

# Cross-lingual ontology matching with CIDER-LM: results for OAEI 2022

Javier Vela, Jorge Gracia

Department of Computer Science and Systems Engineering  
University of Zaragoza  
María de Luna 1, 50018 Zaragoza, Spain

## Abstract

In this paper, the CIDER-LM cross-lingual matching system is presented, as well as the results it achieved during the OAEI (Ontology Alignment Evaluation Initiative) 2022 campaign. This is the first appearance of CIDER-LM in OAEI where it only participated in the MultiFarm track. The matching system uses a pre-trained multilingual language model based on transformers, fine-tuned using the openly available portion of the MultiFarm dataset. The model calculates the vector embeddings of the labels associated to every ontology entity and its context. The confidence degree between matching entities is computed as the cosine similarity between their associated embeddings. CIDER-LM is novel in the use of multilingual language models for cross-lingual ontology matching. Its initial version obtained promising results in the OAEI'22 MultiFarm track, attaining a modest precision but the best overall performance in recall.

## Keywords

Cross-lingual Ontology Matching, Ontology Alignment, Natural Language Processing, Language Models, Transformers, CIDER-LM, Sentence-BERT

## 1. Presentation of the system

CIDER-LM is largely inspired by an earlier system called CIDER-CL [1], the cross-lingual version of CIDER (Context and Inference basED ontology alignER), although CIDER-LM completely redesigns and reimplements it. CIDER-LM stands for Language Model-based CIDER. Differently to its predecessor, which based its cross-lingual capabilities on *cross-lingual explicit semantic analysis* [2], CIDER-LM uses *language models* like BERT [3] based on the Transformer architecture [4]. Transformers are encoder-decoder neural networks with a self attention mechanism. They are very popular for solving *Natural Language Processing* (NLP) tasks. BERT is a language model that leverages the transformer encoder to predict tokens in a sentence when given a context. Numerous studies have been published exploring how BERT can be fine-tuned to solve a wide variety of NLP tasks; and recently, tasks from other fields, such as computer vision. The original BERT was trained using an English corpus.

Using BERT and similar models have proven to be useful in ontology matching, for instance in the biomedical domain [5], but they are largely unexplored for cross-lingual matching until now.

---

Ontology Alignment Evaluation Initiative (OAEI) 2022 campaign, October 23rd or 24th, 2022, Hangzhou, China

✉ 775593@unizar.es (J. Vela); jgracia@unizar.es (J. Gracia)

🌐 <https://github.com/javiervela/> (J. Vela); <http://jgracia.url.ph/web/> (J. Gracia)

🆔 0000-0002-6818-9191 (J. Vela); 0000-0001-6452-7627 (J. Gracia)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

Multilingual language models can be used to that end, which can represent tokens in several languages in the same embedding space. In fact, other BERT variants have been trained with multilingual corpora. One of such a variants, Sentence-BERT (SBERT) [6], produce semantically meaningful embeddings for sentences in different languages, which are aligned in the same embedding space (in contrast to other models that create the embeddings at the token level). These aligned embeddings can be compared to find the similarity between sentences in two different languages. In addition, SBERT monolingual models can be made multilingual using *knowledge distillation*.

CIDER-LM uses SBERT to associate embeddings to the labels of the ontology entities (and their context), which are compared using cosine similarity to create a matching confidence score between them. The choice of sentence embeddings instead of token embeddings (as other multilingual language models do) is motivated by the fact that many ontology labels are conformed by multi-word expressions, thus embeddings that represent whole sentences (and not only atomic tokens) are more suitable to capture their semantic content and to compute similarities between them [7, 6].

### 1.1. State, purpose, general statement

CIDER-LM is a cross-lingual ontology matching system that utilizes a pre-trained multilingual language model based on transformers, fine-tuned using the openly available portion of the MultiFarm dataset<sup>1</sup>. The model calculates the vector embeddings of the labels associated to every ontology entity and its context. The confidence degree between two matching entities is computed as the cosine similarity between their associated embeddings. The generated alignments are 1 to 1 mappings between entities from two input ontologies. The input ontologies need to be in OWL or RDF-S and the provided output is expressed in the Alignment Format<sup>2</sup>. The type of discovered correspondence is “equivalence”, with a confidence degree in  $[0, 1]$ . CLIDER-LM works with ontology classes and properties, not yet with instances.

