Semantic Web 1 (0) 1–4
IOS Press

OntoMatcher: Leveraging Context-Aware Siamese Networks, LLMs and BioBERT for Enhanced Biomedical Ontology Alignment

Zakaria Hamane[a] *, Amina Samih[b] and Abdelhadi Fennan[c]
[a,b,c] *Department of Computer Sciences, Data & Intelligent Systems (DIS) Team, Faculty of Sciences and Techniques, Abdelmalek Essaadi university, Tanger, Morocco*
E-mail:

Abstract. Biomedical ontologies play a crucial role in knowledge representation and standardization within the biomedical domain. With the rapid growth of ontologies, the need for efficient and accurate alignment techniques has become paramount to ensure interoperability between various biomedical systems. Current ontology alignment methods often struggle to cope with the complex and dynamic nature of biomedical terminologies, resulting in suboptimal performance. In this study, we introduce a novel supervised deep learning approach for aligning biomedical ontologies, employing Large Language Models (LLMs) alongside Bidirectional Encoder Representations from Transformers for Biomedical Text Mining (BioBERT), One Dimensional Convolutional Neural Network (1D-CNN), highway networks, bi-directional long short-term memory (Bi-LSTM) and Siamese Network models. This approach captures character-level and contextual information of entities and efficiently incorporates entity descriptions and context embeddings to improve alignment accuracy. The results of our method demonstrate a significant improvement in performance, achieving an F1 score of 0.87 for match/not match classifications and 0.94 for level classifications, outperforming several baselines on benchmark datasets. These results indicate the potential of our approach, employing LLMs for data enrichement and Transformer models for embeddings, in facilitating a more effective alignment of biomedical ontologies. Ultimately, this enhances data integration and interoperability across different biomedical systems.

Keywords: Biomedical Ontologies, Ontology Alignment, BioBERT, Deep Learning, Entity Embeddings, Highway Network, Siamese Network, LLM

1. Introduction

Ontologies are essential for representing information in a particular field [1], including the biomedical field. They provide a standardized vocabulary for annotating and querying data and enable interoperability between different systems. However, the proliferation of ontologies in the biomedical domain has led to a large number of ontologies with overlapping, but slightly different vocabulary sets [2]. This makes it difficult to compare and combine data from different ontologies, as each ontology may represent the same concept with different terminology (e.g., “Adrenocortical Hyperfunction” in MeSH [3] is the same as “Hypercortisolism” in SNOMED [4]). This poses a challenge for NLP-based biomedical applications that require multiple ontologies (e.g., ”Skin Epithelioid Hemangioma” in

*Corresponding author. E-mail: zakaria.hamane1@gmail.com.

NCI [5] is the same as "Epithelioid hemangioma of skin" in SNOMED), but require each concept to be represented by only one entity. Furthermore, the rapid growth of biomedical literature and new discoveries constantly introduce new terms, exacerbating the issue of overlapping vocabularies.

To address this challenge, ontology alignment techniques have been developed to automatically map semantically equivalent entities from one ontology to another [6]. Although these methods have shown effectiveness in ontology alignment, there are still some limitations that need to be addressed. In this paper, we propose a novel approach that tackles these limitations and improves the accuracy and efficiency of ontology alignment in the biomedical domain.

Specifically, we present the following contributions:

- We introduce an innovative neural architecture for ontology alignment that utilizes a combination of BioBert [7], 1D-CNN [8], and highway network to better represent Out of Vocabulary (OOV) [9] terms, which is a limitation of traditional approaches in the biomedical domain.
- Our proposed method also automatically labels parent-child relationships in Unified Medical Language System (UMLS) [10] to enhance the model's performance. This added complexity as opposed to the traditional match or not match categories, allows the model to accurately capture parent-child relationships and improve the accuracy of ontology alignment.
- Furthermore, we leveraged multiple data sources such as Wikipedia and scientific articles, integrating a Large Language Models (LLMs) as an agent to search and query these databases. This approach enriched our ontology with additional descriptions and context of entities, consequently enhancing the coverage of the new features and improving alignment accuracy.

The rest of this paper is structured as follows: Section 2 provides an overview of related works and their limitations; Section 3 introduces the problem definition and offers a comprehensive discussion of the overall architecture of our proposed approach; Section 4 explains the process of obtaining the training data, including parent-child relationship labeling and the use of additional data sources; Section 5 delves into the details of the OntoMatcher approach; Section 6 presents the experimental results, validation, and comparison to baseline methods; and finally, Section 7 concludes the paper with a summary of our findings, limitations, and possible directions for future research.

2. Related work and limitations

Several methods have been proposed for ontology alignment, including rule-based (e.g. Gunter Saake et al. 2005) [11], statistical matching (Fernandez, S. et al.) [12], and deep learning approaches (e.g. Lucy Lu Wang et al. 2018 [13] and Jifang Wu et al. 2020 [14]). However, these methods face some limitations, particularly in the biomedical domain. One limitation is that they may not perform well on OOV) biomedical names, as they rely on word2vec to get the embeddings of the entities [13]. Another limitation is that many papers using supervised deep learning methods for medical ontology matching have focused primarily on comparing the similarity of individual pairs of entities [14] and have not utilized any techniques for comparing the hierarchical or relational structure of the ontologies. This can make it difficult to correctly match entities that have parent-child relationships in well-defined ontologies. Additionally, there is low coverage of the additional information, such as descriptions or contextual information, for many of the medical entities in the training and test data sets.

Our proposed approach innovatively addresses these limitations by incorporating advanced techniques and leveraging diverse data sources to enhance the ontology alignment process. By employing BioBERT, 1D-CNN, and a highway network and Siames Network, our method can better represent OOV terms, which is a significant challenge faced by existing methods. Furthermore, our approach takes into account the hierarchical structure of ontologies and automatically labels parent-child relationships in UMLS, increasing the model's performance and ability to match entities with complex relationships. Finally, we utilize a wide array of data sources, such as Wikipedia articles and scientific literature, to provide supplementary information for medical entities, enabling a more comprehensive understanding of their context and ultimately leading to improved accuracy in ontology alignment.

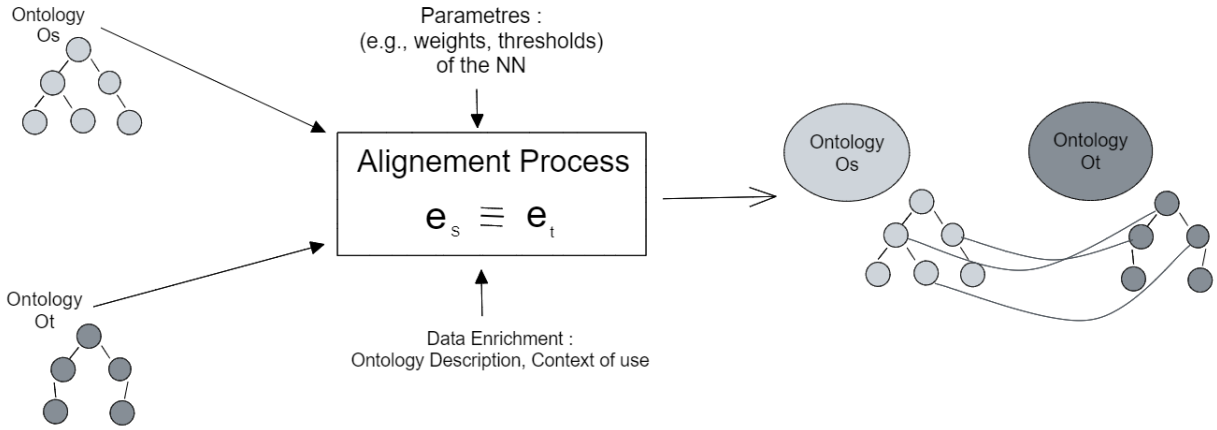

Fig. 1. Ontology Matching Problem - Identifying Semantically Equivalent Entity Pairs in Source and Target Ontologies.

