JSCDM

JOURNAL OF SOFT COMPUTING AND DATA MINING
e-ISSN: 2716-621X

Vol. 7 No. 2 (2026) 32-44
https://publisher.uthm.edu.my/ojs/index.php/jscdm

Biomedical Knowledge Graph Construction Based on Embedding Models and Ontologies

Salah Edine Ech-Chorfi[1]*, Elmoukhtar Zemmouri[1]

[1] *Department of Mathematics and Computer Science,*
National School of Arts and Crafts, Moulay Ismail University, Meknes, 50000, MOROCCO

*Corresponding Author: sa.echchorfi@edu.umi.ac.ma
DOI: https://doi.org/10.30880/jscdm.2026.07.02.003

Article Info

Received: 15 August 2025
Accepted: 5 March 2026
Available online: 30 June 2026

Keywords

Knowledge graph extraction, relation extraction, ontology, graph embedding

Abstract

Many machine learning and linguistic tools assist biomedical practitioners and researchers in fulfilling their daily missions. Text mining techniques, such as Natural Language Processing (NLP), help process large amounts of biomedical data and perform tasks such as information retrieval, question answering, and document recommendation. Linguistic resources, such as ontologies and terminologies, provide a standardized knowledge reference across all biomedical subdomains to resolve ambiguities and unify the semantic biomedical concepts. This paper presents a series of experiments investigating new approaches to training Named Entity Recognition (NER), Entity Linking (EL), and Relation Extraction (RE) models using standard ontologies without relying on annotated benchmark datasets. We aim to improve the trainability and generalization of state-of-the-art tools while preserving decent performance across tasks. We combine BioBERT as a word embedding model, TuckER as a graph embedding model, and the MONDO ontology as a source of standard training data to develop separate NER, EL, and RE components. The best performances were obtained for the RE task with 0.75, 0.94, 0.84, and 1.50 in hits@1, hits@3, MMR, and mean rank metrics, respectively. Lastly, we apply our work in an ontology enrichment application scenario by integrating the newly conceived RE model into a KGE pipeline to extract ontology-aligned relations from text.

1. Introduction

Biomedical Knowledge Graph Extraction (KGE) is a standard Natural Language Processing (NLP) operation that transforms natural language into a knowledge graph. It requires preprocessing the input text and applying Named Entity Recognition to extract biomedical entities that will constitute the nodes of the output knowledge graph (KG). Entity Linking is an optional phase that follows, in which the extracted entities are linked to standard references to standardize and disambiguate the extracted knowledge. Finally, the Relation Extraction step retrieves the relations that connect the extracted entities, forming the edges of the output KG. At the end of the KGE pipeline, we return a KG with nodes representing entities and edges representing relations. Fig. 1 depicts the base process of KGE from natural language text.

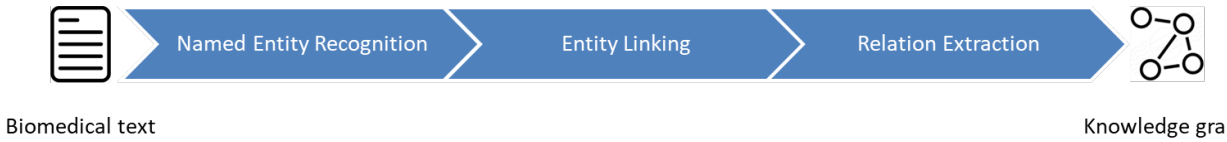

Fig. 1 *A KGE pipeline containing the 3 main steps: NER, EL, and RE. The pipeline processes biomedical natural text and outputs a structured KG*

There are several state-of-the-art tools in the literature that tackle NER, EL, and RE tasks. Their quality and performance are crucial to the overall performance of the KGE pipeline. Supervised methods have proven more robust than semi-supervised and unsupervised methods, with transformer-based deep learning models leading the way. Most existing models are usually finetuned in a supervised manner to perform different tasks such as NER, EL, and RE.

This supervised learning, which yields important state-of-the-art results, also poses a major limitation to the application of these models. The limitation resides in the dependency of the models on training data, both in terms of quantity and quality, as these datasets require a large number of diverse and richly labeled data (NCBI-Disease [1], Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA) [2], Chemical Compound and Drug Name Recognition (CHEMDNER) [3], BioCreative V Chemical-Disease Relation (BC5CDR) [4], and BioCreative VI Chemical-Protein (CHEMPROT)[5]). This may not always be the case, as these datasets are not abundantly available and are hard to conceive from scratch. In the case of NER and RE, the training data also limits the models' outputs, as an NER model can only extract and classify entities into the predetermined types in the dataset, while an RE model can only extract the relations seen in the training dataset. As a result, supervised NER, EL, and RE models cannot be applied beyond their training datasets, limiting their ability to handle diverse text inputs across biomedical subfields.

In this work, we address the models' dependency on training entity types and relations by training on standard, rather than dataset-specific, entity types and relations. Our approach is to provide a more inclusive and representative source of knowledge for the models to exploit. We rely on biomedical ontologies as an external source of standard knowledge. Unlike NER and RE datasets, state-of-the-art ontologies contain the most comprehensive knowledge in each biomedical subfield. However, this form of knowledge is not compatible with the training or application of transformer-based models, as it provides structured, context-free data in the form of concepts and their relations, whereas transformer-based models rely on natural language text to produce accurate, context-aware representations. We use state-of-the-art word embedding models to encode textual data, and graph embedding models to encode the graph-based structure of the ontology.

In summary, we perform a set of experiments to explore combining the advantages of transformer-based word embedding models for text processing, ontologies as a source of domain-specific standard knowledge, and graph embedding models for representing the graph aspects of ontologies. We experimented with different pipeline configurations for each task: NER, EL, and RE. Our aim is to analyse the contribution of domain ontologies in the training of the different steps of the KGE pipeline.

In Section 2, we outline the related tools used in this work, followed by a detailed explanation of the experimental work and the proposed RE method in Section 3. We present the results and analyse the performance and challenges of the discussed theories in Section 4. Finally, in Section 5, we present an application scenario of the proposed RE model in a KGE pipeline.

