Deep Model-Based Security-Aware Entity Alignment Method for Edge-Specific Knowledge Graphs

Jongmo Kim 1, Kunyoung Kim 1, Mye Sohn 1,*, and Gyudong Park 2

1 Department of Industrial Engineering, Sungkyunkwan University, Suwon 16419, Korea; dignityc@skku.edu (J.K.); kimkun0@skku.edu (K.K.)
2 2nd R&D Institute, Agency for Defense Development, Seoul 05771, Korea; iobject@add.re.kr
* Correspondence: myesohn@skku.edu; Tel.: +82-31-290-7605; Fax: +82-31-290-7610

Abstract: This paper proposes a deep model-based entity alignment method for the edge-specific knowledge graphs (KGs) to resolve the semantic heterogeneity between the edge systems’ data. To do so, this paper first analyzes the edge-specific knowledge graphs (KGs) to find unique characteristics. The deep model-based entity alignment method is developed based on their unique characteristics. The proposed method performs the entity alignment using a graph which is not topological but data-centric, to reflect the characteristics of the edge-specific KGs, which are mainly composed of the instance entities rather than the conceptual entities. In addition, two deep models, namely BERT (bidirectional encoder representations from transformers) for the concept entities and GAN (generative adversarial networks) for the instance entities, are applied to model learning. By utilizing the deep models, neural network models that humans cannot interpret, it is possible to secure data on the edge systems. The two learning models trained separately are integrated using a graph-based deep learning model GCN (graph convolution network). Finally, the integrated deep model is utilized to align the entities in the edge-specific KGs. To demonstrate the superiority of the proposed method, we perform the experiment and evaluation compared to the state-of-the-art entity alignment methods with the two experimental datasets from DBpedia, YAGO, and wikidata. In the evaluation metrics of Hits@k, mean rank (MR), and mean reciprocal rank (MRR), the proposed method shows the best predictive and generalization performance for the KG entity alignment.

Keywords: edge computing; data privacy and security; entity alignment; knowledge graph; deep model

1. Introduction

As edge computing becomes more prevalent, the collaboration and collaboration between the multiple edge systems (ESs) required to perform a specific task (e.g., service delivery, data privacy, and security) become important [1–3]. At this time, for the ESs to collaborate and cooperate seamlessly, the heterogeneity of the data they must share, particularly the semantic heterogeneity, must be addressed [4,5]. Until recently, the ontology, which was domain-specific and had a rigid structure, was the primary method for solving the semantic heterogeneity of the data in the ESs [6,7]. However, due to the easy and frequent change of the components of the ESs, their operating environments can be very dynamic [8]. In addition, collaboration between heterogeneous ESs for multiple domains is absolutely required to perform some tasks [9]. It is very difficult to accommodate these changes in the ontology with their security and privacy policies. Thus, it is a non-trivial task to develop an ontology that encompasses such diverse domains for the dynamic and collaborative edge computing.

To overcome the limitations of the ontology, attention is given to solving the semantic heterogeneity of the data of the ESs with a knowledge graph (KG) that can structurally represent human knowledge using entities, relations, and semantic descriptions [10]. By
simply adding some entities and their relationships related to the changed environments to the KG, it can rapidly and easily reflect dynamic changes of the ESs. Furthermore, it can develop diverse domain knowledge models that are not limited to a specific domain [11]. At present, as an early stage of applying the KG, researchers are using edge-specific KGs to resolve the semantic heterogeneity of the internal data necessary to ensure the security and privacy of the ESs [12]. However, as cooperation and collaboration between the ESs become common, they will require a new method of using the edge-specific KGs that can resolve the semantic heterogeneity of data required for their cooperation and collaboration while maintaining the security and data privacy of each ES. To propose the new method, we researched as follows.

First, we explore the unique characteristics of the edge-specific KGs. To resolve the semantic heterogeneity of data used in the multiple heterogeneous ESs, the relations between them, especially the equivalence relation, must be clearly identified. Fortunately, entity alignment, which is identified as the equivalence relation between entities in different KGs, has already been proposed [13]. Unfortunately, the KGs used in the entity alignment are the general-purpose KGs, such as DBpedia and YAGO, with different properties to the edge-specific KGs [14]. So, before proposing an entity alignment method suitable for the edge-specific KGs, we conducted research to find the unique characteristics of the edge-specific KGs.

Second, based on the unique characteristics of the edge-specific KGs, we propose a novel entity alignment method for the edge-specific KGs, named the deep model-based security-aware entity alignment method. The properties of the proposed method are as follows. The deep model is used for learning the semantic relations between the concept and instance entities that are the target of the entity alignment. At this time, two deep models, named BERT (bidirectional encoder representations from transformers) [15] for the concept entities and GAN (generative adversarial networks) [16] for the instance entities are applied to train for each because the properties of the concept and the instance entities are very different. In particular, for the security and privacy of the ES data, the data must not be exposed outside the ESs during the entity alignment process. To do so, we utilize the deep models, neural network models that humans cannot interpret. This is because these neural network models can perform the entity alignment considering the security and privacy of the ES data by hiding the model’s input data and hyperparameters.

Third, this paper proposes a graph-based deep learning model that can integrate the learning models for the BERT for the concept entities and the GAN for the instance entities. Although the deep model learning for the entity alignment was separately performed considering the characteristics of the concept and the instance entities, information sharing between them is required to perform the entity alignment robustly because the instance entities as well as the concept entities may be used to describe the concept entities. Conversely, the concept entities, as well as the instance entities, may be used to describe the instance entities. A flexible graph-based deep learning model named GCN (graph convolution network) is utilized to integrate two deep models trained on two types of entities with different characteristics [17]. So, this paper attempts to merge the learned models for concept and instance entities into one deep model by borrowing graph-based deep learning models and transfer learning approaches.

This paper is organized as follows. In Section 2, we describe the characteristics of the general-purpose KG and the edge-specific KG. Section 3 represents the overall architecture of the deep model-based security-aware entity alignment framework and the detailed process of each module. In Section 4, we conduct the experiments to show the superiority of the proposed methods for KG entity alignment in comparison with the comparative methods. The related studies on the KGs in edge computing and KG entity alignment are summarized in Section 5. Finally, Section 6 puts forth the conclusions and suggests further research on the KG entity alignment framework in edge computing and takes security into consideration.
2. General-Purpose KG vs. Edge-Specific KG

As mentioned earlier, the entity alignment method has already been proposed to increase the utilization of the KGs [18]. Currently, conventional entity alignment is performed using the following characteristics of the KGs. In many applications, the KGs are used as the graph-structured knowledge bases (KBs) that can structurally represent human knowledge using entities, relations, and semantic descriptions [10]. Many KGs, including YAGO, DBpedia, and the Google Knowledge Graph, have already been developed and used [19]. These KGs are the general-purpose KGs that include comprehensive and multidisciplinary knowledge. Therefore, they have following properties. First, the general-purpose KGs are created mainly for concept entities with higher levels of abstraction than instance entities with lower levels of abstraction because they represent knowledge across multiple domains. Currently, when using the general-purpose KG, regardless of the domains, it describes in detail the description of the concept entities themselves, their relations to other concept entities (e.g., taxonomic relations), and their literal and textual descriptions. Second, the general-purpose KGs contain multidisciplinary information and are large-scale [20,21]. Therefore, it is highly likely that the entities to be aligned and their information, such as their taxonomic relations and their literal and textual descriptions, will be included in several general-purpose KGs. Multiple general-purpose KGs may contain the same concept and instance entities simultaneously. Of course, depending on the development intention of the general-purpose KGs, the concept and instance entities may or may not be represented in a similar structure. Third, the general-purpose KGs emerged to increase the utilization of information by sharing and disclosing all information, such as the concepts and instances, to users through the connection of multidisciplinary information. In other words, their degree of information openness is very high.

