Combining Word and Sentence Embeddings with Alignment Extension for Property Matching

Guilherme Sousa[1], Rinaldo Lima[2] and Cassia Trojahn[1]

[1]IRIT: Institut de Recherche en Informatique de Toulouse, France
[2]Universidade Rural de Pernambuco, Recife, Brazil

Abstract

Matching properties still have less performance when compared to classes. Properties frequently involve a higher variation in naming (verb variation, functional words, common synonyms) than classes. Another challenge concerns the variation in property representations. This paper extends a lexical-based property matching approach combining the similarity from word and sentence pre-trained embeddings and alignment extension. The proposed approach performs competitively with state-of-the-art alignment systems on popular benchmarks in the field.

1. Introduction

Property matching is the problem of finding similar properties between two different ontologies or knowledge graphs. While most matching approaches are still dedicated to matching classes, matching properties pose additional challenges as properties frequently involve a higher variation in naming (verb variation, functional words, common synonyms) than classes [1]. Another of the challenges faced by property matching systems adapting to diverse domains is the variation in property representations, which may require dealing with constructors. For instance, the data property `name` can have multiple domains as `#Conference` and `#Person`, being represented as a collection. Approaches for property matching still mostly consider lexical-based methods that compare the similarity of property labels. However, such approaches are not enough to capture their meaning. Recently, representation learning techniques have captured the attention in the field and several systems have explored such models [2, 3], in particular sentence embeddings that are embeddings generated from sentences (and not only individual words), such as TOM [4], Fine-Tom [5] and DAEOM [6]. Sentence embedding models are able to generate different embeddings of the same word if it appears in different contexts [7]. Contrary to sentence models, word embeddings generate fixed embeddings of each word regardless of the context they are used in the sentence. Examples of systems using this type of embedding are OntoEmma [8], or still those from [9].

This paper addresses the problem of property matching with the use of pre-trained word and sentence embedding and alignment extension. The main contributions of the paper are

OM 2023: The 18th International Workshop on Ontology Matching collocated with the 22nd International Semantic Web Conference ISWC-2023 November 7th, 2023, Athens, Greece
guilherme.santos-sousa@irit.fr (G. Sousa); rinaldo.jose@ufrpe.br (R. Lima); cassia.trojahn@irit.fr (C. Trojahn)
0000-0002-2896-2362 (G. Sousa); 0000-0002-1388-4824 (R. Lima); 0000-0003-2840-005X (C. Trojahn)

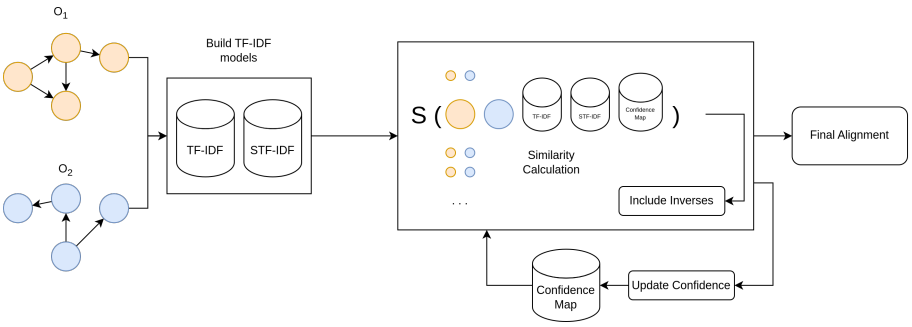

Figure 1: Architecture of proposed property matching system.

as follows: (i) an extension of a purely lexical-based approach [10] that improves its low recall through the use of word embeddings for comparing property domains and sentence embeddings for property labels. (ii) addressing the problem of complex constructors; (iii) adopting a way of combining embeddings and alignment extension strategy [11] which has proven to be useful to capture frequent alignment patterns when similar properties are detected. The proposed system is evaluated on datasets used in the OAEI campaign. The results show that using the proposed techniques can improve property matching when compared to the best systems in the well-known Conference track. The system is available under license MIT on https://github.com/anonym-om/propmatcher.

The rest of this paper is organized as follows. §2 details the proposed architecture and the specific techniques employed. §3 discusses the experiments. Finally, §4 concludes the paper.

