

¹ Syrian Virtual University

²Damascus University

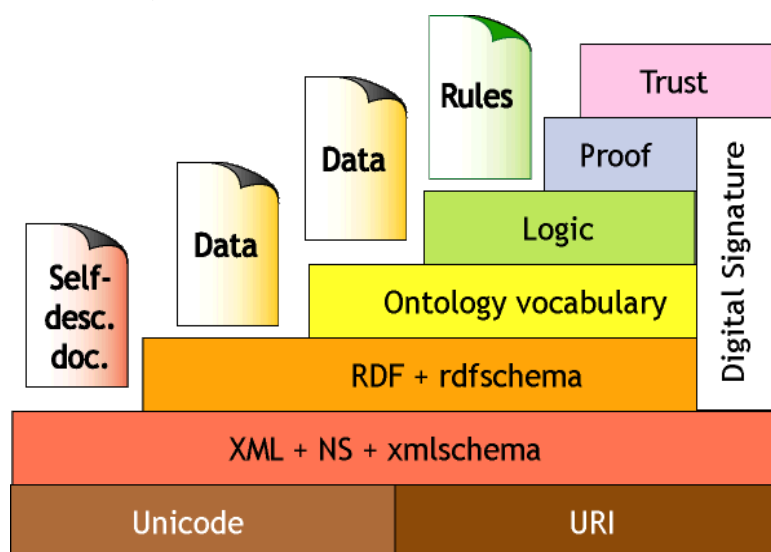
Ontology is the philosophical study of being. It investigates what types of entities exist, and how they are related to one another on the most fundamental level, ontology is the skeleton of the semantic web, many algorithms developed to make the integration of existing ontologies on the world wide web, that these ontologies belong to different domains, and its mandatory to benefit from these ontologies to integrate these ontologies based on domain, we propose a new automatic algorithm for ontologies integrations based on domain, and this algorithm will generate two types of semantic relations between ontologies Is_A relations and Part_Of relations, the proposed algorithm will use word embedding technology to generate Is_A relations, we have used PubMedBERT which is pre trained model from scratch using abstracts from full-text articles from PubMed Central, and wordNet to generate Part_Of relations, we develop this algorithm on the ontologies in medical domain based on the popularity of ontologies in this domain, we have tested our algorithm the accuracy rate was 85 %.

INTRODUCTION

As the figure above we notice that the ontology is the central element in the semantic web layers, in general, each ontology contains four types of entities [2]:

- Object properties (i.e., relationships among individuals)
- Datatype properties (i.e., attributes associated with individuals).

An additional type of entities (annotation properties) is used to add human-readable metadata (such as labels and comments) at different levels of the ontology. Concepts and properties are organized within hierarchies using the is-a relation.



Ontology is defined in multiple languages, the most popular and standard language for describe ontology is Web Ontology Language (OWL) is a language for defining ontologies on the Web. An OWL

Ontology describes a domain in terms of classes, properties, and individuals also it contains descriptions of the characteristics of those objects. OWL ontologies can be used to describe the properties of Web

resources. Ontologies are domain based which means that each domain contains its specific ontologies, currently ontologies exist in all fields, such as engineering, medicine, pharmacy . . . , the medical sector has the biggest number of ontologies, and actually there are many repositories on the internet which contains a large numbers of medical ontologies such the Europe PMC repository which is the partner of PubMed Central (PMC), an ELIXIR core data resource [3], and the repository of choice for many international science contains about 90 medical ontologies in it. In general Ontology merging methods is classified into two main classes:

- Simple merge.
- Full merge.

In simple merge [4] the input ontologies are imported into a new one—constituting a union of input ontologies—and adds bridging axioms, called articulations, translating the alignment between these input ontologies. These added axioms are semantic correspondences. Union of O_1 , O_2 will be $O_3 = O_1 \cup O_2$. The resulting ontology is generally called a bridge ontology, the following figure show the simple merge of ontologies:

