ATBox Results for OAEI 2022

Sven Hertling, Heiko Paulheim

Data and Web Science Group, University of Mannheim, Germany

Abstract

In this paper we present the results of the ATBox matcher (ATMatcher for short) during the OAEI campain 2022. The system is able to match instances (Abox) as well as the schema (Tbox) of given knowledge graphs. It scales to large inputs by using effective and modular approaches implemented in the MELT framework. The system participates for the third time in the OAEI. First, two pipelines for matching the schema and instance are used to generate candidates. Afterwards the instance matches are improved and repaired by reusing the candidate class correspondences.

Keywords

Ontology Matching, Knowledge Graph

1. Presentation of the system

ATBox (also called ATMatcher) is a knowledge graph (KG) matching system which is able to match ontologies (Tbox) including properties and classes as well as instances (Abox). KG matching becomes an important step towards building consolidated large scale knowledge graphs. During the last years, more datasets which require instance alignment are published (like Spimbench, Link Discovery, Common KG, and Knowledge Graph track). ATBox tackles this problem by running independent pipelines for instance and schema matches which could also be parallelized to decrease runtime. In a next step, the schema correspondences are used to improve the instance alignment.

1.1. State, purpose, general statement

In this year we extended the MELT framework[2] by implementing the alignment repair filter[3] of LogMap. Due to time constraints the matching component was not yet integrated in the final system. A comparison between this filter and the other repair component in MELT would be a useful next step. Similar evaluation was also executed in [4]. Therefore, in this years submission there are only minor modifications to the system. To make this paper self contained, we keep the description of the system from last years submission [1] which shown below.

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

CEUR Workshop Proceedings (CEUR-WS.org)

Figure 1: Overview of the ATBox matcher strategy (from [1]).
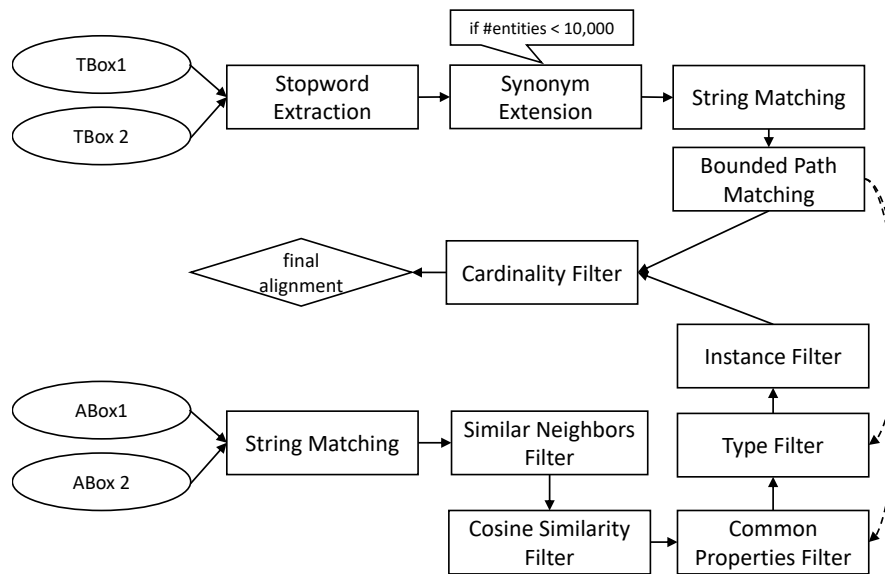
alignment. One of the main differences in comparison to the system submitted last year is the additional bounded path matching for classes.

First have a look at the Tbox matching. It is applied for all classes and properties (`owl:Object-Property`, `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 [5]. The extraction process is detailed in the previous results paper[6] 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.

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 [2]:
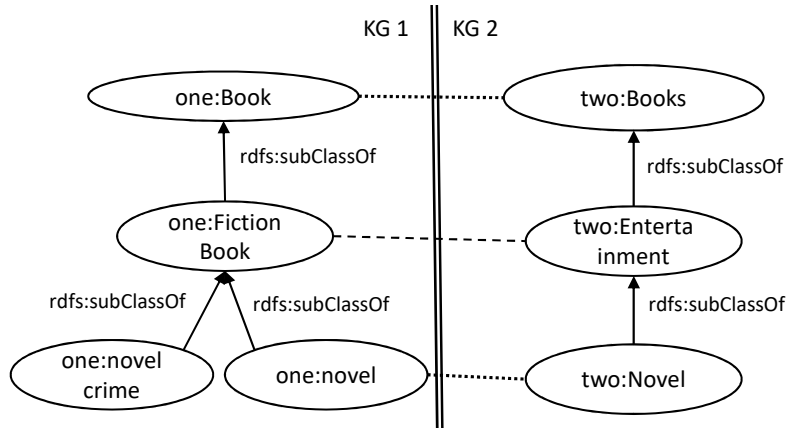
---

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

Figure 2: Bounded path matching of a class hierarchy. The top and bottom lines are already matched classes. The middle line represents a new correspondence (from [1]).

