ATBox Results for OAEI 2022

Sven Hertling[0000−0003−0333−5888] and Heiko Paulheim[0000−0003−4386−8195]

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

Abstract. ATBox matcher is a system for matching instances (Abox) as well as schema (Tbox) of two given KGs. The focus of this matcher is on scalability such that it can easily perform huge tasks like Knowledge Graph and Large Bio track. ATBox participates in the OAEI for the third time. For matching, two pipelines (schema and instance) are used for generating candidates. The schema matches are used to further improve the instance alignments.

Keywords: Ontology Matching · Knowledge Graph

1 Presentation of the system

ATBox (also called ATMatcher) is a system designed for matching not only ontologies/schemas (Tboxes) but also instances (Abox). During the past years of the Ontology alignment Evaluation Initiative (OAEI) more and more tracks which requires instances matches are submitted e.g. Spimbench, Link Discovery, Geolink Cruise, and Knowledge Graph. This results not only in ontology but also knowledge graph matching. The question is, how the schema mappings can improve the instance mappings and vice versa. An alternative approach is to switch the matching of schema and instances over and over again. ATBox solves this problem by matching the schema first and uses this information to improve instance matches.

When talking about knowledge graph matching, another dimension is also the size of the ontologies/KGs. Usually they are much bigger than ontologies which models only a specific domain. Therefore, these matching systems need to scale to larger amounts of instances, classes, and properties. Especially the knowledge graph track needs scalable systems which can deal with such an amount of resources[4]. ATBox uses simple comparison methods to first generate a set of candidates and then increases the precision of the alignment to achieve a high F-Measure.

In this year we extended the MELT framework[5] by implementing the alignment repair filter[12] of LogMap. Due to time constraints the matching component was not yet integrated in the final system.

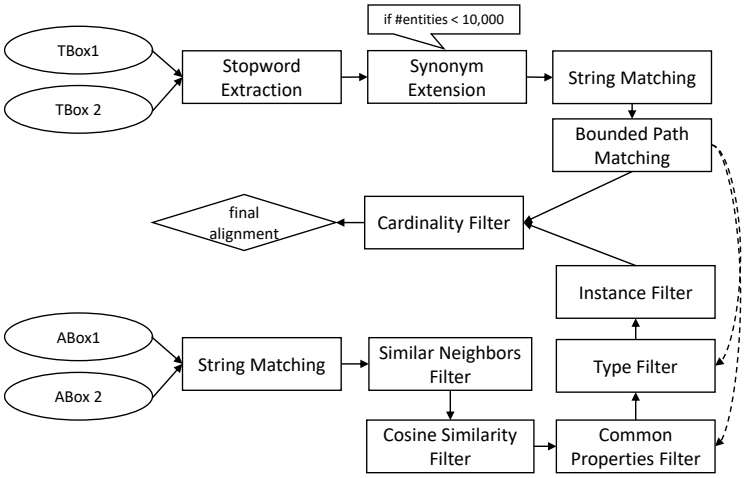

Fig. 1. Overview of the ATBox matcher strategy.

1.1 State, purpose, general statement

The overall matching strategy of ATBox is shown in figure 1. The Tbox and Abox have different processing pipelines but the correspondences are combined in the end to get the final alignment. One of the main differences in comparison to the system submitted last year is the additional bounded path matching for classes.

First have a look at the Tbox matching. It is applied for all classes and properties (`owl:ObjectProperty`, `owl:DatatypeProperty`, and `rdf:Property`). They are retrieved by the jena[1] methods *OntModel.listClasses()* and *OntModel.listAllOntProperties()*.

The first step is to extract KG specific stopwords because in some cases the labels and/or fragments contains tokens which appears very often like `class`, `infobox` etc. If these tokens appears in more than 20 % of all classes/properties, then they are assumed to be stop words.

