International Journal of Innovative Research in Engineering
Volume 7, Issue 6 (November - December 2026), PP: 202-214.
https://www.doi.org/10.59256/ijire.20260706023
www.theijire.com
ISSN No: 2582-8746

Molecular score based entity alignment between diabetes knowledge graphs

Sushmita Singh[1], Manvi Siwach[2]
[1,2]Department of Computer Engineering, J.C. BOSE University of Science and Technology, YMCA, Faridabad, Haryana, India.

How to cite this paper:
Sushmita Singh[1], Manvi Siwach[2], "Molecular score based entity alignment between diabetes knowledge graphs", IJIRE-V7I6-202-214.

Copyright © 2026 by author(s) and Fifth Dimension Research Publication. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/

Abstract: *Entity alignment identifies entities in different knowledge graphs that refer to the same real-world object, and it is a key step in integrating knowledge from multiple sources. In domains such as medicine, the same entity name can carry different meanings depending on how it is linked to its neighbors, and equivalent entities often have neighborhoods of different sizes in different knowledge graphs. Many alignment approaches do not explicitly use the relation connecting an entity to its neighbor, or do not account for these differences in neighborhood size. This paper proposes an embedding-based entity alignment approach that computes a molecular score for each entity by combining the entity's own embedding with an averaged aggregation of its neighbors' embeddings, each paired with the embedding of the relation that connects it. Averaging reduces the effect of unequal neighborhood sizes, and the inclusion of relation embeddings helps distinguish entities that share neighbors but are linked to them through different relations. The approach works in three stages: joint embedding space creation, candidate selection using cosine similarity, and candidate ranking based on the similarity of molecular scores. It is evaluated on a pair of diabetes knowledge graphs, one manually curated from health authority sources and the other extracted from DBpedia, using Hits@1, Hits@3, Hits@10, Mean Rank, and Mean Reciprocal Rank. An ablation study compares entity embeddings alone against variants that add neighbor and relation embeddings. The results show that incorporating neighboring entities together with their connecting relations improves Hits@1 and lowers Mean Rank compared with using entity embeddings alone, indicating that relational context contributes to more accurate alignment.*

Key Words: *entity alignment, knowledge graphs, relation-aware embeddings, neighborhood aggregation, medical knowledge graphs.*

I.INTRODUCTION

Knowledge graphs (KGs) represent real-world entities and the relationships between them as a graph of triples. Although the idea has older roots, KGs gained wide recognition after Google launched its Knowledge Graph in 2012 [46], and they are now used by many companies and research communities for data representation and processing [47]. Open KGs such as DBpedia [3] and Freebase [4] are among the best known. KGs store structured, semi-structured and unstructured data, and they combine reasoning with a knowledge base to derive implicit knowledge from raw, explicit data. Information extraction methods are used to construct a KG from semi-structured or unstructured data, and the data sources should be expanded to populate and diversify the graph [5]. Hence, multiple data sources are merged to create a richer KG.

While combining two or more KGs, entities from one KG are connected with entities in the other [6]. Because the same real-world entity often occurs in both KGs, entities must be aligned before the graphs can be merged. Entity alignment (also called entity matching) is the task of identifying entities in different knowledge bases that refer to the same real-world object or concept [7][8]. It is a way of integrating knowledge from heterogeneous data sources, and a significant step in tasks such as data analysis, knowledge reasoning and KG construction [9], and it is surveyed in [48].

Entity alignment is generally divided into two phases: candidate selection and candidate ranking, after which the best-ranked candidate is selected [5]. Existing approaches use a range of entity features, which we group into atomic features (such as attributes and labels) and molecular features (such as neighbors and relations), as reviewed in Section 2. Recent graph-based methods already model neighbors and relations [25][43][44], but they learn their representations from seed alignments, which are scarce in specialized domains such as medicine. Two issues also remain important when aligning entities from independently built KGs. The first is *ambiguity of relationships*: two entities can share the same neighbor but be connected to it through different relations, so neighbor identity alone is not enough. The second is *molecular non-isomorphism*: equivalent entities often have neighborhoods of different sizes in different KGs, which makes direct comparison of neighborhood information unreliable.

This paper proposes an embedding-based, training-free approach that addresses both issues. It computes a molecular score for each entity by combining the entity's own word embedding with an averaged aggregation of its neighbors'

embeddings, each paired with the embedding of the relation that connects it. Averaging reduces the effect of unequal neighborhood sizes, and pairing neighbors with relations separates entities that share neighbors but differ in relation.

1.1 Motivation of using knowledge graphs in medical domain

The amount of medical text has grown rapidly with the use of electronic health records, health management systems and other digital tools [10]. Medical text is difficult to interpret because it contains specialized concepts, terminology and complex relationships between words and phrases [10], which increases the need for data analytics and natural language processing in this domain. KGs can support the analysis and representation of such relationship-oriented data. Medical text can broadly be categorized as clinical text (for example, electronic medical records [11][12], electronic health records [13][14] and personal health records [15]), biomedical text such as published journals and studies [16][17], and web-based text from search engines, social media and health applications. This paper focuses on biomedical and web-based text for constructing medical KGs. Because these KGs are built from different sources, entity alignment is required to merge their information. We evaluate the proposed approach on two diabetes KGs introduced in our earlier work [49].

1.2 Contribution

The main contributions of this paper are as follows:

- We propose a molecular score for entity alignment that pairs each neighboring entity with its connecting relation and averages over neighbors, addressing relationship ambiguity and molecular non-isomorphism.
- The approach is training-free: entity and relation embeddings come from word embeddings, and no seed alignments are required.
- We evaluate the approach on two diabetes KGs, one manually curated from health authority sources and the other extracted from DBpedia, with an ablation study of the neighbor and relation components [and a comparison with baseline methods].

1.3 Organization of paper

Further, this paper is organized in following manner, section-2 surveys different techniques used for aligning entities. Section-3 discusses the problems identified, section-4 introduces and describes the proposed algorithm. Section-5 demonstrates the ablation study of algorithm over medical text. Finally, section-6 concludes this paper with suggestions and future work.

II. RELATED WORK

Entity alignment has been studied extensively, and approaches differ mainly in which features of an entity they use to compute similarity. Following our earlier review, we group these features into atomic features, which come from the entity itself, and molecular features, which come from its structural surroundings. This section reviews both groups, then recent work based on pre-trained language models and large language models, and alignment in the biomedical domain. It closes by positioning the proposed approach.

2.1 Atomic features

It comprises of All the features of the entity at the Individual or atomic level Which can be extracted singularly from the entity itself. It includes Attributes of the entity and literal Meaning of the entity label. Attributes of an entity Refers to the properties of that entity. Literal meaning refers to the textual meaning of the word.

Degree-Aware Alignment for Entities in Tail [18] uses the concept of degrees in the procedure of entity alignment by introducing long-tail entities (the entities that are connected to less than three entities). These entities tend to make the knowledge graph sparse; they are also infrequently accessed and offer less structural information. With the help of degree-related information, the entities are divided into two categories- long-tail entities and short ones. For long-tail entities, name representation is given more importance and for the second category of entities, structural representation is given more importance. A co-attention feature similarity calculating mechanism is used to do this by creating a similarity matrix between the features of both entities.