As the goal of this paper is to propose the entity alignment method for the edge-specific KGs, we analyze the properties of the edge-specific KGs in terms of the level of abstraction, the degree of data overlap, and the information openness, which are the properties of the general-purpose KGs. The first is the level of abstraction. Unlike the general-purpose KGs, which are developed to perform domain-independent operations, the edge-specific KGs are created to support only the functions of the corresponding ESs [22]. Furthermore, the primary function of the edge-specific KGs is to solve the heterogeneity of the data collected by the sensing devices of the edges; so, they are composed mainly of the instance entities rather than the concept entities. Contrary to the concept entities, as most instance entities are leaf nodes of the KGs they lack taxonomic relations, literals, and textual descriptions. The second is the degree of overlapped data. As the edge-specific KG contains only information related to the specific ESs, the scale is small, and the information contents are very specific. Therefore, the possibility that the edge-specific KGs contain the overlapped information necessary for the entity alignment is very low. The third is the difference in the degree of information openness. The edge-specific KGs disclose some concepts inherited from the general-purpose KGs but suppress external exposure by modeling the users’ private information or the sensory data of the ESs, to be protected as their instances. Therefore, their degree of information openness is very low. In this light, the conventional entity alignment methods, which are conducted using the characteristics of the general-purpose KGs, cannot apply the alignment of the edge-specific KGs. So, this paper proposes a novel entity alignment method specialized in the edge-specific KGs. The comparison results are summarized in Table 1.

Table 1. Comparison of general-purpose KG and edge-specific KGs.

<table>
<thead>
<tr>
<th></th>
<th>General-Purpose KG</th>
<th>Edge-Specific KG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level of abstraction</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Degree of data overlap</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Degree of information openness</td>
<td>High</td>
<td>Low</td>
</tr>
</tbody>
</table>
Based on the above analysis, we discovered the characteristics that the entity alignment method of the edge-specific KGs should have; these are as follows.

First, the characteristics of the edge-specific KGs, which are mainly composed of the instance entities rather than the conceptual entities, should be reflected. For this, rather than the entity alignment using the graph topological similarity of the entities, the entity alignment method using data characteristics should be devised. To do so, the proposed method performs two-way entity alignment for the concept and instance entities using different data-centered deep models.

Second, the concept entities have rich information, such as taxonomic relations, literals, and textual descriptions, but with a very small amount of overlapped information needing to be aligned. Therefore, it is tough to align them only with the information of the concept entities contained in the edge-specific KGs. To reduce the difficulty, this paper attempts to expand the range of information available into external information, such as general-purpose KGs and web documents.

Third, for the security and privacy of the instance entities in the edge-specific KGs, the sensing data should not be exposed to the outside during the alignment process. In this light, this paper uses the GAN, which is a neural network model that humans cannot interpret because it is a black-box model. As a result, it is possible to perform alignment considering the security and privacy of the instance entities by hiding the input data and the hyperparameters of the model. In Section 3, we delve into the entity alignment method of the edge-specific KGs in detail.

### 3. Overall Framework

Based on the characteristics that the entity alignment method of the edge-specific KGs should have, this paper proposes a novel alignment method. The proposed method is conducted in three phases. The first phase identifies the entities from each edge-specific KG into the conceptual and instance entities. To do so, we perform graph clustering utilizing the graph properties of the edge-specific KGs. The second phase, which is the core of the proposed method, performs the learning to align the concept and instance entities. At this time, considering the properties of the concept and instance entities, the former uses a language model, and the latter uses a GAN model to perform learning. Finally, by merging the two learned models, an alignment model for the edge-specific KGs is generated. The overall framework of the proposed method is depicted in Figure 1.

![Figure 1. Framework of deep model-based security-aware entity alignment method.](image-url)
This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation, as well as the experimental conclusions that can be drawn.

3.1. Graph Clustering-Based Concept and Instance Entities Identification

The concept and instance entities are mixed in the general-purpose KG as it is a large dataset that contains sufficient information for entity alignment. There is no problem in performing the entity alignment without a distinction between the concept and the instance entities in the general-purpose KG. On the other hand, for the edge-specific KGs constructed for the specific purposes, the volume of the dataset is significantly smaller than that of the general-purpose KGs. It makes it difficult to learn relationships between entities for the predictive model, which significantly reduces the performance of the entity alignment. To improve the performance, we first distinguish the concept and instance entities from the edge-specific KGs.

To identify the concept and instance entities from the edge-specific KGs, we first explored the structural characteristics of the concept and instance entities in the edge-specific KGs. From the point of view of the graph’s structure, the concept entities have many links and have greater outdegree than indegree because they are used to explain other concepts and instances in the edge-specific KGs. In contrast, instance entities have fewer links and have greater indegree than outdegree because they are the physical objects that reify the concept entities [23]. Using these structural characteristics, we will identify the concept and instance entities in the edge-specific KGs. Centrality is a representative metric that uses the graph’s structure, especially the number of connections with other nodes, to find important nodes included in the graph. The more connections a node has, the more important it is. In other words, a node with a large centrality is an important node. Let us project these centrality properties onto the identification of conceptual and instance entities. So, an entity with a large centrality value, that is, an entity with an outdegree greater than the indegree, becomes a concept entity. In contrast, an entity with a small value, that is, an entity with an outdegree smaller than the indegree, becomes an instance entity.

There are three types of centrality metrics used in the structural analysis of graphs: degree, closeness, and betweenness [24,25]. Degree centrality can determine which entities have the most links. It is calculated based on the number of relations between two arbitrary entities in the edge-specific KGs [26]. However, as it only considers one-hop neighbors, it is difficult to capture the structural difference between the concepts and the instances. To reduce the difficulty, we simultaneously use betweenness centrality and closeness centrality, which can reflect the different aspects of multi-hop neighbor structures. The betweenness centrality is based on the paths of two arbitrary entities, which reflects the density of the path, while the closeness centrality is based on their distances, which reflects the length of the paths. So, we use all three centrality indices simultaneously to capture the structural information of the KGs from multiple perspectives.