2. Proposed approach

The proposed system builds upon the PropString system [10], incorporating embedding similarity and alignment extension techniques. The main architecture is shown in Figure 1. PropString assumes that similar properties share similarities in domains, ranges, and labels, relying on lexical methods. The similarity between domains and ranges is measured using TF-IDF [12], which is effective for class alignment [13]. In order to align property labels, a soft version of TF-IDF [14] with JaroWinkler similarity is applied to the **core concept** of the label, representing the verb or noun with its adjectives if verbs are absent.

Handling of complex constructors The first step of the matching process consists of converting the complex entities into simple entities that have all labels of the complex entity concatenated. With this strategy, complex constructors such as `owl:UnionOf` can be handled. One example of this processing occurs in the property `hasTitle` which has a complex domain with two entities `Conference` and `Paper`. After the processing, the labels of the Conference and Paper are concatenated to generate a single label `Conference_Paper`.

TF-IDF models construction Since the TF-IDF and Soft TF-IDF metrics are global, as they take into account the frequency of words across all entity labels, the construction of these

models takes place prior to the matching process. To construct the TF-IDF and its soft version, a virtual document is created for each entity in the two ontologies in order to generate the final vectors. The virtual document generated for ontology classes is composed of the class name, while, for properties, it includes the property name, domain, and range labels. The labels are split using a tokenizer and converted to lowercase, e.g., `writePaper` with domain `Author` and range `Paper`. These elements will produce the document "author write paper paper". The entire set of documents is then used to construct the frequency models, vocabulary, and IDF. The TF-IDF model was generated using the Scikit-learn library [1]. After building the vocabulary and calculating IDF values, the system determines the final similarity using cosine similarity. For property labels, it employs the Soft TF-IDF approach, which includes similar words exceeding a 0.8 threshold based on the Jaro-Winkler metric. Once TF-IDF models are constructed, the system computes similarity scores for property pairs. The final score is the minimum among three confidence values based on domain, range, and label similarities, ensuring the properties are only similar when all three are above the threshold. If the metric yields zero, an embedding similarity is used as described below.

Embeddings similarity Word embeddings are used to compute the similarity between domains, and sentence ones are used to compute the similarity between property labels. The word embedding similarity is used in domain similarity computation when the compared domains consist of a single word. This pre-trained word embeddings are obtained from the Finnish Internet Parsebank [15]. The initial step in calculating property label similarity consists of removing the last word from the property label if it matches the first word in the range label. This preprocessing is applied because of the common naming pattern in properties that usually include the range in the property label. After that processing, the property label is tagged using a POS tagger. The system then uses the core concept to determine similarity by employing the Soft TF-IDF vectors. To identify the core concept of the property label, the system selects the first verb that consists of more than four characters. If no verb is found, the noun and its adjectives are chosen as the core concept. Similar to the fallback strategy employed in domain similarity, the system applies the same approach to property labels. In this case, a sentence similarity model is employed to generate embedding representations of property labels. The fallback occurs when domain and range similarity is above 0.9 and label similarity is below 0.1. The sentence embedding similarity model [16] used in this work is from the HuggingFace [2] repository. The sentence used for similarity calculation consists of the property label concatenated with the range labels.

Alignment Extension One common practice adopted in the field consists of using the resulting alignments to find new correspondences. This practice is commonly guided by the *locality principle* [11] that states that new correspondences are more likely to be discovered among previously aligned entities. Here, two strategies of alignment extension are adopted. The first strategy involves the adoption of a *Confidence Map* which serves as a key-value store. In this map, the keys are a pair of classes, and the corresponding value represents the similarity

[1] Equations for the model generation can be found at the TfidfTransformer at https://scikit-learn.org/
[2] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2 consulted at 07/09/2023.