2. Background Works

2.1 Word Embedding Models

Word embedding models are text mining tools that represent words as dense, fixed-length vectors, called word embeddings. These representations capture the syntactic and semantic similarities between words, where similar words are closer together in distance or aligned in angle. Early models were Count-based word embedding models. This type of model (Latent Semantic Analysis (LSA) [6], Global Vectors for Word Representation (GloVe) [7]) relies on statistical calculations of word co-occurrence within a text corpus. Neural-based models are another type of word embedding model that leverages shallow neural networks (e.g., Word2Vec [8], FastText [9]) to generate word embeddings by training to bring vectors of similar words closer together based on their local context. The training process is usually unsupervised and applied over large datasets of general or domain-specific natural language text. Deep Neural-based word embedding models follow the same principles as simple neural-based models, but with deeper, multi-layered architectures that provide greater advantages in word representation by supporting long-range context and handling variations in word meanings across natural-language contexts. These models often rely on Recurrent Neural Networks (Long Short-Term Memory (LSTM)

[10]) or Transformers (Bidirectional Encoder Representations from Transformers (BERT) [11], Generative Pre-Training Transformers (GPTs) [12]). RNN-based models process the text word by word in a sequential manner, relying on a hidden state to store information about the previous work to use it in the next calculation. They also need forward and backward passes over the word sequence to capture left- and right-context for all words. On the other hand, transformer-based models rely on the transformer architecture introduced by [13] in 2017, leveraging self-attention and multi-head attention mechanisms, to encode all words or tokens in a sequence at once while taking into consideration their position (order) and their relative dependencies regardless of their distance, which ensures the support of global context in the calculations.

2.2 Graphs and Ontologies

Graphs are data structures used to store information in a structured and interconnected manner. The key components of a graph are nodes and edges, forming a triple (node X, edge, node Y). The nodes represent entities, while the edges represent relations between pairs of nodes. There are several types of graphs. To cite a few, an undirected graph is a graph where edges are symmetrical, while the edges of a directed graph are only true in one direction. For an undirected graph UG(V,E) and A,B ∈ V and e ∈ E where V is the set of nodes and E is the set of edges, (A,e,B) ∈ UG implies that (B,e,A) ∈ UG, while for a directed graph DG(V,E) and A,B ∈ V and e ∈ E where V is the set of nodes and E is the set of edges, (A,e,B) ∈ DG does not necessarily mean that (B,e,A) ∈ DG. A weighted graph assigns weights or costs to each edge to model the quantity or the importance of the relation between a node pair. Labelled graphs provide labelled nodes and edges. Each graph type serves a purpose depending on the use case; weighted graphs are used for shortest-path problems, and undirected graphs can accurately model social networks. In this work, we use knowledge graphs. KGs are labelled, directed graphs that model real-life entities using nodes (e.g., people, locations, diseases, proteins) and the relations between them using edges (e.g., located in, parent of, negatively affects, caused by). Some of the most relevant resources for KGs are ontologies.

Formal ontologies are considered a standard, structured representation of knowledge within a given domain. They model facts using key components such as classes to represent concepts and entities, attributes to enrich the classes with more information (e.g., label, definition, and synonyms), relations to express the link and interaction between the classes, and axioms to define the rules and constraints holding the semantic and scientific integrity of the ontology. Researchers have developed various state-of-the-art ontologies for multiple biomedical subfields. The Gene Ontology (GO) [26] specializes in genes, biological systems, and molecular functions. The Monarch Disease Ontology (MONDO) [15] and Human Disease Ontology (DIOD) [15] both focus on diseases, genotype, and phenotype concepts. The choice of which ontologies to use depends on the field of application of the proposed method, which can vary across users.

2.3 Graph Embedding Models

Similar to word embedding models that return vector representations of words, graph embedding models (graph embedding model) return vector representations of graph nodes, edges, or subgraphs. They encode structural data and graph features of nodes and their relationships into dense vectors that can be further exploited by machine learning algorithms. Graph embedding models are widely used for tasks such as node classification, link prediction, clustering, and recommendation. The core idea of graph embedding models is to learn node and edge embeddings during training to maximize the scores of true triples relative to false ones using a specific scoring function. While the core idea is the same across most state-of-the-art graph embedding models, the scoring function and its calculation are the key differences between models. The choice of a suitable scoring function and, subsequently, an embedding model is based on criteria such as the graph's structure and size, the complexity and nature of the relations, and the computational cost of the model itself. Table 1 lists state-of-the-art graph embedding models, their scoring functions, advantages, and shortcomings.

Table 1 *Comparison of different state-of-the-art graph embedding models, highlighting their core difference, which is the triples scoring function, along with their advantages and challenges. The advantages and shortcomings serve as criteria for determining which model is suitable for a given graph embedding task*

Model	Scoring function	Advantages	Shortcomings
TransE [17]	$f(h, r, t) = -\|\|h + r - t\|\|_2$	- Interpretable - Lightweight	- Inability to model 1-to-N, N-to-N, reflective, and symmetric relations
TransR [18]	$(h, r, t) = -\|\|Mr\ h + r - Mr\ t\|\|_2$	- Separates entity and relation space - Ability to model complex relations	- Important memory use - Slow training

Model	Scoring function	Advantages	Shortcomings
RotatE [19]	$(h, r, t) = -\|\|h \circ r - t\|\|$	- Ability to model symmetric, asymmetric, inverse, and composed relations	- Inability to model N-1 relations
ComplEx [20]	$(h, r, t) = \Re(\langle h, r, \bar{t} \rangle)$	- Ability to model symmetric, asymmetric, inverse, and composed relation	- Complex computation - Complex-valued vectors instead on real -valued vectors
TuckER [21]	$(h, r, t) = W \times_1 h \times_2 r \times_3 t$	- Ability to model symmetric, asymmetric, inverse, and antisymmetric relations	- Great computational cost

In all functions, the symbols h, r, and t represent the vectors of the head entity, relation, and tail entity, respectively. The ||.||2 symbol denotes the Euclidean norm that measures the distance between 2 vectors. In TransR, the Mr symbol is the relation-specific projection matrix. In RotatE, the ∘ means element-wise vector multiplication. The $\Re$ function used in ComplEx retrieves the real part of the complex vector. For TuckER, W and ×n denote the core tensor and mode-n tensor product, respectively.

2.4 Ontology-Based Information Retrieval

Many researchers have shown interest in exploiting ontologies to improve information retrieval (IR) techniques such as NER, EL, and RE. Several state-of-the-art works presented different approaches to integrating ontologies and KGs across various parts of IR models. We go over some of these works to emphasize the different methods used for leveraging the ontology's standard knowledge.