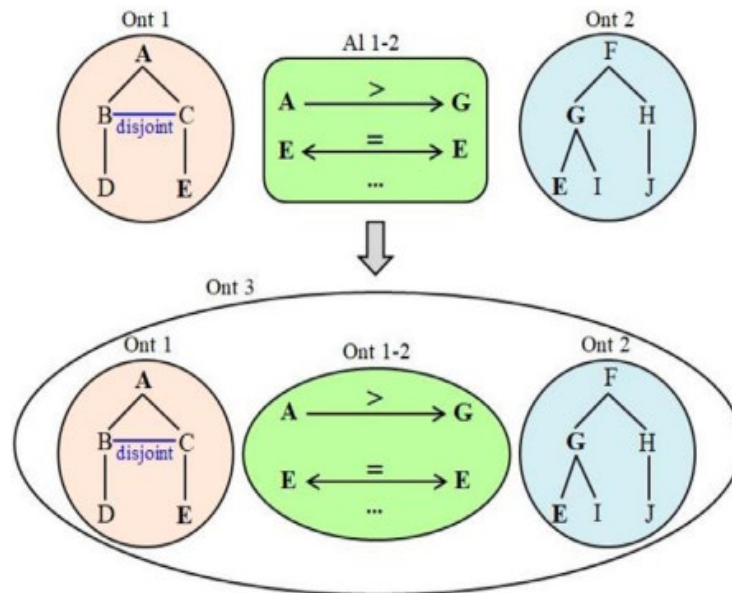


FIGURE 2: Ontology Simple Merge [4]

In full merge the input ontologies are imported into a new ontology—constituting a union of input ontologies—and merges each set of equivalent entities into a single new entity that preserves all their attached description and relations. Ontology integration is essential in semantic web because when these resources (ontologies) are disconnected this will affect the inference capabilities because the inference will not take into consideration the whole information which will be exist in the distributed and disconnected ontologies, and ontologies integration is very important especially in medical domain that by using this integration we can integrate the diseases ontologies with the symptom and diagnosis ontologies.

Related Work

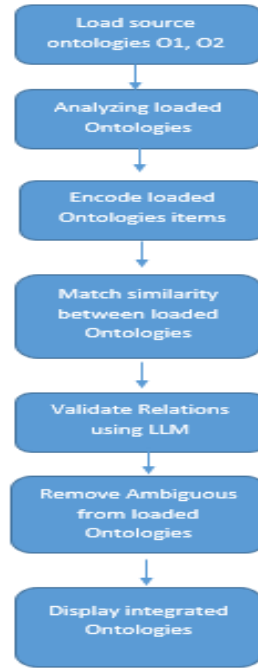
Actually, there are many developed algorithms for ontology integrations, we will talk about these algorithms. In general ontology integration approaches are defined as the following:

- Simple Alignment only contains simple entity-to-entity correspondences. The most used format for representing simple pairwise alignments is the RDF Alignment format2 [5].
- Complex Alignment contains at least one complex correspondence—expressed as a logical rule. In this case, EDOAL3 [6] is the most used format for representing complex pairwise alignments.
- Pairwise Alignment, only involves two ontologies to match: a source ontology and a target ontology; and only contains correspondences linking two entities, each coming from an ontology.

- Holistic Alignment multi alignment [7], involves more than two ontologies and contains at least one correspondence that links different entities coming from more than two input ontologies.
- Oriented Alignment or directional alignment , is the directed or oriented version of a pairwise alignment [4], denoted $A_{O1} \rightarrow O_2$ where the source ontology is O_1 , and the target ontology is O_2 .
- Non-Oriented Alignment is a non-directed or non-oriented pairwise alignment, denoted A or $A_{O1} \leftrightarrow O_2$ and briefly called an alignment. When an alignment is a bidirectional one,
- Compound Alignment [8] involves more than two ontologies and contains at least one complex correspondence that involves an expression composed of named entities from two or more ontologies.
- Ternary Alignment [8] is a special case of a compound alignment. It involves three ontologies: a source and two targets; and contains at least one complex equivalence correspondence between a named entity from the source ontology and an expression involving two named entities from the two target ontologies.

Proposed Algorithm

Our proposed algorithm workflow in shown in the following figure:

**FIGURE 3:** Algorithm Workflow

We will explain in detail all the steps of our proposed algorithm in the following:

Load Source Ontologies

In this step the input ontologies are loaded, here in our proposed algorithm it makes the integration between two ontologies, first ontology (O1), and the second ontology (O2).

Analyzing Loaded Ontologies

In this step the algorithm detects the different items of the loaded ontologies, usually each ontology contains four main types of entities [9]:

- 1-concepts (or classes).
- 2-individuals (or instances of classes).
- 3-object properties (i.e., relationships among individuals).
- 4-datatype properties (i.e., attributes associated with individuals).

5-Additional type of entities (annotation properties) is used to add human-readable metadata (such as labels and comments) at different levels of the ontology, so the algorithm in this step stores each entity type in array, after finishing this step we have for each ontology four arrays which contain concepts, individuals, object properties, datatype properties.

Encode Loaded Ontologies Items

In this step all the entities of the loaded ontologies are encoded, encoding means that these ontology entities are converted to numeric vectors representation, for this purpose we used a medical Sentence Transformers which is a text embedding framework, The initial work is described in Sentence-BERT paper, which describes Sentence Embeddings using Siamese BERT-Networks, in sentence embedding we used PubMedBERT which is pre trained model from scratch using abstracts from PubMed and full-text articles from PubMed Central. This model achieves state-of-the-art performance on many biomedical NLP tasks, and currently holds the top score on the Biomedical

Language Understanding and Reasoning Benchmark, after finishing this step the loaded ontologies are converted to lists of numeric vectors, numeric lists for concepts, numeric lists for individuals, numeric lists for properties, numeric lists for data types.

