Linking Meaning: Semantic Technologies Shaping the Future of AI
B. Spahiu et al. (Eds.)

doi:10.3233/SSW250016

SISMA: Sentence Embedding–Based Ontology Matching with SBERT

Giulio MACILENTI [a,1], Manuel FIORELLI [a] and Armando STELLATO [a]
[a] *Tor Vergata University of Rome, Via del Politecnico 1, Rome 00133, Italy*
ORCiD ID: Manuel Fiorelli https://orcid.org/0000-0001-7079-8941, Armando Stellato https://orcid.org/0000-0001-5374-2807

Abstract.

Purpose: Ontology Matching (OM) has been studied for decades, yet fully automatic solutions remain elusive because ontologies differ in structure, granularity and vocabulary. Nevertheless, the abundant textual content attached to ontology entities suggests that the task could benefit from modern language-representation models. We therefore present the Semantically-Informed Similarity Matching Algorithm (SISMA), a novel system that matches concepts by leveraging the similarity of SBERT embeddings computed over pseudo-sentences extracted from the ontologies.

Methodology: We focus on the task of class and property equivalence. We represent each ontology concept as a set of SBERT embeddings associated with each predicate. For every pair, a similarity matrix is computed and reduced to a score via linear operations with two learnable matrices. These are trained on a dedicated dataset. We evaluated our system on the OAEI benchmark alignments, training on the Conference track and testing on the Circular Economy (CE) and Material Sciences and Engineering (MSE) tracks.

Findings: Our experiments reveal that the SISMA method achieves performance comparable to the state of the art. On the CE track our system achieves a higher F_1-score than the participating systems, while on the MSE track it performs slightly lower. We also compared our results with a baseline across the parameter space, confirming that the training step is key to overall performance.

Value: We have designed, implemented, and evaluated a novel system for ontology matching that achieves performance comparable to state-of-the-art methods. Our approach is readily extensible—primarily by training and testing on additional datasets—and the underlying idea can be realized in alternative ways, for example by replacing the current linear-operator scoring and threshold-filtering approach with a classifier that operates directly on the similarity matrix space.

Keywords. Ontology Matching, Sentence Embedding, Semantic Similarity

1. Introduction

Ontologies are a powerful method for organizing and structuring knowledge, based on the definition and formal representation of entities and their reciprocal relations [1,2]. This kind of structure allows for a robust but at the same time scalable and reusable

[1] Corresponding Author: Gulio Macilenti, giulio.macilenti@students.uniroma2.eu.

storage of information, and is particularly suitable for sharing open data across the web [3]. Ontologies are relevant in several applications, such as knowledge graphs [4] and database integration [5], and in many domains, such as healthcare [6], justice [7], e-commerce [8] or social networks [9].

Unfortunately, partly due to their very definition and their purpose of facilitating data exchanging in an open world, what happens in reality is that there is a proliferation of ontologies describing same or overlapping domains. This can cause problems with respect to interoperability and modularity of existing semantic sources, so ontology alignment (OA) or, more specifically, ontology matching (OM) is needed [10,11]. While ontology alignment is the most general task, aiming to find generic relations between concepts of two different ontologies, ontology matching is usually referred as the particular case in which this relation is a simple equality, and so the objective is to find matches between entities or properties. Although it is not a new task, ontology matching demands a lot of effort and is not easy to automate, most of the time requiring manual control by domain experts. It is therefore natural that there is a great interest in the development and design of matching systems. It is also natural that, since ontologies contain a lot of textual information in the form of annotations, comments and sometimes local names, many approaches try to find relations between concepts through methods originating from natural language processing, such as text embedding [12] and Large Language Models (LLMs) [13], which are currently achieving remarkable results across a wide range of applications in computer science.

It is in this context that we propose the Semantically-Informed Similarity Matching Algorithm-SISMA- a new matching method that exploits the embeddings of all the textual information contained in the ontologies to be matched. Our system embeds pseudo-sentences composed of all the object related to each specific predicate, then computes a similarity matrix between concepts, and refines it through linear operations involving a Weight and an Offset matrix. Full methodological details are provided in Section 3.

Our approach can be instantiated with a variety of embedding models; in this study, however, we evaluate it with SBERT [14], because preliminary tests demonstrated that it is particularly suitable for the matching task. We evaluated our method on three datasets from the popular Ontology Alignment Evaluation Initiative (OAEI) [15]. OAEI is an annual benchmark competition in which various systems compete to produce the best possible alignment on a predefined set of ontology pairs. We trained the Weight and Offset matrices on the OAEI Conference track and tested them on the Circular Economy (CE) and Material Science and Engineering (MSE) tracks. Our results show that the approach is promising: on the CE track the performance is higher than that of the other participating systems, while in the case of MSE it is slightly lower. We think that the method has several opportunities for improvement and that these results show evidence of great potential.

The rest of the paper is organized as follows: in Section 2 we give an overview of similar methods, which leverage text embeddings for ontology matching, in Section 3 we describe our method with all the details, while in Section 4 we report the results of the evaluation on the MSE and CE OAEI tracks, together with the comparison with a simple baseline configuration of our approach, to enlighten the importance of the learning step. Section 5 concludes with the current limitations and future developments of the system.

2. Related Work

Since ontology alignment is a long-standing task, various approaches have been developed over time, resulting in a vast body of literature. The first generation of methods was rule-based and relied primarily on lexical matching, often combined with structural information and filters to manage logical conflicts between potential correspondences. It is worth noting that many of these systems—such as LogMap [16] and AgreementMakerLight—still [17] achieve performance close to the state of the art in many equivalence matching tasks. In the last decade new systems have emerged that incorporate machine learning in various configurations [18,19], culminating in recent models that explicitly prompt large language models (LLMs) to generate or evaluate matches [20,21].

Given that the fundamental building block of our approach is the use of text embeddings—i.e., sentence representations in a high-dimensional vector space where distances reflect semantic similarity—we will focus this related work section on the most significant methods that leverage this technique to align or match concepts in ontologies or, more generally, semantic resources. To the best of our knowledge, Zhang et al. [22] were the first to incorporate word embeddings into ontology alignment. They addressed the shortcomings of similarity measures based on WordNet [23], which suffers from limited coverage of ontology elements. To overcome this, they trained word2vec embeddings [24] on Wikipedia and used cosine similarity between entity names, labels, and comments to identify matches. Their approach was evaluated on the OAEI 2013 benchmark and conference track, as well as on three real-world ontologies, and consistently outperformed WordNet-based methods. Dhouib et al. [25] proposed an ontology alignment method that combines FastText embeddings [26] with a radius-based similarity measure. In their approach, each concept is represented as the average of the vectors corresponding to its labels. This technique achieved state-of-the-art results on the OAEI conference complex alignment benchmark [27] and was validated in a real-world scenario involving the alignment of the Silex ontology—which models skills, occupations, and business sectors—with other related ontologies. Among other notable alignment systems are DeepAlignment [28] and OntoEmma [29], both of which leverage text embeddings and machine learning for concept matching. DeepAlignment enriches pre-trained embeddings by extracting synonymy and antonymy relations from ontological and external sources. OntoEmma, on the other hand, employs a neural architecture that incorporates external definitions and contextual information to enhance entity representations. This system was evaluated on the OAEI largebio SNOMED-NCI task.

