WomboCombo Results for OAEI 2022

Peter Kardos[1], Zsolt Szántó[1] and Richárd Farkas[1]

[1]*University of Szeged*

Abstract

This paper presents the results of the WomboCombo Matcher in the Ontology Alignment Evaluation Initiative (OAEI) 2022. WomboCombo is an ontology matching tool that finds node pairs starting out from simpler exact string matching based steps through more complex neural Language Model based steps. We also train a classifier to differentiate between entities with the same and entities with similar meaning.

Keywords

Ontology Mathching, Ontology Alignment, Language Models,

1. Presentation of the system

1.1. State, purpose, general statement

Word **m**eaning **b**ased matcher **o**ver **Comb**inati**o**ns (WomboCombo) is a multi-stage ontology matching system that uses only textual information to find the same entities in two knowledge graphs. The first step is a simple exact string similarity based pairing process followed by more complex and resource exhausting steps as we progress through the whole system. Later stages utilize pretrained Language Models to find entities with the same meaning but different lexical representations. Each stage has its own output which we then combine for a final alignment (therefore the name *Combo*). WomboCombo was built for and got tested only on the Knowledge Graph track and mainly focused on the instance pairs. This decision is supported by the fact of instance nodes carrying the core information of a graph. Also class and property counts are only a handful, making it easier to correctly pair and most knowledge graphs even miss out on representing classes or properties. WomboCombo is implemented in python and is compatible with the *Matching EvaLuation Toolkit (MELT)* [1] with SEALS packaging.

1.2. Specific techniques used

In this section we explain how each of the WomboCombo's steps work and how they connect to each other. Figure 1 depicts the system architecture. You can see this pipeline visually on figure 1.

OAEI'22: The 17th International Workshop on Ontology Matching, Oct 23–24, 2022, Hangzhou, China

kardos@inf.u-szeged.hu (P. Kardos)

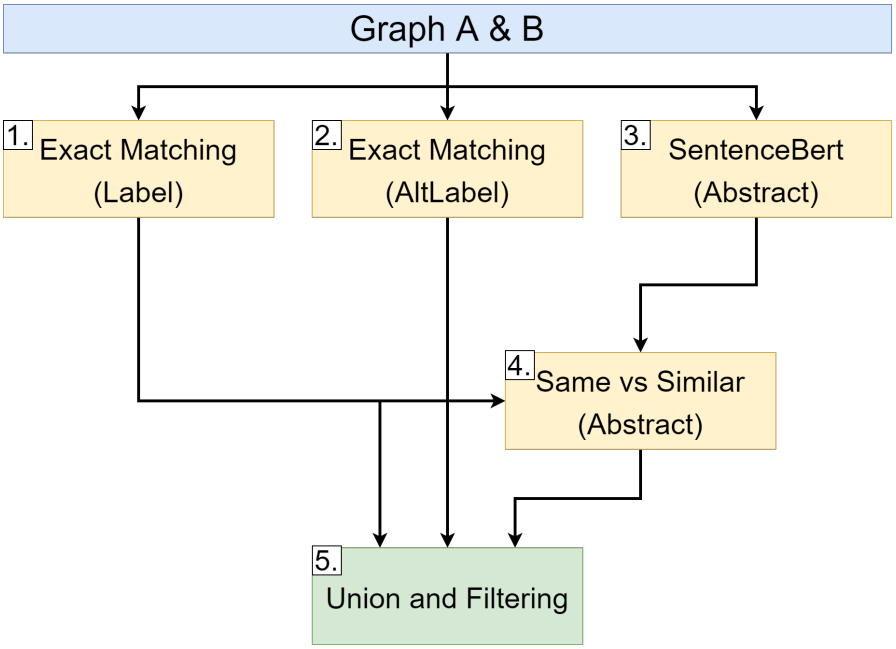

Figure 1: WomboCombo pipeline.

1.2.1. Exact matching module

The exact matching module takes a property as input and given the two graphs it will search for nodes that has the same string representation of this property.

We've also experimented with fuzzy string matching algorithms with different parameters, however these solutions brought more noise with themselves than actual pairs. Therefore we've discarded using fuzzy string matching.

WomboCombo's first two steps use the exact matching module but with a different property. The first step uses the nodes' Label property, the second step uses the nodes' AltLabel property both of them resulting in a high precision matches. Even though the precision is high using the following module's we focus on increasing the recall of our final alignment.

