International Journal of Advanced Smart Convergence Vol.12 No.3 75-88 (2023)
http://dx.doi.org/10.7236/IJASC.2023.12.3.75

IJASC 23-3-10

Ontology Matching Method Based on Word Embedding and Structural Similarity

Hongzhou Duan[1], Yuxiang Sun[2] , and Yongju Lee[3]

[1]PhD Student, School of Computer Science and Engineering, Kyungpook National University, Korea
[2] Doctor, Software Technology Research Center, Kyungpook National University, Korea
[3] Professor, School of Computer Science and Engineering, Kyungpook National University, Korea
E-mail: [1]caixiuming1984@163.com, [2]syx921120@gmail.com, [3]yongju@knu.ac.kr

Abstract

In a specific domain, experts have different understanding of domain knowledge or different purpose of constructing ontology. These will lead to multiple different ontologies in the domain. This phenomenon is called the ontology heterogeneity. For research fields that require cross-ontology operations such as knowledge fusion and knowledge reasoning, the ontology heterogeneity has caused certain difficulties for research. In this paper, we propose a novel ontology matching model that combines word embedding and a concatenated continuous bag-of-words model. Our goal is to improve word vectors and distinguish the semantic similarity and descriptive associations. Moreover, we make the most of textual and structural information from the ontology and external resources. We represent the ontology as a graph and use the SimRank algorithm to calculate the structural similarity. Our approach employs a similarity queue to achieve one-to-many matching results which provide a wider range of insights for subsequent mining and analysis. This enhances and refines the methodology used in ontology matching.

Keywords: *Ontology Heterogeneity, Ontology Alignment, Word Embedding, Text Similarity, Structural Similarity*

1. Introduction

By constructing structured resources that exhibit interrelatedness, ontologies establish connections among Web resources within Semantic Web, consequently augmenting the reusability of domain knowledge in the representation of concepts and their associations [1]. Specialist domain vocabularies and semantic interpretations of their constituent terms are defined through ontologies [2]. While domain ontologies are formulated by domain experts and scholars, variations can arise among different researchers within the same domain due to their distinct understandings of domain knowledge. This lack of the standardized framework results in the existence of multiple ontology models within the same domain, giving rise to the phenomenon known as the ontology heterogeneity. Ontology matching, also referred to as ontology alignment, addresses the challenge of ontology heterogeneity by comparing concepts across two or more ontologies and establishing

Manuscript Received: July. 10, 2023 / Revised: July. 17, 2023 / Accepted: July. 20, 2023
Corresponding Author: Yongju Lee (yongju@knu.ac.kr)
Tel:*** - **** - **** Fax: +82-53-957-4846
Professor, School of Computer Science and Engineering, Kyungpook National University, Korea

mapping relationships among them. By means of ontology matching, ontologies can achieve a mutual comprehension through the utilization of mapping relationships between concepts, facilitating cross-ontology research endeavors [3].

The objective of this paper is to propose a novel ontology alignment model that effectively addresses the issue of ontology heterogeneity among distinct ontologies within a specific professional domain. We use word vectors to enhance the representation of original semantic information associated with concepts, enabling a more precise characterization of inter-concept relationships. Leveraging the advancements in machine learning, deep learning, and related techniques, word vectors offer a promising approach for effectively capturing the original semantic information of concepts and accurately depicting their relationships. Various text embedding methods have been employed in this study to leverage textual information within the ontology, catering to different problem scenarios..

(1) Owing to the training properties of Word2Vec [4], frequently co-occurring words in text tend to possess similar word vectors. However, word vectors of the Word2Vec model lack the capability to effectively differentiate between semantically similar yet distinct concepts. To address this limitation, we employ the SCBOW (Siamese Continuous Bag-of-Words) model [5]. By incorporating additional information, such as ontology entity synonyms, we refine the vectors to more accurately represent the textual information associated with ontology entities. This approach aims to enhance the semantic specificity and descriptive differentiation within the ontology.

(2) In the context of ontology matching tasks, the phenomenon of "word polysemy" is observed when two entities share the same string but do not constitute a matching pair. To address this issue, we employ the BERT (Bidirectional Encoder Representations from Transformers) model [6], which calculates dynamic word vectors by considering contextual information. This enables us to obtain distinct embedding outcomes for the same entity in different contexts. By leveraging the BERT model, we effectively tackle the challenge posed by "word polysemy," ensuring that multiple meanings of words are appropriately captured and resolved during the ontology matching process.

To address the ontology similarity matching problem, the traditional stable matching algorithm (SMA) [7] has been widely used to identify optimal and stable matching solutions accurately. However, SMA is limited to one-to-one matches, whereas ontology matching tasks often involve not only one-to-one matches but also many-to-many matches based on the granularity of ontology construction. As a result, SMA is not suitable for such scenarios. To overcome this limitation, we employ the SimRank method [8].

