WikiV3 results for OAEI 2017

Sven Hertling

Data and Web Science Group, University of Mannheim, Germany
sven@informatik.uni-mannheim.de

Abstract. WikiV3 is the successor of WikiMatch (participated in OAEI 2012 and 2013) which explores Wikipedia as one external knowledgebase for ontology matching. The results show that the matcher is slightly better than matchers based on string equality and can get higher recall values. Moreover due to the construction of the system it is able to compute mappings in a multilingual setup.

1 Presentation of the system

1.1 State, purpose, general statement

WikiV3 is a system which exploits external knowledgebases - in this case Wikipedia. It uses the MediaWiki API and searches pages which corresponds to a given resource. When exploring the interlanguage links of Wikipedia the system is also able to find mapping between ontologies of different languages. These links point from a Wikipedia page to a correspondent page in Wikipedia with a different language. In contrast to the previous version of the matcher (WikiMatch [1] which participated in OAEI 2012 and 2013) all interlanguage links are now stored in Wikidata [1].

Wikidata is a separate project which allows to build a collaboratively edited knowledge base. One part of this project is to centralize the interlanguage links. Thus the text of Wikipedia is used to better map to Wikidata entities than just using the text available in Wikidata. The search engine of Wikipedia is based on Elasticsearch and is wrapped by a MediaWiki plugin called CirrusSearch[2]. The service provided by this plugin is heavily used by this matcher to find corresponding resources.

The general approach is shown in figure 1.

For each resource of the first ontology a list of corresponding Wikidata concepts is generated. A resource can be a class, datatype property or a object property. All of them are handled seperately to ensure that no mapping between different type of resources is generated (e.g. no class is matched to a datatype or object property). In the same way a list of Wikidata IDs (WIDs) is created for the second ontology. If there is at least one WID of a list in ontology 2 appearing in a list of WIDs in ontology 1, then a mapping is created. This will

[1] https://en.wikipedia.org/wiki/Help:Interlanguage_links
[2] https://www.mediawiki.org/wiki/Help:CirrusSearch

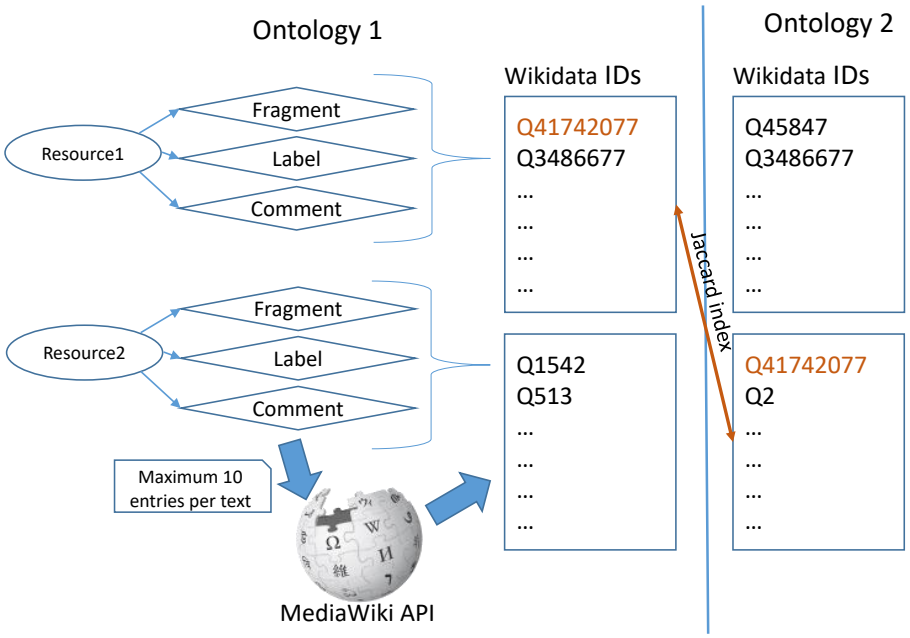

Fig. 1. Matching strategy of WikiV3

result in a n:m mapping which means one concept can be mapped to multiple other concepts. This will be reduced in a further step. The confidence value of a generated mapping is computed by the Jaccard index which is defined as

$$confidence(M) = \frac{WID(Ont1(M)) \cup WID(Ont2(M))}{WID(Ont1(M)) \cap WID(Ont2(M))} \quad (1)$$

where M represents the mapping, Ont1 and Ont2 selects the corresponding resource in Ontology one or two and the function WID returns the set of all Wikidata IDs for the corresponding resource.