Self-aligning pretrained BERT (SapBERT) [22] trains BERT [11] on UMLS [23] entities to learn how to represent similar entities. It trains on a set of triples containing an entity, its hard positive (shares the same UMLS identifier as the entity), and its hard negative (has a different UMLS identifier than the entity), and tries to increase the cosine similarity between the entity and its positive embeddings while decreasing it for the entity and its negative. The model acquires the awareness of synonyms and learns to represent similar words closer together.

On the other hand, the Knowledge Embedding and Pre-trained LanguagE Representation model (KEPLER) [24] uses joint learning on natural language and a set of triples to learn entity embeddings. It trains a pretrained language model (PLM) with 2 objectives: masked language modelling (MLM) for predicting the masked token in a sentence, and knowledge embedding (KE) for learning embeddings of entity descriptions. This method allows the model to capture factual knowledge from the KG in addition to semantic and syntactic features from natural text.

Lastly, the ontology-enhanced joint entity and relation extraction (OntoRE) [25] model infuses knowledge from an ontology into a multi-gate encoder (MGE) architecture to improve NER and RE performance. It consists of serializing the ontology elements using a breadth-first search (BFS), transforming each ontology triple into a sentence with the head label, head description, relation, tail label, and tail description. BFS returns a text representation of all triples separated by a special token. This ontology-derived text is passed through a frozen BERT [11] model to obtain its numerical representation, and then through a second transformer encoder to produce an encoded ontology. This result is integrated with the sentence representation of the MGE's encoder before passing to the MGE layer. The ontology-infused sentence representation can then be used for downstream tasks.

3. Experiments and Proposed Method

In this work, the transformers-based model BERT [11] is chosen as the base word embedding model across all experiments, for its training efficiency, scalability, and state-of-the-art performance. Moreover, we specifically use the BERT variant bioBERT [26], pre-trained on 1M PubMed articles, for this paper, as our focus is the biomedical domain. This pre-training on field-specific data ensures that the model's encoder is already trained to capture the semantic and relational aspects of biomedical words and concepts, resulting in better, more accurate embedding representations.

In the same context, MONDO ontology [15] is used as our source of standard biomedical knowledge. MONDO is a well-maintained, up-to-date, and consistent ontology from the OBO Foundry containing over 37000 concepts and interfaced with other OBO Foundry ontologies such as GO [14], CHEBI [27], PATO [28], CL [29], and RO that help enrich its knowledge and provide meaningful interactions between diseases and other entity types like cells and genes. In application contexts, any ontology can serve as a knowledge base without any constraints. We pre-

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 32-44

process the ontology and transform it from its standard RDF format to a set of triples (subject, predicate, object), also referred to as (head, relation, tail), for further processing. The extraction from MONDO [15] resulted in a set of 46313 triples, divided between training and testing as needed, 37912 unique classes, and 174 unique relations.

Relations in the MONDO [15] triples are considered complex as they contain different kinds of relations, such as symmetric (e.g., connected to, attached to), asymmetric (e.g., is a, develops into), and inverse (e.g., has part, part of), which are hard to model using lightweight geometric graph embedding models. For this purpose, we use the TuckER [21] model to learn and generate graph embeddings for entities and relations in the MONDO [15] triples.

Our goal is to develop standalone models that perform NER, EL, and RE tasks using standard knowledge (entity types and relations) without relying on benchmark datasets. To this end, we perform a series of experiments using the tools detailed in Section 2 to design an NER, EL, or RE component with state-of-the-art performance, while retaining the core advantage of minimizing its dependence on pre-designed datasets. We present in detail the first method for NER, followed by a second method for EL, and, lastly, the method for RE.

3.1 Named Entity Recognition

In this experiment, the bioBERT [26] model is finetuned on the standard NER task using MONDO [15] data. This operation requires a set of entity-annotated sentences. To annotate MONDO entities, each entity is assigned a label that represents its type. The labels are the classes assigned to each node based on its designated root node in the hierarchy, and nodes linked to a labelled node via the relation "is a" are automatically labelled with the same class. Fig. 2 depicts an example of the entity "acute chest syndrome", whose label is retrieved from its highest root node in the hierarchy "disease".

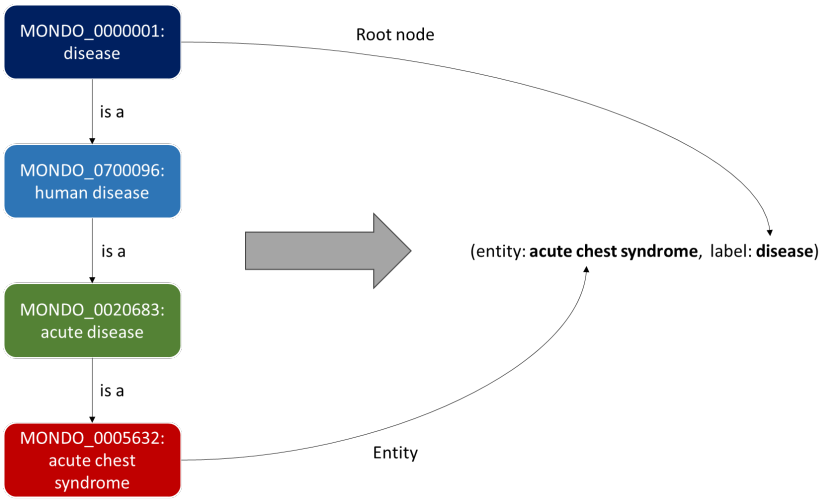

Fig. 2 *Label assignment system based on the hierarchy of entities in the ontology*

This test case focuses on the 4 nodes cited in Table 2 as the root nodes and annotates all child nodes accordingly based on the chain of "is a" relations. This method is not limited to the classes chosen in this work and applies to all ontology classes, depending on the application context.