The fundamental concept of SimRank is that two nodes exhibit similarity if their respective neighbors also exhibit similarity in some way. Nevertheless, the original SimRank method calculates similarity within the same graph. Hence, we utilize the matching results from the preceding step to select a specific number of anchor points for establishing correlations between the two graphs. Furthermore, we incorporate textual similarity between entities as weights to modify the SimRank algorithm. The resulting structural similarity obtained from the adapted SimRank algorithm is then employed to verify the candidate matches obtained in the previous step and identify new matches.

The remaining sections of this paper are organized as follows. In Section 2, an overview of relevant studies is provided. Section 3 presents an ontology matching model based on word embedding, which includes the introduction of the model and the calculation of structural similarity. Section 4 conducts a performance analysis. Finally, in Section 5, conclusions of our study are presented, and future research directions for further exploration are proposed.

2. Related Work

2.1 Word2Vec

Word2Vec, proposed by Mikolov et al. [9], is a technique that utilizes contextual information within a corpus to train word vectors. It involves mapping high-dimensional one-hot encoded vectors to lower-dimensional spaces, enabling the vectorization of words. These trained word vectors are then utilized in subsequent text processing tasks. Word2Vec is structured as a three-layer neural network model comprising input, hidden, and output layers. This approach addresses the limitations of one-hot encoding, which fails to capture semantic information in text and yields large, sparse vectors. The input layer consists of a unique binary-encoded vector of dimension V, with only one dimension containing a value of 1. The hidden layer of the neural network employs a linear unit without an activation function. The output layer employs the SoftMax function to generate results, forming a fully connected network. Through iterative training and parameter adjustment, Word2Vec eventually produces a word embedding matrix, with the word vectors calculated from the output of the hidden layer.

2.2 Siamese Continuous Bag-of-Words Model

Upon comparing word vectors obtained from training the Word2Vec model, we observed that instances of similar text similarity between two entities may actually indicate cases of descriptive association. To ensure that word vectors reflect semantic similarity between the two entities and to mitigate the influence of descriptive association, we incorporated the SCBOW model to enhance the word vectors. The SCBOW model was initially proposed by Tom Kenter as an effective approach to address the limitations of word vectors trained using methods like Word2Vec, which were not specifically optimized for sentence representation tasks. Kenter argued that the context of a sentence within a paragraph, specifically the sentence preceding and the sentence following it, exhibit higher contextual similarity to each other compared to a randomly selected sentence from the paragraph. Recognizing this higher contextual similarity, Kenter constructed the SCBOW model based on the siamese long and short-term memory network (Siamese LSTM) [10].

The model aims to learn semantic representations of concepts by encoding them as word embedding vectors within ontologies using the continuous bag-of-words model. Simultaneously, it utilizes a siamese network structure to assess the similarity between two concepts, with each concept encoded through the same embedding and representation network. During training, the model learns feature representations of matching concepts by minimizing the similarity loss function. This optimization process brings similar concepts closer together in vector space, facilitating more accurate semantic representation and comparison of concepts within ontologies.

2.3 BERT

Word vectors obtained from the Word2Vec model are static, lacking the ability to distinguish between multiple meanings of a word. To address this limitation, the introduction of transformer-based models, such as BERT, has proven to be beneficial. BERT can be seen as an improvement upon the Transformer model. BERT is a bidirectional language model that leverages contextual information by employing stacked encoders from the transformer model [11]. Figure 1 depicts the schematic of BERT, showcasing the presence of multiple stacked transformers [12]. BERT utilizes a bidirectional approach to model language and incorporates contextual information. In the figure, N represents the length of input sentence, E represents input embedding of the oldest word in the sentence, and T represents the feature vector obtained from the *i*-th word after BERT

processing.

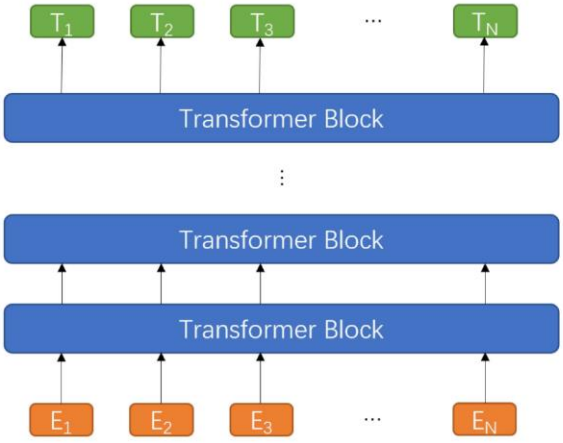

Figure 1. BERT Schematic diagram of the structure.

The key innovation of BERT lies in its bidirectional pre-training approach. Unlike traditional language models that predict the current word based on only the left or right side of the context, BERT utilizes both left and right side contextual information for making predictions. By incorporating this two-way modeling strategy, BERT achieves a deeper understanding of the meaning and context of words in various contexts. This bidirectional approach enhances BERT's ability to capture intricate semantic relationships and dependencies within the text.

2.4 Calculation of Structural Similarity

The calculation of structural similarity between entities in an ontology involves utilizing relationships such as "is-a" and "subclass-of" present in the ontology. By determining the adjacency of each entity based on these relationships, entities in the ontology can be represented as a graph structure. The SimRank algorithm is then employed to discover the similarity between nodes within graph structures extracted from two ontologies.