### 1.2. Specific techniques used

CIDER-LM integrates a fine-tuned version of *distiluse-base-multilingual-cased-v2*<sup>3</sup>, a multilingual pre-trained model from Sentence Transformers<sup>4</sup>. This ontology aligner is implemented in Python and wrapped using the *Matching Evaluation Toolkit* (MELT), a framework for ontology matching and evaluation [8]. The system is packaged as a Docker image implementing the *Web Interface*<sup>5</sup>.

An overall view of the CIDER-LM matching pipeline is shown in Figure 1. In short, the process is as follows. First, the aligner receives a source ( $O_1$ ) and target ( $O_2$ ) ontology. Both ontologies are read into Python objects using the library Owlready2<sup>6</sup> and fed individually into a semantic reasoner, which extends the ontologies by inferring semantic relations not initially

<sup>1</sup><https://www.irit.fr/recherches/MELODI/multifarm/>

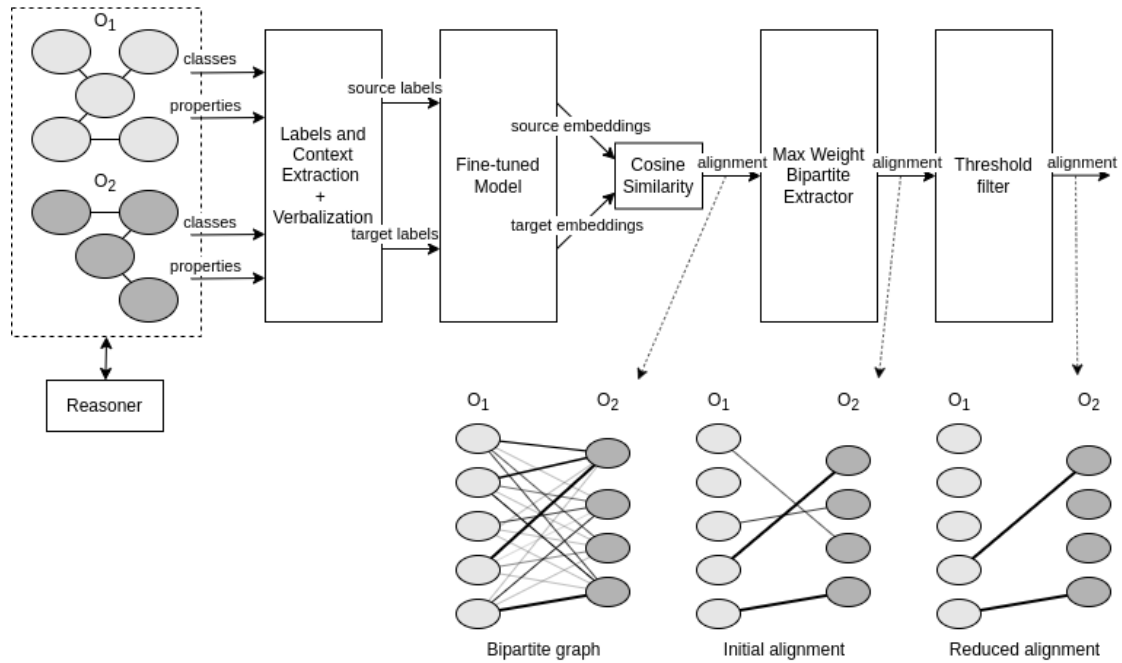
<sup>2</sup><http://alignapi.gforge.inria.fr/format.html>

<sup>3</sup><https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v2>

<sup>4</sup>[https://www.sbert.net/docs/pretrained\\_models.html#multi-lingual-models](https://www.sbert.net/docs/pretrained_models.html#multi-lingual-models)

<sup>5</sup><https://dwslab.github.io/melt/matcher-packaging/web#web-interface-http-matching-interface>

<sup>6</sup><https://owlready2.readthedocs.io/en/v0.37/>



**Figure 1:** CIDER-LM architecture: Given two ontologies  $O_1$  and  $O_2$ , they are expanded by a reasoner. The labels characterising the compared entities are extracted, verbalized, and an embedding is obtained for every one of them. The embeddings are compared using the Cosine Similarity, passed through the Max Weight Bipartite Extractor and the Threshold Filter, obtaining the final alignment.

declared. The classes and properties labels extracted from both ontologies are verbalized using their ontological context. Then, the verbalized labels are passed to the Transformer model that obtains an embedding for each of the entities in the ontologies (i.e., a vector capturing the semantics of the entity). Every embedding from  $O_1$  is compared to every embedding from  $O_2$  using the cosine similarity, forming a bipartite graph. The Maximum Weight Bipartite Extractor algorithm obtains an initial alignment that is reduced by a threshold filter, obtaining the final alignment. The following paragraphs describe each of the involved techniques with some more detail.