3. Problem definition and overall architecture

We start by defining the ontology matching problem as depicted in Figure 1. Given a source ontology O_s and a target ontology O_t, each consisting of a set of entities, find all semantically equivalent entity pairs, i.e., (e_s, e_t) $O_s \times O_t$ e_s e_t, where indicates semantic equivalence. To facilitate the alignment process, we preprocess the entities in O_s and Ot to have the same set of attributes: a name of the entity (e_{name}), a list of alternative names ($e_{alternatives}$), a textual description ($e_{description}$), and a list of usage contexts ($e_{context}$). These attributes capture the essential information about each entity and provide a basis for comparing and aligning entities from different ontologies.

In this paper, we use the following notation:

- O_s and O_t: source and target ontologies, respectively.
- E_s and E_t: sets of entities in ontologies O_s and O_t, respectively.
- (e_s, e_t): a pair of entities in E_s and E_t, respectively.
- Alignment: set of pairs of entities that are semantically equivalent.
- e_{name}, $e_{alternatives}$, $e_{description}$, $e_{context}$: attributes of an entity.
- f: the ontology alignment algorithm, which maps the attributes of entities in E_s and E_t to a binary label indicating whether they are semantically equivalent (1) or not (0).

With this notation in mind, we can now describe our approach for aligning biomedical ontologies using OntoMatcher.

The OntoMatcher system is designed as shown in Figure 2 in a cohesive framework that effectively handles the ontology alignment problem. It can accept two ontologies, O_s and O_t, as inputs and outputs a list of alignments between their respective entities. In scenarios where a neural network is utilized, the Preprocessing and Inference modules are combined within the network, thus streamlining the alignment process:

- Obtaining and Normalizing UMLS Data: This module focuses on the extraction and standardization of ontology data from the UMLS. Details of this module can be found in §3.
- Preprocessing: This module performs essential preprocessing steps on the source and target ontologies, such as embedding representations. For more information on preprocessing techniques, refer to §3.3 and §3.4.
- Inference: The inference module is responsible for determining semantic equivalence between entity pairs using the ontology alignment algorithm f. The details of the inference process are provided in §3.5.

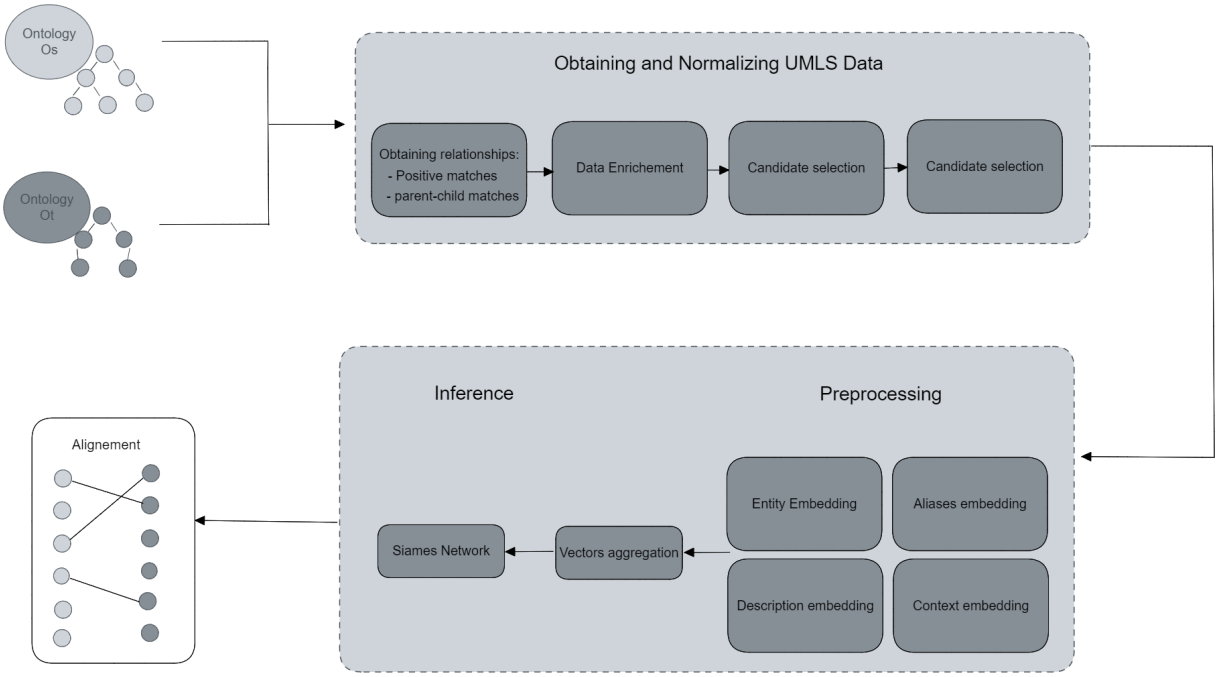

Fig. 2. OntoMatcher System Framework - From Data Normalization to Inference for Ontology Alignment

4. Obtaining labeled data from UMLS

A comprehensive resource for biomedical and health informatics, the UMLS offers a number of tools and databases to make it easier to access electronic health records (EHRs) and other healthcare data. It was created by the National Library of Medicine (NLM) [15] and is intended to speed up information retrieval and NLP in the biomedical field, as of the 2022AB release a massive, multilingual thesaurus with over 17 million names, 4.6 million concepts, and 8.9 million codes from over 26 languages and 182 source vocabularies are called the Metathesaurus [16]. Hierarchical and associative links connect these ideas and phrases.

The UMLS allows users to map concepts and terms from different source vocabularies to a common set of concepts, and therefore it is widely used for tasks such as ontology alignment, information extraction, and NLP in the biomedical domain [17]. UMLS provides a parent-child relationship between the concepts and terms, which is important in ontology alignment, where the parent-child relationships can be used to improve the accuracy of the alignment.

4.1. Extracting training data

To train our ontology alignment models, we utilized the UMLS to label our data as positive, negative, parent, or child match. We identified the following set of ontologies within the UMLS to use as the source of the labeled data: Medical Subject Headings (MeSH), SNOMED (global standards for health terms), Current Procedural Terminology (CPT), Gene Ontology (GO), Hugo Nomenclature (HGNC), Human Phenotype Ontology (HPO), Online Mendelian Inheritance in Man (OMIM), and RxNorm [18].

Although there are more complex relationships present in the UMLS, in this study, we only chose four categories for the alignment process. A positive label indicates that the entities s and t are semantically equivalent as illustrated in Figure 3. Our labeled data takes the form (es, et, l 0, 1, 2, 3) where:

– $l = 1$ indicates positive examples where es et, meaning that entities s and t are semantically equivalent.