The centrality indices we use are represented as follows, respectively.

\[
dc(e_x) = \lambda \cdot \log \left(1 + \frac{\deg^+(e_x)}{\deg^-(e_x)}\right) \tag{1}
\]

\[
bc(e_x) = \frac{\sum_{x \neq y \neq z \in G} a_{yz}(e_x)}{\sigma_{yz}} \tag{2}
\]

\[
cc(e_x) = \frac{1}{\sum_{x \neq y \in G} d(e_x, e_y)} \tag{3}
\]

where \(dc(e_x)\), \(bc(e_x)\), and \(cc(e_x)\) are degree centrality, betweenness centrality, and closeness centrality of entity \(e_x\), respectively. \(\deg^+(e_x)\) and \(\deg^-(e_x)\) are the outdegree and indegree of \(e_x\), respectively. \(\lambda\) is the scale coefficient. In addition, \(\sigma_{yz}\) is total number of
shortest paths from $e_x$ to $e_y$, and $\sigma_{yz}(e_z)$ is the number of those paths that pass through $e_z$, and $d(e_x, e_y)$ is the distance two arbitrary entities $e_x$ and $e_y$ ($e_x, e_y, e_z \in KG, x \neq y \neq z$).

Using calculated centrality values, we perform spectral clustering with the number of clusters $k = 2$, showing the best performance for the data where the dimensionality reduction is essential, such as the edge-specific KGs. Based on the centrality values, the Euclidean distance between two arbitrary entities $e_x$ and $e_y$ ($d(e_x, e_y)$) is calculated as follows.

$$(e_x, e_y) = \sqrt{\left(dc(e_x) - dc(e_y)\right)^2 + \left(bc(e_x) - bc(e_y)\right)^2 + \left(cc(e_x) - cc(e_y)\right)^2} \tag{4}$$

By using the distances between the entities, the affinity matrix of the edge-specific KG ($W$) is calculated. The affinity value $w_{ij}$ for $e_i$ and $e_j$ in $W$ is calculated as follows.

$$w_{ij} = \exp \left(-\frac{d(e_i, e_j)}{2\sigma^2}\right) \tag{5}$$

where $d(e_i, e_j)$ is the distance between $e_i$ and $e_j$, $\sigma$ is the gaussian kernel parameter.

For the two clusters, $C_1$ and $C_2$ ($C_1 \cap C_2 = \emptyset$), which are the result of the spectral clustering, the concept and instance entities are determined. At this time, as a centrality-based distance measure identifies the two clusters, one of the two clusters is a cluster of the entities with a low outdegree compared to the indegree (a.k.a. concept entity), and the other is a cluster of the entities with a high outdegree compared to the indegree (a.k.a. instance entity). The entities that belong to a cluster with a large sum of centrality (smaller) are determined as the concept (instance) entities. Finally, we obtain the concept entities ($ce_x$) and instance entities ($ie_x$).

3.2. Semantic Relation Learning Module

3.2.1. Language Model-Based Learning for the Concept Entities

As mentioned earlier, the concept entities have rich information, such as taxonomic relations, literals, and textual descriptions, but with a very small amount of overlapped information needing to be aligned. To address the lack of overlapped information, this paper proposes a method to learn the equivalence relations between the concept entities using not only KGs but also pre-trained language models and external sources. At this time, the pre-trained language models and external sources are used to resolve the lack of overlapped information by learning the generalized lexical relationships between two concept entities, predicting equivalence relations, and generating additional overlapped information. The language model-based learning is composed of three steps, which are depicted in Figure 2.
Figure 2. Training process for lexical relationship prediction of concept entities.

- **Step 1 Data Collection**

To train the pre-trained language model, enough input data must be secured. However, the data we currently have available are the concept and instance entities identified by the edge-specific KGs. As there are very few concept entities in the edge-specific KGs, it is impossible to train a language model with this alone. To amplify the number of concept entities, we extract the partial graphs related to the concept entities from the general-purpose KGs that inherit some of the concept and instance entities to the edge-specific KGs. However, there are still insufficient data to train a language model with only the concept entities in the general-purpose KGs as they contain only information with a graph structure for the concept entities. To compensate for the lack of training data, we randomly sample vocabulary documents similar to the concept entities from external sources such as web documents. The general-purpose KGs, the concept-related partial graphs, and the web documents are defined as follows.

**Definition 1. General-purpose knowledge graph (KG_{gp}).** \( KG_{gp} \) is an overarching KG that contains a wide variety of domains and their information, which consist of entities, literals, and relations. It can be represented as a set of subject–property–object triples, depicted as follows.

\[
KG_{gp} = \{(s_i, p_i, o_i) | s_i \in E_{gp}, p_i \in R_{gp}, o_i \in E_{gp} \cup L_{gp}\}
\]  

where \( s_i \) represents the subject, \( p_i \) represents the property, \( o_i \) represents the object, \( E_{gp} \) represents the set of entities, \( R_{gp} \) represents the set of relations, and \( L_{gp} \) represents the set of literals contained in \( KG_{gp} \).

**Definition 2. Concept-related partial KGs (pKG_x).** \( pKG_x \) is a partial graph of the \( KG_{gp} \), which is directly connected to a concept entity (ce_x).

\[
pKG_x = \{(s_x, p_x, o_x) | (s_x, o_x) \in \{ce_x \forall x\}, (s_x, p_x, o_x) \in KG_{gp}\}
\]
where \( ce_x \) is an identified entity as a concept entity.

**Definition 3. Sample document (\( D_d \)).** \( D_d \) is the \( d^{th} \) sample document including any concept entity as a word of the document.

\[
D_d = \{..., w_{dt}, ..., \{ce_x | \forall x \} \cap \{w_{dt} | \forall t \} > 0 \}
\]  

(8)  

where \( w_{dt} \) is \( t^{th} \) word of the sample document \( D_d \).

- **Step 2 LDA-based topic modeling**

In this step, only sample documents (\( D_d \)) and partial KGs (\( pKG_x \)) that are semantically similar to the arbitrary concept entities are associated with the specific \( ce_x \). To do so, the LDA topic model is utilized to discover abstract topics and their vectors occurring in a collection of documents, \( D = \{D_d | \forall d \} \). The LDA model takes a set of words as input; the partial KGs and sample documents are transformed into a set of words. As it is simple to transform the sample documents to the set of words, only the process of transforming a partial KG into the set of words is described.

To select and transform only the elements related to the concept entity from the \( pKG_x \) into documents, we utilize ‘owl:sameAs’ or ‘skos:exactMatch’, which indicate that the two URIs are equivalent. To select only the concept entities with the relation ‘owl:sameAs’ or ‘skos:exactMatch’, we randomly select the pairs of concept entities (PCE), which have the relation ‘owl:sameAs’ or ‘skos:exactMatch’ from the \( pKG_x \). The PCE is simply represented as follows.

\[
PCE = \{((ce_x, ce_y) | rel(ce_x, ce_y) = ‘owl:sameAs’ or ‘skos:exactMatch’), \]

\[
\quad ce_x \neq ce_y, (ce_x, ce_y) \equiv (ce_x, ce_y)
\]

(9)

For all pairs of the PCE, we compare the number of hops between the PCE pairs and the \( pKG_x \) elements. Finally, we generate \( pKG'_x \) using only \( pKG_x \) elements connected with \( n \) hops or less. This element of \( pKG'_x \) is used to create a set of words, that is, documents to be input into the LDA topic model. At this time, the word set of \( pKG'_x \) (\( pD_x \)) is constructed by extracting only the entity names included in \( pKG'_x \) as components. It is simply represented as follows.

\[
pD_x = \{..., n(E_{gp}')_{xt}, ..., |E_{gp} \in pKG'_x, \forall x\}
\]

(10)

where \( n(E_{gp}')_{xt} \) represents the \( t^{th} \) name of the entity \( E_{gp} \), which is included in \( pKG'_x \).

