6. Average F-measure, precision and recall for OEAI multifarm track with *Individual Token Search Approach*.

achieved with the biggest Wikipedias (English, French, German and Dutch are larger than 1,000,000 entries, and the results between those four languages are the best ones). Conversely, the worst result is achieved for Czech and Portuguese, where the Czech and the Portuguese Wikipedia are the smallest ones among the languages used in multifarm.

5 Conclusion and Future Work

In this paper, we have presented *WikiMatch* as a matching approach based on a large external resource, namely Wikipedia. We use this information to handle synonyms and determine a score describing the equality of two concepts. This paper showed that a matcher using only one external resource without respect to structural information within the ontologies can also yield good results in OAEI benchmarks.

We can handle all domains which Wikipedia covers. Since Wikipedia is community project, it is maintained by many volunteers around the world and grows to more domains each day. Thus, we do not have to take care about updates of the external resources used in in our matcher.

Moreover, we use language links to match ontologies in different languages. Since these links are maintained by humans and not created by bots, we can heavily rely on these links for translation. In this case it is also possible to cope with matching multilingual ontologies.

The aim of this paper was to explore the possibilities of exploiting Wikipedia as an external resource for ontology matching. Since the results are promising and the approach is capable of tackling many hard-to-handle cases, especially in the multi-lingual area, combining the *WikiMatch* approach with other techniques, such as structure-based matching algorithms, would be a natural step to further exploit the approach's potential.

In our experiments, we have compared two different variants for searching contents in Wikipedia, which lead to similar maximum f-measure values, but behave differently in detail. A suitable combination of both approaches could help generating better overall results.

At the moment, despite using various caches, our approach is not very fast. The most time consuming operation is querying Wikipedia. On the other hand, our approach is purely element based, which allows for efficient distribution of the matching problem to many computers [20]. Developing a parallel version of *WikiMatch* would thus eliminate that problem.

While Wikipedia is for sure one of the largest and encompassing online resource, implementing our approach with other such resources, such as *answers.com*, or exploiting even general web search engines, would be an interesting experiment to further assess the value of Wikipedia as a knowledge resource in ontology matching.

In summary, we have shown that a simple approach with Wikipedia as an external resource can handle many different problems in the ontology matching area. Especially, the matching of multilingual ontologies are covered with this approach. The external resource is never outdated and can be used for all domains covered by Wikipedia. We hope that our work will improve future ontology matcher to get better results in monolingual as well as multilingual matching.

References

1. Aleksovski, Z., ten Kate, W., van Harmelen, F.: Ontology matching using comprehensive ontology as background knowledge. In: 1st International Workshop on Ontology Matching (OM 2006). (2006) 13–24
2. Beisswanger, E.: Exploiting Relation Extraction for Ontology Alignment. In Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., 0007, L.Z., Pan, J.Z., Horrocks, I., Glimm, B., eds.: International Semantic Web Conference (2). Volume 6497 of Lecture Notes in Computer Science., Springer (2010) 289–296
3. Bouma, G.: Cross-lingual Ontology Alignment using EuroWordNet and Wikipedia. In: Proceedings of the 7th Conference on International Language Resources and Evaluation (LREC 2010). (2010) 1023–1028

4. Cheatham, M.: MapSSS Results for OAEI 2011. In: Sixth International Workshop on Ontology Matching (OM 2011). (2011)
5. Cheng, C., Lau, G., Pan, J., Law, K., Jones, A.: Domain-specific ontology mapping by corpus-based semantic similarity. In: NSF CMMI Engineering Research and Innovation Conference. (2008)
6. Cilibrasi, R., Vitanyi, P.: The google similarity distance. Knowledge and Data Engineering, IEEE Transactions on **19**(3) (2007) 370–383
7. Cruz, I.F., Antonelli, F.P., Stroe, C.: AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. PVLDB **2**(2) (2009) 1586–1589
8. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, Berlin, Heidelberg, New York (2007)
9. Fu, B., Brennan, R., O'Sullivan, D.: Cross-Lingual Ontology Mapping – An Investigation of the Impact of Machine Translation. In: Fourth Asian Conference (ASWC 2009). (2009) 1–15
10. Gligorov, R., Aleksovski, Z., ten Kate, W., van Harmelen, F.: Using Google Distance to weight approximate ontology matches. In: 16th International World Wide Web Conference (WWW2007). (2007)
11. Jain, P., Yeh, P., Verma, K., Vasquez, R., Damova, M., Hitzler, P., Sheth, A.: Contextual ontology alignment of lod with an upper ontology: A case study with proton. In: Extended Semantic Web Conference (ESWC 2011), Springer (2011) 80–92
12. Jiménez-Ruiz, E., Grau, B.C.: LogMap: Logic-based and Scalable Ontology Matching. In: 10th International Semantic Web Conference (ISWC 2011). (2011) 273–288
13. Li, J.: LOM: A Lexicon-based Ontology Mapping Tool. In: Proceedings of the Performance Metrics for Intelligent Systems (PerMIS. (2004) 2004
14. Lin, F., Krizhanovsky, A.: Multilingual Ontology Matching based on Wiktionary Data Accessible via SPARQL Endpoint. In: Russian Conference on Digital Libraries 2011 (RCDL'2011). (2011) 1–8
15. Mascardi, V., Locoro, A., Rosso, P.: Automatic Ontology Matching via Upper Ontologies: A Systematic Evaluation. IEEE Transactions on Knowledge and Data Engineering **22** (2010) 609–623
16. Meilicke, C., Castro, R.G., Freitas, F., van Hage, W.R., Montiel-Ponsoda, E., de Azevedo, R.R., Stuckenschmidt, H., Šváb-Zamazal, O., Svátek, V., Tamilin, A., Trojahn, C., Wang, S.: MultiFarm: A Benchmark for Multilingual Ontology Matching. Journal of Web Semantics (2012)
17. Miller, G.A.: WordNet: A Lexical Database for English. Communications of the ACM **38**(11) (1995) 39–41
18. Nakayama, K., Hara, T., Nishio, S.: Wikipedia mining for an association web thesaurus construction. Web Information Systems Engineering–WISE 2007 (2007) 322–334
19. Ngo, D., Bellahsene, Z., Coletta, R.: YAM++ – Results for OAEI 2011. In: Sixth International Workshop on Ontology Matching (OM 2011). (2011)
20. Paulheim, H.: On Applying Matching Tools to Large-scale Ontologies. In: Third International Workshop on Ontology Matching (OM-2008). (2008)
21. Sabou, M., d'Aquin, M., Motta, E.: Exploring the semantic web as background knowledge for ontology matching. Journal on Data Semantics **11** (2008) 156–190
22. Shvaiko, P., Euzenat, J.: Ten Challenges for Ontology Matching. In: On the Move to Meaningful Internet Systems: OTM 2008. (2008) 1164–182
23. Trojahn, C., Quaresma, P., Vieira, R.: An API for Multilingual Ontology Matching. In: Proceedings of the 7th Edition of the Language Resources and Evaluation Conference, Malta (2010) 3830–3835