The synonyms are extracted from the English Wiktionary via DBnary [11]. The extraction process is detailed in the previous results paper[3] similarly to the string matching component. After these components the new bound path matching is executed. This component will match classes which are in between two already matched classes in a hierarchy. Thus it is a structural approach which requires already matched resources. Figure 2 shows an example. The class book is matched to class books and novel to novel. With this information, the class in between is a candidate for another correspondence. Thus it will be added with the average confidence of the other two correspondences.

[1] https://jena.apache.org

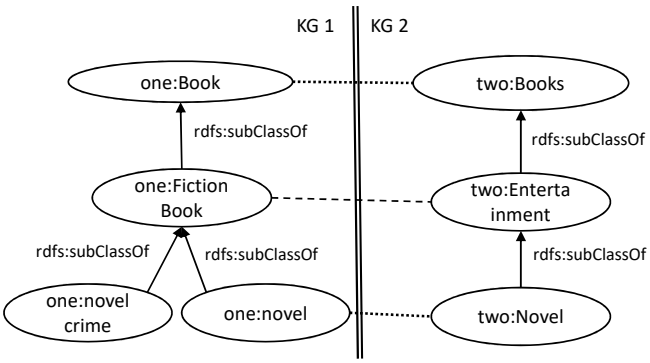

Fig. 2. Bounded path matching of a class hierarchy. The top and bottom lines are already matched classes. The middle line represents a new correspondence.

The instance matching (Abox - shown in the lower part of the figure 1) is kept the same in comparison to the last submission. As a last step, all correspondences are combined and a final cardinality filter ensures a one to one alignment by comparing the confidence scores.

1.2 Specific techniques used

We used the following matching components of MELT [5]:

- ScalableStringProcessingMatcher
- StopwordExtraction
- SimilarNeighborsFilter
- CommonPropertiesFilter
- CosineSimilarityConfidenceMatcher
- SimilarTypeFilter
- NaiveDescendingExtractor
- BoundedPathMatching

1.3 Adaptations made for the evaluation

ATBox matcher is also available as a docker based matcher which runs a HTTP endpoint. The matcher is packaged with the MELT framework[5]. It will generate a docker image which also contains the code for running a small server.

1.4 Link to the system and parameters file

ATBox matcher can be downloaded from
https://www.dropbox.com/s/l344aawh0mw6rjm/atmatcher-1.0-web-latest.tar.gz?dl=0.

2 Results

This section discusses the results of ATBox for each track of OAEI 2022 where the matcher is able to produce results. The following tracks are included: anatomy, conference, bio-ml, commonKG and knowledge graph track.

2.1 Anatomy

The F-Measure is still teh same to last years submission which was 0.794. This still beats the baseline but only by a small margin. The matcher is rather precision oriented and achieves the third highest value after the string baseline, LSMatch, and ALION. Recall should be optimized further than just using synonyms.

2.2 Conference

In the conference track, ATBox matcher has a F-Measure of 0.59 using the rar2-M3 evaluation setup [13] (which is a violation free version of the entailed reference alignment for classes and properties). This is the fourth highest value after LogMap, GraphMatcher, and SEBMatcher. Again the recall (with 0.51) is lower than precision (with 0.69).

2.3 Common Knowledge Graphs

This is a new track which was introduced in OAEI 2021. The task is to align classes between NELL and DBpedia. NELL has 134 classes and 1,184,377 instances whereas DBpedia has 138 classes and 631,461 instances.

ATMatcher is the third best matcher after KGMatcher+ and LogMap with a F-Measure of 0.89. For this track it would help to find classes based on the instances matches as already done by DOME matcher. The currently version of ATMatch only uses the classes to improve the instance correspondences. In the next version we plan to also add this component to increase the capabilities of this matcher.

In the new test case YAGO and Wikidata needs to matched. ATMatcher is again the third best one after KGMatcher+ and Matcha with a F-Measure of 0.87.

2.4 Knowledge Graph

The results of ATBox are similar to previous years because the class hierarchy in this track is not deep. One possibility would be to use the categories (connected with property dcterms:subject[2]) as an additional type of class information.

The F-Measure is 0.85 which is only slightly higher than the baseline using label and alternative label (0.84).