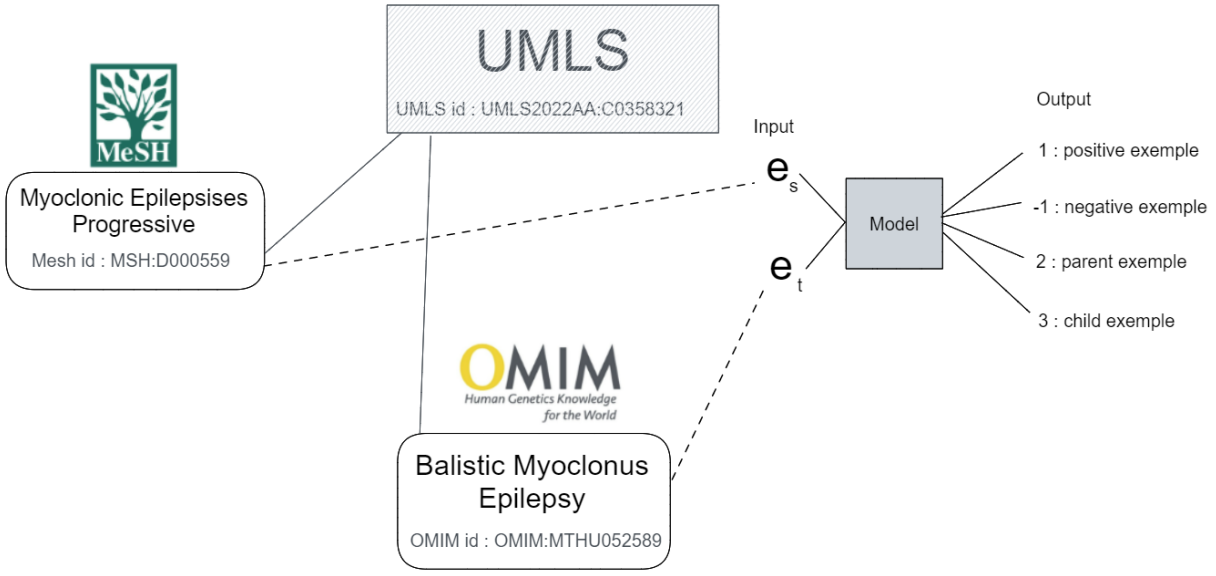

Fig. 3. Four Categories of Entity Relationships in the Alignment Process: Positive, Negative, Parent, and Child

- $l = 0$ represents negative examples, where entities s and t are not equivalent.
- $l = 2$ signifies a parent relationship between entities s and t, with the entity s being the parent of entity t.
- $l = 3$ denotes a child relationship between entities s and t, with the entity s being the child of entity t.

In order to identify positive matches for entities that do not have equivalent names, we employed the use of the UMLS ID. The UMLS ID is a unique identifier that is assigned to entities from various medical ontologies, such as the MeSH and the OMIM databases. For instance, two entities from different ontologies may have different names and unique identifiers, but they may have the same UMLS ID, indicating that they are referring to the same concept. By utilizing the UMLS ID, we were able to accurately match entities despite differences in naming conventions, thus providing a more robust and comprehensive record linkage.

4.2. Deriving parent-child relations

In order to derive parent-child labels for entities represented by $l = 2$ and $l = 3$ respectively, the code first retrieves a list of relation ids that are associated with a specific entity. These relation ids are used to identify the parent-child relationships between entities in the knowledge base. Then we check the type of each relation id and compare it to a predefined list of parent-child relation types (e.g. `part_of`, `subClassOf`)

For each relation id that matches a relation type in the `parent_ralations` list, the second entity id in the relation is considered the parent id of the current entity. Similarly, for each relation id that matches a relation type in `child_relations`, the second entity id in the relation is considered as the child id of the current entity.

4.3. Data Enrichment

Deriving comprehensive $e_{description}$ and $e_{context}$ from ontologies can pose a significant challenge. This study circumvents this problem by leveraging various resources, tools, LLMs, and APIs, with the objective of enhancing $e_{description}$ and $e_{context}$, emphasizing semantic accuracy and relevancy.

Our strategy employed two potent tools provided by Langchain: the GoogleSearchAPIWrapper [19][20] and the WikipediaAPIWrapper [19][21], both supplemented by GPT-3.5, acting as an LLM search agent. The

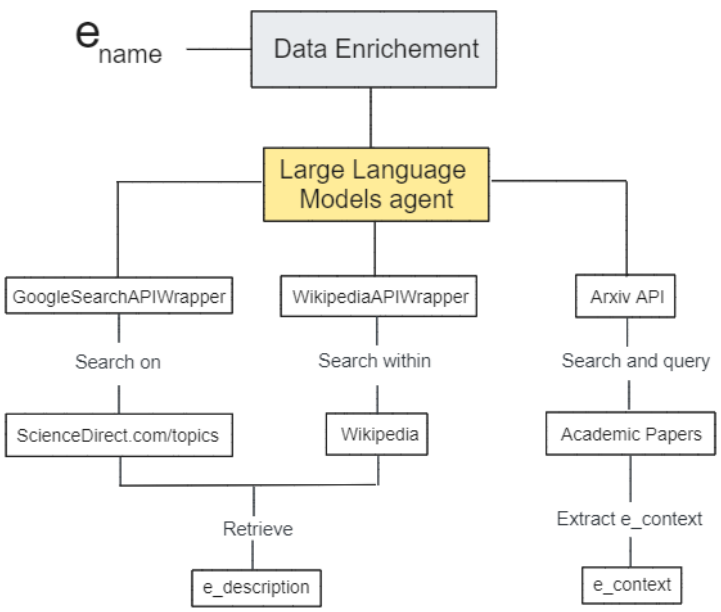

Fig. 4. Overview of the pipeline utilized in the entity name for deriving entity description and context, leveraging LLMs and various APIs.

GoogleSearchAPIWrapper was primarily used as a customized search engine, probing the extensive database of `ScienceDirect.com/topics` [22]. The latter encompasses over 363,000 topic pages and 5.8 million journal articles. This tool facilitated the retrieval of semantically nearest $e_{description}$ to the target e_{name}.

In the event that the initial search on ScienceDirect topics fails to yield an $e_{description}$ our e_{name}, we then enlist the capabilities of the WikipediaAPIWrapper as an additional recourse for augmenting $e_{description}$. This combination of the WikipediaAPIWrapper and an LLM model, specifically designed for conducting extensive searches within Wikipedia and it's effective in retrieving $e_{description}$ that are closely related to the e_{name}.

To ascertain the $e_{context}$ of e_{name}, a different strategy was adopted. We utilized the Arxiv API [23], powered by GPT-3.5 acting as the LLM agent, to extensively search and query academic papers, honing in on sentences where e_{name} were mentioned. This strategy yielded a comprehensive list of sentences detailing the usage of e_{name}, thus enriching our understanding of the e_{name}'s application within a scientific manuscript.

The entire pipeline of this process, from data gathering to context generation, is visualized in Figure 4 below.

4.4. Candidate's selection

Handling large ontologies presents significant computational challenges when attempting to evaluate all potential pairs of source and target entities for alignment purposes. For instance, the total number of potential entity pairs within our training ontologies reaches an astounding $10\hat{8}$. To efficiently decrease the number of candidates for consideration, we utilize embeddings of word tokens appearing in entity names and descriptions.