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

1.3. Adaptations made for the evaluation

ATBox matcher is available as a docker image. When starting a container given the image, an HTTP endpoint is started on port 8080 which fulfills the requirements of the web based interface described in the MELT user guide [2].

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]https://dwslab.github.io/melt/matcher-packaging/web

2.1. Anatomy

The F-Measure didn't change in comparison to last years submission which was expected. It is still 0.794. This beats the baseline but only by a small margin. The matcher is very precision oriented and achieves the third highest value after the string baseline and ALIN. The recall can be optimized by not only using synonyms from wordnet but also other external sources. We hope to create a coherent alignment in the next submission by using the aforementioned repair strategies.

2.2. Conference

In the conference track, ATBox matcher has a F-Measure of 0.59 using the rar2-M3 evaluation setup [7] (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).

This year a new subtrack was also evaluated. The task is to match DBpedia to OntoFarm ontologies. ATBox was one of six systems which was able to solve this task. The F-Measure is 0.55 (same with KGMatcher+ and LSMatch) and only LogMap was better in those test cases.

2.3. Common Knowledge Graphs

In this track the task is to align classes in two given KGs. This allows to use instance matches to find useful class correspondences. In 2021 there was only one task where the input graphs are Nell and DBpedia. This year a new task was added which aligns YAGO and Wikidata.

In the first task, ATBox scored 0.89 (only Matcha, KGMatcher+ is better). In the second task the situation is exactly the same - meaning that the proposed system is on the third place. This also shows that the two matching tasks are very similar to each other.

For this track it would be beneficial if classes matches are created with the help of instances correspondences as already done by DOME matcher. The current version only uses the schema matches to improve the instance alignment.

2.4. Knowledge Graph

In the KG track ATMatcher is the best matching system with an overall F-Measure of 0.84. In previous years, some systems were able to beat this score by 0.03 like Wiktionary matcher.

Due to the fact that scalability is one crucial factor for developing the system, it shows clearly that it is the fastest one after the baselines. The proposed system only needs 19 minutes for all test cases.

3. General comments

3.1. Discussions on the way to improve the proposed system

Due to the fact that the two matching pipelines are independent of each other, the runtime of the system could be further decreased by parallelization.

Another possible way to improve the system is to incorporate a transformer model [8] as already shown in [9]. Furthermore, the created alignments can be logically checked at the end of the pipeline. Possible approaches are the LogMap reapir strategy [3] or the ALCOMO component[10].

Finally, one could not only improve instance matches by schema macthes but also the other way around. This was not implemented in the presented approach but would help especially in the Common KG track.

4. Conclusions

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

Most of the components which are used in ATBox are included in the MELT framework[2] which allows other researchers to reuse and compose components in their own systems.

References

[1] S. Hertling, H. Paulheim, Atbox results for oaei 2021, OM @ ISWC 3063 (2021) 137–143.

[2] S. Hertling, J. Portisch, H. Paulheim, Melt - matching evaluation toolkit, in: SEMANTICS. Karlsruhe., 2019, pp. 231–245.

[3] A. Solimando, E. Jimenez-Ruiz, G. Guerrini, Minimizing conservativity violations in ontology alignments: Algorithms and evaluation, Knowledge and Information Systems 51 (2017) 775–819.

[4] E. Jiménez-Ruiz, C. Meilicke, B. C. Grau, I. Horrocks, Evaluating mapping repair systems with large biomedical ontologies., Description Logics 13 (2013) 246–257.

[5] G. Sérasset, Dbnary: Wiktionary as a lemon-based multilingual lexical resource in rdf, Semantic Web 6 (2015) 355–361.

[6] S. Hertling, H. Paulheim, Atbox results for oaei 2020, OM@ ISWC 2788 (2020) 168–175.

[7] O. Zamazal, V. Svátek, The ten-year ontofarm and its fertilization within the onto-sphere, Journal of Web Semantics 43 (2017) 46–53.

[8] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[9] S. Hertling, J. Portisch, H. Paulheim, Matching with transformers in melt, in: OM@ ISWC, 2021, pp. 13–24.

[10] C. Meilicke, Alignment incoherence in ontology matching (2011).