All the aforementioned approaches rely on word2vec-style embeddings (including FastText), which lack contextual awareness and thus fail to capture nuanced variations in word meaning. The advent of transformer-based architectures [30] addresses this limitation by enabling the generation of contextualized embeddings that can differentiate between word senses. BERT [31] is a prominent model in this category and has been successfully applied to compute semantic similarity between concepts. However, since ontology alignment often involves comparing full sentences rather than individual words, BERT's original architecture is not directly suitable. To perform the embedding of sentences, Reimers and Gurevych introduced Sentence-BERT (SBERT) [14], a modified version of BERT capable of producing semantically meaningful sentence embeddings that can be compared using cosine similarity. SBERT outperformed other sentence embedding models, including GloVe [32], in most Semantic Textual Similarity

(STS) benchmarks. In the context of ontology alignment, one of the first applications of transformer-based embeddings is presented in [33]. Beutel and Boer evaluated multiple alignment strategies using both BERT and SBERT embeddings, as well as traditional word2vec representations, in a practical use case in the labor market: aligning the ESCO and O*NET ontologies. Their findings suggest that BERT-based embeddings generally outperform word2vec, although they also note that the results are not yet sufficient for fully automated alignment in real-world applications. This highlights the need for hybrid approaches that combine automatic and manual alignment. Building on BERT, He et al. [34] introduced BERTMap, a system that predicts ontology mappings using a classifier fine-tuned on semantic corpora derived from ontologies. The mappings are further refined through structural and logical reasoning. Evaluation on a subset of the OAEI LargeBio Track shows that BERTMap frequently surpasses existing methods. In [35], instead, Sousa et al. propose an approach for complex matching that leverages various embedding models together with a smart usage of SPARQL queries. The generation of correspondences is performed by matching similar surroundings of instance sub-graphs. Such a method is tested in four architectural modification on the populated version of the OAEI Conference benchmark and compared to state-of-the-art approaches, obtaining a significative performance improvement.

Finally, we highlight two recent approaches that bear resemblance to our own, albeit with notable differences. The first is TEXTO (TEXT-based Ontology matching system) [36]: in this system concept labels are embedded using GloVe [32], while the accompanying descriptions are processed with SBERT. The final similarity score is computed as a weighted combination of the cosine similarities derived from both features. If this score exceeds a predefined threshold, a match is established. TEXTO was evaluated on the OAEI Common Knowledge Graphs Track, enriched with class descriptions, and a newly aligned dataset between Schema.org and Wikidata, demonstrating promising results. The main difference between TEXTO and our method lies in the type of features used: TEXTO considers only class labels and their descriptions, while omitting other properties available in the ontologies. Secondly, the weights assigned to the two components of the similarity score are not learned, but rather heuristically preset, and, finally, there is not an offset parameter involved in the score computation. This is an important difference because, as we will explain in more detail in the next section, the presence of an offset matrix makes it possible to assign a negative weight to the information carried by a pair of predicates if it falls below a certain threshold. The second system is Natural Language Focused Ontology Alignment (NLFOA) [37]. NLFOA represents semantic and structural information of nodes and relations as pseudo-sentences, that are subsequently embedded with SBERT and compared via cosine similarity. A threshold filter is subsequently applied to identify valid matches. Although it appears conceptually close to ours, this model has two crucial differences. First, all the information related to each concept is aggregated into a single embedding vector and not distributed across multiple embeddings for different predicates. Second, the representation is not obtained with the pre-trained version of SBERT, but instead the embedding model is finetuned using the reference alignments. As a consequence, the model obtains outstanding performance after the learning phase, as reported in the experiments described in the paper, but it heavily depends on the availability of alignment data for the specific dataset pair being evaluated. Our model SISMA also uses a supervised learning approach, but, unlike NLFOA, it does not need examples from the dataset nor from the domain on which it is applied, since it uses the learning phase only to infer generic properties common to all alignments. The cited article also reports the performance in a zero-shot configuration: the F_1-score in this case is at the lower end of that of the systems participating in the OAEI competition, but yet remains valuable as a baseline.

3. Methodology

In this section, we separately describe all the steps of our ontology matching method and the datasets used for training and testing. More details about the datasets can be found on the OAEI website.

3.1. Our Approach

The workflow of our system is illustrated in Figure 1. The first step involves extracting all textual information from the two ontologies to be matched. At the end of this phase, each entity or relation in the two semantic resources is associated with a list of pseudo-sentences, where each sentence corresponds to a specific predicate from the RDF, RDFS, or OWL vocabularies.

These texts are then embedded using SBERT, resulting in a set of vectors from which a similarity matrix S^{AB} is computed for each pair of entities A and B in the two ontologies. A similarity score is subsequently derived from S^{AB} through linear operations involving two additional matrices, W and O. Finally, a thresholding step determines which pairs are similar enough to be considered a match.

Importantly, the matrices W and O represent explainable properties that are, to a good approximation, independent of the specific ontologies being matched. Specifically, W represents the relevance of predicate pair similarity in determining concept correspondences, while O contains information about the cosine similarity threshold that determines whether a predicate pair contributes positively or negatively to the matching decision.

Since the elements of W and O are related to meta-properties derived from upper ontology vocabularies (such as RDF, RDFS, and OWL), these matrices—as well as the threshold—can be learned from a training alignment. As shown in our evaluation, training does not need to be performed on a domain similar to the target one; it is only necessary that the relevant predicates are sufficiently represented in the training datasets.

The following subsections provide more detailed explanations of each processing step.

3.1.1. Text Extraction and Embedding

The first step, as previously described, involves extracting textual information from the two semantic resources undergoing the matching process and embedding it. A simple example illustrating how the method works is provided in Figure 2.