For each source entity, we first retrieve all target entities that have a high similarity score based on their word embeddings. We compute the similarity between word embeddings of source and target entities using cosine similarity. Given the set of shared word embeddings w(s+t) between a source and target entity, we calculate the similarity score for each pair as follows:

$$\text{sim}_{\text{total}} = \sum (w(s) + w(t)) \times \text{sim}(w(s), w(t)) \quad (1)$$

$$\text{where } \text{sim}(w(s), w(t)) = \cos(\theta) = \frac{w(s) \cdot w(t)}{||w(s)||\ ||w(t)||} \tag{2}$$

Here, $\cos(\theta)$ denotes the cosine similarity, $w(s)$ and $w(t)$ represent the word embeddings of the source and target entities, and $||w(s)||$ and $||w(t)||$ are their respective magnitudes. Word embeddings with higher similarity values indicate more relatedness between the entities.

We calculate the contextual similarity score for each target entity and select the top $K = 30$ target entities with the highest values for each source entity, resulting in $|Os| \times K$ candidate pairs. This refined candidate selection method accelerates the matching process by considerably reducing the number of potential pairs to analyze.

For each candidate pair (e_s, e_t), we compute a distinct set of 10 features, diverging from those traditionally used in ontology-matching literature. These features include fuzzy token similarity, lemmatized token similarity, semantic similarity using word embeddings, shared hypernyms, and an assortment of other boolean and probability values. Fuzzy token similarity, for example, evaluates the similarity between tokens using a string-matching technique, such as the Levenshtein distance:

$$\text{Fuzzy\_token\_similarity} = 1 - \left(\frac{\sum(\text{lev\_dist}(a,b) \text{ for } a \in A, b \in B)}{(|A| * |B|)} \right) \tag{3}$$

where A and B are sets of tokens for the source and target entities, respectively, and $\text{lev\_dist}(a, b)$ is the Levenshtein distance between tokens a and b. The Levenshtein distance measures the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform one string into another.

4.5. Derive negative examples

To create the negative examples, for each entity source, we selected multiple entity targets for which there are no name equivalences, first, the TF-IDF of the e_{name} and $e_{description}$ of the entity source is calculated. The mathematical formula for calculating the TF-IDF of e_{name} can be written as:

$$\text{tf}_{\text{name}}(es) = \text{tf}(es_{\text{name}}, es_{\text{description}}) \tag{4}$$

$$\text{idf}_{\text{name}}(es) = \log(\tfrac{N}{\text{df}(es_{\text{name}})}) \tag{5}$$

where $\text{tf}(es_{\text{name}}, es_{\text{description}})$ represents the term frequency of e_{name} in the $e_{description}$ of the source entity, N represents the total number of $e_{description}$ in the corpus, and $\text{df}(es_{\text{name}})$ represents the number of $e_{descriptions}$ where the e_{name} appears.

Next, the weights of all terms in every $e_{description}$ are determined using the TF-IDF values. Once the weights are determined, the process compares the vectors of weights to find similar descriptions. The mathematical formula for calculating the cosine similarity between the source and target entities is:

$$\text{cosine\_similarity}(es, et) = \frac{\text{tf}_{\text{name}}(es) \cdot \text{idf}_{\text{name}}(es) \cdot \text{tf}_{\text{description}}(et) \cdot \text{idf}_{\text{description}}(et)}{||\text{tf}_{\text{name}}(es) \cdot \text{idf}_{\text{name}}(es)|| \cdot ||\text{tf}_{\text{description}}(et) \cdot \text{idf}_{\text{description}}(et)||} \tag{6}$$

We classified the similarity results from least similar to very similar, and took a threshold for entity pairs that had a similarity of less than 0.5. These pairs were labeled as 0.

5. OntoMatcher

This part describes a supervised deep-learning approach for biomedical ontology alignment using OntoMatcher. This approach is based on representing each entity in an ontology as a vector using BioBERT and other techniques and then using these vectors to align entities across different ontologies as described in Preprocessing and inference step of Figure 2.

5.1. Entity name and alternative names embeddings

As schematized in Figure 5, First, the entity name is represented as a sequence of tokens, each of which is encoded using pre-trained BioBERT. To address the issue of out-of-vocabulary (OOV) words, which are common in the biomedical realm due to the emergence of new terms, a character-level 1D-CNN is employed. The 1D-CNN finds a numeric representation of words based on their character-level compositions, mimicking the human ability to understand word parts, such as the prefix "Epi-" in "Epidermis". This allows the model to generate more meaningful representations for OOV words, which would otherwise be assigned random vector values by BioBERT, potentially leading to confusion in the ontology matching model. The BioBERT and 1D-CNN representations are combined using a highway network, which adjusts their relative contributions based on the presence of a word in the BioBERT dictionary. The highway network consists of a transform gate and a carrier gate, where the transform gate is a sigmoid function ranging from 0 to 1, and the carrier gate is equal to 1 minus the transform gate. If a word exists in the BioBERT dictionary, the transform and carrier gates will have similar values, indicating that both representations should be given equal importance. If the word is an OOV, the transform gate will have a lower value, indicating that the 1D-CNN representation should be given more importance. After combining the BioBERT and 1D-CNN representations, a bi-directional long short-term memory (Bi-LSTM) network is used to capture the contextual meaning of words, which is essential for ontology alignment. The word representations generated by the previous steps do not take into account the context in which the words appear, leading to situations where homonyms like "tear" (watery eyes) and "tear" (rip apart) would be assigned the same vector representation. By considering the words both before and after a given word in the sequence, the Bi-LSTM can generate more accurate and meaningful representations for ontology alignment. The hidden layers at both ends of the Bi-LSTM are concatenated, and max pooling is applied to obtain the entity name vector, V_{name}, which represents the embedding of the entity name.

Let N be the entity name token sequence, with each token t_i encoded using pre-trained BioBERT embeddings. The output of the 1D-CNN is denoted as $C(N)$. The concatenation of the BioBERT embeddings and the 1D-CNN output is given by:

$$M = \text{concatenate}(\text{BioBERT}(N), C(N)) \tag{7}$$

The output of the highway network is denoted as $H(M)$, which is then fed into a Bi-LSTM network. Let L_f and L_b be the final hidden states of the forward and backward LSTMs, respectively:

$$V_{\text{name}} = \text{max\_pooling}(\text{concatenate}(L_f, L_b)) \tag{8}$$

This process is repeated for the $e_{alternatives}$ of the entity as depicted in Figure 6. Each alternative name is independently embedded using the same encoder used for the names, yielding a set of vectors. The $V_{\text{alternatives}}$ is the average vector embedding of the alternative names.

Given a set of $e_{alternatives}$ embeddings $A = \{a_1, a_2, \ldots, a_n\}$, the $V_{\text{alternatives}}$ is the average vector embedding of the $e_{alternatives}$:

$$V_{\text{alternatives}} = \frac{1}{n} * \sum_{i=1}^{n} a_i, \text{ where } i \in [1, n] \tag{9}$$