Name	Precision	Recall	F-measure
PropMatch	0.83	**0.52**	**0.64**
AML	**1.0**	0.41	0.58
base (PropString)	**1.0**	0.28	0.44
LogMap	0.62	0.28	0.39
GMap	0.56	0.2	0.29
Wikitionary	0.24	0.28	0.26
TOM	0.27	0.24	0.25
ALOD2Vec	0.22	0.3	0.25
LogMapLt	0.24	0.22	0.23
FineTOM	0.24	0.22	0.23
OTMapOnto	0.13	0.48	0.2
edna	0.21	0.11	0.14
StringEquiv	0.07	0.02	0.03

Table 1
Result of the property matching systems that participate in the OAEI competition 2021.

between them. Every time a property alignment is added to the resulting set of alignments, the domains in each property form a pair that is added to the Confidence Map. Then, for each property evaluated by the similarity metric, the system checks if the associated domain pair is present in the Confidence Map. If found, the domain confidence is increased by 0.66. Due to the iterative nature of the system evaluation, multiple iterations are required for the influence of the Confidence Map to be reflected in all the compared property pairs. The number of iterations is a system hyper-parameter that can vary between ontologies. The second strategy involves including the inverse of aligned properties in the final alignment set if they are present. These inverse properties are more likely to have alignments, and the Confidence Map is subsequently updated accordingly. Finally, to ensure a 1-1 set, if there are multiple correspondences involving a particular property, only the pair with the highest similarity is retained, while the others are discarded. This step helps maintain a more concise and accurate alignment set.

3. Experiments

Evaluation on the OAEI Conference dataset The first experiments have been conducted on the OAEI Conference [3] dataset. This dataset comprises 21 alignment pairs between 7 different ontologies. 7 reference alignments do not contain any property. The properties in this dataset are represented by nodes containing `rdfs:domain` and `rdfs:range` information. The evaluation focuses solely on the pairs that contain property reference alignments. The baseline systems are those that participated in the campaign, which is compared with the original system (PropString) and the proposed system (PropMatch). The similarity threshold used by the proposed system in all evaluations is 0.65. The system achieved the best F-measure (Table 1) with this value when tested with a threshold ranging from 0 to 1 with increments of 0.05.

[3] http://oaei.ontologymatching.org/2021/conference/index.html

	mcu-marvel			malpha-mbeta			malpha-stexpand			starwars-swg			starwars-swtor		
Pair	Prec	Rec	F-M	Prec	Rec	F-M	Prec	Rec	F-M	Prec	Rec	F-M	Prec	Rec	F-M
AMD	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ATMatcher	0.91	0.91	**0.91**	0.98	**0.92**	**0.95**	0.95	0.95	**0.95**	**1.00**	**1.00**	**1.00**	**1.00**	**0.98**	**0.99**
BaselineLabel	**1.00**	0.36	0.53	**1.00**	0.34	0.51	**0.97**	0.68	0.80	**1.00**	**1.00**	**1.00**	**1.00**	**0.98**	**0.99**
KGMatcher	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LogMap	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LSMatch	0.82	0.82	0.82	0.62	0.58	0.60	0.62	0.61	0.62	0.72	0.65	0.68	0.88	0.79	0.83
Matcha	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
PropMatch	0.66	**1.00**	0.79	0.51	**0.92**	0.66	0.53	**0.97**	0.69	0.44	**1.00**	0.61	0.49	0.96	0.65

Table 2
Results of the systems that participated in property-specific tasks in the Knowledge Graph tracks. The best results in each column are in bold.

Evaluation on other datasets PropMatch has also been evaluated on the OAEI Knowledge Graph track[4]. In this track, 8 knowledge graphs are aligned in 5 pairs. Different from Conference, the knowledge graphs in this track represent properties as a predicate that connects instances to other values. In this structure, multiple instances can share the same property. For example, authors, films, and enterprises can have the same property name with different ranges. This can be viewed as the property having a complex domain composed of multiple entities. In order to address this specificity, the system selects the most frequent domain-range pair composed of types of instances present in the domain and range that the property connects to act as the property's single domain and range. PropMatch, when not utilizing domain and range similarities, had better performance compared to the other systems. It was run with a threshold value of 0.969 and a single iteration, as the Confidence Map is not updated during the process (Table 2). Out of the 8 systems evaluated for property alignment in this track, only 4 systems were able to generate alignments. Among these systems, PropMatch demonstrates superior overall results compared to AMD and LogMap when considering alignment across tracks. It is evident that PropMatch surpasses the baseline performance in the pairs *mcu-marvel* and *malpha-mbeta*, while also achieving the highest recall in the pairs *mcu-marvel*, *malpha-mbeta*, *malpha-stexpand*, and *starwars-swg*. Furthermore, PropMatch outperforms LSMatch in terms of f-measure in the pairs *malpha-mbeta* and *malpha-stexpand*. However, the system still has low precision in the pairs *starwars-swg* and *starwars-swtor*.