Table 2 *Summary of the 4 classes picked for this work, along with their respective node IDs and labels in the MONDO tree*

Node ID	Node label	Class
MONDO:0000001	disease	Disease
GO:0008150	biological_process	Biological process
MONDO:0021178	injury	Injury
CL:0000000	cell	Cell

After entity typing, 2 NER datasets of annotated sentences are synthesized based on MONDO triples [15]. For the first dataset, the subject, predicate, and object of each triple are concatenated to produce simple and short sentences with labelled entities. For the second synthetic data, the triples are passed through a large language model (LLM) to augment the sentences with natural language while preserving the meaning of the original triples and the labelled mentions. Examples of synthesized sentences are listed in Table 3. The newly conceived datasets

are split into 46113 sentences for NER training and 200 sentences for testing. In summary, this approach allows us to generate a dynamic NER dataset where the sentences are based on true factual triples from a standard biomedical resource, and the classes are dynamically set based on the chosen root nodes.

Table 3 *Examples of synthesized sentences for the MONDO triples. Concatenated sentences are the result of concatenating the subject, predicate, and object of the triples. While LLM augmented sentences are generated using an off-the-shelf LLM based on the triples*

Triple	Concatenated sentence	LLM augmented sentence
(NEK9, is a, gene)	NEK9 is a gene	NEK9 can be described as a kind of gene
(lymphedema-distichiasis syndrome, disease has feature, Distichiasis)	lymphedema-distichiasis syndrome disease has feature Distichiasis	A characteristic feature of lymphedema-distichiasis syndrome is the presence of Distichiasis
(lymphedema-distichiasis syndrome, has material basis in germline mutation in, FOXC2)	lymphedema-distichiasis syndrome has material basis in germline mutation in FOXC2	A germline mutation in FOXC2 is the genetic foundation for lymphedema-distichiasis syndrome

BioBERT [26] is finetuned on the 2 versions of the synthetic datasets for 15 epochs using the same hyperparameters featured in BioBERT's original paper. A summary of the NER component innerworkings is depicted in Fig. 3.

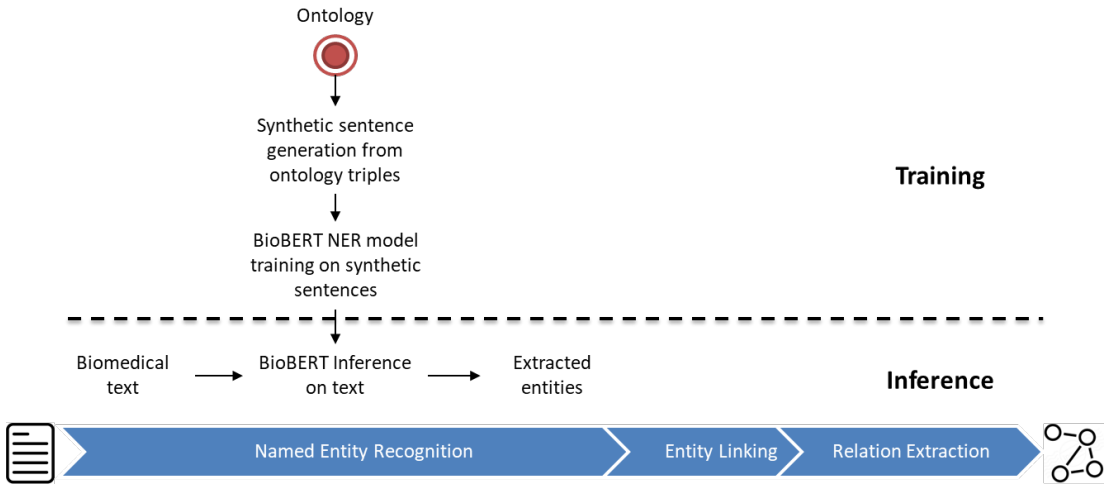

Fig. 3 *The new process of the NER component, requiring training on ontology data once, is integrated in the original KGE pipeline in the inference phase*

3.2 Entity Linking

The second setup tackles EL and exploits ontology graph embeddings to link newly encountered entities to their corresponding standard concepts in the ontology. The pipeline is divided into a training and inference phase. In the training phase, we generate embeddings for all concepts in the ontology using the bioBERT [26] encoder and the graph embedding model Tucker [21], which is trained on the ontology triples. The first embeddings, represented in the bioBERT [26] space, hold the knowledge incubated in bioBERT [26] during its pre-training on biomedical text, while the second embeddings, represented in the Tucker [21] space, contain the knowledge of hierarchy and interconnection of the concepts within the ontology. This approach involves training a projection model to map the embeddings from both sources into a shared space, minimizing the distance between the projected vectors for each embedding pair. As a result, the model learns to project the bioBERT [26] and TuckER [21] embeddings of a given ontology concept into a shared space, where the new projected vectors are closer together than those from a different concept. In the inference phase, we encode the extracted mentions using the same pre-trained bioBERT model [26], and then feed the bioBERT [26] embeddings to the projector, which projects them into the shared space. Next, the distance between each new projected vector and the Tucker [21] embeddings' projections is calculated. Lastly, the projector returns a ranked list of ontology concept candidates

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 32-44

for each input mention based on their distance. Fig. 4 depicts the projection operation of BioBERT and TuckER embeddings.

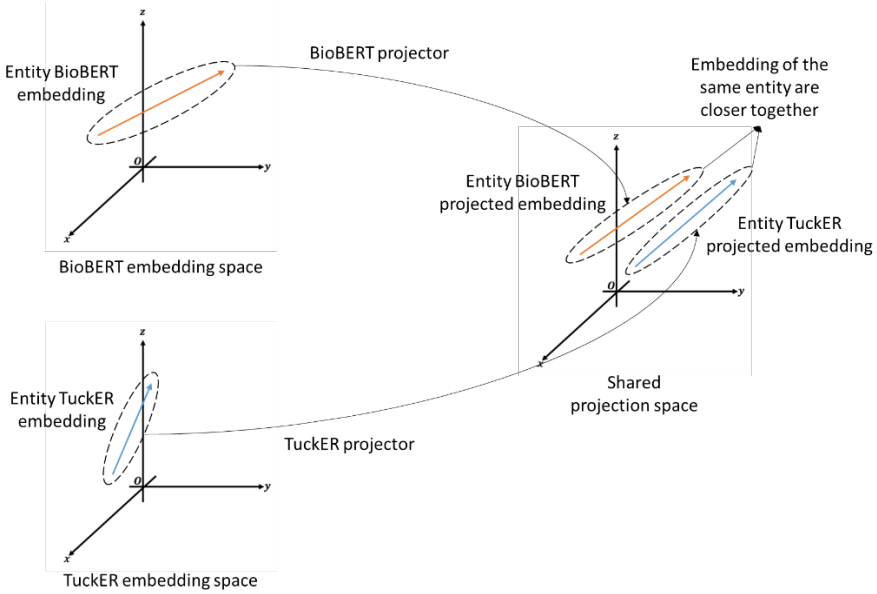

Fig. 4 *Projection operation linking BioBERT and TuckER embeddings into a shared projection space. The 3D spaces are visualisation examples, all embedding vector spaces' dimensions are 768*