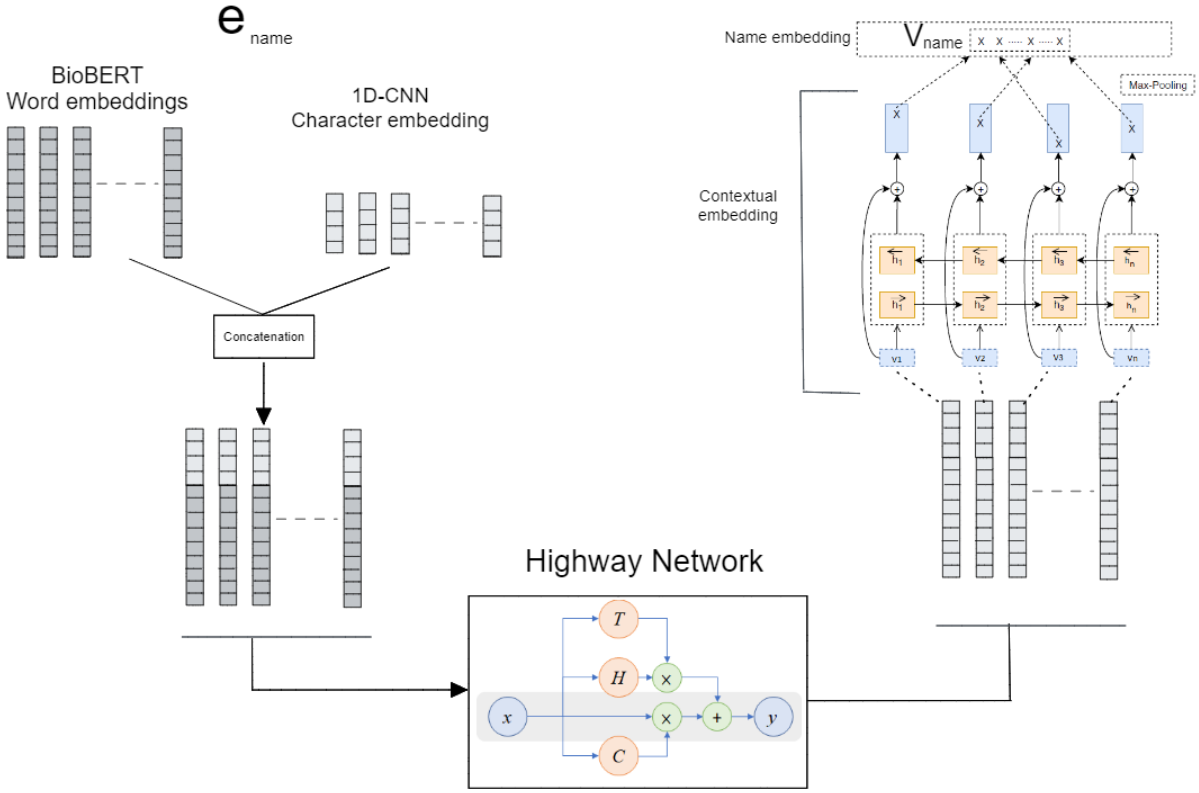

Fig. 5. Entity Name Embeddings Pipeline - Combining BioBERT, 1D-CNN, and Bi-LSTM for Enhanced Ontology Alignment

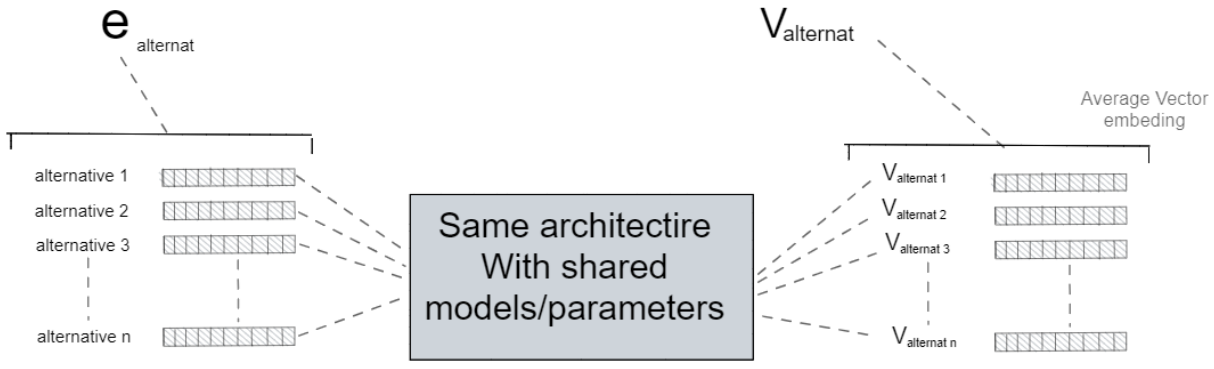

Fig. 6. Alternative Name Embeddings Process - Averaging Vectors for Enhanced Entity Representation

5.2. *Entity Description and context embeddings*

To represent the entity description as a vector, as described in Figure 7 we first take the entity description as a sequence of tokens, each encoded using pre-trained BioBERT embeddings. This sequence is then fed into a bi-directional LSTM network. The description vector is the output of max pooling applied to the concatenation of the

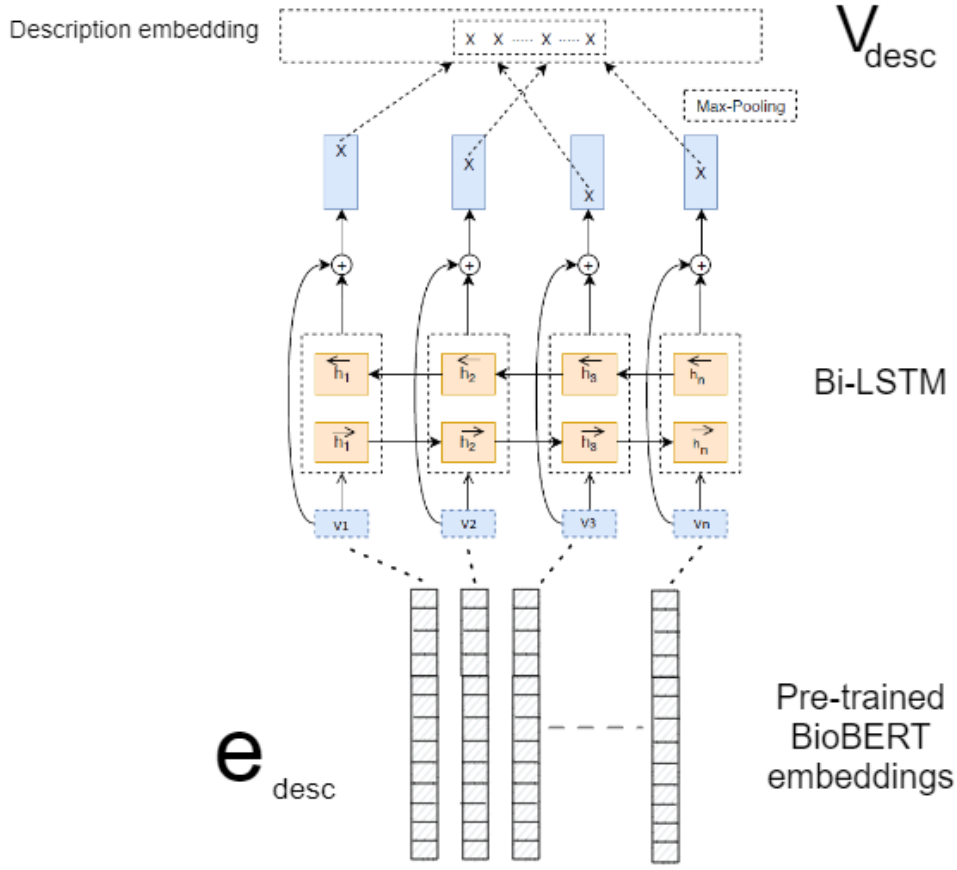

Fig. 7. Entity Description Vector Representation - Combining BioBERT and Bi-LSTM for Effective Description Embeddings

final hidden states in the forward and backward LSTMs.