For each concept in the two ontologies, we extract all triples where the concept appears as the subject. From this set of triples, we generate pseudo-sentences by collecting all the objects associated with the same predicate, lexicalizing them, and concatenating the resulting strings with a comma. An alternative approach would have involved embedding each object individually and then aggregating them (e.g., via averaging) to obtain

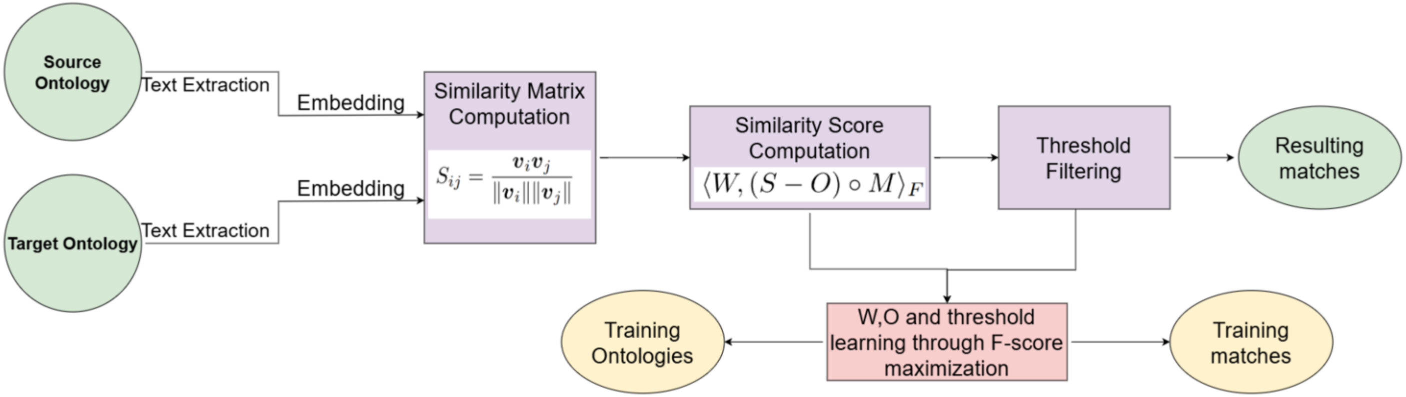
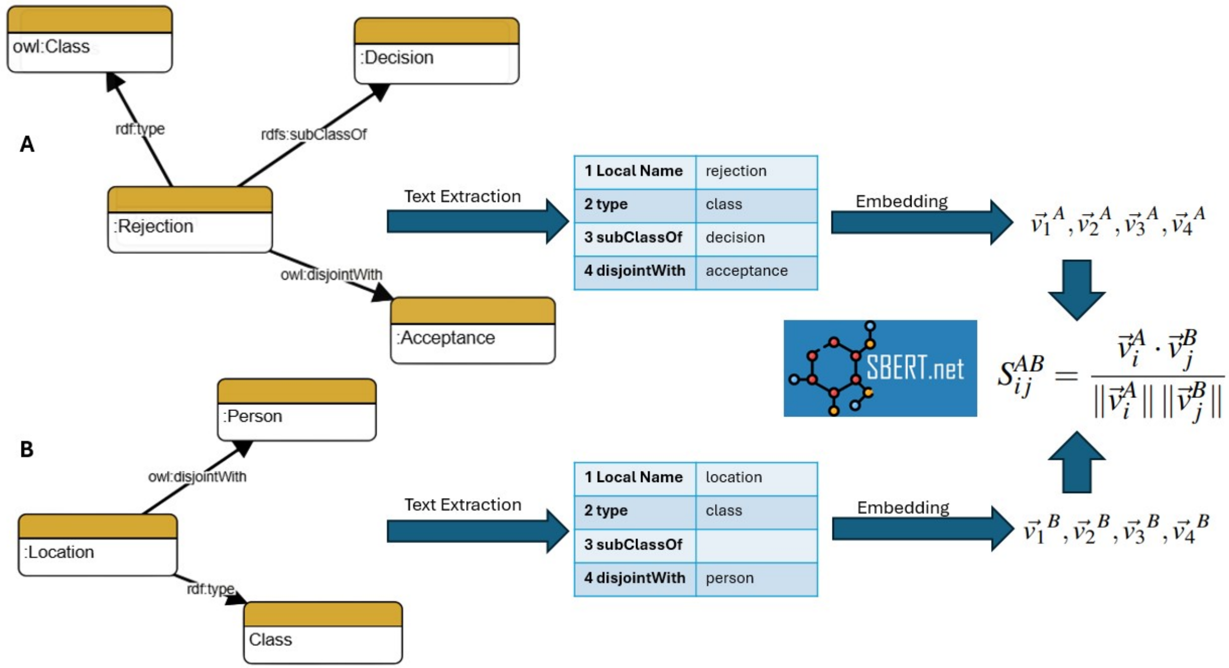

Source Ontology
Text Extraction
Embedding
Target Ontology
Text Extraction
Embedding
Similarity Matrix Computation
$S_{ij} = \frac{\boldsymbol{v}_i \boldsymbol{v}_j}{\|\boldsymbol{v}_i\| \|\boldsymbol{v}_j\|}$
Similarity Score Computation
$\langle W, (S - O) \circ M \rangle_F$
Threshold Filtering
Resulting matches
W,O and threshold learning through F-score maximization
Training Ontologies
Training matches

Figure 1. Outline of the main components of the SISMA workflow.



1 Local Name	rejection
2 type	class
3 subClassOf	decision
4 disjointWith	acceptance

$$\vec{v_1}^A, \vec{v_2}^A, \vec{v_3}^A, \vec{v_4}^A$$

$$S_{ij}^{AB} = \frac{\vec{v}_i^A \cdot \vec{v}_j^B}{\|\vec{v}_i^A\| \, \|\vec{v}_j^B\|}$$

1 Local Name	location
2 type	class
3 subClassOf	
4 disjointWith	person

$$\vec{v_1}^B, \vec{v_2}^B, \vec{v_3}^B, \vec{v_4}^B$$

Figure 2. An example of text extraction, embedding and similarity matrix computation for two different entities. Both concepts belong to ontologies from the OAEI conference track: the first is from the *cmt* ontology, while the second is from the *ekaw* ontology. Since the corresponding predicate is not present for entity B, $\vec{v_3}^B$ is the zero vector.

a single representative vector. Nonetheless, this method is computationally more expensive and still entails some issues, as it requires using sentence embedding models even for single words: this results in averaging embeddings of multiple pseudo-sentences. We opted for embedding all the information in a single vector, considering it a more elegant and promising solution.

With regard to the predicates considered, although in principle it is possible to include all those appearing in the triples of the target ontologies, in this study we restrict ourselves to a predefined list consisting of all properties in the RDF, RDFS, and OWL vocabularies, with the addition of a synthetic predicate designed to capture the local name. This choice is not limiting, as in most cases—including all those used for training and evaluation—these predicates cover the full set of those actually used in the ontologies. However some semantic resources may define and use custom annotation properties. Consequently it becomes important to include all occurring predicates, provided that appropriate training data is available to support them.