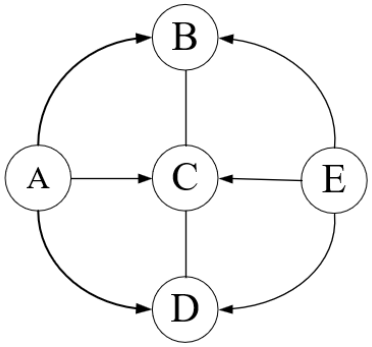

Figure 2. SimRank example.

The fundamental principle of the SimRank algorithm is that similar objects are referenced by other similar objects [13]. When applied to the context of graphs, the SimRank algorithm operates as follows; if neighboring

nodes of two nodes exhibit a certain degree of similarity, it implies that the two nodes themselves possess a certain level of similarity. To illustrate, consider a graph with five nodes (A, B, C, D, E). If node A is adjacent to nodes B, C, and D, and node E is also adjacent to nodes B, C, and D, as depicted in Figure 2, we can conclude that nodes A and E share a certain degree of similarity. It is worth noting that a node is most similar to itself. The similarity of points (a, b) in SimRank is shown in equation (1):

$$S(a,b) = \begin{cases} 1 & a = b \\ \frac{C}{|N(a)||(b)|}\sum_{i=1}^{|N(a)|}\sum_{j=1}^{|N(b)|} S\left(N_i(a), N_j(b)\right) & |N(a)||N(b)| \,!=\, 0 \\ 0 & |N(a)||N(b)| \,=\, 0 \end{cases} \quad (1)$$

When two nodes (a, b) represent the same entity (a = b), their similarity is inherently the highest, with a similarity value of 1. However, when a != b, the similarity between a and b is calculated based on the following approach. It is determined by taking the average similarity of all combinations between neighbors of nodes a and b, multiplied by a constant coefficient c. If either node a or node b has no neighbors (i.e., the cardinality of the set of neighbors for either node is zero), indicating the absence of similar nodes in their respective neighborhoods, the similarity between a and b is 0.

3. Ontology Matching Method Based on Word Embedding

3.1 Model Overview

The conceptual representation of the ontology alignment model grounded in word embedding is illustrated in Figure 3. The model consists of three sequential stages: data acquisition and preprocessing, computation of textual similarity, and alignment of results based on textual similarity. These stages ultimately lead to the derivation of the final alignment outcomes.

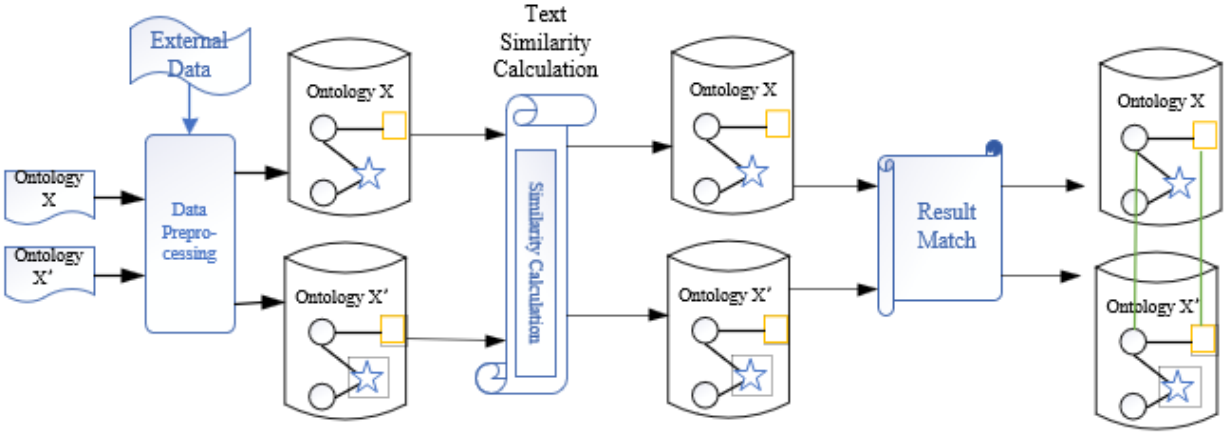

Figure 3. Schematic diagram for ontology matching model based on word embedding.

Ontology files of the OWL format are parsed to extract the necessary textual and structural information required for the ontology matching process. Additionally, external information is obtained from various Web sites such as ConceptNet and WikiSynonyms to enhance the experimental process. ConceptNet, developed by MIT, is a freely available semantic network that aids computers in understanding the meaning of natural

language words [14]. ConceptNet offers valuable information on synonyms, multilingual translations, and example sentences associated with concepts. It provides an API interface, enabling users to access information about query concepts through the official API. WikiSynonyms, on the other hand, leverages Wikipedia redirects to facilitate the discovery of term synonyms [15]. It also offers an API query service, allowing users to retrieve external data by constructing specific links based on the queried word. The WikiSynonyms library contains a collection of synonyms for various terms.