The retrieval of WIDs for one resource is now described in more detail. The goal is to generate a list of WIDs which represents a given resource. In the best case there is a WID which directly represents the resource but most of the time there will be only Wikidata entries which partially represents the concept. For achieving that goal, the search API of Wikipedia is used[3].

We queried the search API for all labels, comments and for the fragment of the URI for each resource. The text length is reduced in case it is longer than 300 characters because otherwise the endpoint do not process the query. Furthermore we do not consult the endpoint if 50% of the characters are numbers. Due to the fact that the search endpoint is sensitive to tokenization (compare results from

[3] `https://www.mediawiki.org/wiki/API:Search`

"Review_preference"[4] and "Review preference"[5]), the text is tokenized (using the following characters as a splitting point:",;:()?!._- "). Afterwards all tokens are joined with a single whitespace.

The search URI[6] is parameterized and the language variable is replaced with the ISO 639-1 language code of the literal. In case there is no language tag the default language of the ontology is used (the most used language of all literals). The variable text is replaced with the processed string of the literal. With this query the suggestions of Wikipedia are also explored. Thus misspellings can be detected and fixed.

The results of this API call are Wikipedia page titles. These are converted to WIDs by using the page properties call[7] and the remaining variable joinedTitles is replaced with the Wikipedia page titles. For faster processing all queries are cached.

After comparing the WID lists from each ontology the result is a n:m mapping of the concepts with a computed confidence value which is used in a second step to increase the precision of the matcher. This step will filter all mappings below a given threshold. There are two different thresholds depending if the matching task is multilingual or not. This is detected through the default languages of both ontologies. If they differ then the threshold is not applied because in a multilingual setup the recall would drop drastically. In monolingual setup we choose a threshold of 0.28 which means that more than a quarter of the WIDs of two resources have to match.

The confidence filter does not ensure that we get a 1:1 mapping. Therefore an additional cardinality filter is applied. In case there is an n:m mapping it chooses the one with the best confidence score. As a last step all mappings which do not have the same host URI as the majority of the ontology will be deleted. This ensures that the final mapping does not contain trivial mappings.

1.2 Specific techniques used

The main technique is the usage of Wikipedia API as an external source to find mappings in Wikidata. With this information it is possible to also deal with a multilingual ontology matching setup. The filter steps of the postprocessing ensures a 1:1 mapping which is generally applicable.

1.3 Adaptations made for the evaluation

The only adaption of the system is the threshold setting. In a multilingual setup the threshold is not applied whereas in all other cases a value of 0.28 is used. In

[4] `http://en.wikipedia.org/w/index.php?search=Review_preference`

[5] `http://en.wikipedia.org/w/index.php?search=Review+preference`

[6] `https://{language}.wikipedia.org/w/api.php?action=query&list=search&format=json&srsearch={text}&srinfo=suggestion&srlimit=10&srprop=&srwhat=text`

[7] `https://{language}.wikipedia.org/w/api.php?action=query&prop=pageprops&format=json&titles={joinedTitles}&ppprop=wikibase_item`

context of the matching system this value represents the overlap in percentage of two sets consisting of WIDs representing a resource.

1.4 Link to the system and parameters file

The WikiV3 tool can be downloaded from
`https://www.dropbox.com/s/kqthgvci2onj472/WikiV3.zip`.

2 Results

2.1 Anatomy

WikiV3 has by far the highest runtime due to Wikipedia API calls (nearly 37 minutes). In comparison to the string equivalence base line the system has only a little bit higher F-measure (+0.036) but a better recall (+0.112).

The system is able to match the follwing resources but only with a low threshold.