The lexicalization procedure varies depending on the nature of the object: if it is a literal, the textual content is used directly; if it is an IRI, our first choice is to use the local name. While the inclusion of local names may introduce noise in cases where they are not semantically informative, this did not pose a significant problem in our evaluation. The OAEI ontologies used for testing and training assign meaningful local names to entities. However, if this issue becomes problematic for other resources, it is possible to mitigate it by applying simple filtering strategies—for example, discarding local names that contain more than 50 % digits, as proposed in [21]. In this cases, to lexicalize the concept it is necessary to retrieve a label (such as rdfs:label, skos:prefLabel, or skos-xl:prefLabel) depending on what the ontology provides. Blank nodes are completely ignored in this implementation, however it is straightforward to consider them through the same procedure if they are labeled.

As a result of this process, we obtain textual chunks that correspond to the different predicates associated with each concept. These texts are then preprocessed by splitting CamelCase expressions—common in IRI local names—into individual tokens, lower casing and removing special characters. Subsequently, the resulting sentences are separately embedded using SBERT [2] to obtain vector representations suitable for downstream similarity computation. At the end of text extraction and embedding, each concept in both target and source ontologies is represented by a set of vectors indexed by the previously mentioned predicates,as shown in Figure 2.

3.1.2. Similarity Matrix, Score Computation and Threshold Filtering

Given the set of vectors obtained in the previous section, we compute a similarity matrix S for any pair of concepts. Specifically, for a concept A in the source ontology and a concept B in the target ontology, we define the matrix element S_{ij}^{AB} as:

$$S_{ij}^{AB} = \frac{\vec{v}_i^A \cdot \vec{v}_j^B}{||\vec{v}_i^A|| \cdot ||\vec{v}_j^B||} \tag{1}$$

where $\vec{v}_i^A$ denotes the embedding vector of the objects linked to concept A by predicate i, and $\vec{v}_j^B$ is the corresponding vector for concept B and predicate j. In other words,

[2] The exact model is *all-mpnet-base-v2*

the elements of S represent the cosine similarities between vectors associated with different predicates. By definition, S is symmetric and remains invariant under the interchange of concepts.

Computing such a matrix entails iterating over all the properties (i.e., predicates) of the given concepts, represented as triples, and measuring their pairwise similarity. For instance, if both concepts A and B are classes, the corresponding triples—`A rdf:type owl:Class` and `B rdf:type owl:Class`—will yield a similarity value of 1 in the respective entry of S. Conversely, if A is a subclass of *Animal* (`A rdf:subClassOf Animal`) and B is a subclass of *Material* (`B rdf:subClassOf Material`), the resulting similarity will be lower for that specific entry.

The core idea of our approach is that treating each predicate separately preserves substantially more information than aggregating all textual content into a single sentence embedding, as done in [37]. In addition, we assume that not only similarities between identical predicates (i.e., the diagonal elements of S) are relevant—cross-predicate similarities may also contribute meaningfully to the matching process. At this point, we sum all the similarities stored in the S matrix to compute a score, which is then compared against a threshold to determine a match.

When aggregating all the elements of S, it is essential to weight them appropriately, to reflect at least two key aspects. First, not all predicates are equally important. For instance, the similarity between primary labels is typically more informative than the similarity between superclasses. Many non-diagonal elements should, in fact, receive a weight of zero. As an illustration, consider objects linked by `rdf:type`: they often bear no meaningful relation to those linked by `rdf:subClassOf`, and similar situations occur with other predicate pairs. The different importance of various predicates is captured in our model by a weight matrix W over the space of predicates.

Second, certain entries in S may provide negative evidence—that is, they may indicate that two concepts do not match. For example, matching concepts typically share the same `rdf:type`; thus, a similarity value below one for this predicate should not contribute positively to the overall score, but rather negatively. We model this property by an offset matrix O: for each specific predicate pair, an offset value between zero and one is subtracted from the similarity score to reflect the penalization.

All these considerations can be summarized in a simple formula that describes our system. Given a normalized weight matrix W, an offset matrix O, and a threshold T, a match between two entities represented by S is established if and only if:

$$\langle W, (S - O) \circ \delta_{S>0} \rangle_F > T \tag{2}$$

Here, $\langle \cdot, \cdot \rangle_F$ denotes the Frobenius inner product between matrices, and the symbol $\circ$ represents the element-wise (Hadamard) product. $\delta_{S>0}$ is the mask matrix, where each entry is equal to 1 if the corresponding entry in matrix S is non-zero and 0 otherwise. It ensures that the offset is subtracted only when a given pair of predicates is present. In index notation, the above expression becomes:

$$\sum_{i,j} W_{ij} \cdot (S_{ij} - O_{ij}) \cdot \delta_{S_{ij}>0} > T \tag{3}$$

It is questionable whether non-diagonal elements—for example, the similarity between the label of one class and the description string of another—contribute significantly to the matching process, and whether a particular choice of values for W and O improves performance compared to an unweighted sum. Both issues will be addressed in our evaluation, where we compare our approach to one baseline model that simply uses an identity matrix for W, and another diagonal model that completely discards contributions from non-identical predicates while still retaining learned weights for W and O. The next section describes how W, O, and T are learned from a training dataset.

3.1.3. Learned Parameters: Weight Matrix, Offset, and Threshold

As we stated and argued in the previous section, W, O, and T are parameters not related to the specific pair of ontologies involved in the matching process, but rather general properties that refer to the meaning of predicates and their relevance for tracing correspondences. As a consequence, we can learn their values by optimizing the F_1-score of the alignment that our model is able to generate, provided that we have at our disposal one or more reference alignments between some training ontologies. The relation between the training datasets and the target alignment does not have to be strict. The ontologies used can describe different domains, since semantic similarity is encoded in the pre-trained SBERT embeddings. The only requirement is that both the training set and the test set use similar predicates, so that the weights learned by the model are meaningful for the new matching task. More details about the datasets will be given in the next section.

We used simulated annealing [38] to learn the values for W, O, and T: the F_1-score of the training alignment serves as the objective function. In principle, it is possible to optimize the parameters without placing any constraint on the number of matches generated by SISMA. However, we adopted a different protocol. We constrained the model to propose exactly as many correspondences as there are in the reference alignment, so that recall equals precision and the F_1-score during training. In the end, only a portion of the target matches is found, and the threshold is chosen as the lowest among the scores of the correct pairs. We adopted this configuration because it avoids a disproportion between recall and precision, imposing their equality. Otherwise, the system often settles into a high-precision local minimum, compromising its generalization ability and overall performance, as confirmed by our experiments.