4. Conclusion

This paper has proposed to combine lexical-based similarity metrics to embeddings and alignment extension. It achieved not only the best results in the Conference track but also competitive results in the Knowledge Graph track. The system can, however, be improved in several directions: incorporating other models that can encode more information (graph embeddings) that can not only effectively encode features but also compare semantic similarity; a confidence map built for the property alignment could contribute to improving the results in the related task of class alignment; and discovery of complex correspondences involving properties and

[4]http://oaei.ontologymatching.org/2022/knowledgegraph/index.html

transformation functions has to be addressed.

References

[1] M. Cheatham, C. Pesquita, D. Oliveira, H. B. McCurdy, The properties of property alignment on the semantic web, Int. J. Metadata Semant. Ontologies 13 (2018) 42–56.

[2] P. Kolyvakis, A. Kalousis, B. Smith, D. Kiritsis, Biomedical ontology alignment: an approach based on representation learning, J. Biomed. Semant. 9 (2018) 21:1–21:20.

[3] Y. He, J. Chen, D. Antonyrajah, I. Horrocks, BERTMap: A BERT-Based Ontology Alignment System, in: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, February 22 - March 1, 2022, 2022, pp. 5684–5691.

[4] D. Kossack, N. Borg, L. Knorr, J. Portisch, TOM matcher results for OAEI 2021, in: Proceedings of the 16th International Workshop on Ontology Matching, 2021, pp. 193–198.

[5] L. Knorr, J. Portisch, Fine-TOM matcher results for OAEI 2021, in: Proceedings of the 16th International Workshop on Ontology Matching, October 25, 2021, 2021, pp. 144–151.

[6] J. Wu, J. Lv, H. Guo, S. Ma, Daeom: A deep attentional embedding approach for biomedical ontology matching, Applied Sciences 10 (2020) 7909.

[7] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics, 2019, pp. 4171–4186.

[8] L. L. Wang, C. Bhagavatula, M. Neumann, K. Lo, C. Wilhelm, W. Ammar, Ontology alignment in the biomedical domain using entity definitions and context, in: Proceedings of the BioNLP 2018 workshop, Melbourne, Australia, July 19, 2018, 2018, pp. 47–55.

[9] J. Wu, J. Lv, H. Guo, S. Ma, Ontology matching by jointly encoding terminological description and network structure, in: CCIOT 2020: 5th International Conference on Cloud Computing and Internet of Things, Okinawa, Japan, September, 2020, pp. 77–85.

[10] M. Cheatham, P. Hitzler, The properties of property alignment, in: Proceedings of the 9th International Workshop on Ontology Matching, October 20, 2014, 2014, pp. 13–24.

[11] E. Jiménez-Ruiz, B. C. Grau, LogMap: Logic-Based and Scalable Ontology Matching, in: Proceedings of the 10th International Semantic Web Conference, 2011, pp. 273–288.

[12] A. N. Aizawa, An information-theoretic perspective of TF-IDF measures, Inf. Process. Manag. 39 (2003) 45–65.

[13] M. Cheatham, P. Hitzler, String similarity metrics for ontology alignment, in: Proceedings of the 12th International Semantic Web Conference, 2013, pp. 294–309.

[14] W. Cohen, P. Ravikumar, S. Fienberg, A comparison of string metrics for matching names and records, in: KDD workshop on data cleaning and object consolidation, 2003, pp. 73–78.

[15] J. Luotolahti, J. Kanerva, V. Laippala, S. Pyysalo, F. Ginter, Towards universal web parsebanks, in: Proceedings of the Third International Conference on Dependency Linguistics, DepLing 2015, August 24-26 2015, Uppsala University, Uppsala, Sweden, 2015, pp. 211–220.

[16] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, M. Zhou, Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, in: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.