### 1.2.1. Ontology Reasoning

CIDER-LM performs a preliminary reasoning of both source and target ontologies using HerMiT OWL Reasoner<sup>7</sup>. This step expands the semantic relations in the ontologies, inferring new knowledge not initially asserted. However, including the reasoning in the pipeline increases considerably the execution time of CIDER-LM. This step is optional, but we kept it for OAEI-22.

<sup>7</sup><http://www.hermit-reasoner.com/>

### 1.2.2. Verbalization of entities

CIDER-LM considers ontology classes and properties. For both types of entities, their associated labels are extracted to build the corresponding sentence embeddings. In order to make the calculated embedding more representative of the semantics of the analyzed ontology entity, the labels of other entities that are part of its ontological context are considered. The verbalisation process is needed so the language model can process the set of labels (coming from the entities and its ontological context) as an individual sentence.

In its current version, CIDER-LM builds the ontological context of an entity based on its neighbouring entities, treating classes and properties differently (we use '+' as the string concatenation operator):

**Class Verbalization.** For a class  $a$  and its label  $L(a)$ , the set of parent classes and child classes in the hierarchy of the ontology are  $(P_a)$  and  $(C_a)$ , respectively. The verbalization of the class is the concatenation of the following:

- $L(a) + ', '$ ;
- for every class in  $P_a$ :  $L(a) + ' is a ' + L(P_a^i) + ', '$
- for every class in  $C_a$ :  $L(C_a^i) + ' is a ' + L(a) + ', '$

**Property Verbalization.** For a property  $a$  and its label  $L(a)$ , the sets of its domain and range classes in the ontology are  $(D_a)$  and  $(R_a)$ , respectively. The verbalization of the property is the concatenation of the following:

- $L(a) + ', '$
- for every entity in  $D_a$ :  $L(a) + ' has domain ' + L(D_a^i) + ', '$
- for every entity in  $R_a$ :  $L(a) + ' has range ' + L(R_a^i) + ', '$

The verbalized sentences from labels are concatenated with particles in English, independently of the language of the ontology. During the preliminary evaluation of the system, we found evidence that the particular language used for concatenating the labels was not very relevant, even if it differed from the language of the ontology. This will need further exploration in the future.

### 1.2.3. Fine-tuned language model

CIDER-LM relies on distiluse-base-multilingual-cased-v2, which is pre-trained on Semantic Textual Similarity (STS) and uses the SBERT architecture. The model is the knowledge distilled version of the Universal Sentence Encoder and supports more than 50 languages. A checkpoint of the distiluse-base-multilingual-cased-v2 model has been downloaded from HuggingFace<sup>8</sup> and fine-tuned for CIDER-LM, using the *SentenceTransformers*<sup>9</sup> Python framework.

In fact, CIDER-LM uses a fine tuned version of the model, specialized on the task of obtaining similarities between two entity labels in different languages. Given a set of pairs of entity labels obtained from the training set and a *true confidence* of 1 or 0, indicating if the entities are a

<sup>8</sup><https://huggingface.co/distiluse-base-multilingual-cased-v2>

<sup>9</sup><https://www.sbert.net/index.html>

match or not; the model is trained on reducing the CosineSimilarityLoss between the *predicted* and *true confidence*. We use the SenteceTransformers framework and a basic training pipeline to obtain the fine-tuned model used on the matching system.

The *cosine similarity* metric is used to find the distance between the vectors (embeddings) associated to the entities coming from the two different ontologies, in the common embeddings space. This offers a measure of how similar two sentences (the verbalised sets of entity labels) are, resulting in the confidence degree associated to the possible matching between the two entities.