Another important observation is that this method has proven more effective at jointly learning W, O, and T. Indeed, while the first two matrices are independent of the specific alignment, the final threshold T can vary significantly depending on the granularity or specificity of the alignment under consideration. Our preliminary tests indicated that learning T without applying constraints on the number of matches proposed during training can lead to overfitting. One solution to mitigate this issue is to apply various *a posteriori* heuristics to the distribution of scores, that is, not directly using the threshold learned by the model, but applying an appropriate transformation of it with respect to the distribution of scores on the new dataset. However, the simplest and most effective approach was to adopt the previously described bounded protocol from the outset.

The optimization of W and O starts from a default configuration where W is the identity matrix and O is the zero matrix. Values of W are constrained to be positive and to sum to one, while the entries of O are constrained between zero and one. We conducted several tests to establish the hyperparameters for the learning process: our final configuration uses an initial temperature of $T = 0.15$, a decrease rate of $\alpha = 0.986$, and a total number of steps $N = 2100$. In our experiments, a plateau is typically reached around $N = 1500$. At each step, one move on the W matrix and one move on the O matrix is performed. We repeated the tests multiple times to ensure that optimization yields consistent results for the learned matrices and performance.

3.2. Datasets

In this section, we report key details regarding the datasets used for training and evaluation. All three datasets are sourced from the Ontology Alignment Evaluation Initiative.

For training, we employed the OAEI Conference track, which consists of 21 alignments corresponding to the complete alignment space among seven ontologies describing the same domain—namely, the organization of a scientific conference. As a reference alignment, we used the one identified as `ra1`, which is publicly available alongside the datasets on the official OAEI website. The 21 alignments collectively contain 305 correct correspondences (including both classes and properties) over a total search space of 270759 possible pairs.

Performance evaluation was conducted on two separate tracks: Circular Economy (CE) and Material Science and Engineering (MSE). The CE track [39,40] involves the alignment between two ontologies in the Circular Economy domain: the Circular Economy Ontology Network (CEON) and the Sustainable Bioeconomy and Bioproducts Ontology (BiOnto). This alignment includes 18 correct matches within a search space of 407,040 candidate pairs.

The MSE benchmark addresses the materials science and engineering domain and comprises three test cases. We focused on the second test case, which involves aligning the MaterialInformation and MatOnto ontologies. In this alignment, the search space includes 590633 possible matches, of which only 302 are correct. It is important to note that participants in this track are provided with two background knowledge resources intended to serve as semantic bridges between ontology concepts: a Periodic Table Dictionary (created using Protégé) and the European Materials Modelling Ontology (EMMO). SISMA does not utilize this additional information. This should be considered when comparing its performance with other systems, as many of the reference correspondences rely on the ability to associate atomic symbols from the periodic table with their natural language labels—an operation that these background resources are specifically designed to support.

It is also important to briefly explain the rationale behind our selection of alignment tasks. We aimed to use medium-sized ontologies to ensure evaluations that are both meaningful and computationally manageable, especially considering the quadratic growth of the search space, resulting from all the possible pairs of concepts. The datasets were deliberately selected from distinct domains, and we ensured the presence of established evaluation benchmarks to support reliable comparisons. Furthermore, we prioritized ontologies in which local names were semantically meaningful, as previously discussed, in order to simplify the implementation. We also intentionally excluded datasets employing SKOS-XL, as our goal was to focus exclusively on predicates defined within RDF, RDFS, and OWL. Although SKOS-XL labels could significantly enhance alignment—especially in systems like ours that rely on lexical information—we chose to evaluate our method in a more general and challenging context. This decision was made to avoid the risk of overestimating performance on potentially biased or cherry-picked test cases.

4. Evaluation

In this section, we report the results of the experiments conducted to evaluate the performance of our method. The evaluation is organized into two subsections: the first compares SISMA with state-of-the-art systems participating in the corresponding OAEI tracks; the second analyzes its performance against a baseline, by varying the threshold used in the last step of filtering.

We evaluate our method in two different settings. The first, referred to as SISMA, corresponds to the standard configuration described in detail in the previous sections. The second, denoted as SISMA-D, refers to a variant in which only the diagonal elements of the similarity matrix S—i.e., those corresponding to identical predicates—are considered.

4.1. Comparison with State-of-the-Art Systems

As previously mentioned, the Weight matrix W and the Offset matrix O were trained on the OAEI Conference track. We then evaluated the performance of the method on the Circular Economy and Material Science and Engineering tracks. The results in terms of recall, precision, and F_1-score for these datasets are reported in Tables 1, 2, and 3, respectively, and are compared with the most recent results available for the corresponding tracks.

Although our main evaluation focuses on the CE and MSE tracks, we also report results on the training dataset, compared with those of other systems participating in the conference track. This choice is motivated by the fact that, as previously explained, the optimization of W and O is not tied to properties specific to the domains or ontologies present in the training set. Consequently, while the risk of overfitting cannot be completely ruled out, performance on the training data still provides a reasonable approximation of the method's expected average performance during testing. In fact, the training process does not exploit any information beyond the relative importance of the various predicates.

As shown in Table 1, SISMA-D ranks second in terms of F_1-score among the systems participating in the Conference track, while SISMA ranks sixth.

In Table 2, it is possible to observe that both SISMA configurations significantly outperform the systems participating in the CE track. The diagonal configuration still achieves the highest F_1-score. On this particular dataset, all approaches reach the same Recall value; it is their Precision that determines the difference in overall performance.

Finally, in Table 3 are reported the results on the MSE track. Here our system has a performance slightly lower than the others. The main difference, however, lies in Recall, while the Precision of our method remains consistently higher. It is worth noting again that the systems participating in the MSE track have access to two additional resources, including a periodic table dictionary. Many of the reference alignment matches involve relations between concepts representing atoms—expressed in MaterialInformation using periodic table acronyms, and in MatOnto using textual labels. It is therefore reasonable to assume that the lack of access to such additional information in our experiment may account for the lower Recall observed in this case. Even on the MSE track, SISMA-D performs slightly better than SISMA. A more detailed comparison between the two configurations of our model will be provided at the end of the next section, where we

Table 1. Comparison between the performance obtained by our method after training and the participants in the OAEI 2023 conference track[41]. The best results are highlighted in bold.