Let D be the entity description token sequence, with each token t_i encoded using pre-trained BioBERT embeddings, and let H_f and H_b be the final hidden states of the forward and backward LSTMs, respectively:

$$\text{description\_vector} = \text{max\_pooling}(\text{concatenate}(H_f, H_b)) \qquad (10)$$

The same process is applied to the usage context of the entity as shown in Figure 8, with the resulting vector representing the average of all the v_contexts.

Given a set of context embeddings $C = \{c_1, c_2, \ldots, c_m\}$, the resulting vector representing the average of all the v_contexts:

$$V_{\text{contexts}} = \frac{1}{m} \sum c_i, \text{ where } i \in [1, m] \qquad (11)$$

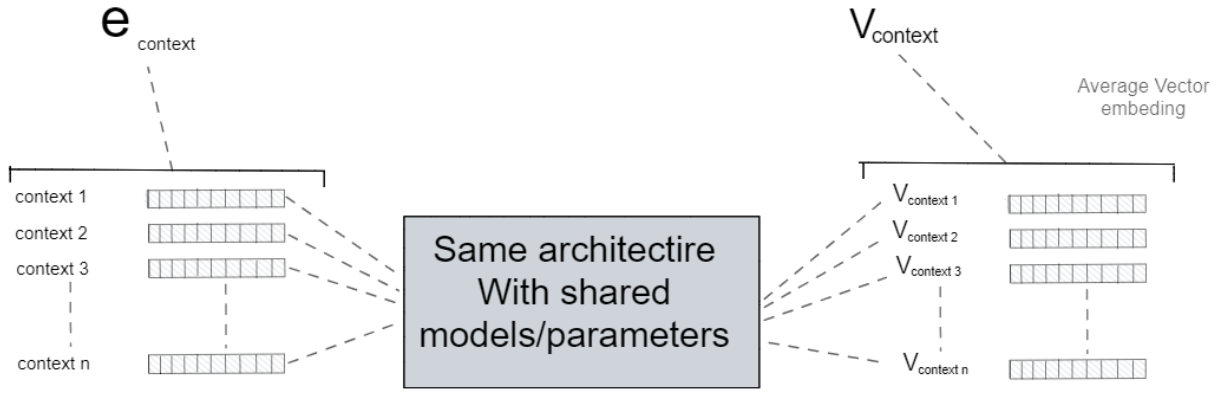

Fig. 8. Usage Context Embedding Process: Averaging Context Vectors for Comprehensive Entity Contexts Representation

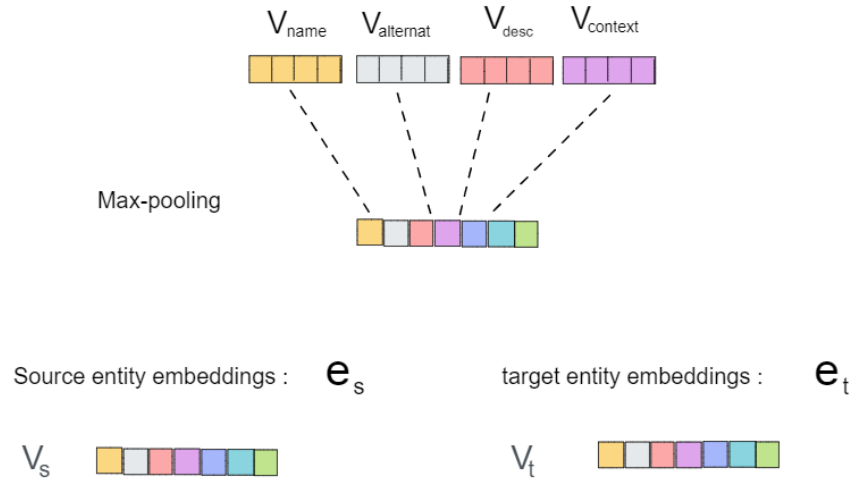

Fig. 9. Creating Entity Source and Target Vector Representation - Combining Name, Alternatives, Description, and Context for Holistic Entity Representation

5.3. Siamese network

The vector representation of e_{name}, $e_{alternatives}$, $e_{description}$, and $e_{context}$ are concatenated to create the entity embedding as shown in Figure 9. The source and target entity embeddings are the output of max pooling from all the vectors.

$$entity\_embedding = \max\_pooling(\text{concatenate}(V_{\text{name}}, V_{\text{alternatives}}, V_{\text{description}}, V_{\text{contexts}})) \quad (12)$$

The source and target vectors are then passed through a Siamese network [24], as illustrated in Figure 10. The network's objective is to determine the similarity between the two vectors. The Siamese network comprises two identical subnetworks with shared parameters, producing mirrored outputs. These subnetworks yield fully connected layers, and the Euclidean distance between them is calculated. Values closer to 1 indicate a higher similarity, while values closer to 0 indicate less similarity.

Let S and T be the source and target entity embeddings, and let F_s and F_t be the outputs of the Siamese subnetworks for the source and target embeddings, respectively.

$$F_s = \text{SiameseSubnet}(S) \qquad (13)$$

$$F_t = \text{SiameseSubnet}(T) \qquad (14)$$

$$\textit{binary\_relationship} = \text{SiameseNetwork}(F_s, F_t) \qquad (15)$$

Here, *binary_relationship* equals 1 if the entities are matching and 0 otherwise. This step allows us to initially classify entity pairs into matching and non-matching categories.

During training, the goal is to minimize a contrastive loss function that encourages the network to learn similar feature representations for matching entities and dissimilar representations for non-matching entities. The contrastive loss function is defined as:

$$L = \frac{1}{2N} \sum_{i=1}^{N} (1 - y_i) \cdot d_i^2 + y_i \cdot \max(0, m - d_i)^2 \qquad (16)$$

where N is the number of training pairs, y_i is a binary label indicating whether the entities in the i-th pair are matching (1) or not (0), d_i is the Euclidean distance between the feature representations for the i-th pair of entities ($d_i = ||F_{s_i} - F_{t_i}||_2$), and m is a margin that determines how far apart dissimilar entities should be in the feature space.

Next, the embeddings of matching pairs (F_s, F_t), that represent only relationship labels 1, 2, and 3 mentioned in §4.1, are used as input to an XGBoost [25] classifier to determine the type of relationship between the source and target entities.

Let $G(F_s, F_t)$ be the output of the XGBoost classifier for the embeddings F_s and F_t. The XGBoost classifier is trained on the embeddings generated by the Siamese network for the matching pairs $\{(F_{s_i}, F_{t_i}, r_1)\}$ for $i = 1, \ldots, n$, where F_{s_i} and F_{t_i} are the outputs of the Siamese subnetworks for the source and target embeddings of the i-th training example, and r_1 is the corresponding relationship label = 1. The hierarchical_relationship between the source and target entities is determined by the XGBoost classifier as:

$$\textit{hierarchical\_relationship} = \text{XGBoost}(F_s, F_t) \qquad (17)$$

where *hierarchical_relationship* is 2 if the source entity is the parent of the target entity, 3 if the source entity is the child of the target entity, and 1 if it is an exact match.

XGBoost is an ensemble learning algorithm that utilizes gradient boosting to optimize decision trees. The underlying principle behind XGBoost is to iteratively add weak learners (decision trees) to the model, minimizing the objective function given by:

$$\text{Obj}(\theta) = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \qquad (18)$$

where $\ell(y_i, \hat{y}_i)$ represents the loss function comparing the true label y_i and the predicted label $\hat{y}_i$, $\Omega(f_k)$ is a regularization term for each weak learner f_k, and K is the total number of weak learners. The objective function consists of a loss term that measures the similarity between the true and predicted labels and a regularization term that controls the complexity of the model to prevent overfitting.