#### 1.2.4. Maximum Weight Bipartite Extractor

CIDER-LM obtains the matching confidence for every pair of entities from the target and source ontology. Once the confidence degrees have been determined, the alignment can be considered as a bipartite graph with an edge (the confidence in a match) from every node (entity) from the source ontology to the target ontology. Using the implementation of the Hungarian algorithm in the *scipy* Python library, the maximum weight matching in bipartite graphs problem is solved.

#### 1.2.5. Threshold filter

After having a complete alignment, a threshold filter is used to remove all the correspondences with low confidence from the alignment. Every correspondence with a confidence lower than the threshold value is removed. The threshold value can be used to direct the results obtained by the system: a higher value will promote precision, while a lower value will favour recall. CIDER-LM applies a threshold value of 0.5 that, according to the results, promotes recall.

### 1.3. Adaptations made for the evaluation

To participate in the OAEI campaign, we have wrapped the Python implementation of CIDER-LM around the `MatcherCLI` Java class from the MELT framework. The wrapping enables the MELT evaluation and the packaging plugin, which is used to create the Docker container image for the submission to the OAEI.

CIDER-LM does a preliminary substitution of `'&xsd;date'` for `'&xsd;dateTime'` and `'http://www.w3.org/2001/XMLSchema#date'` for `'http://www.w3.org/2001/XMLSchema#dateTime'` because the dataset of ontologies used for the evaluation was not recognized initially by the Hermit reasoner as OWL Version 2, which is the only version taken by the reasoner. Performing the substitution fixes the error reading the ontology.

#### 1.4. Link to the system and parameters file

The system implementation is hosted in a GitHub repository<sup>10</sup>. The container image that executes the web server with the matching system is also in GitHub in the *Packages* section.

---

<sup>10</sup><https://github.com/sid-unizar/CIDER-LM>

**Table 1**

MultiFarm aggregated results per matcher for different ontologies

System	Time(Min)	Precision	F-measure	Recall
CIDER-LM	~ 157	0.16	0.25	<b>0.58</b>
LSMatch	~ 33	0.24	0.038	0.021
LSMatch Multilingual	~ 69	0.68	<b>0.47</b>	0.36
LogMap	~ 9	<b>0.72</b>	0.44	0.31
LogMapLt	~ 175	0.24	0.038	0.02
Matcher	~ 2	0.00082	0.000082	0.000043

### 1.5. Link to the set of provided alignments (in align format)

The resultant alignments will be provided by the organisers (see <https://oaei.ontologymatching.org/2022/results/>).

## 2. Results

In its first participation in the OAEI campaign (2022), CIDER-LM contributed to the MultiFarm track only. The reason is that this tool is primarily targeted to cross-lingual ontology matching. However, the method is able to produce monolingual mappings as well, so we do not discard participation in other tracks in future OAEI editions.

Details about the test ontologies, the evaluation process, and the complete results for the MultiFarm track can be found at the OAEI'22 website<sup>11</sup>. The results reported by the OAEI organisers on October 9th of 2022 describe the precision, recall and F-measure of the alignments produced by each of the participant systems. The aggregated results for completing the 55x24 matching tasks are shown in Table 1.

## 3. General comments

The following subsections contain some remarks and comments about the results obtained and the evaluation process.

### 3.1. Comments on the results

The obtained results in MultiFarm are intermediate in terms of F-measure (third-best result out of six participants), and are also very good in terms of recall, having attained the best result of any OAEI edition for the MultiFarm “different ontologies” sub-task<sup>12</sup>.

The results of CIDER-LM largely improve the ones obtained by its predecessor tool CIDER-CL [1] for the MultiFarm different ontologies sub-task, which were  $P = 0.16$ ,  $R = 0.19$ , and  $F = 0.17$

<sup>11</sup><https://oaei.ontologymatching.org/2022/multifarm/index.html>

<sup>12</sup>Only in OAEI'12 higher values can be found, but from two matchers that gave nearly every possible combination as result, thus resulting in close to zero values of precision and F-measure

The results of CIDER-LM in the OAEI multilingual track are still modest, especially in precision, but the fact that even the best systems do not score very high illustrates the difficulty of the problem. For instance, the F-measure attained in MultiFarm was never higher than 0.47 in any OAEI edition.

### **3.2. Discussions on the way to improve the proposed system**

The evaluation from the OAEI campaign shows that the CIDER-LM matching system has potential to improve its current results in several ways. For instance, the current version includes as context to obtain the embeddings: the entity labels, the labels of parents and children for classes, and the labels of domain and range for properties. A future version could include more features in the verbalization of the labels.