Cross-lingual Entity Alignment via Joint Attribute-Preserving Embedding [19] proposes a joint attribute-preserving embedding model for aligning the entities from two knowledge bases. Structure embedding is applied simultaneously on two knowledge graphs, to create a unified vector space. The model uses structure embedding and attribute embedding for entity alignment. TransE embedding technique is used for structure embedding. It is used to learn relationship pairs from the input knowledge graphs and create overlay relationship graphs. After this attribute embeddings are used to refine the triples and align the entities.

Linguistic Frames as Support for Entity Alignment in Knowledge Graphs [20] suggests the concept of entity alignment by enriching the entity embeddings with the literal information related to the entity itself. Literal information refers to the non-figurative context or sense of a word or a set of words. This approach considers literals from attributes as well as literals from triplets. The proposed approach exploits linguistic frames and their inclusion in the entity alignment process. Lexical patterns are discovered between the triplets which capture varied correspondence and correlations between literals.

EASA: Entity Alignment Algorithm Based on Semantic Aggregation and Attribute Attention [21] introduces EASA an entity alignment algorithm that uses semantic aggregation, which is deduced from attributes of the entities and attribute values; which improvises the process of entity alignment. EASA uses attribute attention which highlights the discrimination of

importance among the multiple attributes of a single entity. For example, if an entity has two attributes- name and sex the probability of finding similar sex is higher as compared to similar names. Therefore, the attention given to these biased attributes should be varied as well and hence weights are calculated for different attributes belonging to a particular entity. Based on these weights, semantic aggregation of entities is computed. In the next phase, similarity scores are calculated between the entities of different knowledge graphs. And finally, the entities with the highest similarity score are aligned together.

CTEA: Context and Topic Enhanced Entity Alignment for knowledge Graphs [22] proposes a framework for entity alignment which emphasizes on the context of entities for alignment. In the training Entity topic learning module employs a topic model (a tool of NLP and machine learning which identifies a "topic" for a set of words) over the attribute names for text clustering and learning semantics of the entity. The next module is structure embedding which learns the structure of entities and relationships by using translation-based embeddings and then scores the triplets to measure the plausibility of the triplets. The third module is Context modelling which uses a multichannel Convolutional Neural Network to shape the context of an entity. While ranking the target entities for the source entities, CTEA considers the combined embeddings and representations as well as topic distribution of the attributes as the base criteria. Table-1 reports the analysis of different entity alignment approaches.

Text-based methods go further by using pre-trained language models. BERT-INT [41] encodes entity names and descriptions with BERT, and recent large language model frameworks such as ChatEA [42] use names, descriptions and graph structure in LLM prompts. These methods give strong results but depend on large models, and they also use neighborhood information, so they sit between the two feature groups.

2.2 Molecular features

It comprises of all the features that are derived from the structural surroundings of the entity. It includes relationships, degree of the entity, neighborhood of entity and semantics of the entity. Semantics of an entity refers to the Meaning associated with interpretation of that word in natural language Generally derived from the context surrounding the word.

Cross-lingual Knowledge Graph Alignment via Graph Convolutional Networks [23] uses Graph Convolutional Network (GCN) to learn the structure of knowledge graphs by encoding the information about the neighborhood of entities. GCN uses structure and attribute embeddings of both the knowledge graphs (to be aligned) and merges them into a unified vector space. Entity alignments are predicted based on the distances between entities from two knowledge graphs in the unified vector space.

Entity Alignment between Knowledge Graphs Using Attribute Embeddings [24] proposes an entity alignment model based upon attribute embeddings. The model consists of three submodules- predicate alignment module, embedding learning module, and entity alignment module. The first module tries to identify partially similar predicates to get a unified vector space for relationship embeddings. The core part lies in the second module that involves learning the entity embeddings through two ways- structure embedding that utilizes the relationship triplets (subject, predicate, object) and; attributes embeddings that utilize the attribute triplets (subject, predicate, attribute). Finally, in the third module, a similarity score is calculated for every pair of entities and pairs with a minimum threshold similarity score are selected.

Relation-Aware Entity Alignment for Heterogeneous Knowledge Graphs [25] proposes Relation-aware Dual Graph Convolutional Network (RDGCN) to discover the entity pairs (created from different knowledge graphs) that refer to the same real-world entity. RDGCN first creates a primal graph by applying a simple UNION operation between the knowledge graphs. In the next stage, a dual graph is created as a counterpart of the primal graph, which represents the connectivity measure of two relationships (or triplets) from different graphs. In dual graph, each vertex represents the triplet (or relationship) of the primal graph and a vertex is connected to another vertex if they both have either a common head (subject) or tail (object). Weights are also assigned to the edges in a dual graph, based on similar heads and tails. Lastly, a graph attention mechanism is used to obtain vertex representation in the dual relation graph to obtain entity pairs from the distance between these vertex representations.

Collective Multi-type Entity Alignment Between Knowledge Graphs [26] proposes the usage of GNN – Graph Neural Network in entity alignment and introduces a Collective Graph neural network for multi-type entity Alignment, called CGMuAlign. It collects positive as well as negative evidence from the neighborhood recursively to derive inductive knowledge and attain precision in the identified entity sets (from heterogeneous knowledge graphs) for a single real-world entity.

More recent graph methods model relations explicitly or handle uneven neighborhoods. AliNet [43] aggregates multi-hop neighbors with a gating mechanism to handle non-isomorphic neighborhood structures, and Dual-AMN [44] uses relational attention for faster alignment. These methods, together with RDGCN [25], learn relation and neighbor representations from seed alignments. The present work instead pairs each neighbor with its relation embedding from a shared word embedding space and averages over neighbors, without training on seed alignments.

Feature classification	Paper	Data analytics techniques used	Feature criteria
Atomic features	Weixin Zeng et. al.	Structure embedding (From entity embedding), Textual and Semantic similarity matrix.	Degree of the entity, frequently visited entities

	Z Sun et. al.	TransE for Structure embeddings and attribute embeddings	Structure and attributes of entities
	Gustavo de Assis Costa et. al.	Linguistic frames (from Natural Language Processing)	The attribute of entities, entity embeddings
	Li-an Huang et. al.	Stochastic gradient descent (from entity embeddings), Semantic aggregation.	Attributes of entities, attribute attention
	Zhihuan Yan et. al.	Text clustering (from text mining), Topic modelling (from NLP), Multichannel CNN (Convolutional Neural Network).	Context of the entities; Topics of entities
	Xiaobin Tang et al.	BERT-based interaction model (fine-tuned BERT for name/description matching, combined with neighbor and attribute interactions)	Entity names and descriptions, attributes, neighbors
Molecular features	Zhichun Wang et. al.	GCN (Graph Convolutional Network) for learning structure of KGs	Structure and attributes of entities; Neighborhood of entities
	Bayu Distiawan Trisedya et. al.	Attribute embeddings (as well as entity embedding)	Predicates/relationships, learned embeddings
	Yuting Wu et. al.	Dual Graph Convolutional Network	Relationships between the entities (scored by creating a dual graph of relationships)
	Qi Zhu et. al.	GNN (Graph Neural Network)	Neighborhood of entities
	Zequn Sun et al.	GNN with gated multi-hop neighborhood aggregation	Multi-hop neighborhood, non-isomorphic neighborhood structure and attention
	Xin Mao et al.	Dual attention matching network (relational attention and proxy matching attention), normalized hard sample mining	Relations and neighborhood of entities