Match Similarity Between Ontology Embeddings

In this step the algorithm tries to find the relations between classes in the loaded ontologies, the similarity matching tries to find the following relations between loaded ontologies:

- Is_A relations
- Part_of relations

Is_A relation meaning:

A is_a B should be read to mean: Every instance of A is an instance of B.

Part_Of relation meaning:

A part_of B should be read to mean: Every instance of A is part of some instance of B. For example.

Find Is_A relations:

So, the proposed algorithm first try to find the Is_A relations between classes/entities, the Is_A relation is detected between two classes if the similarity score between these two classes numeric vectors (vectors which are the classes encoding vectors) are greater than the similarity threshold (which set to 0.7), the similarity score between classes numeric vectors is calculated by applying cosine similarity formula as the following:

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}},$$

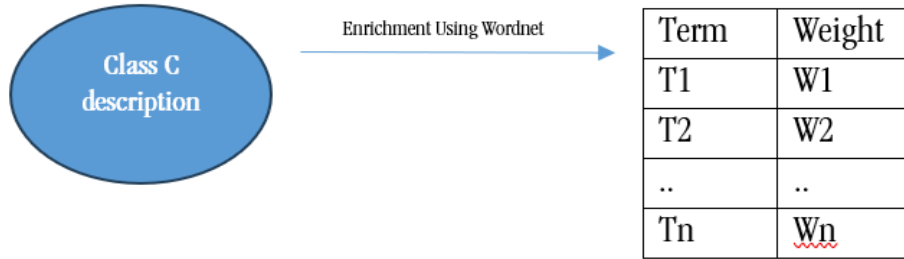
Where:

- A: is the numeric vector of the first class C1 which belongs to O1

- B: is the numeric vector of the second class C2 which belongs to O2

Find Part_Of relations:

In this step the algorithm tries to find the Part_Of relations between loaded ontologies classes as the following: Enrich each class description (this step is done by using wordNet as a general ontology



After enrichment is done to all classes descriptions, then the algorithm calculates the semantic similarity between two classes based on the following formula [10]:

$$F(SR_{i_1}, SR_{i_2}) = \frac{\sum_{c \in (SR_{i_1} \cap SR_{i_2})} W_{SR_{i_1}}(c) * W_{SR_{i_2}}(c)}{|SR_{i_1} \cap SR_{i_2}|}$$

Where:

SR_{i1}: is the semantic representation of the first-Class description.

SR_{i2}: is the semantic representation of the second-Class description.

W_{SR i 1}: is the function return the weight of the word in SR_{i1}

W_{SR i 2}: is the function return the weight of the word in SR_{i2}

c: Are all shared terms between the two semantic representations

To normalize the similarity score we divide it on the number of the shared terms between the two semantic representations of the selected classes. After calculating the similarity score between the two classes the algorithm decides that these two classes must be related to each other by part of relation if the similarity score is greater than 0.5.

Validate Relations Using Large Language Model

In this step the algorithm validates the generated relations between ontologies concepts (which are generated from previous step), the validation will be applied on the two types of relations, the validation is applied by using large language model (LLM) Gemini Pro 002 as the following:

- Validate Is_A relations:

In this step the Is_A relations are validated using Gemini Pro 002 with the following prompt:

Is {Concept1} a sub concept of {Concept2}? or Is the stated relationship accurate?

Where:

- Concept1: is the first concept in the is_a relation
- Concept2: is the second concept in the is_a relation

- Validates Part_Of relations:

In this step the Part_Of relations are validated using Gemini Pro 002 with the following prompt:

source), the algorithm tokenizes the class description and for each token the algorithm finds the synonym of it from WordNet and add it to the class bag of words, by the end of this step each class/concept will be converted to a bag of words list(class description words and its synonyms), which contains the term/token and its weight(the weight is the count of the term in the bag of words after enrichment).

Is {Concept1} a part of {Concept2}? or Is the stated relationship accurate?

By the end of this step the relations which not approved by the LLM will be deleted from the source and the target ontologies.