Third, using all the sample documents, \( D_d \), and the transformed documents, \( pD_x \) (\( \forall x \)), the LDA-based topic modelling is performed. At this time, the documents are short texts, such as paragraphs of the research publications or Wikipedia, which are randomly collected from well-structured and refined sources and have enough semantic information.

We derive the topic distribution vectors for each document from \( D_d \) and \( pD_x \) using the LDA-based topic model. The results of the topic modeling are as follows.

\[
d\theta_m = \{d\theta_{m,1}, d\theta_{m,2}, ..., d\theta_{m,k}, ..., d\theta_{m,K}\} \text{ for } m = 1, 2, ..., M
\]

\[
e\theta_{p,n} = \{e\theta_{p,n,1}, e\theta_{p,n,2}, ..., e\theta_{p,n,k}, ..., e\theta_{p,n,K}\} \text{ for } n = 1, 2, ..., N
\]

(11)

and \( p = 1, 2, ..., P \)

where \( K \) is the number of topics, \( d\theta_m \) represents the topic distribution vector of the document \( d_m \) in \( p = \{pD_x \} \forall x \) and \( D_d \), \( d\theta_{m,k} \) is the probability of the topic \( k \) of \( d_m \) \((\sum_{k=1}^{K}d\theta_{m,k} = 1 \text{ for all } m = 1, 2, ..., M)\), and \( e\theta_{p,n,k} \) is the probability of the topic \( k \) of \( n(E_{gp}')_{xt} \) \((\sum_{k=1}^{K}e\theta_{p,n,k} = 1 \text{ for all } n = 1, 2, ..., N \text{ and } p = 1, 2, ..., P)\).
However, although both the partial KGs and the sample documents are generated based on the concept entities, it is impossible to directly compare their relations because their sources are different. To solve the problem, using the topic vectors generated from the LDA topic model, the concept entities with the most similar topics to the sample documents are found, and the corresponding documents are attached. The process for finding similar topics is as follows.

To discover the documents related to the concept entity $ce_x$ (for $\forall x$), we calculate the topic similarity between the documents $D_d$ and $pD_x$ using Hellinger distance, which can derive the similarity between probability distributions. The topic similarity between $e_{\theta_{p,n}}$ and $d_{\theta_m}$ is calculated as follows.

$$sim(e_{\theta_{p,n}}, d_{\theta_m}) = \frac{1}{\sqrt{2}} \sqrt{\sum_{k=1}^{K} (e_{\theta_{p,n,k}} - d_{\theta_{m,k}})^2}$$

Based on this similarity measure, the related document of $D_d$ can be discovered with maximum of $sim(e_{\theta_{p,n}}, d_{\theta_m})$ in the document distributions.

- **Step 3 Training the Prediction Model for Lexical relationships**

In the third step, the lexical relations between two arbitrary concept entities are learned using documents attached as a training dataset. Furthermore, a language model capable of predicting equivalence relations is trained. To create a dataset that can learn the equivalence relation, we utilize the sub-structure of the general-purpose KGs. In the sub-structure, we give the label ‘1’ to pairs of documents attached to concept entities with an equivalence relation, and otherwise, we give the label ‘0’. Finally, the language model is learned using the label-annotated datasets. At this time, the transfer learning is performed using a pre-trained language model such as BERT to improve the learning speed and generalization performance.

To train the prediction model of lexical relation between the entities, we utilize the related documents of the previous model. First, we select two arbitrary entities ($e_i$ and $e_j$) from $PCE$ with the random selection of one related document ($e_{d,i,a}$ and $e_{d,j,b}$) for each entity. At this time, the label is 1 if the two entities have an equivalence relationship ($e_{i,j} \in PCE$) and 0 otherwise. Second, the entity name and the related document are concatenated and encoded as vectors using the BERT model [27]. Third, the prediction model is trained by using the encoded vectors of the two entities with their related documents. As the prediction model, we adopt the attention network model that is widely used for classification tasks. The prediction model uses the ReLU function as an activation function, which has the advantages of sparse activation as well as a low computational burden.

3.2.2. GAN-Based Semantic Alignment Learning for Instances

In the conventional entity alignment method, the primary information sources about the instance entities are literal or data properties (called attributes) [28]. Unlike the general-purpose KGs, the edge-specific KGs are likely to contain much larger-scale numerical data, such as sensor data in the instance entities, because the ESs operate on the low-end, directly exchange, and store sensor data. However, even if there is a lot of data about the instance entities, it is tough to access because the data of the ESs requires a very high level of security, such as for the information about a specific space or personal privacy. In addition, due to the characteristics of the ESs, which separately store data in multiple repositories, it is also challenging to gather the data required for the entity alignment of the instance entities.

To overcome the difficulties, this paper utilizes the generative adversarial networks (GAN)-based unsupervised instance alignment method for the instance entities. The GAN, as a deep model, has a significant advantage in security because it shields data from being directly interpretable. In addition, it shows an excellent performance in predicting and generating insufficient data as a generative model. Based on the properties of the
GAN, this paper obtains the GAN model that can predict the entire data pattern after learning the GAN with subsets of data samples. Finally, this paper performs embedding for the instance entity alignment using the predicted data that the GAN model creates. The learning process of the GAN is as follows.

To train the GAN model, we use instance-related data subsets collected from sensors or devices, such as literal, attribute, or signal data. The data subset is defined as follows.

**Definition 4. Data Subset \((X_x)\).** \(X_x\) is a subset of data samples related to the \(x^{th}\) instance of entity \(ie_x\)

\[
X_x = \{\cdots, x_{xt}, \cdots\}, 1 \leq t \leq n
\]  
(13)

where \(x_{xt}\) is \(t^{th}\) data value of \(X_x\).

At this time, to perform the alignment of two arbitrary \(ie_x\) and \(ie_{x'}\), it is necessary to know whether the populations of \(ie_x\) and \(ie_{x'}\) are equal. However, the volumes of \(X_x\) and \(X_{x'}\), which are the data subsets of \(ie_x\) and \(ie_{x'}\), are tiny because the data in the edge systems are stored distributionally. In addition, for data privacy and security reasons, blocking the external disclosure of the edge systems' data is a factor that shrinks the volumes of \(X_x\) and \(X_{x'}\). Therefore, it is challenging to infer the similarity of the entire populations of the two entities with only the similarity of the distributions of \(X_x\) and \(X_{x'}\), which are small in scale. This means that accurate alignment between \(ie_x\) and \(ie_{x'}\) is difficult. To overcome the difficulty, we use the GAN model, which can learn the distribution of the whole populations. At this time, the GAN model uses the data subset \(X_{x}(for \forall x)\) as its input. Finally, using the reconstructed data subsets \(\hat{X}_x\) and \(\hat{X}_{x'}\) derived from the inferred populations, the entity alignment between \(ie_x\) and \(ie_{x'}\) is conducted. The GAN consists of two parts: the generator \(G\) and the discriminator \(D\) [29]. The loss function of the discriminator is defined as follows.

\[
\min_G \max_D V(D,G) = \mathbb{E}_{x \sim P_{data}(X_x)}[\log D(x)] + \mathbb{E}_{z \sim P_z}[\log (1 - D(G(z)))]
\]  
(14)

where \(P_{data}(X_x)\) is the real data distribution from \(X_x\), \(P(z)\) is a prior distribution on noise vector \(z\), \(D(x)\) denotes the probability that \(x\) comes from the real data, \(\mathbb{E}_{x \sim P_{data}(X_x)}\) is the expectation of \(x\) from \(P_{data}(X_x)\), and \(\mathbb{E}_{z \sim P_z}\) is the expectation of the \(z\) sample from the noise.