Table 1 Entity alignment approaches review analysis

Atomic features provide the entity level information about the entity but molecular features provide Augmented information as it comprises not just the entity level data but the information from its surroundings as well. Hence Molecular features of the entity are considered better than just the atomic features. However, there are some gaps that exist in the explored molecular features of an entity which are discussed in the next section

III. PROBLEM IDENTIFICATION

Several approaches use molecular features, such as neighbors and relations, to match entities. However, two issues remain important when aligning entities from independently built knowledge graphs:

- **Ambiguity of Relationship-**

Two entities can share the same neighbor while that neighbor plays a different role for each. If only neighbors are compared, such entities look similar even though they are not equivalent.

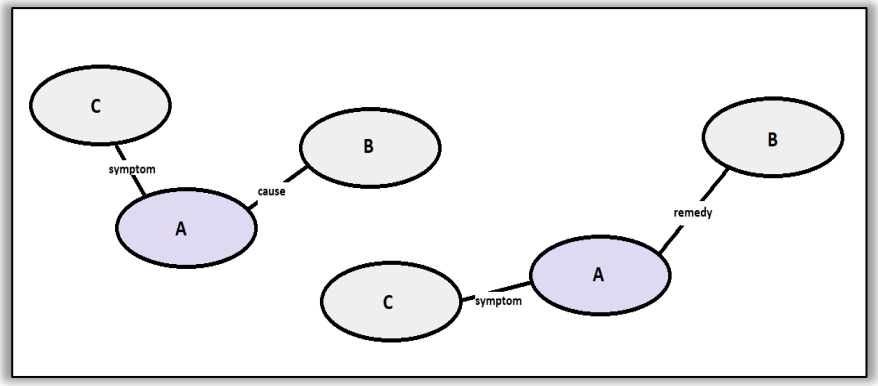


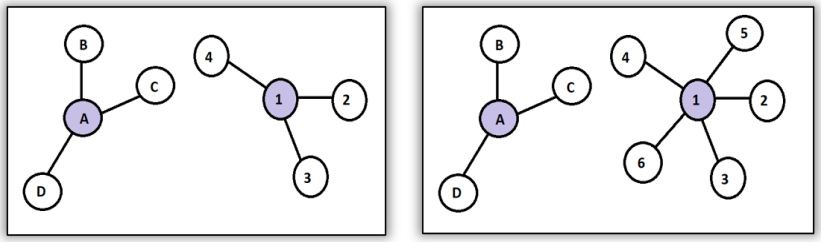
Fig.2 Relationship ambiguity

In Fig. 2, B is a neighbor of both entities A, but it is the cause of A in the first graph and the remedy of A in the second. The connecting relation therefore carries information about the role of the neighbor and should be used together with

it. Because independently built graphs name relations differently, the relation is treated as soft evidence through its embedding, not as an exact label match.

- **Molecular non-isomorphism**

In graph theory, two graphs are isomorphic if there is a one-to-one mapping between their vertices that preserves adjacency. A necessary condition is that they have the same number of vertices, the same number of edges and the same degree sequence. In this paper, the term is used in a relaxed sense for the neighborhood of an entity. Two entities are molecular-isomorphic when they have the same number of neighboring entities, as shown in Fig. 3(a), so their neighborhood information can be compared directly. Molecular non-isomorphism occurs when the equivalent entities have different numbers of neighbors in the two knowledge graphs, as shown in Fig. 3(b). This is common when the knowledge graphs are built from different sources, and it makes direct comparison of neighborhood information unreliable.



(a) (b)

Fig.3 (a) Entities with the same neighborhood size (molecular-isomorphic); (b) entities with different neighborhood sizes (molecular non-isomorphic)

Both issues are addressed by the molecular score proposed in Section 4, which pairs each neighbor with its connecting relation and averages over the neighbors.

IV. MOLECULAR SCORE-BASED ENTITY ALIGNMENT

Molecular features of the entity are considered better than just the atomic features for aligning the entities as they comprise of not just the entity level data but the information from its surroundings as well. So, in this algorithm entities are aligned by calculating similarity between the molecular scores of entities. It adapts the idea of message passing from graph neural networks: the molecular score of an entity is computed from its own embedding together with an aggregation of its neighbors and the relations that connect them. This section explains the proposed approach in detail. Further subsections give basic definitions related to the algorithm, describes the architecture of the proposed algorithm in detail.

4.1Definitions

i. Knowledge graphs:

The knowledge graph is represented using

$$KG_i\ (V, E)$$

where V represents vertices that portray entities or concepts and E represents edges that portray the relationships.

ii. Entity:

The entity is a real-world object or concept which is represented in the form of vertex/node. In this paper, *'e_i'* is used to denote entities.

iii. Relationship:

The relationships in a knowledge graph are represented as triplets-

$$T\ (S, L, O|\ \{S \in V;\ L \in E;\ O \in V\})$$

(Subject, Relationship or Link, Object)

where the subject (represented by S) and object (represented by O) define the entities and predicate/relationship (represented by L) states the name of the connection between subject and object.

4.2 Architecture

The core element in the process of aligning two entities from two knowledge graphs is to calculate the similarity between entities [27]. Consider two knowledge graphs 'KG_1' and 'KG_2'. Now every entity that belongs to the knowledge graph 'KG_1' is compared with every entity in 'KG_2'. This comparison is resolved by calculating similarity between the entities. In order to calculate the semantic similarity, the proposed approach consists of three phases – joint embedding (vector) space creation, candidate selection and candidate ranking. Figure-4 illustrates the architecture of the algorithm.

In the first phase, the triplets are extracted from both the knowledge graphs. After extracting them, a joint vector space is created out of the corpus of two knowledge graphs. The second phase is candidate selection, in which for every entity in 'KG_1' candidate entities are selected from the entities of 'KG_2'. A set of candidate entities is obtained for each entity of 'KG_1'. Next, in the third phase, i.e., in candidate ranking all the obtained candidate sets are sorted by calculating and

comparing molecular scores of the entities. The top-k ranked entities are selected from each set of candidate entities.

