AgreementMakerDeep Results for OAEI2021

Zhu Wang[1] and Isabel F. Cruz[1] *

ADVIS Lab, Dept of Computer Science
University of Illinois at Chicago, Chicago IL 60607, USA
zwang260@uic.com
isabelcfcruz@gmail.com

Abstract. AgreementMakerDeep (AMD) is a new flexible and extensible ontology matching system with knowledge graph embedding techniques. AMD learns from classes and their relations between classes by constructing vector representations into the low dimensional embedding space with knowledge graph embedding methods. The results demonstrate that AMD achieves a competitive performance in a few OAEI tracks, but AMD has limitations for property and instance matching.

1 Presentation of the system

AgreementMakerDeep (AMD) is a new ontology matching system inspired by AgreementMaker [2, 3], AgreementMakerLight (AML) [7] and BootEA [19]. This year is the first time that AMD participates in OAEI. It is designed with the main goal of higher efficiency for ontology matching problems by applying knowledge graph embedding methods.

1.1 State,purpose, general statement

Ontology matching aims to establish semantic correspondences or relationships between concepts or properties of different ontologies [6]. There is a wide range of algorithms developed for ontology matching, such as those that use lexical similarity with linguistic techniques [12], partition large ontology sets based on structural proximity [10], or detect graph similarity [5, 14]. However, such strategies may be time consuming [9], may use sparse and a high-dimensional training space [17], and may vary with the domains [1].

AMD mainly utilizes string-based techniques [4] and lexical matching algorithms [15], but adopts the representative learning models [8] to capture the relations as structural information with a translation vector between two classes.

2 Specific Techniques Used

The architecture of AMD is shown in figure 1, including ontology parsing, string and lexical matching, knowledge graph embedding, model learning and candidate selection.

* This paper is dedicated to the memory of Isabel F. Cruz.

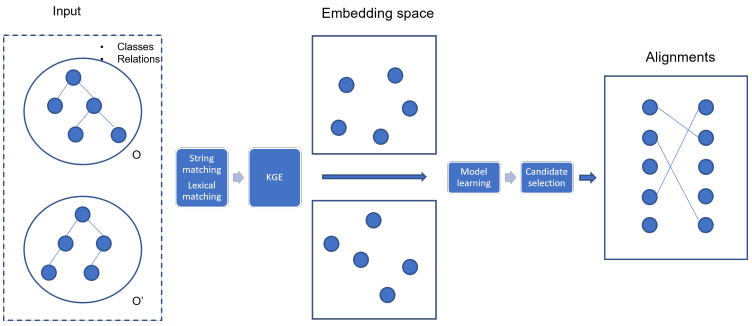

Fig. 1. The framework of AMD.

Ontology parsing owlready2 [11] is used to extract meta information of classes from the source and target ontology, such as super/sub-classes, labels, annotations, partof and disjointwith. BeautifulSoup [16] is used to extract synonyms.

String and lexical matching We apply several text per-processing techniques like stop-words removal and tokenization on class labels and annotations. AMD uses the Base Similarity Matcher (BSM) [5] and lexical matching algorithms to obtain a baseline class alignment.

Knowledge graph embedding and model learning We characterize the structure information of ontologies by relations translated from one class to another class using a modified TransR [13] model into relational embedding spaces.

Problem Formulation Given two ontologies O and O', we construct knowledge graph X and Y, and define the correspondence between two concepts as following triplets $T_{c,c'} = < c, r, c' >$, where r is the relation between c and c'. The problem is to find mapping set M = $\{(c_x, c_y) \epsilon X \times Y | c_x \equiv c_y\}$. In this study, we focus on one-to-one alignment and the relation between concepts is equality.

Let $\vec{v}(c_x)$= $\{v_1, v_2, ...v_m\}$ and $\vec{v}(c_y)$= $\{v'_1, v'_2, ...v'_n\}$ be two d-dimensional vectors sets of size m and n, we compute their distance with simple cosine similarity by d($\vec{v}(c_x), \vec{v}(c_y)$) = 1-sim($\vec{v}(c_x), \vec{v}(c_y)$) as follows:

$$sim(\vec{v}(c_x), \vec{v}(c_y)) = \sum_{i=1} \arg\max_j cos(\vec{v}(c_x), \vec{v}(c_y)) \tag{1}$$

We define the probability of the aligned labels between concepts c_x and c_y by $p(c_y|c_x)$ as follows:

$$p(c_y|c_x) = \sigma sim(\vec{v}(c_x), \vec{v}(c_y)) \tag{2}$$

where σ is the sigmoid function.