The trained GAN for each instance is used to attain a reconstructed data sample \(\hat{X}_x\). Using the \(\hat{X}_x\), the equivalence relationships between the instances are identified. The GAN-based instance alignment process is depicted in Figure 3. In Figure 3, an embedding layer is added to obtain fixed dimensional vectors for \(\hat{X}_x\). The simple description of the process of instance alignment using GAN is as follows. All data subsets of the target instance \((ie_x, \forall x)\) to perform entity alignment are input to the GAN model. The trained GAN is used to reconstruct the data for all instances. Lastly, it estimates the distribution of instances using reconstructed data and real data and performs entity alignment by inputting the estimated distribution \((\hat{X}_x)\) to the embedding layer.
3.3. Graph-Based Merged Deep Model Learning Module

To perform the edge-specific entity alignment, this module creates the robust entity alignment model for the various structures and the lack of information of the edge-specific KGs by sharing and merging the learned deep models. At this time, it is impossible to directly merge the learning results of the concept and instance entities because the learning was performed using two deep models named the language model and the GAN, with completely different properties. To merge the two heterogeneous deep models, an embedding vector is generated using the learning information of the deep models and then projected onto the graph structure. As a next step, this paper uses a GCN (graph convolution network)-based merged deep model that can perform the entity alignment of the edge-specific KGs by learning only the structural relations of the graph.

As the GNN shows high performance in exploiting graph structure information for many entity alignment tasks, we adopt the GCN model to capture the structural information of the concepts and the instances in the KGs. A GCN model consists of multiple GCN layers. In a GCN model, the input of the $l$th layer is the entity feature matrix, $H(l) \in \mathbb{R}^{n \times d(l)}$, where $n$ is the number of nodes, and $d(l)$ is the number of features in the $l$th layer. The output of the $l$th layer is a new feature matrix $H^{(l+1)}$. The convolutional computation of $H^{(l+1)}$ is as follows.

$$H^{(l+1)} = \sigma(AH^{(l)}W^{(l)})$$  \hspace{1cm} (15)

where $A$ is a normalized adjacency matrix with a self-connected input graph, $H^{(l)}$ is the hidden states, $W^{(l)}$ is the weight of the $l$th layer, and $\sigma(\cdot)$ is a nonlinear activation function.

Two GCN models for the concepts and instance entities are trained independently because the spaces that explain the concepts and instances are different. Moreover, it is easy to find the optimal hyperplane, and the complexity can be lowered by independently learning. The inputs of the GCN models for the concept and instance entities are the two types of embeddings for the concept and instance entities by the language model-based trained model and the GAN-based trained model, respectively. After the training of the two GCN models, we can obtain the embeddings of the structures of the concept and the instance entities as output by training the GCN models. Finally, the two types of embeddings are concatenated to discover their shared spaces, which cannot be identified by the concept or the instance entities alone. The concatenated embeddings are used for the
merged model to get merged embeddings for the alignment task. The details of the merged model are depicted in Figure 4.

Figure 4. Procedure of graph-based merged deep model learning.

The space found by deep learning is almost impossible to be explained without the information of the training dataset and the dimensions and hyperparameters of the models. Therefore, our proposed framework can contribute to the security and privacy issue by only predicting the alignment using the merged model, without directly exposing the structures of the KGs.
4. Experiments and Performance Evaluation

In this section, we first explain the experimental setup, including a summary of the experimental datasets and the comparative methods. We then report the experimental results and analysis compared to the other entity alignment methods. We used a processor with Intel(R) Xeon(R) CPU @ 2.30GHz, a Teslar P100, and 12 GB memory provided by Google Colab Pro.

4.1. Experimental Datasets

In order to evaluate our proposed framework, we used two datasets, DBP-YG 15K and DBP-WD 15K, sampled from DBpedia-YAGO and DBpedia-Wikidata, respectively [30]. The statistical data of the two datasets are listed in Table 2. The DBP-YG and DBP-WD include attribute triples representing the attribute information of the entities and relation triples representing the relations between the entities, respectively. At this time, the predicates and objects of the attribute triples contain many words describing the actual entities. On the other hand, the relation triples can only non-symbolically express only entities and their equivalence relationships, but no vocabularies can explain their meaning.

Table 2. Statistical data of DBP-YG 15k and DBP-WD 15k.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>#Entities</th>
<th>#Attrs</th>
<th>#Attr. Triples</th>
<th>#Rel</th>
<th>#Rel. Triples</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBP-YG 15k</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBpedia</td>
<td>15,000</td>
<td>39,520</td>
<td>52,093</td>
<td>17,368</td>
<td>30,291</td>
</tr>
<tr>
<td>YAGO</td>
<td>15,000</td>
<td>117,622</td>
<td>117,114</td>
<td>15,859</td>
<td>26,638</td>
</tr>
<tr>
<td>DBP-WD 15k</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBpedia</td>
<td>15,000</td>
<td>42,294</td>
<td>52,134</td>
<td>19,132</td>
<td>38,265</td>
</tr>
<tr>
<td>Wikidata</td>
<td>15,000</td>
<td>133,090</td>
<td>138,246</td>
<td>19,324</td>
<td>42,746</td>
</tr>
</tbody>
</table>

Comparison Methods

We conducted an experiment to evaluate the superiority of our proposed framework compared to the existing entity alignment methods. The details of the comparison methods are presented as follows.

1. MTransE: MtransE is an EA method which learns mapping between two separate embedding spaces of different KGs [31].
2. TransD: TransD is an embedding method which extends TransE to model complex relations by projecting the entities into a relation-related space [32].
3. RotatE: RotatE is an embedding method which represents entities as complex vectors and relations as rotations in a complex vector space [33].
4. ConvE: ConvE is an embedding method which is the representative multi-layer CNN-based architecture for link prediction [34].
5. AlignE: AlignE is a self-training entity alignment method which embeds two KGs in a unified space and iteratively labels newly identified entity alignment as supervision [30].
6. AttrE: AttrE generates attribute character embeddings to shift the entity embeddings from different two knowledge graphs into the same space [14].
7. GCN-a: GCN-Align is an entity alignment method which employs GCN to model entities to exploit their neighborhood information [30].