In the text similarity computation stage, we utilize Word2Vec and BERT methodologies to assign semantic representations to the text elements in the ontology. Word2Vec transforms textual fragments into vectorized forms with semiotic content. On the other hand, BERT generates dynamic word vectors based on contextual cues, allowing for the distinction of multiple meanings of certain words. However, Word2Vec-constructed vectors may not effectively differentiate between description association and semantic similarity. To address this, we introduce the twin continuous bag-of-words model, which enhances the word vectors and makes them more suitable for the ontology alignment task explained in this paper. Finally, during the alignment stage, the pursuit of steadfast and accurate alignment outcomes necessitates the application of a stable matching algorithm for the calculation of alignments.

3.2 Text Similarity Calculation

Within conventional ontology matching frameworks, entity similarity is primarily evaluated using a string-oriented approach, often utilizing metrics such as edit distance. However, this method, which is based on string similarity computation, lacks comprehensiveness as it fails to capture semantic nuances between entities. For example, synonyms like 'tumor' and 'neoplasm', both referring to a type of malignancy, are unfortunately considered distant from a string-matching perspective, hindering accurate matching.

To surmount this quandary, a promising avenue entails embedding each entity within a vector space, facilitating comparative analysis. The most straightforward vectorization method is the one-hot coding. This technique constructs a vocabulary inclusive of all terms present within the dataset. Each term within the vocabulary corresponds to a distinct dimension within the vector space. Consequently, the vector representation for a term, when encoded using one-hot coding, assumes a '1' value exclusively within the dimension corresponding to the term, while all other dimensions are rendered '0'. Despite of its efficacy in the text-to-vector transformation, it becomes that the vector's dimensions equate to the vocabulary's size. In scenarios involving expansive datasets, the resultant vectors grow exceedingly large in dimensionality, propagating sparsity. This, in turn, precipitates inefficiencies in both storage and computation.

Word vectors derived from training with the Word2Vec model transcend the limitations of one-hot encoding by mapping terms to a lower-dimensional vector space, encapsulating semantic nuances from the training dataset. Word2Vec's training process necessitates the integration of contextual word information, operationalized through the introduction of a window size parameter. This parameter delineates the scope of context during training. Notably, when the window size exceeds 1, the Word2Vec model trains word vectors. It emerges that when words share proximity in context and recurrently co-occur, their training data tends to converge, fostering the similarity of resultant word vectors. This phenomenon is referred to as "descriptive association", indicative of terms frequently utilized together. Word2Vec-trained word vectors exhibit similarity in two aspects: semantic meaning and descriptive association. Consider the instance where "harness" is commonly coupled with "horse", fostering nearness in their word vectors. Nonetheless, their semantic significance remains disparate. In the realm of ontology matching, the aim is to ascertain semantic congruity among entities, rather than discerning descriptive relationships. Consequently, an enhancement to the

preliminary word vectors becomes requisite. In this pursuit, the twin continuous bag-of-words (CBOW) model is employed for vector refinement.

The occurrence of word polysemy surfaced during ontology matching endeavors. Polysemy herein denotes instances where identical strings do not culminate in definitive matches. Notably, the static nature of Word2Vec-derived and twin CBOW-enhanced word vectors impedes the differentiation of word meanings within context. This prompted the adoption of BERT for word vector embedding. BERT's dynamic word vectors are contingent upon context and proficiently distinguishing multitudinous word meanings. The BERT model requires contextual information to generate the corresponding word vectors. To establish an appropriate context for the concepts to be embedded, we construct a concept sentence for each concept provided in the ontology. This concept sentence serves as the contextual background for generating the word vectors, as illustrated in equation (2).

$$Concept\ sentence = name + info \tag{2}$$

The variable info refers to the information pertaining to a concept. It is important to note that different ontologies can possess varying levels of information about a specific concept and leading to the information asymmetry. For instance, relying solely on the definition of an entity found within an ontology to construct a contextual context does not guarantee that the ontology provides a comprehensive definition for every entity.

3.3 Word Vector Enhancement

As described in Section 2.2, Tim Kenter proposed the twin CBOW model with the principal aim of elevating the final computation of sentence vectors. This is achieved through averaging word vectors, thereby augmenting the values inherent in initial word vectors. In the context of ontology matching, the computational process for deriving entity embeddings mirrors the approach utilized for sentences. This involves summating and averaging word vectors for each constituent word, culminating in the ultimate embedding outcome. The emulation of sentence embedding serves as a blueprint for identifying appropriate positive and negative examples, facilitating the application of the twin CBOW model to enhance word vectors. The ontology matching model, being unsupervised in nature, mandates the creation of self-constructed positive and negative example datasets for training. Guided by the ontology matching task's objectives, the impetus for enhancing word vectors centers on ensuring that proximate semantic similarity results between two entities authentically signify semantic similarity, rather than mere descriptive associations. This entails a distinction between semantic similarity and descriptive associations. Consequently, synonyms of entities are employed as positive examples within the training process. Conversely, entities that exhibit nearness in semantic similarity, yet diverge from synonymity or close synonymity, function as negative examples. This yields word vectors attuned to the nuances of ontology matching and, by extension, entity embedding outcomes. In this manner, the transformation of word embedding methodologies, initially designed to optimize sentence representations, has been adeptly extended to enhance entity representations, thereby bolstering the ontology matching task.