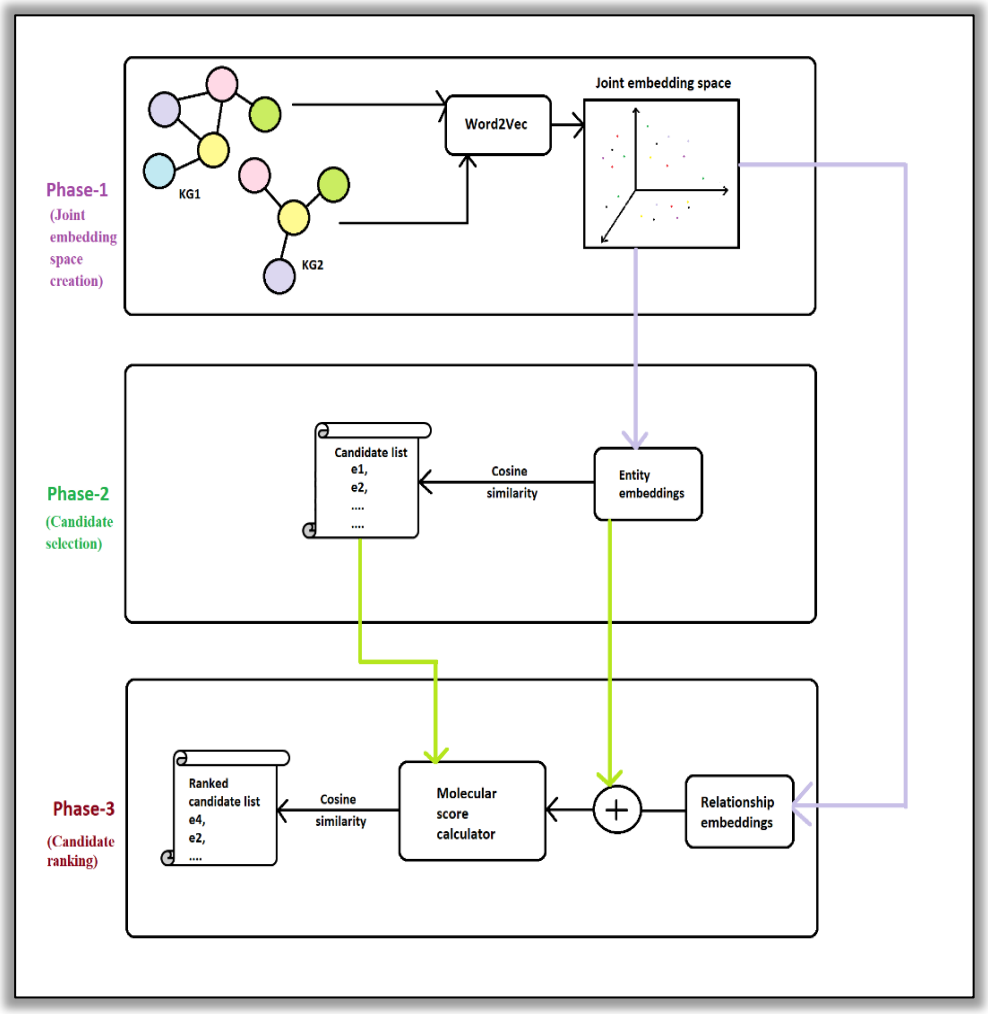


Fig.4 Architecture of Molecular score based entity alignment

4.2.1Joint embedding space creation

The first phase finds out the embeddings for every entity and relationship in the knowledge graphs and creates a joint embedding (vector) space. Word embeddings are numerical representation of words in the form of vectors. These real-valued vectors encode the meaning of word and are obtained by different type of methods. This algorithm uses Word2Vec model to extract entity embeddings and relationship embeddings. Word2vec is a neural network model which is used to create distributed vector representations of words based on an underlying corpus [28]. In this work, vectors pre-trained on the Google News dataset, which contain approximately 3M words and phrases, are used, so no seed alignments are required. The composition of the network is simple and contains an input layer, one hidden layer, and an output layer.
There are two models in *word2vec*- CBOW(continuous bag of words) and skipgram [29]. Figure-5 portrays the working of CBOW and skipgram models.

- **Continuous Bag of Words (CBOW)**

The CBOW model predicts the current word using the neighboring context words. It takes the context words occurring nearby as input, passes them through a hidden layer called embedding layer and predicts the original word.

- **Skipgram model**

The Skipgram model predicts the neighboring context words given a target word. Each word from the corpus is taken as the input, it is then passed through the hidden layer called embedding layer. Finally, it predicts the context words as output.

The number of the neighboring context words in input layer in CBOW model and in output layer in Skipgram model is decided by the window size. For example, consider a text- "COVID-19 has been found to affect the cardiovascular system". Suppose the window size is 2, then the training samples generated are depicted in Fifure-6, where 'green word' represents the target word and 'yellow words' represent the window frames. Figure-7 is a snippet of embeddings/vectors generated by

the skipgram model of word2vec.

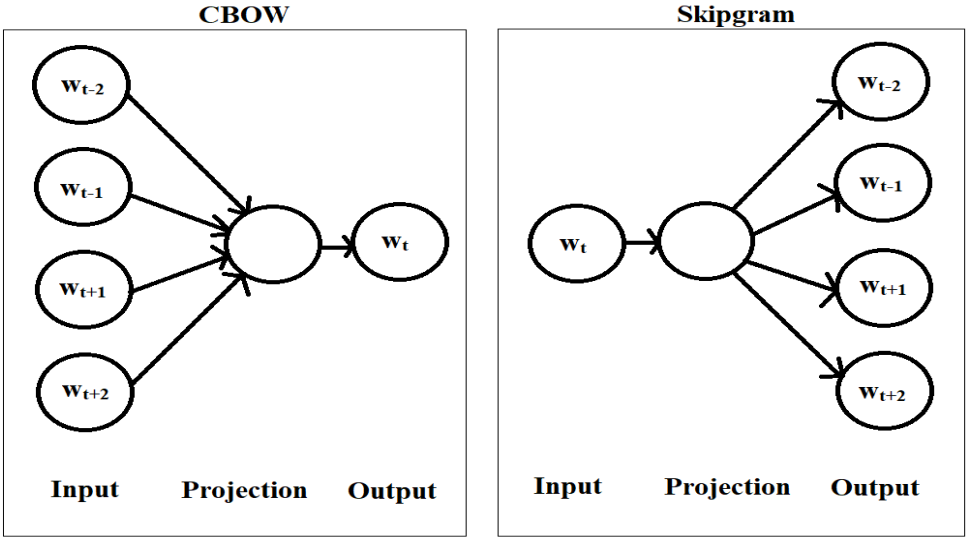


Fig.5 Word2Vec- CBOW and Skipgram models

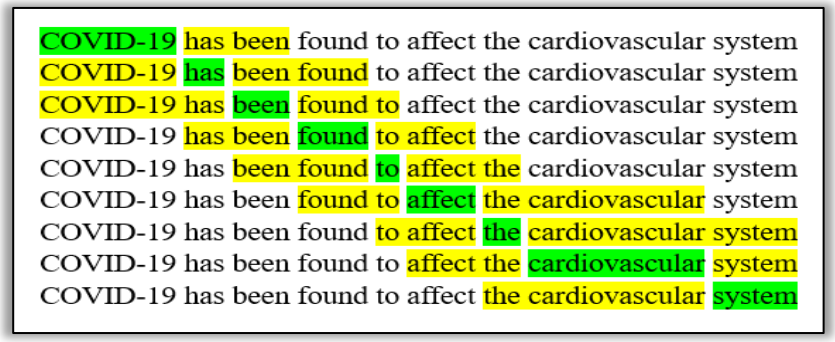


Fig.6 Example training samples for Word2Vec

```
covid-19 [-0.0960355   0.05007293 -0.08759587 -0.04391825 -0.000351
-0.00296183 -0.0766124   0.09614742 0.04982056 0.09233143]
found [-0.07511582 -0.00930042 0.09538119 -0.07319167 -0.0233377
-0.01937742 0.08077435 -0.05930896 0.00045161 -0.04753735]
affect [ 0.07311766 0.05070262 0.06757693 0.00762866 0.06350889
-0.03405366 -0.00946403 0.05768573 -0.07521639 -0.03936105]
cardiovascular [ 0.07380505 -0.01533473 -0.04536615 0.06554051
-0.0486016 -0.01816018 0.0287658   0.00991874 -0.08285215 -0.09448819]
system [-0.00536227 0.0023643   0.0510335   0.09009273 -0.0930295
-0.07116809 0.06458871 0.08972988 -0.05015428 -0.03763373]
```