The projection model, a multi-layer perceptron (MLP), is trained to align pairs of bioBERT [26] and TuckER [21] embeddings into a new, homogeneous vector space across all 37912 classes. The training dataset consists of the full class set to distill the knowledge from the whole ontology and apply it to both seen and unseen concepts in application contexts. The model projects the 768-dimensional BioBERT [26] vectors into a shared vector space, with 2 hidden layers, each followed by a ReLU function and a dropout layer, and finally an output layer of the same dimension. During inference, this approach encodes the entity mention with bioBERT, projects its embedding into the shared vector space, and calculates its distance from all ontology embeddings' projections. The ontology entity respective to the TuckER embedding with the smallest distance is returned as the final entity for EL. In this setup, we trained the projection model for 30 epochs with a learning rate of 1e-5. The EL model pipeline is depicted in Fig. 5.

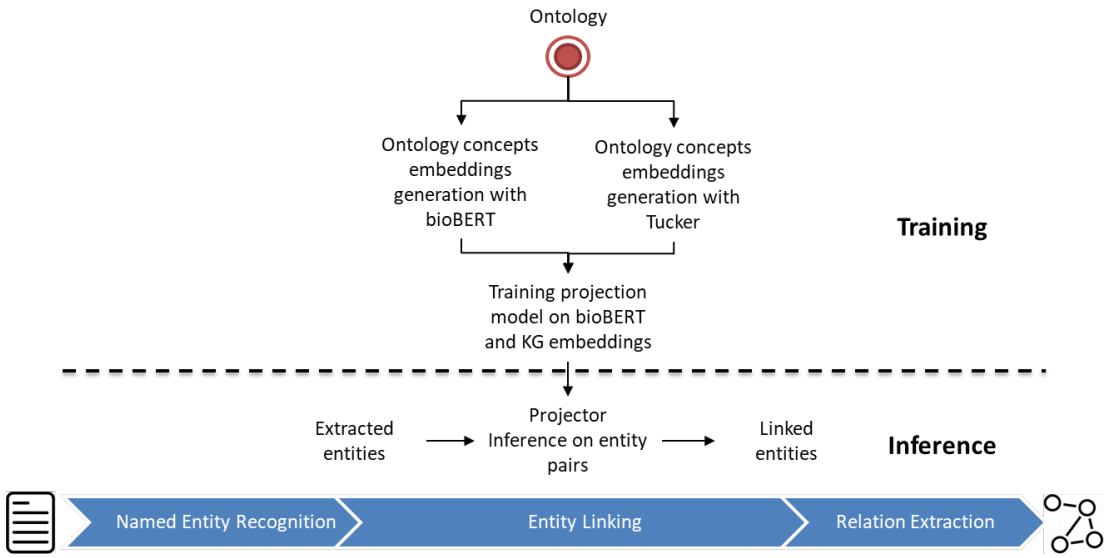

Fig. 5 *The new process of the EL component, requiring training on ontology data once, is integrated in the original KGE pipeline in the inference phase*

3.3 Relation Extraction

The method depicted in Fig. 6, focusing on the RE task, explores a new approach using the Tucker scoring function to predict the correct relation for any given subject-object pair. The RE task is treated as a ranking problem: for each entity pair, the Tucker scoring function computes a score for each possible relation , then ranks the relations by score, placing the correct relation at the top. It's worth noting that the Tucker [21] scoring function evaluates the correctness of the entire triple and considers combinations of all heads, relationships, and tails, as given in Equation 1.

$$score(h, r, t) = \mathcal{W} \times_1 h \times_2 r \times_3 t \quad (1)$$

This does not necessarily mean that the triple with the correct relation will get the highest score among a set of triples C combining a fixed head and tail with all relations in the set of relations R, as expressed in Equation 2.

$$(h, r^*, t) \in C \nRightarrow tucker\_score(h, r^*, t) > tucker\_score(h, r, t), \forall r \in R \backslash \{r^*\} \quad (2)$$

The Tucker [21] scoring function is not suitable for relation ranking, although it is used for link prediction. For this reason, this approach leverages the core concept of TuckER [21] without applying it directly to the MONDO [15] triples. We initialize a new learnable core tensor and relation embeddings with random values, while initializing entity embeddings with fixed bioBERT [26] embeddings of the ontology's classes. Next, a new scoring function (Equation 1), similar to TuckER's [21], is trained with an objective that ensures triples with correct relations are scored higher than those with false relations. In this scenario, the training updates the core tensor and relation embeddings until they correctly model the ontology's triples. This approach differs from the classic TuckER [21] application in 2 ways. The first difference is that the scoring function is relation-oriented, explicitly feeding it incorrect triples with false relations and training it to score correct triples higher than incorrect ones. The second difference is that only the core tensor and relation embeddings are learned while the entity embeddings remain unchanged to preserve the meaning in the bioBERT embedding space, instead of all of them being learned in the case of TuckER [21]. This technique allows us to control the source of entity embeddings and to apply the scoring function to bioBERT [26] vectors for previously unseen entities. In the inference stage, bioBERT [26] generates vector embeddings for the entity pairs extracted in the NER step. The experimental method then constructs triples for each pair with all relations and feeds them to the adapted scoring function. For each pair, the scoring function predicts the correct relation by ranking all relations and assigning the highest score to the most correct prediction. In the RE experiment, the MONDO [15] triples set is split into a smaller dataset containing 2100 train triples and 400 test triples divided on 9 relations (is a, disease has location, part of, has part, disease caused by disruption of, has material basis in germline mutation in, develops from, disease has feature, disease has infectious agent) for computational purposes. The training configuration consists of training for 30 epochs with a batch size of 32. We also use the Adam optimizer with a learning rate of 1e-3. The RE component is publicly available on a Github repository[1].