Knowledge graph embedding In AMD, we apply a modified TransR method which translates concepts and relations into concept space and relation-specify

concept spaces, since there are multiple relations in the ontologies e.g subclassof and disjointwith. In the original TransR, the projected vectors are defined as $c_r = cM_r, c'_r = c'M_r$, and the score function as $f_r(c, c') = \|c_r + r - c'_r\|_2^2$ [13]. Inspired by Sun et al. [19], the absolute scores of positive triples are lower than the negative ones, so we modify the loss function by using two γ hyper-parameters as follows:

$$L = \sum_{(c_x,r,c'_x)\in S_1} \sum_{(c_y,r,c'_y)\in S_2} max(0, (f_r(c_x, c'_x) - \gamma_1) - \mu(f(c_y, c'_y) + \gamma_2) \quad (3)$$

where $\gamma_1, \gamma_2, \mu > 0$ and $\gamma_2 > \gamma_1$, S is the positive triples set and S' is the negative triples set. We set different γ values to ensure absolutely low margin loss scores in the positive triples for reducing the drift of the embedding and also keep the function of the margin-based ranking loss.

During the process that computes vectors, we need to generate negative triples. Following the work of Sun et al. [19] and Li et al. [12], we refine the uniform negative sampling by choosing from the k-nearest neighbors in the embedding space, and setting constraints of select candidates excluding from the subclassOf or disjointWith related concepts. In this way, we can avoid vector sparsity and obtain better quality of vector representations for the concepts.

Candidate selection We select candidates based on a threshold of the classes knowledge graph embedding vectors similarity, and then compare the similarity with baseline if the pairs are in baseline result sets.

2.1 Parameter settings

In AMD, we use stochastic gradient descent as the optimizer and configure hyper-parameters as listed: dimensions are set to 200 for the vectors. The learning rate is among {0.01,0.02,0.001}, and mini-batch is {5,10}. γ_1 = {0.01,0.05,0.1}, γ_2 = {0.5,1.0}. The number of nearest neighbors for negative sampling is {5,10,20}.

From the local evaluation results on the Anatomy track, the best parameter set is as follows: the learning rate is 0.01, mini-batch is 10, γ_1 is 0.01, γ_2 is 0.5 and 10 nearest neighbors for the negative sampling.

2.2 Adaptations made for the evaluation

Our framework uses Python with Tenserflow[1] and RDFLib [2], and is packed for SEALS using MELT. We use the best parameter set in local alignments for the OAEI submission, see section 2.1.

[1] https://www.tensorflow.org/
[2] https://github.com/RDFLib

3 Results

3.1 Anatomy

The Anatomy track results of AMD are shown in Table 1. AMD returns 1167 correspondences in 3 seconds. The result shows that AMD can be competitive among the top promising matching systems, especially in terms of runtime and precision. AMD is the second fastest system in this track and a slightly higher(0.004) precision than AML.

Table 1. AgreementMakerDeep results in the Anatomy track.

	Runtime	Precision	Recall	F-measure
AMD	3	0.96	0.739	0.835

3.2 Conference

The Conference track results of AMD are shown in Table 2. As expected, the performance of AMD in the conference track is not good, with the F-measure only slightly higher when comparing baseline method(StringEquiv). AMD shows a lack of ability to extract and match the properties in M2 and M3 evaluation variants. However, AMD has higher values in term of Precision in most tasks.

Table 2. AgreementMakerDeep results in the Conference track.

	Precision	Recall	F-measure
ra1-M1	0.87	0.51	0.64
ra1-M3	0.87	0.43	0.58
ra2-M1	0.82	0.48	0.59
ra2-M3	0.82	0.39	0.53
rar2-M1	0.81	0.48	0.6
rar2-M3	0.81	0.41	0.54

3.3 Largebio

Table 3 shows results of AMD in the Large BioMed track. Our workstation was able to complete one task and finished the other larger tasks with an out of memory exception. In FMA-NCI small fragment task, AMD achieves a promising performance with a F-measure of 0.906.

3.4 Knowledge Graph

AMD is able to complete two of the five tasks with a runtime of 37 minutes, and AMD only returns class correspondences with a precision of 1.0.

Table 3. AgreementMakerDeep results in the Largebio track.

	Precision	Recall	F-measure
FMA-NCI small	0.973	0.848	0.906

4 General comments

4.1 Comments on the result

2021 is the first time that AMD participates in OAEI, and performs promising results. Overall, the results show that AMD is able to complete several tasks in different domains on class-level matching in a timely manner. It is a fast system in most of tracks. Hence, we have shown that knowledge graph embedding is helpful to decrease computation time and that it leads to a competitive performance in term of F-measure in Anatomy and LargeBio tracks. AMD has consistently had higher precision than AML in a few tracks. However, AMD is still under development that it is only able to return class correspondences. Moreover, AMD has memory issues for large scale datasets and is not able to match properties and instances in the current stage.