The twin CBOW model encompassed within this framework comprises four layers of a neural network: input layer, mapping layer, cosine layer, and prediction layer. Leveraging Word2Vec-trained word vectors as the foundation, subsequent refinements engender the initial word embedding matrix, iteratively updated during training. The resultant matrix constitutes the augmented word vectors. Within the input layer, the unit of input transitions from sentences to entities. For each entity 'e' within the ontology dataset, its synonyms serve as

positive examples, while concepts exhibiting heightened similarity—yet not synonyms—are enlisted as negative examples, computed from the initial word vectors. These positive and negative examples are jointly inputted. Subsequently, each word within the entity undergoes transformation via the word vector matrix, converting one-hot encoding into word vector format. This transformed input feeds into the mapping layer. The mapping layer involves the summation and averaging of word vectors representing entity constituents, culminating in an output vector that serves as the entity's representation in the subsequent neural network layer. The cosine layer calculates direct cosine similarity among entity vectors, functioning as a metric for entity-positive and entity-negative example relationships. In the prediction layer, normalized cosine similarity values (post-Softmax normalization) serve as the final model prediction.

Integrating the SCBOW model into the word embedding paradigm for optimal entity representation inherently yields an analogous formulation for the anticipated similarity distribution, akin to the optimal sentence representation task, delineated by Equation (3).

$$p(S_i, S_j) = \begin{cases} \frac{1}{|s_i^+|}, & if\ S_j \in S_i^+ \\ 0, & if\ S_j \in S_i^- \end{cases} \tag{3}$$

3.4 Result Matching

Stable matching algorithms find diverse applications. Consider a scenario comprising a group of medical students and a set of hospitals. Each medical student and hospital possesses distinct preferences and requirements. Medical students desire assignment to hospitals aligning with their preferences, while hospitals seek medical students meeting their specific criteria. Nevertheless, given the limited number of available positions, a matching procedure is necessitated to determine the final allocation. In such instances, the implementation of a stable matching algorithm ensures the stability of pairings between medical students and hospitals. Medical students express their preferences by applying to their preferred hospitals, while hospitals select medical students based on their backgrounds and preferences. In the event of a medical student's rejection, they continue to apply to their subsequent preferred hospitals until a successful match is achieved.

A match is deemed unstable if the following conditions hold: within the first set, an element A already matched prefers element C in the second set over element B, which is already matched by A. Simultaneously, element C prefers A over element D, with D already matched. In essence, when no matches (A, B) exist, and both A and B prefer each other more than their current matches, the match is considered stable. For instance, consider two ontologies denoted as 1 and 2, each containing elements {a, b, c} and {A, B, C}. The preference ordering for these elements is as follows: {a: B A C, b: C B A, c: A C B}, {A: b a c, B: c b a, C: a c b}. The pairing (aB, bC, cA) exemplifies a set of stable matching outcomes.

Through analysis, it becomes apparent that the two entities within the ontology utilize vector representations derived from textual information to compute cosine similarity, indicating the degree of mutual preference. This cosine similarity serves as the foundation for sorting and ordering the entities. By doing so, the challenge of entity matching is effectively addressed. The matching algorithm concludes when all medical students are matched with hospitals. Nevertheless, in cases where the last medical student's preference for the unmatched hospital and the unmatched student is low, a match still transpires. This dynamic can potentially result in disparate entities in the final matching result. To mitigate this, we introduce a threshold, termed "death". This threshold ensures that two entities are only added to the match list when their similarity surpasses the "death" threshold, thus enhancing the quality of the matching outcomes.

3.5 Structural Similarity Calculation

In the ontology matching task, it is indeed possible to have one-to-many matching results due to differences in granularity between ontologies. To handle such cases, a traditional matching method is employed by setting a threshold value (t) for similarity. If the similarity between two entities exceeds the threshold, they are considered as a match and added to the candidate match results. This approach allows for the identification of one-to-many matches between entities [16]. In a subsequent step, the structural similarity is checked to ensure that the candidate matches are valid. This step involves evaluating the similarity of the graph structures surrounding the matched entities. By considering both textual and structural similarities, the matching algorithm can identify both one-to-one and one-to-many matches, allowing for a comprehensive comparison with the results obtained from the stable matching algorithm. To address this situation, we consider using the structural information in the ontology to calculate the structural similarity to check the matching results. The schematic diagram of the ontology matching model with the introduction of structural similarity is shown in Figure 4.