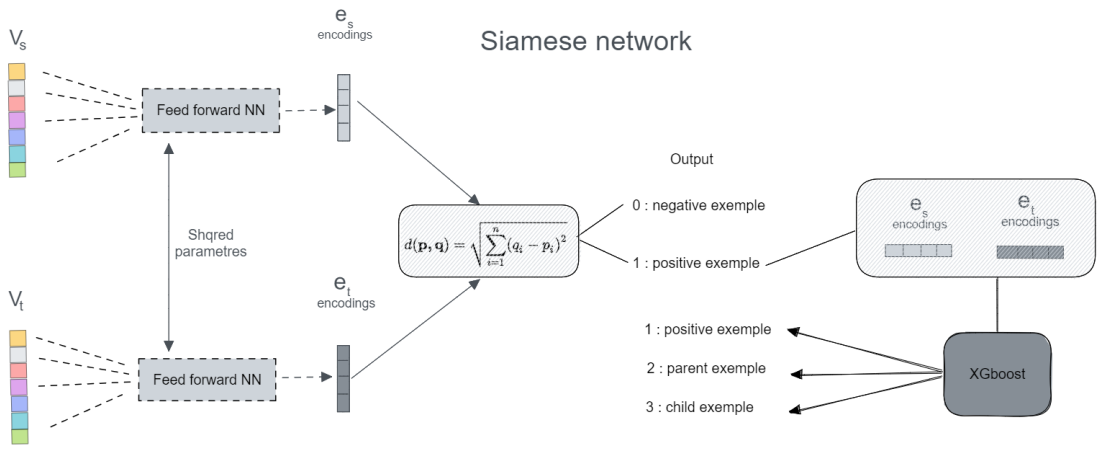

Fig. 10. Siamese Network for Entity Similarity - Assessing Semantic Equivalence of Source and Target Entity Embeddings

Algorithm 1 OntoMatcher Pipeline: Preprocessing, Embedding, and Siamese Network for Identifying Entity Alignments

Source ontology O_s
Target ontology O_t
A list of alignments between entities in O_s and O_t
procedure PREPROCESS ENTITIES IN O_s AND O_t:
    Extract entity attributes: e_{name}, $e_{alternatives}$, $e_{description}$, $e_{contexts}$
end procedure
for each entity e in $O_s \cup O_t$ **do**
    Embed e_{name} and $e_{alternatives}$:
    Tokenize e_{name} and $e_{alternatives}$ using pre-trained BioBERT
    Encode OOV words using 1D-CNN character embedding
    Combine BioBERT and 1D-CNN embeddings using highway network
    Pass combined embeddings through Bi-LSTM and apply max-pooling to obtain e_{name} vector V_{name}
    Repeat for $e_{alternatives}$ and obtain the average vector $V_{alternatives}$
    Embed $e_{description}$ and $e_{contexts}$:
    Tokenize $e_{description}$ and $e_{contexts}$ using pre-trained BioBERT
    Pass tokenized embeddings through Bi-LSTM and apply max-pooling to obtain $e_{description}$ vector $V_{description}$
    Repeat for $e_{contexts}$ and obtain the average vector $V_{contexts}$
    Aggregate vectors:
    Concatenate V_{name}, $V_{alternatives}$, $V_{description}$, $V_{contexts}$, and other relevant vectors to obtain the entity vector V_e
end for
for each pair of entities (e_s, e_t) in $O_s \times O_t$ **do**
    Pass V_{e_s} and V_{e_t} through a Siamese network
    Apply the XGBoost classifier to obtain a similarity score
end for
procedure DETERMINE ALIGNMENTS:
    Set a similarity threshold
    Report pairs of entities with similarity scores above the threshold as alignments
end procedure

Table 1
Performance comparison of OntoMatcher pipeline stages with different training vectors

Model (Name, Desc, Context)	Accuracy	Precision	Recall	F1 Score
OntoMatcher_Siamese_Name_Desc_Context	0.841	0.986	0.786	0.875
OntoMatcher_XGBoost_Name_Desc_Context	0.963	0.975	0.926	0.948
Model (Name, Desc)				
OntoMatcher_Siamese_Name_Desc	0.834	0.976	0.786	0.871
OntoMatcher_XGBoost_Name_Desc	0.958	0.946	0.906	0.925
Model (Name)				
OntoMatcher_Siamese_Name	0.813	0.936	0.791	0.857
OntoMatcher_XGBoost_Name	0.942	0.962	0.772	0.823

6. Experiments

In our experiments, we engaged with the OntoMatcher pipeline, which comprises two main stages. The first stage uses a Siamese Network (OntoMatcher_Siamese), and the second stage employs an XGBoost classifier (OntoMatcher_XGBoost). We explored several variations of the OntoMatcher pipeline, incrementally enriching the entity pair attributes used to train these stages:

- OntoMatcher_SN_Name: OntoMatcher with Siamese network and input vectors containing only the vector representation of entity names.
- OntoMatcher_SN_Name_Desc: OntoMatcher with Siamese network and input vectors containing vector representations of entity names and descriptions.
- OntoMatcher_SN_Name_Desc_Context: OntoMatcher with Siamese network and input vectors containing vector representations of entity names, descriptions, and contexts.

The data used in these experiments comprised the training section of the UMLS-derived labeled data to train our model. For evaluation, we used the test portion of the UMLS-derived data as well as the Ontology Alignment Evaluation Initiative (OAEI) biomedical subtask SNOMED-NCI task. The UMLS test set includes 7479 positive and negative mappings, and the OAEI reference alignments include 18844 equivalent mappings.

The performance of each model was evaluated in terms of F1 score, accuracy, and recall. The results are summarized in Table 1.

Analysis:

- Both stages of the OntoMatcher pipeline OntoMatcher_Siamese and OntoMatcher_XGBoost—demonstrate a pattern of improved performance as entity vectors become richer. This indicates that supplementing entity vectors with not just names, but also descriptions and contexts, can significantly boost the matching outcomes.
- Interestingly, enriching the entity vectors with additional information retrieved from external resources such as Wikipedia, ScienceDirect Topics, and various research papers using GPT-3.5 as an LLM agent, resulted in a noticeable improvement in the F1-score. This suggests that leveraging AI capabilities to incorporate external knowledge into entity representations could be a promising strategy for enhancing ontology matching performance.
- Despite the enhancements achieved with enrichment, it's important to mention that both stages of the OntoMatcher pipeline deliver satisfactory performance even when the entity vectors include only names. This reflects the robustness and adaptability of these stages, demonstrating their ability to handle varying conditions of input data.

7. Conclusion

Biomedical ontologies, while invaluable in the field, face certain limitations, such as the out-of-vocabulary issue due to the continuous emergence of new compound words and terms in the ever-evolving biomedical domain. With this in mind, our study proposes a novel supervised deep learning approach for aligning biomedical ontologies by leveraging the strengths of BioBERT, highway networks, and bi-directional long short-term memory (Bi-LSTM) models. The method effectively captures character-level and contextual information of entities and incorporates entity descriptions and context embeddings to enhance alignment accuracy. Our approach demonstrates significant performance improvements, achieving an F1 score of 0.87 for match/not match classifications and 0.94 for level classifications, which surpasses several baselines on benchmark datasets.