[1] https://github.com/echchorfisalahedine/BioBERT_TuckER_based_RE_component

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 32-44

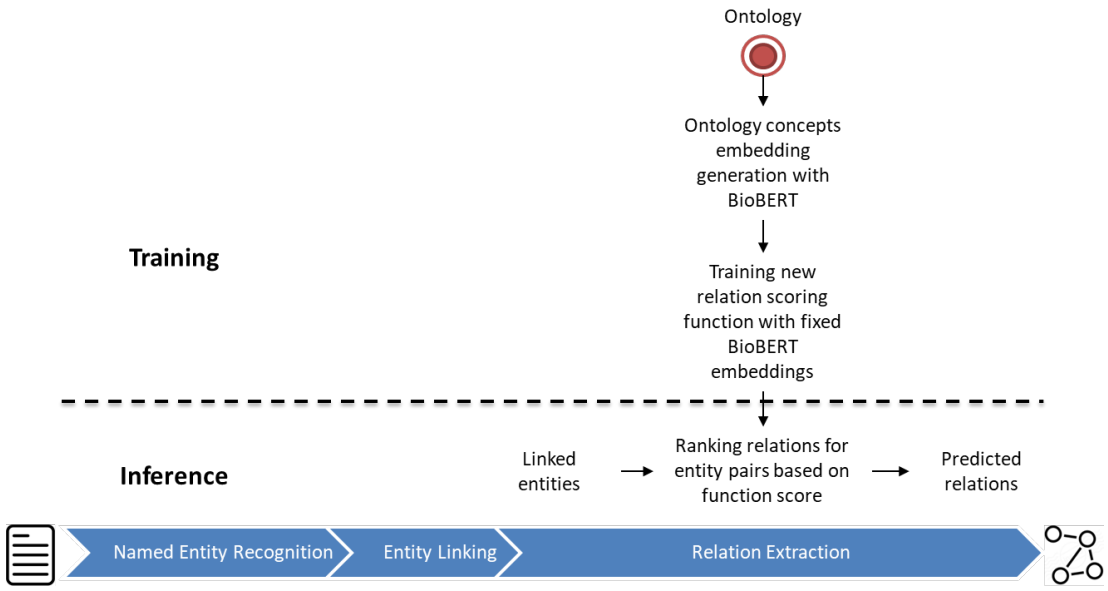

Fig. 6 *The new process of the RE component, requiring training on ontology data once, is integrated in the original KGE pipeline in the inference phase*

4. Results and Discussion

In the NER experiment, the BioBERT [26] model trained solely on augmented sentences struggles to identify the correct tag for each word within the extracted entity, often confusing BIO tags at the entity boundary. For example, the model tends to assign the I-Disease tag to the first word of an entity rather than the correct B-Disease tag. The model also struggles to predict the boundaries of a given entity, as it tends to include words around the entity rather than extracting only its correct words ("People with tuberculosis" is extracted instead of "tuberculosis").

After analysis, it is determined that this behavior is caused by the lack of linguistic diversity in the training sentences, whether concatenated or LLM-generated, whereas state-of-the-art annotated NER datasets are based on real biomedical text from sources such as scientific papers and medical electronic records.

This experiment concludes that using a structured knowledge source, such as ontologies, although standard and comprehensive, is not optimal for NER applications that use finetuned word embedding models. These models rely heavily on learning the natural text composition and language diversity from training data, a key feature absent in our case of augmented sentences.

The EL experiment identified a major challenge in our approach that significantly affected the performance of the proposed theory. Upon inspection of the loss variation during training (Fig. 7), it is apparent that the loss only fluctuates by 0.0012 between the highest and lowest values.

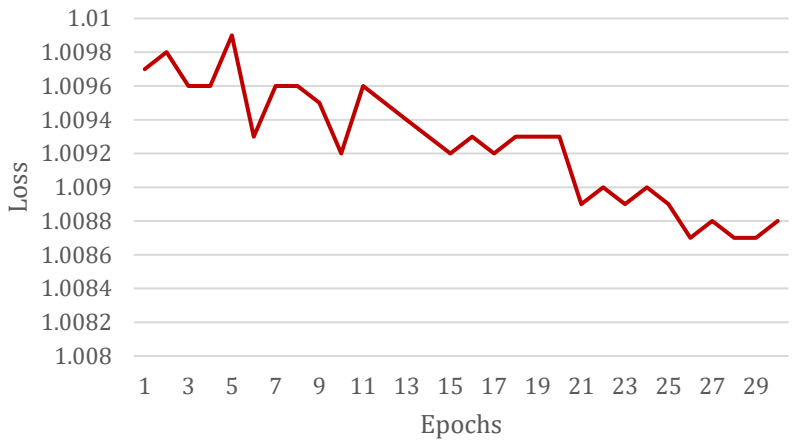

Fig. 7 *Loss variation of the EL model throughout the training epochs*

This indicates that the model stagnates for most of the training and fails to learn a useful pattern for the projection. The first problem lies in the small variance of both BioBERT [26] embeddings and TuckER [21] embeddings, which were calculated to be 2.4788e-05 and 3.8502, respectively. These low-variance values, combined with the projection step, result in a further decline in variance, estimated at 1.9511e-3. This causes the projected vectors to collapse, making it harder for the distance calculation function to distinguish between different projected entity embeddings. The distance calculation function ultimately fails to return the correct corresponding ontology concept because all projected vectors are compact. This experiment established that any solution based on projection, distance, or cosine similarity is not optimal for this setting, and the main concern is mitigating the impact of low BioBERT [26] embedding variance and vector collapse after projection.

In the RE experiment, the approach yielded state-of-the-art results in relation ranking, as shown in Table 4. This method avoids the major challenges of the projection approach and the classic TuckER [21] scoring function by directly integrating fixed BioBERT [26] entity embeddings and training a new scoring function to learn both relation embeddings and a core tensor that captures the interactions between entities in the triples. Our approach tackles RE as a ranking problem. For this reason, the evaluation of its performance is carried out using the following ranking-specific metrics:

- Hits@K: Calculates the number of predictions where the model successfully ranked the correct relation in the top K positions.
- Mean rank: Calculates the average rank of the correct relations across all test triples.
- Mean reciprocal rank (MRR): Calculates the average of the mean rank, which represents how high the correct relation is ranked in each prediction, across all test triples.
- Normalized discounted cumulative gain (NDCG): Evaluates how good the ranking is. It increases when relevant predictions are ranked first, while it decreases when they are ranked last. In our case, the only relevant predictions are the correct relations.
- Area under the ROC curve (AUC-ROC): Measures how well the model ranks the correct relations higher than wrong relations. This aligns with our training objective of ranking true triples above false triples.