Matcher	Recall	Precision	F1-score
SISMA-D	0.61	0.79	0.69
SISMA	0.58	0.58	0.58
LogMapLt	0.47	0.68	0.56
Matcha	0.62	0.62	0.62
GraphMatcher	**0.77**	0.71	**0.74**
OLaLa	0.61	0.59	0.60
ALIN	0.44	0.82	0.57
edna	0.45	0.74	0.56
LSMatch	0.41	**0.83**	0.55
LogMap	0.56	0.76	0.64
SORBETMtch	0.61	0.73	0.66
AMD	0.41	0.82	0.55
PropMatch	0.08	0.86	0.29
StringEquiv	0.41	0.76	0.53
TOMATO	0.47	0.57	0.52

Table 2. Comparison of performance between our system and the participants in the OAEI 2024 CE track[42]. The best results are highlighted in bold.

System	Recall	Precision	F1
SISMA	**0.611**	0.440	0.512
SISMA-D	**0.611**	**0.550**	**0.579**
Matcha	**0.611**	0.393	0.478
LogMap	0.500	0.391	0.439
LogMapLt	**0.611**	0.379	0.468

Table 3. Comparison of performance between our system and the participants in the OAEI 2023 MSE track[41]. The best results are highlighted in bold.

System	Recall	Precision	F1
SISMA	0.172	0.852	0.287
SISMA-D	0.172	**0.945**	0.291
Matcha	**0.219**	0.756	**0.339**
LogMap	0.195	0.881	0.320
LogMapLt	0.189	0.851	0.309

discuss why—despite the diagonal version performing better—it remains important to consider the entire similarity matrix S for ontology matching and alignment.

We conclude this section with a general remark: while relying on a novel architecture and being trained on a relatively limited amount of data, our system achieves performance that is comparable to that of state-of-the-art approaches. We believe that this result represents a promising first step for future research in this direction.

4.2. *Comparison with Baseline Across Parameter Space*

In this section we present a comparative analysis of the general SISMA model, its variant that uses only the diagonal of the similarity matrix (SISMA–D), and a baseline method that we will introduce below. The comparison is carried out by varying the threshold T.

The aims of this study are threefold. First, because tuning T is non-trivial, we verify that the value obtained with the protocol described in Section 3.1.3 yields performance close to the global optimum. Second, we provide a detailed comparison between SISMA and SISMA–D, since the latter, counterintuitively, achieves a higher F_1-score in all of our previous tests. Third, a baseline is required to quantify the advantage of jointly learning the weight matrices W and O from data.

The baseline is defined as follows. The Weight matrix W is set to the identity and scaled so that its elements sum to one, leaving all off-diagonal entries equal to zero. The Offset matrix O is likewise restricted to the diagonal, with every entry fixed to the constant value 0.33. This configuration therefore preserves only pairs of identical predicates, without assigning relative weights to individual pairs or applying predicate-specific thresholds to decide whether a cosine similarity contributes positively or negatively to the match. The value of 0.33 was chosen after some preliminary tests: it was experimentally observed that it corresponds to the cosine similarity value below which two concepts have no appreciable semantic proximity. In practice, therefore, our baseline corresponds to a version of SISMA-D in which there has been no learning step.

As already said, the comparison is performed while varying the threshold. However, the scale of significant values of T changes between SISMA, SISMA-D and the baseline: consequently, for a clearer comparison, we rescale the thresholds as follows. For each model we set $T = 0$ at the point where recall equals 1 and $T = 1$ at the point where recall falls to 0. This linear transformation allows all three methods to be plotted on the same graph and compared across the full span of relevant thresholds.

The results of our experiments are summarized in Figures 3 and 4. First, the F_1-score peaks occur near the threshold T estimated from the training data, demonstrating that our tuning procedure is effective for the threshold as well as for the matrices W and O. Second, on both datasets SISMA and SISMA–D outperform the baseline over most of the threshold range, with the greatest margin in the high-precision region. The peak F_1-score obtained by the two variants is 50–100 % higher than that of the baseline, underscoring the benefit of learning non-uniform predicate parameters from data.

With respect to the comparison between SISMA and SISMA–D, this further analysis confirms that the diagonal variant performs better on both the CE and MSE datasets. At first glance this outcome is counter-intuitive: SISMA, by exploiting the full similarity matrix, should in principle have an advantage. The evidence suggests instead that the parameters learned for the off-diagonal entries of W and O mostly capture noise, resulting in overfitting. Nevertheless, we do not regard this finding as sufficient reason to discard the full-matrix approach in favor of SISMA–D. Several considerations prevent us from drawing such a definitive preference. First, the performance gap between the two variants is modest. Second, as already mentioned, when two ontologies employ locally defined annotation properties, cross-predicate similarities become crucial, so the off-diagonal terms exploited by SISMA encode useful information. More broadly, such cross-predicate contributions could prove valuable in difficult matching scenarios or in the presence of missing data, provided they are activated under an appropriate selection rule. We therefore argue that further training and evaluation on larger, more diverse benchmarks is needed to assess more precisely the strengths and limitations of both models.

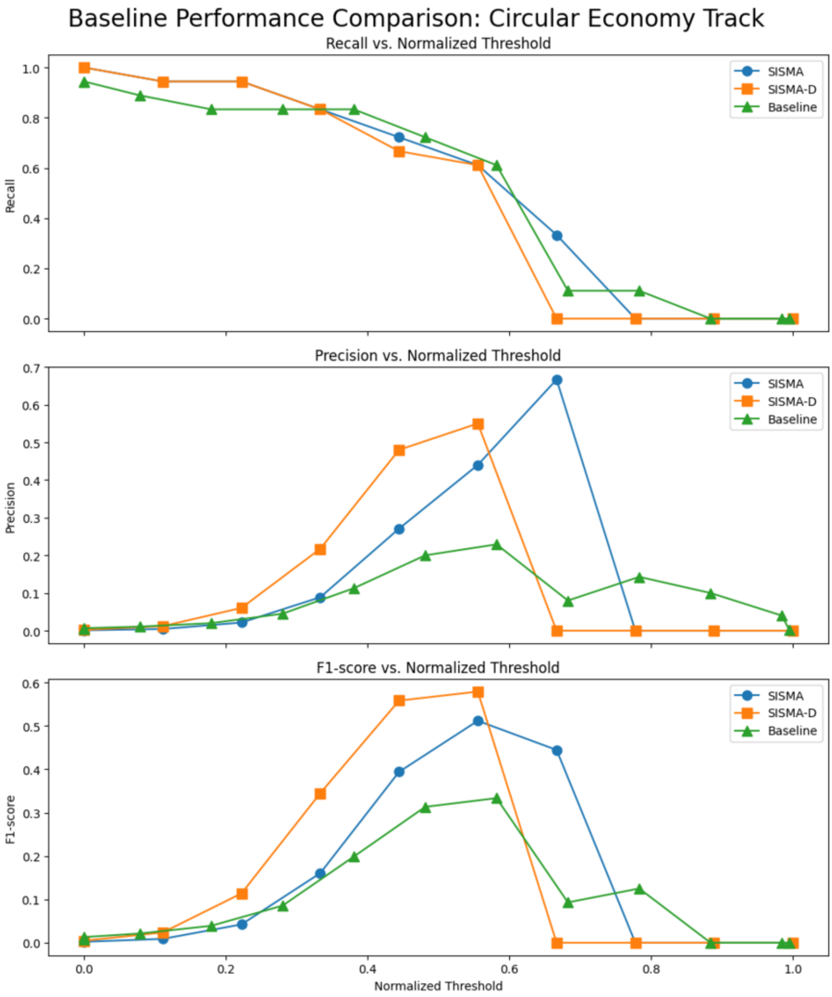


Figure 3. Comparison of SISMA, SISMA–D, and the baseline in terms of recall, precision, and F_1-score, for the CE track. The x-axis shows the normalized threshold.