We conducted an experiment by varying the sampling ratio from 10 to 55% in increments of 5%. By convention, we chose Hit@k (k = 1, 5, 10), mean rank (MR), and mean reciprocal rank (MRR) as the evaluation metrics.
4.2. Experimental Results and Evaluation

The results of the experiment on the full datasets are illustrated in Table 3. As shown in Table 3, our proposed framework outperforms other comparison models. As our proposed method exploits semantic information with a language model, it captured the differences between entities better than the other comparison models. Moreover, the alignment models using attributes including our proposed model show better performances than the models using only relations. This is because the entities in KG have multiple aspects of features, which help in identifying equivalence relations.

Table 3. Experimental results on DBP-YG 15k and DBP-WD 15k.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Hits@1</th>
<th>Hits@5</th>
<th>Hits@10</th>
<th>MRR</th>
<th>Hits@1</th>
<th>Hits@5</th>
<th>Hits@10</th>
<th>MRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTransE</td>
<td>47.343</td>
<td>68.771</td>
<td>74.305</td>
<td>249.55</td>
<td>0.5687</td>
<td>25.048</td>
<td>45.257</td>
<td>53.343</td>
</tr>
<tr>
<td>TransD</td>
<td>31.124</td>
<td>45.495</td>
<td>49.219</td>
<td>1320.94</td>
<td>0.3773</td>
<td>20.99</td>
<td>34.562</td>
<td>40.114</td>
</tr>
<tr>
<td>RotatE</td>
<td>45.743</td>
<td>66.495</td>
<td>71.952</td>
<td>553.97</td>
<td>0.5491</td>
<td>26.533</td>
<td>46.981</td>
<td>55.81</td>
</tr>
<tr>
<td>ConvE</td>
<td>5.886</td>
<td>10.152</td>
<td>11.429</td>
<td>4213.87</td>
<td>0.0788</td>
<td>14</td>
<td>25.648</td>
<td>30.086</td>
</tr>
<tr>
<td>AlignE</td>
<td>26.933</td>
<td>41.362</td>
<td>45.762</td>
<td>452.331</td>
<td>0.3364</td>
<td>16.219</td>
<td>28.095</td>
<td>34.457</td>
</tr>
<tr>
<td>AttriE</td>
<td>28.79</td>
<td>67.41</td>
<td>71.886</td>
<td>901.239</td>
<td>0.5693</td>
<td>11.41</td>
<td>13.21</td>
<td>30.752</td>
</tr>
<tr>
<td>GCN-a</td>
<td>46.638</td>
<td>62.895</td>
<td>66.314</td>
<td>1110.54</td>
<td>0.5383</td>
<td>26.59</td>
<td>48.248</td>
<td>62.552</td>
</tr>
<tr>
<td>proposed</td>
<td>51.324</td>
<td>68.571</td>
<td>71.468</td>
<td>950.118</td>
<td>0.5883</td>
<td>33.257</td>
<td>57.19</td>
<td>64.752</td>
</tr>
</tbody>
</table>

Figures 5 and 6 show the experimental results by changing the sampling rate for the two experimental datasets: DBP-YG 15k and DBP-WD 15k. There are a lot of data on the people-centered relationship in the DBP-YG 15k. In the case of people relationships, the complexity of the label is quite high because the types are quite diverse. Thus, the complexity of the label greatly increases according to the data sampling rate. While the value of Hits@1 increases with the stability, the value of Hits@10 increases slowly as the sampling rate increases. The value of Hits@10 increases steeply up to the sampling rate of 30%, and then the increase slows down after that the rate. This is because the entropy of the label increases as the complexity of the label increases. The increase rate of the Hits@K value of DBP-WD 15k was much slower. The proposed method showed excellent performance at almost all sampling rates and the closest performance with the GCN-a and AlignE. In the case of Hits@5 and Hits@10, the performance of AlignE is lower than the proposed method. It means that the generalization performance of AlignE is low.

Figure 5. Comparison of the Hits@k with changing sampling rate (10–55%).
On the other hand, for the MRR, the value of MRR increases rapidly as the sampling rate increases. Unlike Hits@k, the MR and MRR are indicators that can evaluate the generalization performance and stability because they calculate average scores for the overall results. At this time, the lower the MR, the better the value, and the higher the MRR, the better. As depicted in Figure 5, the proposed method performs better than the other methods at most sampling rates. In particular, the index of Hits@10 in the DBP-WD 15k dataset indirectly suggests that the generalization performance of the AlingE method is somewhat low. Furthermore, the MRR value also suggests that the performance of AlingE is low. As a result of comparing Hits@k, it can be confirmed that the proposed method showed high accuracy compared to the other methods. Moreover, it showed a high generalization performance from the perspective of the MR and MRR. In summary, it can be said that the entity alignment performance of the proposed method is excellent. The experimental results are summarized in Tables 4 and 5 in detail.