Despite our encouraging results, we acknowledge certain limitations. Primarily, we rely on pre-trained BioBERT embeddings, which might not encompass all current biomedical terms. Future work could involve updating the BioBERT model with newer, domain-specific data, or using advanced models like Google's Med-PALM 2. Our system's scalability is also a concern as it was designed and tested on a single laptop, limiting the size of processable ontologies. Future iterations could focus on optimization for distributed or cloud-based computing.

Moreover, while our approach effectively leverages names, descriptions, and context of entities, it could potentially benefit from incorporating additional forms of information. For example, it may be worth exploring the utilization of graph structures and graph embeddings. Graph structures could provide a more holistic view of the relationships and interactions between different entities, and graph embeddings could capture complex relational patterns in a dense vector space, potentially offering richer and more nuanced inputs for the alignment process. Finally, integrating more intricate relationships between entities, beyond exact matches or parent-child relationships, could enhance alignment complexity. By addressing these limitations, we expect future iterations of our work to further improve biomedical ontology alignment.

The scientific contributions of our work are multifaceted. First, we have introduced a novel deep learning approach that efficiently captures both character-level and contextual information of biomedical entities. Second, we have demonstrated the effectiveness of our method in achieving high alignment accuracy, outperforming existing baseline methods. Lastly, our study highlights the potential of leveraging advanced natural language processing techniques, such as BioBERT, to address complex challenges in the biomedical domain. Up to this point, our approach holds promise in addressing the critical need for accurate and efficient alignment of biomedical ontologies, paving the way for improved data integration and interoperability across various biomedical systems. We believe that our work contributes valuable insights to the field of ontology alignment and has the potential to inspire further advancements in this important area of research.

Acknowledgements

I would like to express my sincere gratitude to my parents for their constant support and encouragement throughout my research. I am also grateful to Dr. Amina Samih and Pr. Abdelhadi Fennan for their valuable guidance, expertise, and mentorship, which significantly contributed to the success of this work.

References

[1] Euzenat, J., Shvaiko, P. (2013). *Ontology Matching*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-38721-0

[2] Faria, D., Pesquita, C., Mott, I., Martins, C., Couto, F. M., Cruz, I. F. (2018). Tackling the challenges of matching biomedical ontologies. Journal of Biomedical Semantics, 9, Article number: 4. https://doi.org/10.1186/s13326-017-0170-9

[3] U.S. National Library of Medicine. (n.d.). *Medical Subject Headings (MeSH®)*. Retrieved from https://www.nlm.nih.gov/mesh/

[4] SNOMED International. (n.d.). *SNOMED CT*. Retrieved from https://www.nlm.nih.gov/healthit/snomedct/index.html.

[5] National Institutes of Health. (n.d.). *National Cancer Institute (NCI)*. Retrieved from https://www.nih.gov/about-nih/what-we-do/nih-almanac/national-cancer-institute-nci.

[6] Otero-Cerdeira, L., Rodríguez-Martínez, F. J., Gómez-Rodríguez, A. (2015). Ontology Matching: A Literature Review. *Expert Systems with Applications*, 42(2), 949-971. http://dx.doi.org/10.1016/j.eswa.2014.08.032

[7] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., Kang, J. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, 36(4), 1234-1240. https://doi.org/10.1093/bioinformatics/btz682

[8] Qiao, Y., Wang, Y., Ma, C., Yang, J. (2020). Short-term traffic flow prediction based on 1DCNN-LSTM neural network structure. *Modern Physics Letters B*, 34(28), 2150042. https://doi.org/10.1142/S0217984921500421

[9] Lochter, J. V., Silva, R. M., Almeida, T. A. (2022). Multi-level out-of-vocabulary words handling approach. *Knowledge-Based Systems*, 251, 108911. https://doi.org/10.1016/j.knosys.2022.108911

[10] Héon, M., Aubut, J., Gaudreau, S. (2017). UMLS-OWL: an OWL 2 Translation of the Unified Medical Language System (UMLS®) Semantic-Network and Metathesarus for Publishing in the Semantic Web. In *International Workshop on the Semantic Web, 2017*. Retrieved from https://iswc2017.semanticweb.org/wp-content/uploads/papers/PostersDemos/paper546.pdf

[11] Saake, G., Sattler, K.-U., Conrad, S. (2005). Rule-based schema matching for ontology-based mediators. *Journal of Applied Logic*, Volume 3, Issue 2, Pages 253-270. https://doi.org/10.1016/j.jal.2004.07.021

[12] Fernández, S., Velasco, J. R., Marsa-Maestre, I., Lopez-Carmona, M. A. (2012). FuzzyAlign - A Fuzzy Method for Ontology Alignment. In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development (KEOD)* (pp. 98-107), Barcelona, Spain. https://doi.org/10.5220/0004139500980107

[13] Lucy Lu Wang, Chandra Bhagavatula, Mark Neumann, Kyle Lo, Chris Wilhelm, Waleed Ammar. (2018). Ontology alignment in the biomedical domain using entity definitions and context. In *Proceedings of the BioNLP 2018 workshop* (pp. 47–55), Melbourne, Australia. Association for Computational Linguistics. DOI:10.18653/v1/W18-2306

[14] Wu, J., Lv, J., Guo, H., Ma, S. (2020). DAEOM: A Deep Attentional Embedding Approach for Biomedical Ontology Matching. *Applied Sciences*, 10(21), 7909. https://doi.org/10.3390/app10217909

[15] National Library of Medicine. (n.d.). *About NLM*. Retrieved from https://www.nlm.nih.gov/about/index.html

[16] U.S. National Library of Medicine. (2022). *UMLS Metathesaurus: Statistics*. Retrieved from https://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/statistics/index.html

[17] Demner-Fushman, D., Mork, J. G., Shooshan, S. E., Aronson, A. R. (2010). UMLS content views appropriate for NLP processing of the biomedical literature vs. clinical text. *Journal of Biomedical Informatics*, 43(4), 587-594. https://doi.org/10.1016/j.jbi.2010.02.005

[18] U.S. National Library of Medicine. (n.d.). *UMLS Source Release Documentation*. Retrieved from https://www.nlm.nih.gov/research/umls/sourcereleasedocs/index.html

[19] LangChain Python Library (Version 0.0.181). (n.d.). *Agents.Tools Module*. Retrieved from LangChain website: https://python.langchain.com/en/latest/modules/agents/tools.html

[20] Google Developers. (n.d.). *Custom Search JSON API: Overview*. Retrieved from Google Developers website: https://developers.google.com/custom-search/v1/overview

[21] Goldsmith, R. (n.d.). *Wikipedia: A Pythonic wrapper for the Wikipedia API*. Retrieved from GitHub: https://github.com/goldsmith/Wikipedia

[22] Elsevier. (n.d.). *ScienceDirect Topics: Discover*. Retrieved from https://www.sciencedirect.com/topics

[23] Schwab, L. (n.d.). *arxiv.py: Python wrapper for the arXiv API*. Retrieved from GitHub: https://github.com/lukasschwab/arxiv.py

[24] Xue, X., Jiang, C., Zhu, H. (2021). Matching Ontologies Through Siamese Neural Network. Mobile Multimedia Communications. MobiMedia 2021. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 394. Springer, Cham. https://doi.org/10.1007/978-3-030-89814-4_52

[25] Chen, T., Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794). ACM. https://doi.org/10.1145/2939672.2939785