The current results show a clear unbalance between precision and recall. A higher threshold value would help promote the precision measure, further achieving a higher F-measure. The results also show the execution time of the tool is greater than the other participants. Removing the use of a reasoner would greatly reduce the time, with only a small impact in the performance of the results as seen in our own evaluation.

Furthermore, a more careful study of the fine-tuning process could lead to an improvement of the CIDER-LM alignment results. Involving more sophisticated techniques on the training and validation of the fine-tuned model can reduce overfitting and provide a more general model that behaves better with ontologies different from the ones seen in training.

### **3.3. Comments on the OAEI procedure**

The MELT wrapper for the Python matching system has proven to be easy to comprehend and implement, and useful for encapsulating the tool and later creating the web server container image for the OAEI submission. We consider that the inclusion of the MELT framework is a significant advancement in OAEI to allow the integration and participation of non Java-based implementations.

## **4. Conclusion**

This paper presented the first version of CIDER-LM, which explores for the first time the potential of multilingual language models on the task of finding cross-lingual alignments between ontologies in different languages. The system uses SBERT, a multilingual language model based on the Transformer architecture. It was evaluated in the OAEI'22 MultiFarm track, achieving intermediate results in terms of F-measure and very good results in terms of recall. Despite there is many room for further improvements, we consider that CIDER-LM results have proved the viability of using multilingual language models for this task.

In future versions, more ontology features will be considered to build the ontological context, new verbalisation strategies will be analyzed, and a more careful study of the fine-tuning process will be carried out, to attain a better and more general model for cross-lingual ontology matching.

## Acknowledgments

This paper is result of a collaboration grant 2021-22 at the Department of Computer Science and Systems Engineering, University of Zaragoza, funded by the Ministry of Education and Professional Training (Spain). It has been also partially supported by the Spanish project PID2020-113903RB-I00 (AEI/FEDER, UE), by DGA/FEDER, and by the *Agencia Estatal de Investigación* of the Spanish Ministry of Economy and Competitiveness and the European Social Fund through the “Ramón y Cajal” program (RYC2019-028112-I).

## References

- [1] J. Gracia, K. Asooja, Monolingual and cross-lingual ontology matching with CIDER-CL: Evaluation report for OAEI 2013, in: Proc. of 8th Ontology Matching Workshop (OM’13), at 12th International Semantic Web Conference (ISWC’13), volume 1111, CEUR-WS, ISSN-1613-0073, Sydney (Australia), 2013.
- [2] P. Sorg, P. Cimiano, Exploiting Wikipedia for cross-lingual and multilingual information retrieval, *Data & Knowledge Engineering* 74 (2012) 26–45. doi:10.1016/j.datak.2012.02.003.
- [3] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Stroudsburg, PA, USA, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.
- [4] A. Vaswani, G. Brain, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention Is All You Need, in: Proc. of 31st Conference on Neural Information Processing Systems (NIPS 2017), 2017.
- [5] Y. He, J. Chen, D. Antonyrajah, I. Horrocks, Biomedical Ontology Alignment with BERT, in: Proc. of 16th International Workshop on Ontology Matching (OM’21) co-located with the 20th International Semantic Web Conference (ISWC 2021), CEUR-WS, 2021, pp. 1–12.
- [6] N. Reimers, I. Gurevych, Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, in: Proc. of EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, 2019, pp. 3982–3992. doi:10.18653/v1/D19-1410. arXiv:1908.10084.
- [7] S. Neutela, M. H. de Boerb, Towards Automatic Ontology Alignment using BERT, in: Proc. of the AAAI 2021 Spring Symposium on Combining Machine Learning and Knowledge Engineering (AAAI-MAKE 2021), CEUR-WS, Palo Alto, California, USA, 2021.
- [8] S. Hertling, J. Portisch, H. Paulheim, MELT - matching evaluation toolkit, in: Semantic Systems. The Power of AI and Knowledge Graphs - 15th International Conference, SEMAN-TiCS 2019, Karlsruhe, Germany, September 9-12, 2019, Proceedings, 2019, pp. 231–245. URL: [https://doi.org/10.1007/978-3-030-33220-4\\_17](https://doi.org/10.1007/978-3-030-33220-4_17). doi:10.1007/978-3-030-33220-4\_17.