Table 1. True positive matches in Anatomy

left label	confidence	right label
osseus spiral lamina	0.2857	Lamina_Spiralis_Ossea
thoracic vertebra 9	0.3333	T9_Vertebra
trigeminal V spinal sensory nucleus	0.3333	Nucleus_of_the_Spinal_Tract_ of_the_Trigeminal_Nerve
zygomatic bone	0.3333	Zygomatic_Arch
lumbar vertebra 2	0.3333	L2_Vertebra
nasopharyngeal tonsil	0.3333	Pharyngeal_Tonsil
endocrine pancreas secretion	0.3636	Pancreatic_Endocrine_Secretion
synovium [8]	0.4000	Synovial_Membrane
xiphoid cartilage [9]	0.4286	Xiphoid_Process

If the text is more and more equal then the confidence will also arise. But these examples can be clearly also found by string comparison approaches [3].

2.2 Conference

In conference track the situation is same as in anatomy. WikiV3 is slightly better than the string equivalence baseline (+0.02 F-measure in ra1-M1). Nevertheless it finds correspondences like `http://iasted#Sponsor` = `http://sigkdd#Sponzor` (different spelling) and `http://iasted#Student_registration_fee` = `http://sigkdd#Registration_Student` (different fragment text).

[8] `https://en.wikipedia.org/wiki/Synovial_membrane`

[9] `https://en.wikipedia.org/w/index.php?search=xiphoid+cartilage&title=Special:Search`

2.3 Multifarm

The results are not available up to now.

The system is able to find mappings (exemplary for english-german) like

Table 2. True positive matches in Multifarm

left label	right label
Autor@de	author@en
Konferenz@de	conference@en
hat E-Mailadresse@de	has email@en
Dokument@de	document@en

3 General comments

3.1 Comments on the results

The overall results shows that WikiV3 is able to beat at least the string equivalence matching approaches in terms of F-measure. The recall values are higher than the one of the baselines but could be even higher.

The main drawback of the system is that most of the resources in the ontologies are not described by exactly one concept in Wikipedia (and thus Wikidata). Furthermore the Elasticsearch cluster can only deal with small misspellings and not with semantic equivalent terms or more sophisticated approaches like rewriting the query or applying any machine learning approaches. But this allows reproducible results when fixing a specific version of the cirrussearch dumps.

3.2 Discussions on the way to improve the proposed system

One improvement concern the runtime of WikiV3. Each call to Wikipedia API costs a lot of time. For a future version of this matcher it would be possible to replicate the cirrussearch dumps [10] with the given setting[11] and mapping[12] files. Querying this Elasticsearch cluster is also possible due to the ability to retrieve the corresponding query [13]. With this information a in-depth analysis of the results are feasible. This setup enables a change of the index settings and preprocessing steps to further improve the results.

[10] `https://dumps.wikimedia.org/other/cirrussearch/`

[11] `https://en.wikipedia.org/w/api.php?action=cirrus-settings-dump&formatversion=2`

[12] `https://en.wikipedia.org/w/api.php?action=cirrus-mapping-dump&formatversion=2`

[13] `https://en.wikipedia.org/w/index.php?title=Special:Search&cirrusDumpQuery=&search=cat+dog+chicken`

In the classification of elementary matching approaches [2] the system works at the syntactic element-level and do not use any graph or model based techniques. This is a desired property for this matching system but it can be extended to also use structural information.

4 Conclusions

In this paper we analyzed the results for WikiV3 - an ontology matching system which explores Wikipedia as an external knowledge base. It is able to find more correspondences than a simple string comparison approach. Nevertheless it is only slightly better than that in terms of F-measure. Thus such a mapping approach can be used as a intermediate step to increase the recall also in multilingual setups.

References

1. Hertling, S., Paulheim, H.: Wikimatch - using wikipedia for ontology matching. In: Ontology Matching : Proceedings of the 7th International Workshop on Ontology Matching (OM-2012) collocated with the 11th International Semantic Web Conference (ISWC-2012). vol. 946, pp. 37–48. RWTH, Aachen (2012), http://ub-madoc.bib.uni-mannheim.de/33071/
2. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. In: Spaccapietra, S. (ed.) Journal on Data Semantics IV, Lecture Notes in Computer Science, vol. 3730, pp. 146–171. Springer Berlin Heidelberg (2005)
3. Zhou, L., Cheatham, M.: A replication study: understanding what drives the performance in wikimatch. In: Ontology Matching : Proceedings of the 12th International Workshop on Ontology Matching collocated with the 16th International Semantic Web Conference (ISWC-2017) (2017), to appear