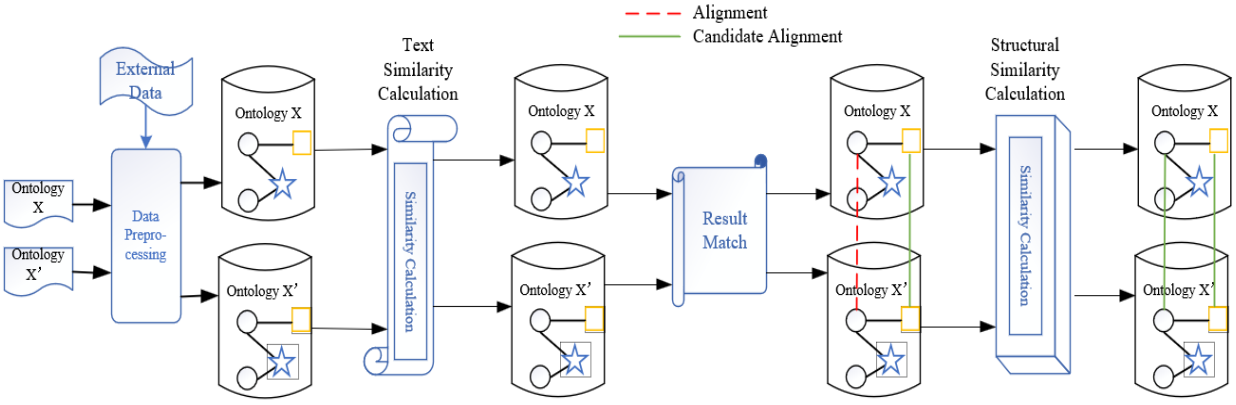

Figure 4. Ontology matching model based on structural similarity.

In the ontology matching process, the SimRank algorithm is utilized to calculate the structural similarity between entities. Ontologies are first transformed into graph structures using their structural information. Then, the text similarity information obtained from the previous step is used to establish connections between the two graph structures, combining them into a single graph. The SimRank algorithm is applied to measure the structural similarity between entities in two ontologies. It assesses the similarity based on the similarity of their neighboring nodes in the graph. By considering the structural similarity and the textual similarity obtained previously, the algorithm filters the final matching results. In this model, the nodes in the initial anchor matching, which have been filtered based on their constant similarity calculated using textual information, are selected as the results. This selection ensures that the structural information within each subgraph does not influence the structural information in the other subgraph. The process is depicted in Figure 5, illustrating how the model integrates textual and structural information to obtain the final matching results.

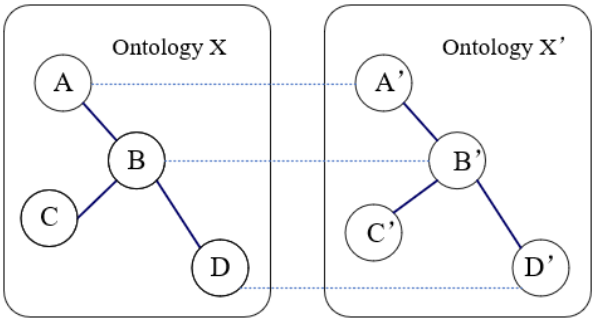

Figure 5. Ontology matching using SimRank algorithm.

Equation (4) indicates how the similarity between two nodes is calculated in the ontology matching process. When the two nodes are the initial anchor matches obtained in the previous step, their similarity is determined by calculating the cosine similarity using word embedding techniques. This captures the textual similarity between the nodes. However, if the two nodes are not initial anchor matches, their similarity is computed differently. It is calculated by summing the similarities between their neighboring nodes and then averaging the resulting similarity. This approach takes into account the structural information by considering the similarity of neighboring nodes. In cases where one of the nodes is isolated, meaning it has no neighboring nodes connected to it, the structural similarity between the two nodes is very low, resulting in a similarity value of 0. This reflects the lack of structural resemblance between an isolated node and other nodes in the graph.

$$S(a,b) = \begin{cases} sim(a,b) & if(a,b) \in \mathrm{A} \\ \frac{C}{|N(a)||N(b)|}\sum_{i=1}^{|N(a)|}\sum_{j=1}^{|N(b)|} S\left(N_i(a), N_j(b)\right) & |N(a)||N(b)| \,!= 0 \\ 0 & |N(a)||N(b)| = 0 \end{cases} \quad (4)$$

The SimRank algorithm is written in matrix form, as shown in equation (5). The last two terms of the formula are designed to preserve the original similarity values between nodes that have non-zero initial similarity. This ensures that the structural similarities within the two subgraphs do not affect each other. In the formula, P(M, Ro) represents a matrix that contains the non-zero elements of the Ro matrix from the matrix M. By multiplying P(M, Ro) with the damping factor and applying element-wise operations, the last two terms ensure that the similarity values of the nodes with non-zero initial similarity remain unchanged. This prevents the structural similarities within the individual subgraphs from interfering with each other during the similarity calculation process.

$$R = CW^T RW + R_0 - P(CW^T RW, R_0) \quad (5)$$

4. Performance Analysis

4.1 Experimental Datasets

FMA (Foundational Model of Anatomy) [17] is an information resource integrated into the distributed framework for anatomical information. It was initiated, developed, and maintained by the Structural

Informatics Group at the University of Washington since 1994. FMA serves as a computer-based knowledge source in the field of biomedical informatics. It is focusing on the representation of classes, types, and relationships essential for describing the phenotypic structure of the human body. It is designed to be comprehensible to humans while also being parseable and interpretable by computer systems. FMA specifically functions as a specialized domain ontology, capturing explicit knowledge in human anatomy and providing a framework that can be applied and expanded to other species as well.