Fig.7 Snippet of entity embedding generated from Word2Vec

These models perform two different prediction tasks, but the word embeddings or vectors are obtained from the hidden layer of the neural network. The networks are fed with a large text corpus as the input, and the hidden layer converts them to vector representations of words. And finally, these vector representations predict the target output. *Word2vec* models are efficient in representing the words as vectors that retain the contextual similarity between words. This algorithm uses the skipgram model for extracting the word vectors, since skipgram represents infrequent words well, which is useful for specialized medical terms. A joint embedding space is created which contains the vector representation of entities and relationships from each of the knowledge graphs [30][31]. The outputs of this phase are entity embeddings (vector representations of entities) and relationship embeddings (vector representations of relationships). These outputs act as inputs for the next two phases.

4.2.2 Candidate selection

This phase takes entity embeddings of all entities of both the knowledge graphs as input from the previous step. Then the next step is to compute the semantic similarity between the entities from both the knowledge graphs. This is done using cosine similarity function [32].

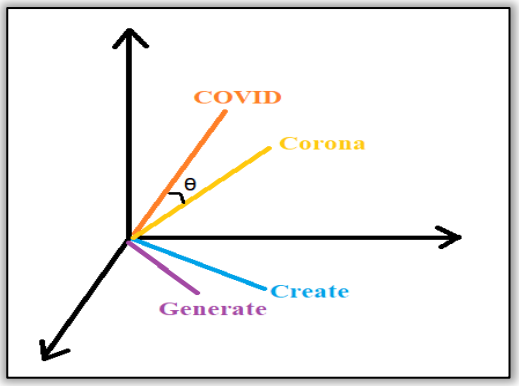


Fig.8 Vector representation of example words

In an n-dimensional vector space cosine similarity between two vectors is the cosine of the angle between them. As shown in the figure-8 similar words tend to appear closer in the vector space. "COVID" appears closer to "Corona" but distant from "Create". For vectors A and B cosine similarity is calculated as follows-

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

After this for each entity say 'e_i' in the first knowledge graph a set of entities is created called the candidate set 'CS_i'. Now each entity has a candidate set that contains the 100 entities of KG_2 with the highest cosine similarity to the entity 'e_i'. The output of this step is a set of candidate sets which contains all the candidate sets 'CS_i' corresponding to every entity of the first knowledge graph i.e., 'KG_1'.

Algorithm-1: Candidate selection

Input:
Set of entities from the dataset- E
$E = \{e_i: i = 1 \text{ to } n\}$
where n is the number of entities.
Output:
Set of Candidate sets – CS
Process:

1. For i = 1 to n
2. {
3. Select the 100 entities of KG_2 with the highest cosine similarity to ei in the joint embedding space as the candidate set CS_i.
4. $CS_i = \{X_{ij}: j\ 1 \text{ to } 100\}$
5. }

4.2.3 Candidate ranking

In this phase the objective is to rank the entities in the candidate set 'CS_i' based on their similarity with the entity 'e_i'. For this the proposed algorithm employs the concept of message passing or information diffusion from the graph neural network to calculate the similarity between entities. A graph neural network is a neural network that operates directly on graph-structured data, where nodes and edges carry information. In message passing, each node updates its representation by aggregating the representations of its neighbors. The proposed algorithm adapts this idea in a single step (one hop) and without learnable parameters. In candidate ranking, for each of the entities in the candidate set the neighborhood of the entities are explored. In order to integrate the neighborhood information, the algorithm adapts the concept of message passing in graph neural networks.

The proposed algorithm calculates a molecular score for all the entities in the candidate set. The concept of molecular score is adapted from [33][34]. Molecular score has two components- Regional score- R_{ij} and Semantic vector- V_{ij}. The first component is the regional score which is obtained from neighboring entities and the links that connect entity to its neighbors.

Generally, while considering the neighborhood of the entity, the relationship between the entity and its neighboring entity is ignored. However, the relationship plays a significant role in the semantic alignment of the entities. So, to calculate the regional score the element-wise products of the embeddings of all the neighboring entities with the embeddings of their

respective relationships/links are added-

$$R_{ij} = \sum(N_{ijk} * L_{ijk}) \quad (2)$$

where, where k = 1 to m, m is the number of neighboring nodes of "X_{ij}", "N_{ijk}" represents the vector embedding of neighboring nodes of "X_{ij}", "L_{ijk}" represents the vector embedding of corresponding links of neighbor "N_{ijk}" with "X_{ij}". Now, in equation (2) the value of "m" might change for different entities which lead to the issue of non-isomorphic entities. Therefore, instead of just adding, the products are averaged out. Hence, the equation (2) is modified as follows

$$R_{ij} = \frac{\sum(N_{ijk} * L_{ijk})}{m} \quad (3)$$

Equation (3) represents the first component i.e., regional score of an entity. The neighbors of an entity are the entities directly connected to it in its own knowledge graph [in either direction]. If an entity has no neighbors (m = 0), R_{ij} is set to the zero vector. The second component is the semantic vector of the entity itself which was obtained in the first phase i.e., candidate selection (using word2vec). It is represented as "V_{ij}".
Since R_{ij} and V_{ij} are vectors of the same dimension, the molecular score is also a vector.

The molecular score is calculated as follows-

$$M_{ij} = R_{ij} + V_{ij} \quad (4)$$

Next step is to calculate the similarity between molecular scores of entity "e_i" and all the candidate entities "X_{ij}" in the set "CS_i". In order to calculate the similarity, cosine similarity is used between them.
Similarity between the molecular score of entity "e_i" and molecular scores of all the candidate entities "X_{ij}" is calculated as-

$$Similarity\ (m_i, m_{ij}) = cos(\theta) = \frac{m_i . m_{ij}}{||m_i|| ||m_{ij}||} = \frac{\sum m_i m_{ij}}{\sqrt{\sum m_i^2}\sqrt{\sum m_{ij}^2}} \quad (5)$$

where "m_i" is the molecular score of "e_i",
"m_{ij}" is the molecular score of "x_{ij}".
The candidate "x_{ij}" with the highest similarity coefficient Similarity (m_i, m_{ij}) is ranked first and is aligned with entity "e_i".