5. Conclusion

We have designed, implemented and tested a new ontology matching system: SISMA. Our approach is based on well-established techniques, such as text embedding and semantic similarity, but organizes them in a totally new predicate-aware strategy. Our experiments show that this method achieves results comparable and sometimes superior to the state of the art on three datasets from the OAEI benchmark.

More specifically, SISMA and its variant SISMA-D clearly outperform all other competing systems in the OAEI Circular Economy track. In the Materials Science and Engineering track—where they had no access to background knowledge—their scores are only slightly below those of the top-performing methods. Learning for the system's parameters has been carried on the OAEI conference track: although performance on the training set is not fully indicative, the results are nevertheless encouraging on these

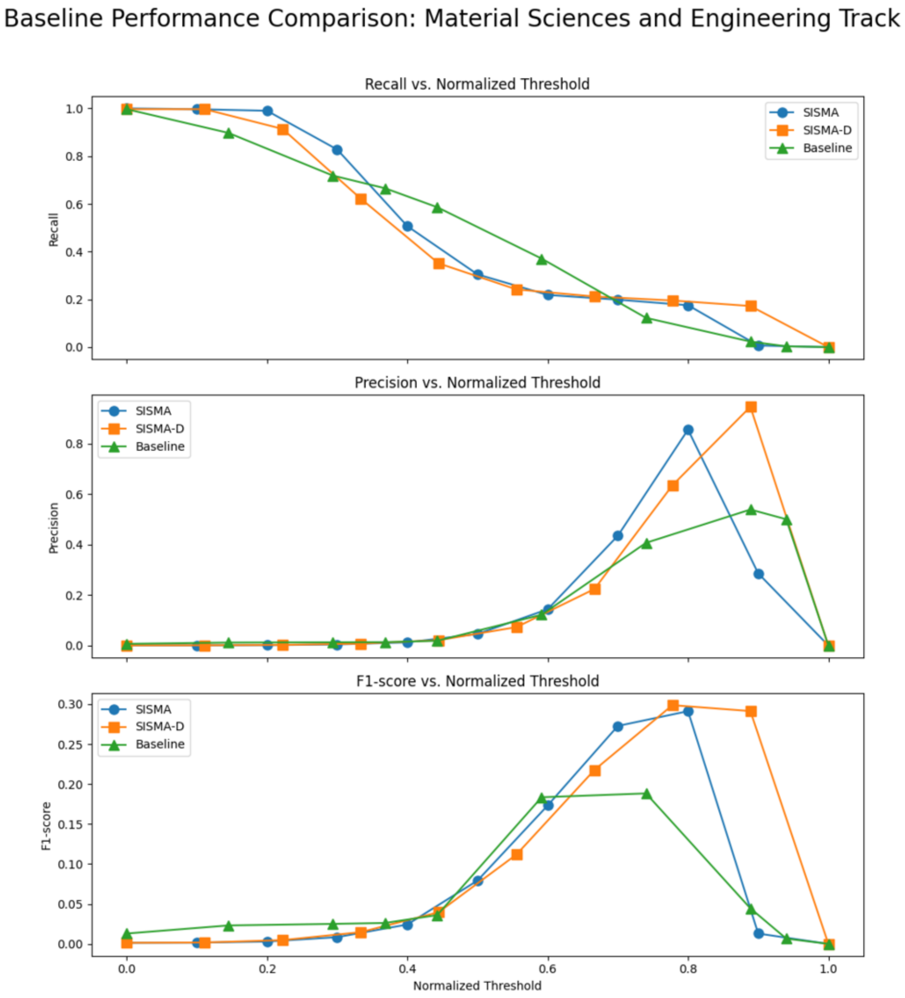


Figure 4. Comparison of SISMA, SISMA–D, and the baseline in terms of recall, precision, and F_1-score, for the MSE track. The x-axis shows the normalized threshold.

benchmark alignments too. In addition, we conducted an exploration of the parameter space and compared both variants of our model against a baseline, consistently observing higher performance.

Several limitations and possible extensions remain. A first objective is to train and evaluate the model on new datasets, expanding the predicate set to include SKOS-XL properties, or, more generally, all predicates occurring in the two ontologies to be matched. This could also help to clarify the comparative performance of the SISMA–D variant. As for more substantial changes to our approach, two promising modifications can be explored. First, the model could be extended to consider triples in which the concepts to be matched occur as objects as well as subjects. Second, the similarity matrix could be processed by a non-linear classifier instead of the linear transformations adopted in this work. Although this would reduce interpretability, it might enable the discovery of richer relations beyond one-to-one correspondences, thus shifting the focus from ontology matching to ontology alignment. In any case, we believe that the evaluations carried out in this article already indicate that SISMA is a powerful ontology matching method, capable of competing with the state of the art, and worthy of further development.

References

[1] Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications. *The knowledge engineering review*, 11(2), 93-136.

[2] Davies, J., Studer, R., & Warren, P. (Eds.). (2006). *Semantic Web technologies: trends and research in ontology-based systems*. John Wiley & Sons.

[3] Schmachtenberg, M., Bizer, C., & Paulheim, H. (2014). Adoption of the linked data best practices in different topical domains. In *The Semantic Web–ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I 13* (pp. 245-260). Springer International Publishing.

[4] Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G. D., Gutierrez, C., ... & Zimmermann, A. (2021). Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4), 1-37.

[5] De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., & Rosati, R. (2017). Using ontologies for semantic data integration. In *A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years* (pp. 187-202). Cham: Springer International Publishing.

[6] Luschi, A., Petraccone, C., Fico, G., Pecchia, L., & Iadanza, E. (2023). Semantic ontologies for complex healthcare structures: a scoping review. *IEEE Access*, 11, 19228-19246.

[7] Van Engers, T., Boer, A., Breuker, J., Valente, A., & Winkels, R. (2008). *Ontologies in the legal domain. Digital Government: E-Government Research, Case Studies, and Implementation*, 233-261.