SNOMED CT (Systematized Nomenclature of Medicine - Clinical Terms) is a computer-processable collection of medical terms that is maintained by the SNOMED International organization. It provides comprehensive information on numbering, terminology, synonyms, and definitions for clinical documentation and medical terminology [18]. SNOMED CT is recognized as the world's most extensive multilingual clinical medical terminology [19]. It covers a wide range of conditions, diagnoses, body structures, drugs, devices, and specimens.

NCIT (National Cancer Institute Thesaurus) is a publicly available terminology collection developed by the US National Cancer Institute. It is based on a descriptive logic approach [20]. NCIT offers several features, including a stable and unique code for each biomedical concept, information on preferred terms, synonyms, and external source codes. It contains over 100,000 distinct text definitions and more than 500,000 cross-linkages between concepts. Moreover, NCIT provides an extensive set of terms and definitions based on descriptive logic.

The accuracy rate is a commonly used evaluation metric for binary classification tasks in ontology matching. It measures the proportion of correct predictions made by the model, considering both true positive (TP) and true negative (TN) cases. It represents the effectiveness of the model in accurately classifying positive cases and negative cases. A higher accuracy rate indicates better performance in ontology matching, as it reflects the model's ability to make correct predictions. Recall calculates how many results that are actually positive examples are correctly predicted by the model, i.e. the effect of an answer check all. The F1 value is the result of combining precision and recall, and is the summed average of the two. The calculation formula of the accuracy, recall, and F1value is shown in equation (6).

$$Precision = \frac{TP}{TP+FP} \qquad Recall = \frac{TP}{TP+FN} \qquad F1 = \frac{2*Precision*Recall}{Precision+Recall} \tag{6}$$

4.2 Experimental Results

In the analysis of the matching answers provided by the datasets (FMA-NCIT and FMA-SNOMED), we observe that significant proportion of matches was the one-to-many type, as indicated in Table 1. However, the stable marriage matching algorithm used in the study can only find one-to-one matches, which restricts the maximum recall value achievable by the algorithm. This limitation prompted the exploration of alternative matching methods using a set threshold, in order to compare their performance with the stable matching algorithm.

Table 1. Statistics of matching results

Dataset	Number of Matches	Number of 1:N Matches
FMA-NCIT	2679	349
FMA-SNOMED	5915	782

After the initial word vector was improved using the SCBOW model, entities were matched using the stable matching algorithm and the traditional method of setting thresholds, respectively, and the structural similarity was calibrated using SimRank to obtain the final matching results, which are shown in Table 2 for the FMA-NCIT dataset and Table 3 for the FMA-SNOMED dataset.

Table 2. Matching results for FMA-NCIT dataset

	Precision	Recall	F1
SCBOW+AtableMarriage	0.905	0.752	0.817
SCBOW+SimpleMatch+SimRank	0.852	0.779	0.802
SCBOW+AtableMarriage+SimRank	0.891	0.753	0.818
LogMap	0.914	0.837	0.872

Table 3 Matching results for FMA-SNOMED dataset

	Precision	Recall	F1
SCBOW+AtableMarriage	0.767	0.699	0.731
SCBOW+SimpleMatch+SimRank	0.645	0.703	0.689
SCBOW+AtableMarriage+SimRank	0.715	0.688	0.730
LogMap	0.917	0.671	0.778

The matching methods using SimpleMatch and SimRank are not as effective as using the stable marriage matching algorithm directly, as the former requires too many thresholds to be set manually, introduces too much noise and does not filter this noise out effectively, resulting in poorer final results. However, the stable matching algorithm is exclusively designed for one-to-one matching outcomes. With the integration of the SimRank method, we can identify one-to-many matching results. In the context of ontology matching, it becomes evident that the structural similarity-based matching approach holds superiority.

5. Conclusion

In this paper, an ontology matching model based on word embedding and structural similarity was proposed and experimentally validated. For the textual information in the ontology, the SCBOW model is used to improve the word vector in order to better calculate the similarity between entities. Meanwhile, in order to solve the problem of "multiple meanings of words" in ontology matching, this paper uses the contextual information of the ontology to construct dynamic word vectors and applies the BERT model to embed the entity names. For the structural information in the ontology, this paper transforms the ontology into a graph structure and uses the SimRank algorithm to calculate the structural similarity between entities. The experimental results show that the proposed model is effective in ontology matching tasks and lays the foundation for cross-ontology research work such as knowledge fusion and knowledge inference. However, further analysis of the experimental results also reveals that there is still some room for improvement of the model, which can be further optimized and improved to enhance its performance and effectiveness.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation

of Korea (NRF) funded by the Ministry of Education (No. 2016R1D1A1B02008553). This study was supported by the BK21 FOUR project (AI-driven Convergence Software Education Research Program) funded by the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea (4199990214394)

References

[1] H. Zhu, X. Xue, C. Jiang, and H. Ren, "Multiobjective Sensor Ontology Matching Technique with User Preference Metrics," Wirel. Commun. Mob. Comput. 2021 (2021), pp. 1–9, https://doi.org/10.1155/2021/5594553.