Remove Ambiguous from Loaded Ontologies

The aim of this step is to remove many to many relations between first ontology O1, and the second Ontology O2, by converting it to one-to-one relations by removing the many to many relations and keep only one to one relation, that if the algorithm finds many relations between two classes it will delete them and keep only one relation based on the following criteria:

- If the type of these relations is same and from type Is_A relations then the algorithm will keep only the relation which has the highest score.
- If the type of these relations is same and from type Part_Of relations then the algorithm will keep only the relation which has the highest
- If the type of these relations is different than the algorithm will keep only the relation which has the highest score and which type is Is_A

Display Integrated Ontologies

Here the system views the origin ontologies and the new established relationships between these ontologies classes.

Evaluation

We have evaluated our proposed algorithm on two medical ontologies.

- International Classification of Diseases ontology [11] which is International Statistical Classification of Diseases and Related Health Problems.
- Disease Ontology which has been developed as a standardized ontology for human disease with the purpose of providing the biomedical community with consistent, reusable, and sustainable descriptions of human disease terms.

Evaluation Metrics

We have chosen the following metric for evaluation our proposed algorithm.

$$\text{Precision} = (\text{automatic relations} \cap \text{manual relations}) / \text{automatic relations}$$

Where:

Automatic Relations: is the relation set which generated by the algorithm

Manual Relations: is the relation set which generated by human expert.

Based on the precision formula above, our developed algorithm reached 85% accuracy rate.

Discussion

In this paper we have developed a new algorithm for automatic ontology integration, the developed algorithm can be used in multiple domains not just in the medical domain and the developed algorithm tries to decrease the complexity and the ambiguity of generated relationships between the input ontologies, that our developed algorithm takes two ontologies as input and generates one ontology which keep the structure of the input ontologies and creates relationships between ontologies concepts, and it not affected by the language which used to describe the input ontologies (OWL,RDF) because our algorithm loads the input ontologies and determines the types of the objects in these ontologies and then tries to build relationships between ontologies concepts, our developed algorithm build two types of relationships "IS-A" relations and "Part-Of"

relations, our developed algorithm depends on concepts embedding and also it used WordNet as a public domain ontology and the algorithm determines the relationship between concepts based on similarity score thresholds.

Conclusion

In this paper we proposed an automatic ontology integration algorithm we used for simple integration that our algorithm remains the existing ontology's structure and tries to find a new relation between input ontologies, our developed algorithm have used two methodologies to find two types of relations which are Is_A relations and Part_Of relations, to detect the first type of relations we have used medical text embedding and then cosine similarity formula to find these types of relations, and for detecting the second type of relations we have used a semantic representation to the concepts using wordNet as a public ontology resource. And then we proposed a formula to calculate the similarity between concepts semantically, by the end of the proposed algorithm job the source ontologies will keep as they are and new connections between these ontologies will be defined according to algorithm methodology in relations construction, our proposed algorithm reached to 85 accuracy rate, in the future will can enhance the accuracy of this algorithm by utilizing medical large language models in the integration process.

References

- [1] M.-R. Koivunen and E. Miller, "W3c semantic web activity," *Semantic Web Kick-Off in Finland*, vol. 2, pp. 27-44, 2001.
- [2] J. r. m. Euzenat and P. Shvaiko, "Ontology matching," 2013.
- [3] C. Durinx *et al.*, "Identifying ELIXIR core data resources," *F1000Research*, vol. 5, pp. ELIXIR-2422, 2017.
- [4] I. Osman, S. B. Yahia, and G. Diallo, "Ontology integration: approaches and challenging issues," *Information Fusion*, vol. 71, pp. 38-63, 2021.
- [5] J. David, J. Euzenat, F. Scharffe, and C. Trojahn dos Santos, "The alignment API 4.0," *Semantic web*, vol. 2, no. 1, pp. 3-10, 2011.
- [6] J. Euzenat, J. David, and M. Atencia, "EDOAL: Expressive and Declarative Ontology Alignment Language," [Online] <http://alignapi.gforge.inria.fr/edoal.html>, 2011.
- [7] I. Osman, S. F. Pileggi, S. B. Yahia, and G. Diallo, "An alignment-based implementation of a holistic ontology integration method," *MethodsX*, vol. 8, p. 101460, 2021.
- [8] C. Pesquita, M. Cheatham, D. Faria, J. Barros, E. Santos, and F. M. Couto, "Building reference alignments for compound matching of multiple ontologies using OBO cross-products," in *OM*, 2014, pp. 172-173.
- [9] J. Euzenat and P. Shvaiko, *Ontology matching*. Springer, 2007.
- [10] K. Omar, B. Alkhatib, M. Dashash, and F. Alhassan, "The Implementation of using Medical ontologies in Plagiarism detection," *International Review on Computers and Software (IRECOS)*, vol. 11, no. 3, pp. 232-238, 2016.
- [11] W. H. Organization, "International classification of diseases, version 10," *Geneva: World Health Organization*, 2010.



This is an open-access article distributed under the terms of the Creative Commons Attribution License.