**Figure 6.** Performance Comparison of the MR and MRR with changing sampling rate (10~55%).

**Table 4.** Experimental results of DBP-YG 15k.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Hits@1</th>
<th>Hits@5</th>
<th>Hits@10</th>
<th>MR</th>
<th>MRR</th>
<th>Hits@1</th>
<th>Hits@5</th>
<th>Hits@10</th>
<th>MR</th>
<th>MRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTransE</td>
<td>5.219</td>
<td>11.743</td>
<td>15.133</td>
<td>899.475</td>
<td>0.087115</td>
<td>6.238</td>
<td>12.752</td>
<td>15.943</td>
<td>971.901</td>
<td>0.097091</td>
</tr>
<tr>
<td>TransD</td>
<td>1.229</td>
<td>2.362</td>
<td>2.81</td>
<td>4621.611</td>
<td>0.018651</td>
<td>6.238</td>
<td>12.752</td>
<td>15.943</td>
<td>971.901</td>
<td>0.097091</td>
</tr>
<tr>
<td>RotatE</td>
<td>1.743</td>
<td>4.771</td>
<td>6.238</td>
<td>3606.116</td>
<td>0.033357</td>
<td>2.733</td>
<td>7.105</td>
<td>9.41</td>
<td>3078.689</td>
<td>0.050155</td>
</tr>
<tr>
<td>ConvE</td>
<td>0.41</td>
<td>0.838</td>
<td>1.124</td>
<td>4840.592</td>
<td>0.007365</td>
<td>0.438</td>
<td>0.981</td>
<td>1.305</td>
<td>4883.341</td>
<td>0.007749</td>
</tr>
<tr>
<td>AlignE</td>
<td>0.276</td>
<td>0.762</td>
<td>1.076</td>
<td>4062.907</td>
<td>0.006292</td>
<td>1.2</td>
<td>2.476</td>
<td>3.295</td>
<td>2647.197</td>
<td>0.020338</td>
</tr>
<tr>
<td>AttrE</td>
<td>3.571</td>
<td>10.048</td>
<td>12.343</td>
<td>3454.863</td>
<td>0.064590</td>
<td>4.943</td>
<td>12.552</td>
<td>15.305</td>
<td>2831.649</td>
<td>0.090642</td>
</tr>
<tr>
<td>GCN-a</td>
<td>3.819</td>
<td>8.552</td>
<td>11.314</td>
<td>2494.948</td>
<td>0.062887</td>
<td>7.838</td>
<td>14.61</td>
<td>17.571</td>
<td>2262.543</td>
<td>0.112049</td>
</tr>
<tr>
<td>proposed</td>
<td>3.489</td>
<td>8.162</td>
<td>10.581</td>
<td>2477.388</td>
<td>0.059166</td>
<td>5.981</td>
<td>11.829</td>
<td>14.419</td>
<td>2400.914</td>
<td>0.089217</td>
</tr>
<tr>
<td>MTransE</td>
<td>9.657</td>
<td>19.657</td>
<td>24.457</td>
<td>742.268</td>
<td>0.148495</td>
<td>12.667</td>
<td>25.21</td>
<td>31.019</td>
<td>638.397</td>
<td>0.189461</td>
</tr>
<tr>
<td>TransD</td>
<td>2.714</td>
<td>4.79</td>
<td>5.571</td>
<td>4292.445</td>
<td>0.037679</td>
<td>4.581</td>
<td>8.857</td>
<td>10</td>
<td>3783.594</td>
<td>0.065699</td>
</tr>
<tr>
<td>RotatE</td>
<td>3.61</td>
<td>9.333</td>
<td>12.2</td>
<td>2797.711</td>
<td>0.065277</td>
<td>5.867</td>
<td>14.019</td>
<td>18.086</td>
<td>2207.615</td>
<td>0.099234</td>
</tr>
<tr>
<td>ConvE</td>
<td>0.657</td>
<td>1.381</td>
<td>1.762</td>
<td>4825.704</td>
<td>0.011041</td>
<td>1.124</td>
<td>2.39</td>
<td>2.829</td>
<td>4626.725</td>
<td>0.017701</td>
</tr>
<tr>
<td>AlignE</td>
<td>1.714</td>
<td>3.733</td>
<td>4.838</td>
<td>2587.264</td>
<td>0.029026</td>
<td>1.752</td>
<td>3.838</td>
<td>5.248</td>
<td>2643.471</td>
<td>0.030305</td>
</tr>
<tr>
<td>AttrE</td>
<td>6.829</td>
<td>15.962</td>
<td>18.99</td>
<td>2828.178</td>
<td>0.109796</td>
<td>10.676</td>
<td>23.476</td>
<td>28.086</td>
<td>2168.71</td>
<td>0.164928</td>
</tr>
<tr>
<td>GCN-a</td>
<td>11.19</td>
<td>20.267</td>
<td>23.819</td>
<td>2014.289</td>
<td>0.155226</td>
<td>15.867</td>
<td>27.029</td>
<td>30.4</td>
<td>1873.262</td>
<td>0.209795</td>
</tr>
<tr>
<td>proposed</td>
<td>10.495</td>
<td>19.19</td>
<td>22.895</td>
<td>2057.617</td>
<td>0.146866</td>
<td>15.267</td>
<td>25.81</td>
<td>29.495</td>
<td>1897.483</td>
<td>0.202825</td>
</tr>
<tr>
<td>MTransE</td>
<td>14.571</td>
<td>27.21</td>
<td>32.752</td>
<td>659.049</td>
<td>0.207981</td>
<td>18.114</td>
<td>34.143</td>
<td>40.724</td>
<td>535.993</td>
<td>0.258659</td>
</tr>
<tr>
<td>TransD</td>
<td>5.629</td>
<td>9.857</td>
<td>11.486</td>
<td>3608.386</td>
<td>0.076879</td>
<td>7.057</td>
<td>13</td>
<td>14.695</td>
<td>3375.369</td>
<td>0.099554</td>
</tr>
</tbody>
</table>
Table 5. Experimental results of DBP-WD 15k

<table>
<thead>
<tr>
<th>Methods</th>
<th>Hits@1</th>
<th>Hits@5</th>
<th>Hits@10</th>
<th>MR</th>
<th>Hits@1</th>
<th>Hits@5</th>
<th>Hits@10</th>
<th>MR</th>
<th>MRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlignE</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
</tr>
<tr>
<td>GCN-a</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
</tr>
<tr>
<td>proposed</td>
<td>25.962</td>
<td>37.829</td>
<td>25.962</td>
<td>37.829</td>
<td>25.962</td>
<td>37.829</td>
<td>25.962</td>
<td>37.829</td>
<td>25.962</td>
</tr>
</tbody>
</table>

Table 6. Experimental results of DBP-WD 15k

<table>
<thead>
<tr>
<th>Methods</th>
<th>Hits@1</th>
<th>Hits@5</th>
<th>Hits@10</th>
<th>MR</th>
<th>Hits@1</th>
<th>Hits@5</th>
<th>Hits@10</th>
<th>MR</th>
<th>MRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlignE</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
</tr>
<tr>
<td>GCN-a</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
</tr>
<tr>
<td>proposed</td>
<td>25.962</td>
<td>37.829</td>
<td>25.962</td>
<td>37.829</td>
<td>25.962</td>
<td>37.829</td>
<td>25.962</td>
<td>37.829</td>
<td>25.962</td>
</tr>
</tbody>
</table>

Table 7. Experimental results of DBP-WD 15k

<table>
<thead>
<tr>
<th>Methods</th>
<th>Hits@1</th>
<th>Hits@5</th>
<th>Hits@10</th>
<th>MR</th>
<th>Hits@1</th>
<th>Hits@5</th>
<th>Hits@10</th>
<th>MR</th>
<th>MRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlignE</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
<td>0.781</td>
</tr>
<tr>
<td>GCN-a</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
<td>2.041</td>
</tr>
<tr>
<td>proposed</td>
<td>25.962</td>
<td>37.829</td>
<td>25.962</td>
<td>37.829</td>
<td>25.962</td>
<td>37.829</td>
<td>25.962</td>
<td>37.829</td>
<td>25.962</td>
</tr>
</tbody>
</table>
5. Related Works

5.1. KGs for Edge Computing

Many recent studies have been performed to develop and utilize KGs to support ESs. The functions of these KGs can be categorized into three types: (1) service data/information management, (2) device/sensor maintenance, and (3) security/privacy of the ES. Based on these categories, the related works on KGs for edge computing are summarized in Table 6.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Functions</th>
<th>50%</th>
<th>55%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provide data integration and reasoning services to support data management in ES</td>
<td>O</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>Provide a group recommendation service of network document resources in ES</td>
<td>O</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Perform a news recommendation service in edge computing environment</td>
<td>O</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Provide interactive services of vehicles, such as traffic flow prediction and route arrangement</td>
<td>O</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>Data representation of the edge computing devices in manufacturing systems</td>
<td>O</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>Perform edge analytics to manage limited resources in ES</td>
<td>X</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>Integrate knowledge for industrial automation systems in edge computing environment</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Make the consumption of KGs more reliable and faster in ESs</td>
<td>O</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>Detect anomalous activity in industrial systems</td>
<td>X</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>Provide security-aware data model and semantics in the dynamic collaboration environment</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

O: satisfied, X: dissatisfied.