[2] P. Shvaiko and J. Euzenat, "Ontology Matching: State of the Art and Future Challenges," IEEE Transactions on knowledge and data engineering, Vol. 25, No. 1, pp. 158-176, 2011, https://doi.org/10.1109/TKDE.2011.253.

[3] P. Kolyvakis, A. Kalousis, and D. Kiritsis, "Deep Alignment: Unsupervised Ontology Matching with Refined Word Vectors," Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, pp. 787-798, 2018, https://doi.org/10.3389/fgene.2022.893409.

[4] F. Li, L. Liao, L. Zhang, X. Zhu, B. Zhang, and Z. Wang, "An Efficient Approach for Measuring Semantic Similarity Combining WordNet and Wikipedia," IEEE Access, Vol. 8, pp. 184318-184338, 2020, https://doi.org/10.1109/ACCESS.2020.3025611.

[5] Y. Yana, Q. Dong, and Y. Ruiteng, "A Quantum-like Text Representation based on Syntax Tree for Fuzzy Semantic Analysis," Journal of Intelligent & Fuzzy Systems, Vol. 44, No. 6, pp. 9977-9991, 2023, https://doi.org/10.3233/JIFS-223499.

[6] S. Neutel and M. D. Boer, "Towards Automatic Ontology Alignment using BERT," AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering, 2021, https://doi.org/10.48550/arXiv.2112.02682.

[7] X. Xue and J. Zhang, "Matching Large-scale Biomedical Ontologies with Central Concept based Partitioning Algorithm and Adaptive Compact Evolutionary Algorithm," Appl. Soft Comput. 106, 107343, 2021, https://doi.org/10.1016/j.asoc.2021.107343.

[8] W. Yu, J. McCann, C. Zhang, and H. Ferhatosmanoglu, "Scaling High-quality Pairwise Link-based Similarity Retrieval on Billion-edge Graphs," ACM Transactions on Information Systems (TOIS), Vol. 40, No. 4, pp. 1-45, 2022, https://doi.org/10.1145/3495209.

[9] T. Mikolov, I. Sutskever, K. Chen, et al., "Distributed Representations of Words and Phrases and Their Compositionality," 2013, arXiv preprint arXiv:1310.4546, https://doi.org/10.48550/arXiv.1310.4546.

[10] J. Mueller and A. Thyagarajan, "Siamese Recurrent Architectures for Learning Sentence Similarity," Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30, No. 1, 2016, https://doi.org/10.1609/aaai.v30i1.10350.

[11] J. Devlin, M. W. Chang, K. Lee, et al., "Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding," 2018, arXiv preprint arXiv:1810.04805, https://doi.org/10.48550/arXiv.1810.04805.

[12] Vaswani, N. Shazeer, N. Parmar, et al., "Attention is All You Need, 2017, arXiv preprint arXiv: 1706.03762, https://doi.org/10.48550/arXiv.1706.03762.

[13] G. Jeh and J. Widom, "SimRank: A Measure of Structural-Context Similarity," Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 538-543, 2002, https://doi.org/10.1145/775047.775126

[14] R. Speer, J. Chin, C. Havasi, "ConceptNet 5.5: An Open Multilingual Graph of General Knowledge," Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31, No. 1, 2017, https://doi.org/10.48550/arXiv.1612.03975.

[15] W. Dakka and P. G. Ipeirotis, "Automatic Extraction of Useful Facet Hierarchies from Text Databases," Proceedings of the 2008 IEEE 24th International Conference on Data Engineering, pp. 466-475, 2008, https://doi.orgg/10.1109/1CDE.2008.4497455.

[16] M. Zhao, S. Zhang, W. Li, et al., "Matching Biomedical Ontologies based on Formal Concept Analysis," Journal of biomedical semantics, Vol. 9, No. 1, pp. l-27, 2018, https://doi.org/10.1186/s13326-018-0178-9

[17] C. Rosse and J. L. V. Mejino, "The Foundational Model of Anatomy Ontology, Anatomy Ontologies for Bioinformatics, Springer, London, pp. 59-117, 2008.

[18] D. Lee, R. Cornet, F. Lau, et al., "A Survey of SNOMED CT Implementations," Journal of Biomedical Informatics, Vol. 46, No. 1, pp. 87-96, 2013, https://doi.org/10.1016/j.jbi.2012.09.006

[19] T. Benson, Principles of Health Interoperability HL7 and SNOMED, London, England, Springer, 2012. ISBN 978-1-4471-2800-7.

[20] J. Golbeck, G. Fragoso, F. Hartel, et al., "The National Cancer Institute's Thesaurus and Ontology," Journal of Web Semantics, First Look 1_1_4, 2003, http://dx.doi.org/10.2139/ssrn.3199007.