Table 4 *The results obtained in the test phase of the RE model on the 400 triples in the test set*

Metrics	Values
Hits@1	0.7500
Hits@3	0.9425
MRR	0.8458
Mean rank	1.5050
NDCG	0.8826
AUC-ROC	0.9309

Of the three experiments presented in Section 3, the RE approach is the only one qualified to be proposed as an RE method for general application, whereas the NER and EL approaches still require further research to overcome the highlighted challenges.

5. Application of the Relation Extraction Model

The proposed RE method serves as an RE component in a KGE pipeline, allowing it to predict the correct relation from a set of standard biomedical relations, regardless of the biomedical ontology used for training. At a higher level, the approach can be applied across different application contexts, such as link prediction, KG completion, and ontology enrichment from biomedical text.

For the application scenario, we integrate the new RE model into the KGE pipeline previously described in [30]. The pipeline consists of 4 modules, slightly adapted to the current use case. First, we perform NER using BioBERT [26] to extract 4 entity types: Disease, Drug, Gene, and Cell. Next, we link the extracted entities into MONDO [15] (previously UMLS [23]) using a Spacy custom entity linker based on the ontology file. Then, the old RE component, trained on relations from the Biorel dataset [31], is replaced with the new RE model and feed it the linked-entity pairs to predict their respective relations. Lastly, a curation step of the triple predictions is performed by filtering out incorrect triples based on their class combinations (e.g., the relation "has material basis in germline mutation in" is true only between the Disease-Gene class pair). Table 5 cites all the combinations of classes for each relation used for curation, on which predictions are discarded if not respected. On the other hand, Fig. 8 depicts a subgraph constructed by the pipeline using an input of biomedical text.

J. of Soft Computing and Data Mining Vol. 7 No. 2 (2026) p. 32-44

Table 5 *This table shows the possible combinations of the classes for each relation used for predictions filtering*

Relation	Head class	Tail class
Is a	Same class	
Part of / Has part	Drug	Drug
	Cell	Cell
	Gene	Gene
Disease has location	Disease	Cell
Disease has feature	Disease	Disease
Disease has infectious agent	Disease	Disease
Disease caused by disruption	Disease	Gene
Develops from	Disease	Disease
Has material basis in germline mutation in	Disease	Gene

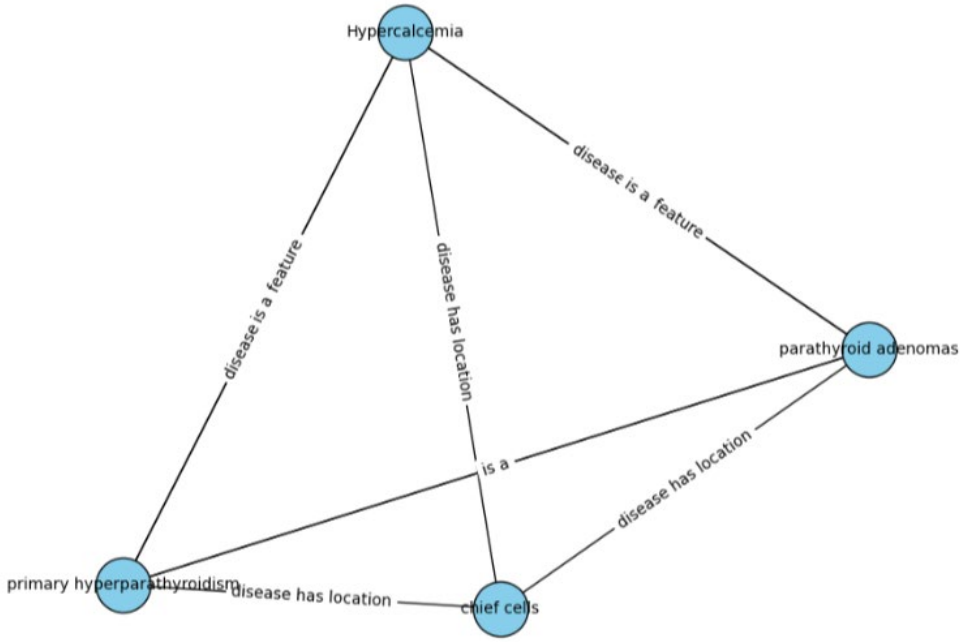

Fig. 8 *A subgraph extracted from the original output graph of the KGE pipeline while equipped with the new RE component. The entities, extracted using the old NER system, are linked by standard MONDO relations*

6. Conclusion

The goal of this work is to explore the possibility of conceiving NER, EL, and RE models that do not rely on training on predefined biomedical datasets that limit the scope of extracted entities and predicted relations but instead are based on biomedical ontologies containing standard knowledge in the form of standard biomedical concepts and relations. Furthermore, the proposed models should yield state-of-the-art results and perform well in real-life biomedical text applications. For this reason, several experiments are carried out with different configurations for all NER, EL, and RE tasks. The NER experiment revealed challenges regarding the diversity of the synthetic sentences used for the word embedding model's training, which limits its ability to provide meaningful representations of biomedical entities. The projection approach was not suitable for the EL task due to the compactness of word embedding model vectors and graph embedding models, which led to the projected vectors collapsing and affecting the model's ability to differentiate between entities' embeddings. In the case of RE, the

proposed approach successfully learned and classified correct relations for unseen subject-object pairs, relying solely on ontology triples. The experiments revealed challenges and limitations that need to be addressed in future research on NER and EL approaches, as well as improvements to implement in RE.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

All authors contributed significantly to this work. ***Conceptualization:*** *Ech-Chorfi Salah Edine;* ***Methodology:*** *Ech-Chorfi Salah Edine;* ***Experiments and formal analysis:*** *Ech-Chorfi Salah Edine, Zemmouri Elmoukhtar;* ***Writing - original draft preparation:*** *Ech-Chorfi Salah Edine;* ***Writing - review and editing:*** *Zemmouri Elmoukhtar; Supervision: Zemmouri Elmoukhtar.*

References

[1] Doğan, R. I., Leaman, R., & Lu, Z. (2014). NCBI disease corpus: a resource for disease name recognition and concept normalization. Journal of biomedical informatics, 47, 1–10.