1.2.2. SentenceBert module

This module loads a pretrained SentenceBert model and outputs a vector embedding to all of the nodes in the two graphs using their textual description (abstract). For each of the nodes of graph A the X most similar nodes from the other graph will be paired using cosine similarity. The main purpose of the module is to get a pool of pairs where the nodes have similar meaning.

This is the 3rd step in our pipeline where we load *all-MiniLM-L6-v2* [2] model and use the abstract properties text value trimmed to the first two sentence to get a vector representation. Maximum the top 6 pairs were gathered for each node and discarded all pairs below 0.6 cosine similarity treshold. In case a graph have no abstract property this module is inactive.

1.2.3. Same vs Similar module

The Same vs Similar module is the most resource exhausting one. Our goal with this module is to train a classifier that can differentiate between two nodes that are similar versus two nodes that represent the same concept. We achieve this by automatically creating a training dataset with 2 classes {*same, similar*} and training a Language Model based classifier. Based on a candidate pair pool the trained model discards the predicted similar pairs and returns the remaining.

In our submission the selected Language Model was *albert-base* [3] pretrained on the MRPC task which is a sentence similarity task. The pairs we've considered same meaning (positive) were the exact matching pairs on the Label property (1st steps output). We've generated 1 similar pair (negative) to each of these using the SentenceBert module where we replaced one of the nodes of the pair with the highest ranking node that wasn't the positive. We can say that these negative pairs might contain noise, but as the task only has 1:1 gold pairs the noise should be minimal. We used the abstract property to get the textual information for each node. As for the training process a batch size of 1 and a learning rate of 10^{-5} were used. The training process was let to run for 100 epochs, but an EarlyStop with 5 patience could shut it down before that which suggest that we split the dataset to train and evaluation sets. To get the output alignment of this module we used the trained classifier to filter the Sentence Bert module's output.

1.2.4. Union and Filtering module

This module can union different alignments considering the confidence of each candidate. As the gold pairs are all 1:1 matches we run a Top 1 filtering based on the confidence score of each pair. In our pipeline the final alignment is calculated from 3 alignment pools maintaining the order: Exact Matching over Labels, Exact Matching over AltLabels and the Same vs Similar module's output. If a pair is selected into the final alignment we pay attention to not include any additional connections to these nodes while merging the different pair sets.

2. Results

2.1. Knowledge Graph

WomboCombo was not evaluated due to the organizers reporting TypeError when running the system, but we report the scores achieved on the Knowledge Graph V4 test cases calculated on our local machine using the official library and evaluation code. Our score achieved on the marvel dataset is much lower than the other datasets due to missing abstract fields for most of the nodes.

3. General comments

We have tested our system with different parameters for example whether cutting the abstracts, using SentenceBert to vectorize over labels/altlabels or even tried different pretrained language models. We could not find one parameter set that works best on all the 5 datasets, mostly

	Prec.	Rec.	F-m.
marvelcinematicuniverse-marvel	0.877	0.663	0.755
memoryalpha-memorybeta	0.932	0.899	0.915
memoryalpha-stexpanded	0.968	0.940	0.954
starwars-swg	0.872	0.775	0.821
starwars-swtor	0.930	0.906	0.918

Table 1
The results of Knowledge Graph track 2022.

different parameters worked better on certain datasets. For submission we have selected the parameters with the best mean scores over the datasets.

WomboCombo is not the best choice when matching properties or classes as these nodes do not have abstract fields most of the time. Therefore only the exact match pairs could be found in the resulting alignment. This was a big issue on the Marvel datasets as even the instances had no abstract property in 90% of the nodes.

4. Conclusions

In this paper, we presented the WomboCombo matching system and its results in the OAEI 2022 campaign. The system participated only in the Knowledge Graph track. Our solution only considers the textual information of a node and creates pairs using a multi-step process that includes exact string matching and more complex neural Language Model based steps as well. The results show that these complex steps can successfully find not so trivial pairs boosting the most basic matchers.

References

[1] S. Hertling, J. Portisch, H. Paulheim, Melt - matching evaluation toolkit, in: M. Acosta, P. Cudré-Mauroux, M. Maleshkova, T. Pellegrini, H. Sack, Y. Sure-Vetter (Eds.), Semantic Systems. The Power of AI and Knowledge Graphs, Springer International Publishing, Cham, 2019, pp. 231–245.

[2] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, CoRR abs/1908.10084 (2019). URL: http://arxiv.org/abs/1908.10084. arXiv:1908.10084.

[3] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: A lite BERT for self-supervised learning of language representations, CoRR abs/1909.11942 (2019). URL: http://arxiv.org/abs/1909.11942. arXiv:1909.11942.