Xu [22] proposed a KG inference-enabled data management system for edge computing, named SuccinctEdge, which is a compact, decompression-free, self-index, in-memory RDF store that can answer SPARQL queries, including those requiring reasoning services associated with some ontology. Wu [35] devised a group recommendation system for network document resource exploration using the KG and LSTM in edge computing. Yao [36] developed a news recommendation algorithm in an edge computing environment using KG and GNN. Shi [37] proposed a KG-empowered reasoning model, named
TKGERM, to reason traffic information with multi-source data for edge computing-enabled IoV (Internet of Vehicles). Liu [12] devised a KG-based data representation method for IIoT (Industrial Internet of Things)-enabled cognitive manufacturing. Zhang [38] proposed an edge analytics method using KG construction and application to manage infrastructure resources such as CPU usage, memory capacity, network bandwidth, and the operating system of relevant devices. Doldy [39] proposed an energy-based model of neuro-symbolic reasoning on KGs to characterize industrial automation systems, integrating knowledge from different domains such as industrial automation, communications and cybersecurity. Marx [40] presented Knowledge Box (KBox), an approach for transparently shifting query execution on KGs to the edge, to make the consumption of KGs more reliable and faster in ESs. Garrido [41] applied machine learning on ES-specific KGs to detect intrusion and anomalous activity in industrial systems. Even though adopting KGs in the edge computing environment is gaining popularity, these KGs are developed for their specific purposes so that it is hard to directly utilize these KGs to support cooperation and collaboration between ESs. To effectively resolve the semantic heterogeneity of the data of each ES, the entities in different KGs should be aligned.

5.2. Entity Alignment for KG

5.2.1. Entity Alignment Methods

Entity alignment, which links the entities from different KGs that indicate the same real-world object, is gaining popularity in fusing knowledge from heterogeneous KGs. Entity alignment methods can be categorized into (1) translation-based models and (2) GNN-based models [19]. Translation-based models [14,30,31,42–44] perform entity alignment by utilizing translation-based embedding models, mostly TransE [45], which learn vector representations of the entities and relations. MTransE [31] generates the embedding of the entities and relations using TransE and provides transitions for each embedding vector to its counterparts in other spaces. BootEA [30] adopted the bootstrapping approach, which iteratively labels likely entity alignment as training data to learn embeddings and employs an alignment editing method to reduce error. AttrE [14] generates attribute character embeddings that shift the entity embeddings of two different KGs into the same space by calculating the similarity of the entities based on their attributes. ITransE/IPTransE [42] iteratively align entities via joint knowledge embeddings by encoding the entities and relations of different KGs into a single semantic space. Currently, ITransE utilizes TransE to learn embeddings while IPTransE utilizes PTransE [46]. JAPE [43] jointly embeds the structures of two KGs into a unified vector space and refines them by leveraging attribute correlations. KDCoE [44] iteratively trains two component embedding models on multilingual KG structures and entity descriptions, respectively.

GNN-based models utilize GNN to generate embeddings and predict alignment and can be divided into GCN-based models [47–52] and GAT-based models [53–55]. GCN-Align [52] trains GCNs to perform the embedding of the entities of different KGs into the same vector space. MuGNN [51] utilizes different channels for each KG to be robust to structural differences. GMNN [50] introduces the topic entity graph, which is a local subgraph of the entity, to represent entities with their contextual information in KG. RDGCN [49] incorporates relation information by attentive interaction between the knowledge graph and its counterpart. HGCN [48] jointly learns entity and relation representations and does not require pre-aligned relations. EMGCN [47] utilizes multi-order graph convolutional networks to perform end-to-end, unsupervised entity alignment. KECG [55] utilizes GAT to embed entities into a single vector space by utilizing inner-graph structure and intra-graph alignment information. MRGGA [54] directly models cross-lingual entity embeddings by attending to the node’s incoming and outgoing neighbors and its connected relations. RAGA [53] adopts a self-attention mechanism to spread entity information to the relations and then aggregates relation information back to the entities.
Even though these alignment methods perform well in general-purpose KGs, such as DBpedia and YAGO, they cannot be directly applied in ES-specific KGs as their heterogeneity is much severe than that of general-purpose KGs. To align entities of ES-specific KGs, a novel method is needed which considers the unique characteristics of the ES-specific KGs.

5.2.2. Entity Alignment Applications

Entity alignment is utilized for many purposes related to enabling the integration of knowledge of heterogeneous sources. Zhang [56] utilized the entity alignment method to integrate knowledge related to maritime dangerous goods. Chong [27] tried to support the interoperability of depressive disorder-related knowledge. Hu [57] performed entity alignment to obtain semantic information from a biomedical knowledge base. Zhou [58] proposed an alignment method of point of interest (POI)-related entities to discover identical POIs in location-based services. Chen [59] tries to integrate multi-source heterogeneous electricity power data by aligning entities in multiple power knowledge graphs. Yang [60] proposed an entity alignment method of power grid-dispatching knowledge graphs. Zhu [61] tried to identify similar IoT devices in different networks by using an alignment method. These studies perform the alignment of entities related to their own purposes. However, in many cases, knowledge integration among different domains is needed, such as healthcare services and transportation. To meet this requirement, a novel alignment method is needed to perform alignment between KGs developed for different domains.

6. Conclusions and Further Research

This paper proposes the deep model-based dynamic entity alignment framework for edge-specific KGs in the edge computing environment. The contributions of this paper can be summarized as follows. First, we applied various deep learning models to align the domain-specific KGs. Second, as the domain-specific KGs of the edge computing environment do not have sufficient shared information, we proposed a method to perform alignment of the concepts from the different KGs by utilizing external information from general-purpose KG and documents. Finally, to ensure security and privacy of the domain-specific KGs of the edge computing environment, we proposed the GAN-based unsupervised alignment methods of the instances. The neural network models such as the GAN are black-box models that humans cannot interpret. So, they can ensure security by hiding the input data and hyperparameters of the model.

As a result, this paper proposes a novel entity alignment method suitable for the edge computing to overcome the limitations due to existing KG and entity alignment techniques not being suitable for real-world applications to edge computing environments. We analyzed the characteristics of edge-specific KGs that are different to general-purpose KGs, considering the edge computing and collaboration environment, and devised the entity alignment technique with the language model and GAN. In addition, various experiments were performed to evaluate the superiority of the proposed method, and it was proved that it exhibits a better performance than the conventional methods.

However, our framework has several limitations. As vector representation does not consider the structure of the KG and containing terms, the position of the entities projected by the language model may be biased. Moreover, in order to quickly respond to new types of cyberattacks, not only resources composed of formal sentences, but also noisy and fragmented text resources should also be used. To overcome these limitations, we will additionally use graph embedding methods which consider graph structures and utilize the language model which is more robust to short and noisy texts.
Author Contributions: Conceptualization, J.K. and M.S.; methodology, J.K., K.K. and M.S.; software, J.K.; validation, M.S. and G.P.; data curation, J.K.; writing—original draft preparation, J.K., K.K. and M.S.; writing—review and editing, M.S. and G.P.; visualization, J.K.; supervision, M.S.; project administration, M.S.; funding acquisition, G.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Agency for Defense Development in Korea grant number UE201114ED.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: This research is supported by the C2 integrating and interfacing technologies laboratory of the Agency for Defense Development (UE201114ED).

Conflicts of Interest: The authors declare no conflict of interest.

References