4.2 Improvements

The current development of AMD touches on several aspects. Besides considering properties and instances matching, we will utilize joint embedding to combine contextualized knowledge graph embeddings like coKE and BERT and additional knowledge resources such as WebIsA [18] as a lexicon database. Moreover, we will adapt AMD with different data types parsing and parameters selections for different tracks.

5 Conclusions

In this paper, we have introduced a new ontology matching system called AMD. We adapted a modified transR model to fit the ontology matching problem: thus, we learn low-dimensional embeddings for each class and relation to capture the hidden semantics of ontologies, rather than measuring the similarities between classes directly, as in other traditional systems. AMD makes full use of the textual and structure knowledge of ontologies. The results demonstrate the high efficiency and the promising performance of our proposed matching method as compared to other systems results in several tracks.

Bibliography

[1] Michelle Cheatham and Pascal Hitzler. String similarity metrics for ontology alignment. In *International semantic web conference*, pages 294–309. Springer, 2013.

[2] Isabel F. Cruz, Flavio Palandri Antonelli, and Cosmin Stroe. AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. *PVLDB*, 2(2):1586–1589, 2009.

[3] Isabel F. Cruz, Flavio Palandri Antonelli, and Cosmin Stroe. Efficient Selection of Mappings and Automatic Quality-driven Combination of Matching Methods. In *International Workshop on Ontology Matching*, volume 551 of *CEUR Workshop Proceedings*, pages 49–60, 2009.

[4] Isabel F. Cruz, Flavio Palandri Antonelli, Cosmin Stroe, Ulas Keles, and Angela Maduko. Using AgreementMaker to Align Ontologies for OAEI 2009: Overview, Results, and Outlook. In *International Workshop on Ontology Matching*, volume 551 of *CEUR Workshop Proceedings*, pages 135–146, 2009.

[5] Isabel F. Cruz and William Sunna. Structural Alignment Methods with Applications to Geospatial Ontologies. *Transactions in GIS, Special Issue on Semantic Similarity Measurement and Geospatial Applications*, 12(6):683–711, 2008.

[6] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer-Verlag, Heidelberg (DE), 2007.

[7] Daniel Faria, Catia Pesquita, Emanuel Santos, Matteo Palmonari, Isabel F. Cruz, and Francisco M. Couto. The AgreementMakerLight Ontology Matching System. In *International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE)*, pages 527–541. Springer, 2013.

[8] Junheng Hao, Muhao Chen, Wenchao Yu, Yizhou Sun, and Wei Wang. Universal Representation Learning of Knowledge Bases by Jointly Embedding Instances and Ontological Concepts. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1709–1719, 2019.

[9] Prodromos Kolyvakis, Alexandros Kalousis, and Dimitris Kiritsis. Deepalignment: Unsupervised ontology matching with refined word vectors. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 787–798, 2018.

[10] Amir Laadhar, Faiza Ghozzi, Imen Megdiche, Franck Ravat, Olivier Teste, and Faiez Gargouri. Partitioning and Local Matching Learning of Large Biomedical Ontologies. In *ACM SIGAPP Symposium on Applied Computing*, pages 2285–2292, 2019.

[11] Jean-Baptiste Lamy. Owlready: Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies. *Artificial intelligence in medicine*, 80:11–28, 2017.

[12] Weizhuo Li, Xuxiang Duan, Meng Wang, XiaoPing Zhang, and Guilin Qi. Multi-view embedding for biomedical ontology matching. *OM@ ISWC*, 2536:13–24, 2019.

[13] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI Conference on Artificial Intelligence*, 2015.

[14] Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching. In *IEEE International Conference on Data Engineering (ICDE)*, pages 117–128, 2002.

[15] Catia Pesquita, Daniel Faria, Cosmin Stroe, Emanuel Santos, Isabel F Cruz, and Francisco M Couto. What's in a 'nym'? synonyms in biomedical ontology matching. In *International Semantic Web Conference*, pages 526–541. Springer, 2013.

[16] Leonard Richardson. Beautiful soup documentation. *April*, 2007.

[17] Petar Ristoski, Jessica Rosati, Tommaso Di Noia, Renato De Leone, and Heiko Paulheim. RDF2Vec: RDF Graph Embeddings Their Applications. *Semantic Web*, 10(4):721–752, 2019.

[18] Julian Seitner, Christian Bizer, Kai Eckert, Stefano Faralli, Robert Meusel, Heiko Paulheim, and Simone Paolo Ponzetto. A large database of hypernymy relations extracted from the web. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 360–367, 2016.

[19] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. Bootstrapping Entity Alignment with Knowledge Graph Embedding. In *IJCAI*, volume 18, pages 4396–4402, 2018.