[2] http://purl.org/dc/terms/subject

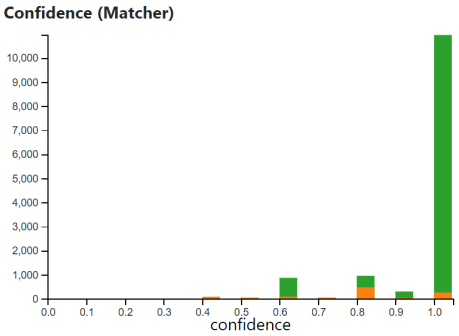

Fig. 3. Confidence values of correspondences for KG track. Green bar is number of true positives and orange bar is number of false positives.

Regarding the runtime, ATMatcher is the fastest one with only 19 minutes for all test cases. Only the baselines are faster which need usually 11 minutes.

The confidences of the overall KG track alignment are visualized in figure 3 (generated with MELT dashboard[9]). The different hard coded confidence values can be seen very well and show that 0.4 and 0.5 has many false positives similar to 0.8.

3 General comments

3.1 Discussions on the way to improve the proposed system

We would like to extend the matching pipeline with further components such as transformer[1,6] based comparison between a textual representation of resources. This only works if already created correspondences needs a precise confidence based on text but does not retrieve any new correspondences because of the complexity to compare all resources in a cross product manner. One way to mitigate this problem is to use sentence transformers[10]. They embed the text in a high dimensional space and thus allows to retrieve the top-k neighbors of a given resource.

Due to the fact that most of the returned alignments are not consistent with the ontology, we also plan to include some alignment repair steps [7] like the ALCOMO component[8].

In case the resources have attached images, it would be also interesting to compare those as well e.g. in the KG track are instances with an image displaying the concept. With a visual comparison (like same persons etc) the confidence of a correspondence can be further increased.

Furthermore the schema matches could be improved with the help of instance correspondences as already shown in the DOME matcher [2].

4 Conclusions

In this paper, we have analyzed the results of ATBox matcher in OAEI 2022. It shows that the system is very scalable and can generate class, property and instance alignments.

Most of the used matching components are furthermore included in the MELT framework[5] to allow other system developers to reuse them.

References

1. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
2. Hertling, S., Paulheim, H.: Dome results for oaei 2019. OM@ ISWC **2536**, 123–130 (2019)
3. Hertling, S., Paulheim, H.: Atbox results for oaei 2020. OM@ ISWC **2788**, 168–175 (2020)
4. Hertling, S., Paulheim, H.: The knowledge graph track at oaei - gold standards, baselines, and the golden hammer bias. In: The Semantic Web: ESWC 2020. pp. 343–359 (2020)
5. Hertling, S., Portisch, J., Paulheim, H.: Melt - matching evaluation toolkit. In: SEMANTICS. Karlsruhe. (2019)
6. Hertling, S., Portisch, J., Paulheim, H.: Matching with transformers in melt. In: OM@ ISWC (2021)
7. Jiménez-Ruiz, E., Meilicke, C., Grau, B.C., Horrocks, I.: Evaluating mapping repair systems with large biomedical ontologies. Description Logics **13**, 246–257 (2013)
8. Meilicke, C.: Alignment incoherence in ontology matching (2011)
9. Portisch, J., Hertling, S., Paulheim, H.: Visual analysis of ontology matching results with the melt dashboard. In: European Semantic Web Conference. pp. 186–190. Springer (2020)
10. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: EMNLP (2019)
11. Sérasset, G.: Dbnary: Wiktionary as a lemon-based multilingual lexical resource in rdf. Semantic Web **6**(4), 355–361 (2015)
12. Solimando, A., Jimenez-Ruiz, E., Guerrini, G.: Minimizing conservativity violations in ontology alignments: Algorithms and evaluation. Knowledge and Information Systems **51**(3), 775–819 (2017)
13. Zamazal, O., Svátek, V.: The ten-year ontofarm and its fertilization within the onto-sphere. Journal of Web Semantics **43**, 46–53 (2017)