Algorithm-2: Candidate Ranking

Input:
Candidate set 'CS_i' for each entity 'e_i'
Output:
Ranked Candidate set for each entity.
Process:

1. For j= 1 to 100
2. {
3. Calculate a molecular score "M_{ij}" for each "X_{ij}" in the candidate set "CS_i".
4. Regional score-

$$R_{ij} = \frac{\sum(N_{ijk} * L_{ijk})}{m}$$

where k = 1 to m, m is the number of neighboring nodes of "x_{ij}", "n_{ijk}" represents the vector embedding of neighboring nodes of "x_{ij}", "l_{ijk}" represents the vector embedding of corresponding links of neighbor "n_{ijk}" with "x_{ij}".

5. Semantic vector

V_{ij} = *vector embedding of "X_{ij}"*

6. Molecular Score

$M_{ij} = R_{ij} + V_{ij}$

7. }

8. Compute the molecular similarity between the entity and the entities in its candidate set to rank the candidate entities in the candidate set in descending order of similarity-

$$Similarity\ (m_i, m_{ij}) = cos(\theta) = \frac{m_i . m_{ij}}{||m_i|| ||m_{ij}||} = \frac{\sum m_i m_{ij}}{\sqrt{\sum m_i^2}\sqrt{\sum m_{ij}^2}}$$

V.EXPERIMENTS

This section presents the experimental study conducted to evaluate the performance of the proposed algorithm. The evaluation was carried out through an ablation study, for which two knowledge graphs were constructed within the domain of diabetes. The construction of the first knowledge graph is based on the careful compilation of data obtained from

authoritative government websites specializing in diabetes-related information, such as [35], [36], [37], and [38]. Figure-1 provides a visual representation for the manually curated knowledge graph. This comprehensive framework captures the depth and diversity of diabetes knowledge, offering a holistic perspective on its multifaceted landscape.

Simultaneously, the second knowledge graph is constructed using a different approach, leveraging the extensive knowledge available in DBpedia, a well-established semantic repository derived from Wikipedia. As one of the oldest and most widely recognized knowledge graphs, DBpedia [39] was developed to extract information from Wikipedia and transform it into a structured web of data that can be semantically accessed and queried. This process involves converting Wikipedia content into RDF (Resource Description Framework) triples, generating a multidomain RDF dataset. The dataset is then interconnected with other open datasets using RDF links. To extract relevant triplets that capture the semantic relationships within the data, advanced querying techniques are applied using the OpenLink Virtuoso SPARQL Query Editor [40]. Figure 2 provides a comprehensive representation for this second knowledge graph, seamlessly integrating insights obtained from DBpedia.

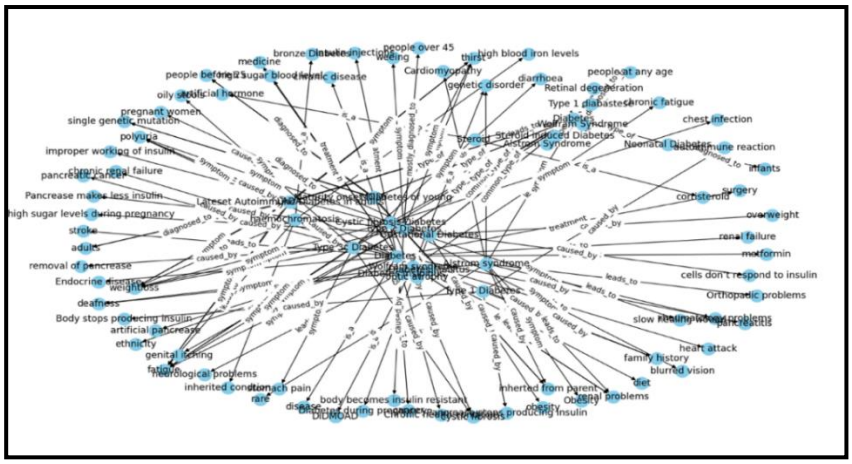


Figure-9: Manual Knowledge Graph

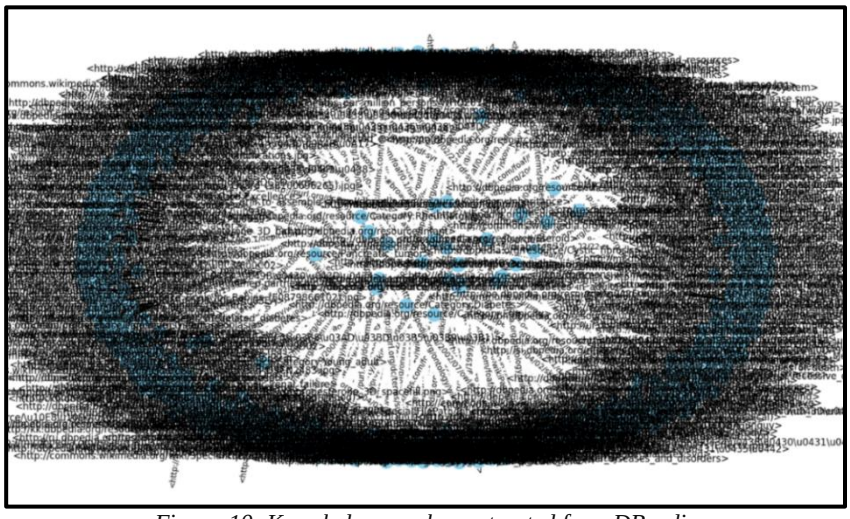


Figure-10: Knowledge graph constructed from DBpedia

The results are evaluated using hits@k (where k= 1,3,10), MRR (Mean Reciprocal Rank), and MR (Mean Rank). Hits@k, MR and MRR provide a measure of how well a model performs in identifying relevant items within a limited subset of predictions. Hits@k is a metric used to evaluate the performance of machine learning models, particularly in the context of ranking and retrieval tasks such as link prediction in knowledge graphs. It measures the proportion of times the correct item appears in the top k predictions of a model. For each query, the model generates a ranked list of potential links. If the true link is among the top k predicted links, it's a hit. If the model correctly includes the true link in the top 'k' predictions for 'x' out of 'y' queries, then Hits@k would be:

$$Hits@k = \frac{x}{y}$$

Mean Rank (MR) calculates the average rank position of the true item (e.g., the correct link) in the list of predictions. For each query, the model generates a ranked list of predictions and rank of true item is identified. Average of all such ranks across all the queries is the mean rank. If there are 'n' number of total queries and 'ir' represents the rank of true item for 'i^{th}' query then MR is calculated as follows:

$$MR = \frac{1}{n}\sum_{i=1}^{n} \text{ir}$$

Mean Reciprocal Rank (MRR) is another evaluation metric that provides a more fine-grained measure of ranking performance. For the same parameters as in MR, MRR is calculated as follows:

$$MR = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{ir}$$

The results are shown in Table-4. Figure-11 and Figure-12 depicts the results of the conducted ablation study.