[2] JNLPBA '04: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (Association for Computational Linguistics, USA, 2004)

[3] Krallinger, M. et al. The chemdner corpus of chemicals and drugs and its annotation principles. Journal of Cheminformatics 7, S2 (2015)

[4] Li, J. et al. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. Database 2016, baw068 (2016)

[5] Taboureau, O. et al. Chemprot: a disease chemical biology database. Nucleic Acids Research 39, D367–D372 (2010). URL https://doi.org/10.1093/nar/gkq906

[6] Foltz, P. Latent semantic analysis for text-based research. Behavior Research Methods 28, 197–202 (1996)

[7] Pennington, J., Socher, R. & Manning, C.Glove: Global vectors for word representation, Vol. 14, 1532–1543 (2014)

[8] Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient estimation of word representations in vector space (2013). URL https://arxiv.org/abs/1301.3781. arXiv:1301.3781

[9] Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T. Enriching word vectors with subword information (2017). URL https://arxiv.org/abs/1607.04606. arXiv:1607.04606

[10] Hochreiter, S. & Schmidhuber, J. Long short-term memory. Neural Computation 9, 1735–1780 (1997)

[11] Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Burstein, J., Doran, C. & Solorio, T. (eds) BERT: Pre-training of deep bidirectional transformers for language understanding. (eds Burstein, J., Doran, C. & Solorio, T.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 4171–4186 (Association for Computational Linguistics, Minneapolis, Minnesota, 2019). https://aclanthology.org/N19-1423/

[12] Radford, A., Narasimhan, K., Salimans, T. & Sutskever, I. Improving language understanding by generative pre-training (2018)

[13] Vaswani, A. et al. Guyon, I. et al. (eds) Attention is all you need. (eds Guyon, I. et al.) Advances in Neural Information Processing Systems, Vol. 30 (Curran Associates, Inc., 2017)

[14] Ashburner, M. et al. Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium. Nat Genet 25, 25–29 (2000).

[15] Vasilevsky, N. A. et al. Mondo: Unifying diseases for the world, by the world. medRxiv (2022). https://www.medrxiv.org/content/early/2022/05/03/2022.04.13.22273750.

[16] Schriml, L. et al. The human disease ontology 2022 update. Nucleic Acids Research 50 (2021).

[17] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J. & Yakhnenko, O. Burges, C., Bottou, L., Welling, M., Ghahramani, Z. & Weinberger, K. (eds) Translating embeddings for modeling multi-relational data. (eds Burges, C., Bottou, L., Welling, M., Ghahramani, Z. & Weinberger, K.) Advances in Neural Information Processing Systems, Vol. 26 (Curran Associates, Inc., 2013). https://proceedings. neurips.cc/paper files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf.

[18] Lin, Y., Liu, Z., Sun, M., Liu, Y. & Zhu, X. Learning entity and relation embeddings for knowledge graph completion. Proceedings of the AAAI Conference on Artificial Intelligence 29 (2015). https://ojs.aaai.org/index.php/AAAI/article/view/9491

[19] Sun, Z., Deng, Z.-H., Nie, J.-y. & Tang, J. Rotate: Knowledge graph embedding by relational rotation in complex space (2019)

[20] Trouillon, T., Welbl, J., Riedel, S., Gaussier, E. & Bouchard, G. Balcan, M. F. & Weinberger, K. Q. (eds) Complex embeddings for simple link prediction. (eds Balcan, M. F. & Weinberger, K. Q.) Proceedings of The 33rd International Conference on Machine Learning, Vol. 48 of Proceedings of Machine Learning Research, 2071–2080 (PMLR, New York, New York, USA, 2016). https://proceedings.mlr.press/v48/trouillon16.html

[21] Balazevic, I., Allen, C. & Hospedales, T. Tucker: Tensor factorization for knowledge graph completion (Association for Computational Linguistics, 2019). URL http://dx.doi.org/10.18653/v1/D19-1522

[22] Liu F., Shareghi E., Meng Z., Basaldella M., and Collier N.. 2021. Self-Alignment Pretraining for Biomedical Entity Representations. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4228–4238, Online. Association for Computational Linguistics.

[23] Bodenreider, O. The unified medical laguage system (umls): Integrating biomedical terminology. Nucleic acids research 32, D267–70 (2004)

[24] Wang X., Gao T., Zhu Z., Zhang Z., Liu Z., Li J., and Tang J.. 2021. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. Transactions of the Association for Computational Linguistics, 9:176–194.

[25] Xiong, X., Wang, C., Liu, Y., Li, S. (2023). Enhancing Ontology Knowledge for Domain-Specific Joint Entity and Relation Extraction. In: Sun, M., et al. Chinese Computational Linguistics. CCL 2023. Lecture Notes in Computer Science, vol 14232. Springer, Singapore. https://doi.org/10.1007/978-981-99-6207-5_15

[26] Lee, J. et al. Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics 36, 1234–1240 (2019). URL https://doi.org/10.1093/bioinformatics/btz682

[27] Degtyarenko, K. et al. Chebi: A database and ontology for chemical entities of biological interest. Nucleic acids research 36, D344–50 (2008)

[28] Gkoutos, G. V., Schofield, P. N. & Hoehndorf, R. The anatomy of phenotype ontologies: principles, properties and applications. Briefings in Bioinformatics 19, 1008–1021 (2017). https://doi.org/10.1093/bib/bbx035

[29] Diehl, A. et al. The cell ontology 2016: enhanced content, modularization, and ontology interoperability. Journal of biomedical semantics 7, 44 (2016)

[30] Ech-chorfi, S. E. & Zemmouri, E. Towards an extensible pipeline for biomedical knowledge graph construction from scientific papers. IAENG International Journal of Computer Science 52, 625 – 636 (2025). https://www.scopus.com/inward/record.uri?eid=2-s2.0-85219304457&partnerID=40&md5=f62490f191841972814f90026530a0c5

[31] Xing, R., Luo, J. & Song, T. Biorel: A large-scale dataset for biomedical relation extraction, 1801–1808 (2019)