[8] Fensel, D., McGuiness, D. L., Schulten, E., Ng, W. K., Lim, G. P., & Yan, G. (2001). Ontologies and electronic commerce. *IEEE Intelligent Systems*, 16(1), 8-14.

[9] Finin, T., Ding, L., Zhou, L., & Joshi, A. (2005). *Social networking on the semantic web. The Learning Organization*, 12(5), 418-435.

[10] Ehrig, M. Ontology alignment: bridging the semantic gap (Vol. 4). Springer Science & Business Media (2006).

[11] Shvaiko, P., & Euzenat, J. Ontology matching: state of the art and future challenges. *IEEE Transactions on knowledge and data engineering*, 25(1), 158–176 (2011).

[12] Incitti, F., Urli, F., & Snidaro, L. (2023). Beyond word embeddings: A survey. *Information Fusion*, 89, 418-436.

[13] Kumar, P. (2024). Large language models (LLMs): survey, technical frameworks, and future challenges. *Artificial Intelligence Review*, 57(10), 260.

[14] Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using siamese BERT networks. arXiv preprint arXiv:1908.10084 (2019)

[15] Euzenat, J., & Shvaiko, P. (2007). *Ontology matching* (Vol. 18). Heidelberg: springer.

[16] Jiménez-Ruiz, E., & Cuenca Grau, B. (2011, October). Logmap: Logic-based and scalable ontology matching. In *International Semantic Web Conference* (pp. 273-288). Berlin, Heidelberg: Springer Berlin Heidelberg.

[17] Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I. F., & Couto, F. M. (2013). The agreement-makerlight ontology matching system. In *On the Move to Meaningful Internet Systems: OTM 2013 Conferences: Confederated International Conferences: CoopIS, DOA-Trusted Cloud, and ODBASE 2013, Graz, Austria, September 9-13, 2013. Proceedings* (pp. 527-541). Springer Berlin Heidelberg.

[18] Ngo, D., Bellahsene, Z., & Coletta, R. (2012). Yam++-a combination of graph matching and machine learning approach to ontology alignment task. *Journal of Web Semantics*, 16(16).

[19] Djeddi, W. E., & Khadir, M. T. (2013). Ontology alignment using artificial neural network for large-scale ontologies. *International Journal of Metadata, Semantics and Ontologies* 16, 8(1), 75-92.

[20] Norouzi, S. S., Mahdavinejad, M. S., & Hitzler, P. Conversational Ontology Alignment with ChatGPT. CoRR abs/2308.09217 (2023).

[21] Hertling, S., & Paulheim, H. OLaLa: Ontology matching with large language models. *Proceedings of the 12th Knowledge Capture Conference 2023* (2023), 131–139.

[22] Zhang, Y., Wang, X., Lai, S., He, S., Liu, K., Zhao, J., Lv, X.: Ontology matching with word embeddings. In: Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, pp. 34–45. Springer (2014)

[23] Pedersen, T., Patwardhan, S., Michelizzi, J.: Wordnet Similarity: Measuring the relatedness of concepts. In: HLT-NAACL 2004, pp. 38–41. Association for Computational Linguistics (2004)

[24] Mikolov, T.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)

[25] Dhouib, M.T., Zucker, C.F., Tettamanzi, A.G.: An ontology alignment approach combining word embedding and the radius measure. In: International Conference on Semantic Systems, pp. 191–197. Springer (2019)

[26] Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics 5, 135–146 (2017)

[27] Thieblin, E.: Task-oriented complex alignments on conference organisation (2019). https://figshare.com/articles/dataset/Complexalignmentdatasetonconferenceorganisation/4986368/8

[28] Kolyvakis, P., Kalousis, A., Kiritsis, D.: Deepalignment: Unsupervised ontology matching with refined word vectors. In: NAACL 2018, pp. 787–798

[29] Wang, L.L., Bhagavatula, C., Neumann, M., Lo, K., Wilhelm, C., Ammar, W.: Ontology alignment in the biomedical domain using entity definitions and context. arXiv preprint arXiv:1806.07976 (2018)

[30] Vaswani, A., et al.: Attention is all you need. Advances in Neural Information Processing Systems 30 (2017)

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

[32] Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543 (2014)

[33] Neutel, S., de Boer, M.H.T.: Towards automatic ontology alignment using BERT. In: AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering (2021)

[34] He, Y., Chen, J., Antonyrajah, D., Horrocks, I.: BERTMap: A BERT-based ontology alignment system. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 5, pp. 5684–5691 (2022)

[35] Sousa, G., Lima, R., & Trojahn, C. (2025). Complex ontology matching with large language model embeddings. *arXiv preprint arXiv:2502.13619*.

[36] Peng, Y., Alam, M., Bonald, T.: Ontology Matching using Textual Class Descriptions. In: OM@ISWC 2023 (2023)

[37] Schneider, F., Dash, S., Bagchi, S., Mihindukulasooriya, N., & Gliozzo, A. M. (2023, December). Nlfoa: Natural language focused ontology alignment. In Proceedings of the 12th Knowledge Capture Conference 2023 (pp. 114-121).

[38] Bertsimas, D., & Tsitsiklis, J. (1993). Simulated annealing. *Statistical science*, 8(1), 10-15.

[39] Li, H., Blomqvist, E., & Lambrix, P. (2024). Initial and experimental ontology alignment results in the circular economy domain. In *Proceedings of the 2nd International Workshop on Knowledge Graphs for Sustainability* (KG4S2024), CEUR-WS. org.

[40] Blomqvist E, Li H, Keskisärkkä R, Lindecrantz M, Abd Nikooie Pour M, Li Y, Lambrix P, ”Cross-domain Modelling – A Network of Core Ontologies for the Circular Economy”, *The 14th Workshop on Ontology Design and Patterns (WOP 2023) at the 22nd International Semantic Web Conference* (ISWC 2023), Athens, Greece, 2023

[41] Abd Nikooie Pour, M., Algergawy, A., Buche, P., Castro, L. J., Chen, J., Coulet, A., ... & Zhou, L. (2023). Results of the Ontology Alignment Evaluation Initiative 2023. In *CEUR workshop proceedings* (Vol. 3591, pp. 97-139). RWTH Aachen.

[42] Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C., Hertling, S., Li, H., Shvaiko, P., & Euzenat, J. (2025). Proceedings of the 19th International Workshop on Ontology Matchingco-located with the 23rd International Semantic Web Conference (ISWC 2024). In *The 19th International Workshop on Ontology Matching co-located with the 23rd International Semantic Web Conference* (ISWC 2024). CEUR-WS. org.