Model	MRR	MR	Hits@1 (%)	Hits@3 (%)	Hits@10 (%)
EE	0.485	63.2	40.0	52.4	58.6
EE+NE	0.322	65.6	23.1	35.5	46.6
EE+(NE+RRE)	0.265	64.8	16.2	29.7	43.1
EE+(NE*RRE)	0.481	59.4	40.3	50.7	61.7

Table 4 (a) Result of ablation study on Diabetes dataset (mean over seed embeddings)

Model	MRR	MR	Hits@1 (%)	Hits@3 (%)	Hits@10 (%)
EE	0.007	1.4	1.4	1.5	0.0
EE+NE	0.011	1.9	1.5	2.6	1.2
EE+(NE+RRE)	0.013	2.6	2.0	1.4	1.7
EE+(NE*RRE)	0.027	1.6	4.0	2.0	3.1

(b) Result of ablation study on Diabetes dataset (Standard deviation over seed embeddings)

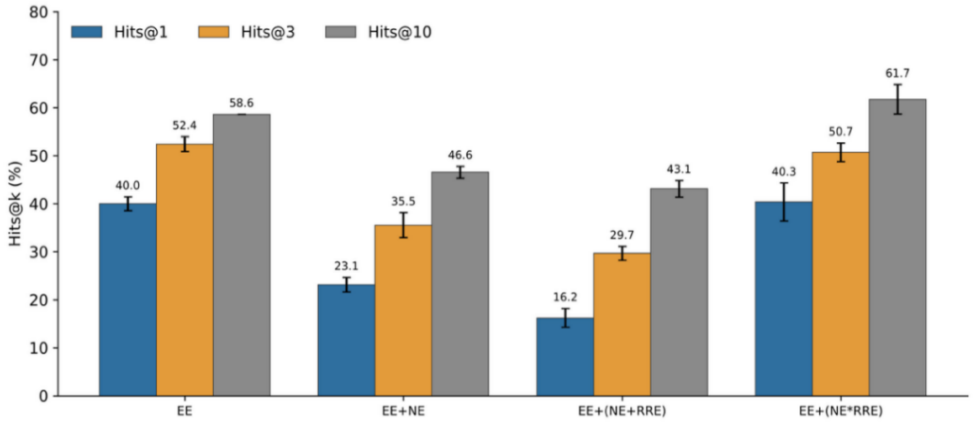


Figure-11: Hits@k for the experiment on diabetes dataset

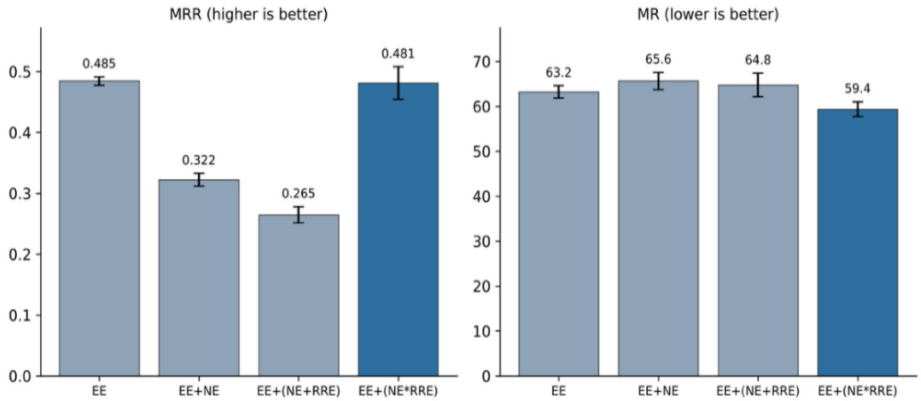


Figure-12: MRR and MR for the experiment on diabetes dataset

VI. CONCLUSION

Entity alignment identifies entities in different knowledge graphs that refer to the same real-world object. This paper examined how the neighborhood of an entity, together with the relations that connect it to its neighbors, can support alignment when the graphs are built independently. Two issues were identified: neighbor-role ambiguity, where entities share a neighbor

but are linked to it through different relations, and molecular non-isomorphism, where equivalent entities have neighborhoods of different sizes. To address them, the paper proposed an approach that needs no seed alignments. It computes a molecular score for each entity from its word embedding and an average over its neighbors, each combined with the embedding of the connecting relation. The approach was evaluated on two diabetes knowledge graphs, one manually curated and one extracted from DBpedia, with an ablation of the neighbor and relation components. Combining each neighbor with its relation through an element-wise product gave the lowest mean rank (59.4 against 63.2 for entity embeddings alone) while keeping MRR and Hits@1 at the level of entity embeddings. Adding the neighbor vectors, or the sum of neighbor and relation vectors, performed worse on every metric, possibly because the aggregated term outweighs the entity's own vector. The gain in mean rank is modest, and the evaluation uses a small number of gold pairs from a single domain, so the results show that relation-aware neighborhood information can help when it is combined with the entity carefully, not that it improves alignment in general. Future work includes evaluating on larger and additional knowledge graphs, comparing with trained relation-aware alignment methods, using domain-specific or contextual embeddings, handling medical acronyms and multi-word terms, and weighting neighbors and relations instead of averaging them.

References

1. Edward W. Schneider Course Modularization Applied: The Interface System and Its Implications for Sequence Control and Data Analysis. In: Association for the Development of Instructional Systems (ADIS), Chicago, Illinois, April 1972.
2. Barry Schwartz Google's Freebase to Close After Migrating To Wikidata: Knowledge Graph Impact? Search Engine Roundtable. Retrieved December 10, 2017
3. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. G. Ives DBpedia: A nucleus for a web of open data. In: Proc. Int. Semantic Web Conf., 2007, pp. 722–735.
4. K. D. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor Freebase: A collaboratively created graph database for structuring human knowledge. In: Proc. ACM SIGMOD Int. Conf. Manage. Data, 2008, pp. 1247–1250.
5. Sushmita Singh and Manvi Siwach Handling Heterogeneous Data in Knowledge Graphs: A Survey. In: Journal of Web Engineering, 2022.
6. Xin Luna Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Kevin Murphy, Shaohua Sun, Wei Zhang From Data Fusion to Knowledge Fusion. In: Proceedings of the VLDB Endowment, Vol. 7, No. 10 VLDB Endowment 21508097/14/06. 2014.
7. Palash Goyal and Emilio Ferrara Graph Embedding Techniques, Applications, and Performance: A Survey. In: Knowledge-Based Systems. 2018
8. Yichen Song, Aiping Li, Yan Jia, Jiuming Huang, Xiaojuan Zhao Knowledge Fusion: Introduction of Concepts and Techniques. In: IEEE Fourth International Conference on Data Science in Cyberspace (DSC). 2019.
9. Xiaojuan Zhao, Yan Jia, Aiping Li, Rong Jiang, Yichen Song In: World Wide Web (2020)
10. L. Shi, S. Li, X. Yang, J. Qi, G. Pan and B. Zhou Semantic health knowledge graph: Semantic integration of heterogeneous medical knowledge and services. In: BioMed Research International 2017, 1–12
11. E H Shortliffe Evolution of electronic medical records. In: Academic medicine, AAMC, 1999
12. Richard Hillestad, James Bigelow, Anthony Bower, Federico Girosi, Robin Meili, Richard Scoville, and Roger Taylor Can Electronic Medical Record Systems Transform Health Care? Potential Health Benefits, Savings, And Costs. In: Health affairs, Volume 24, 2005.
13. A. Hoerbst, E. Ammenwerth Electronic health records A Systematic Review on Quality Requirements. In: Methods of Information Medicine 2010; 49(04): 320-336
14. R S Evans Electronic health records: then, now and in the future. In: Yearbook of Medical Informatics 2016; 25(S 01): S48-S61
15. Huaqiong Wang, Xiaoyu Miao and Pan Yang Design and Implementation of Personal Health Record Systems based on Knowledge Graph. In: 2018 9th International Conference on Information Technology in Medicine and Education, IEEE computer society.
16. Antonio Messina, Haikal Pribadi, Jo Stichbury, Michelangelo Bucci, Szymon Klarman, and Alfonso Urso BioGrakn: A Knowledge Graph-Based Semantic Database for Biomedical Sciences. In: Advances in Intelligent Systems and Computing · July 2017.
17. Maria-Esther Vidal, Samaneh Jozashoori, Ahmad Sakor Semantic Data Integration Techniques for Transforming Big Biomedical Data into Actionable Knowledge. In: IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS). 2019.
18. Weixin Zeng, Xiang Zhao, Wei Wang, Jiuyang Tang, Zhen Tan Degree-Aware Alignment for Entities in Tail. In Proceedings of ACM Conference, Washington, DC, USA, July 2017
19. Z Sun, W Hu, C Li Cross-lingual Entity Alignment via Joint Attribute-Preserving Embedding. In: International Semantic Web Conference, 2017 – Springer
20. Gustavo de Assis Costa, José Maria Parente de Oliveira Linguistic Frames as Support for Entity Alignment in Knowledge Graphs. In: 20th International Conference on Information Integration and Web-based Applications & Services (iiWAS 2018). ACM, New York, NY, USA.
21. Li-an Huang and Xiangfeng Luo EASA: Entity Alignment Algorithm Based on Semantic Aggregation and Attribute Attention. In IEEE Access, 2020
22. Zhihuan Yan, Rong Peng, Yaqian Wang, Weidong Li CTEA: Context and Topic Enhanced Entity Alignment for knowledge Graphs. In: Neurocomputing, Elsevier, 2020.
23. Zhichun Wang, Qingsong Lv, Xiaohan Lan, Yu Zhang Cross-lingual Knowledge Graph Alignment via Graph Convolutional Networks. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 349–357 Brussels, Belgium, October 31 - November 4, 2018. c 2018 Association for Computational Linguistics
24. Bayu Distiawan Trisedya, Jianzhong Qi and Rui Zhang Entity Alignment between Knowledge Graphs Using Attribute Embeddings. In: 2019, Association for the Advancement of Artificial Intelligence
25. Yuting Wu , Xiao Liu , Yansong Feng , Zheng Wang, Rui Yan and Dongyan Zhao Relation-Aware Entity Alignment for Heterogeneous Knowledge Graphs. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-2019)
26. Qi Zhu, Hao Wei, Bunyamin Sisman, Da Zheng, Christos Faloutsos, Xin Luna Dong, Jiawei Han Collective Multi-type Entity Alignment Between Knowledge Graphs. In: Proceedings of The Web Conference 2020 (WWW'20), April 20–24, 2020, Taipei, Taiwan. ACM, New York, NY, USA.

27. Dhivya Chandrasekaran and Vijay Mago Evolution of Semantic Similarity—A Survey. In: ACM Comput. Surv. 54, 2, Article 41 (February 2021)
28. Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean Efficient Estimation of Word Representations in Vector Space. In ICLR Conference , September 2013.
29. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 December, 2013.
30. P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov Enriching word vectors with subword information. In: Trans. Assoc. Comput. Linguistics, vol. 5, pp. 135–146, 2017.
31. Y. Hao, Y. Zhang, S. He, K. Liu, and J. Zhao A joint embedding method for entity alignment of knowledge bases. In China Conference on Knowledge Graph and Semantic Computing. Springer, 2016, pp. 3–14.
32. Sunilkumar P, Athira P Shaji A survey on cosine similarity. In: ICAI3, IEEE, 2019.
33. Diego Collarana, Mikhail Galkin, Ignacio Traverso-Ribón, Maria-Esther Vidal, Christoph Lange, Sören Auer MINTE: Semantically Integrating RDF Graphs. In: WIMS, ACM, 2017
34. Diego Collarana, Mikhail Galkin,Christoph Lange, Simon Scerri, Soren Auer and Maria A Vidal Synthesizing knowledge graphs from web sources with the MINTE+ framework. In: Proceedings of ISWC, 2018.
35. https://www.who.int
36. https://www.cdc.gov
37. https://www.diabetes.org.uk
38. https://www.nih.gov
39. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. G. Ives DBpedia: A nucleus for a web of open data. In: Proc. Int. Semantic Web Conf., 2007, pp. 722–735.
40. https://vos.openlinksw.com/owiki/wiki/VOS/VOSSPARQL
41. Xiaobin Tang, Jing Zhang, Bo Chen, Yang Yang, Hong Chen, Cuiping Li. BERT-INT: A BERT-based Interaction Model For Knowledge Graph Alignment. In: Proceedings of IJCAI, 2020.
42. Xuhui Jiang, Yinghan Shen, Zhichao Shi, Chengjin Xu, Wei Li, Zixuan Li, Jian Guo, Huawei Shen, Yuanzhuo Wang. Unlocking the Power of Large Language Models for Entity Alignment. arXiv:2402.15048, 2024.
43. Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, Yuzhong Qu. Knowledge Graph Alignment Network with Gated Multi-hop Neighborhood Aggregation. In: Proceedings of AAAI, 2020.
44. Xin Mao, Wenting Wang, Yuanbin Wu, Man Lan. Boosting the Speed of Entity Alignment 10×: Dual Attention Matching Network with Normalized Hard Sample Mining. In: Proceedings of The Web Conference (WWW), 2021.
45. Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, Chengkai Li. A Benchmarking Study of Embedding-based Entity Alignment for Knowledge Graphs. Proceedings of the VLDB Endowment, 13(11), 2020.
46. Amit Singhal. Introducing the Knowledge Graph: things, not strings. Official Google Blog, May 2012.
47. Aidan Hogan, Eva Blomqvist, Michael Cochez, et al. Knowledge Graphs. ACM Computing Surveys, 54(4), Article 71, 2021.
48. Kaisheng Zeng, Chengjiang Li, Lei Hou, Juanzi Li, Ling Feng. A Comprehensive Survey of Entity Alignment for Knowledge Graphs. AI Open, 2, 2021.
49. Sushmita Singh, Manvi Siwach. Evaluating diabetes dataset for knowledge graph embedding based link prediction. In: Data & Knowledge Engineering, Volume 157, Article 102414, May 2025. DOI: 10.1016/j.datak.2025.102414.