applied sciences

MDPI

Article

Leveraging Multi-Modal Information for Cross-Lingual Entity Matching across Knowledge Graphs

Tianxing Wu [1,*], Chaoyu Gao [1], Lin Li [1] and Yuxiang Wang [2]

[1] School of Computer Science and Engineering, Southeast University, Nanjing 211189, China
[2] School of Computer and Software, Hangzhou Dianzi University, Hangzhou 310018, China
* Correspondence: tianxingwu@seu.edu.cn

Abstract: In recent years, the scale of knowledge graphs and the number of entities have grown rapidly. Entity matching across different knowledge graphs has become an urgent problem to be solved for knowledge fusion. With the importance of entity matching being increasingly evident, the use of representation learning technologies to find matched entities has attracted extensive attention due to the computability of vector representations. However, existing studies on representation learning technologies cannot make full use of knowledge graph relevant multi-modal information. In this paper, we propose a new cross-lingual entity matching method (called CLEM) with knowledge graph representation learning on rich multi-modal information. The core is the multi-view intact space learning method to integrate embeddings of multi-modal information for matching entities. Experimental results on cross-lingual datasets show the superiority and competitiveness of our proposed method.

Keywords: knowledge graph; cross-lingual entity matching; knowledge graph embedding; representation learning

Citation: Wu, T.; Gao, C.; Li, L.; Wang, Y. Leveraging Multi-Modal Information for Cross-Lingual Entity Matching across Knowledge Graphs. *Appl. Sci.* **2022**, *12*, 10107. https://doi.org/10.3390/app121910107

Academic Editor: Valentino Santucci

Received: 5 September 2022
Accepted: 7 October 2022
Published: 8 October 2022

1. Introduction

Knowledge graphs [1] are attracting attention from both academics and industries due to their power to model structured information and professional knowledge. In recent years, many high-quality knowledge graphs have been built, such as Microsoft Concept Graph [2], NELL [3], Zhishi.me [4], etc. Knowledge graphs have been widely used in many vertical fields [5], e.g., finance, medical and e-commerce. Knowledge graphs and their related technologies have gradually become indispensable basic technologies in the era of artificial intelligence.

However, a knowledge graph can be freely constructed by any organization or individual according to their own needs and languages. Therefore, the data in a knowledge graph can be multilingual and diverse, and there is a large amount of overlapping knowledge and complementary information across different knowledge graphs. To better utilize such various knowledge, knowledge graph matching has attracted the attention of more and more researchers. The purpose of knowledge graph matching is to integrate different knowledge graphs to form a global knowledge base and establish interoperability between the applications based on different knowledge graphs, especially in the fields of information retrieval, machine reading and knowledge base question answering [5].

The research of knowledge graph matching starts with traditional ontology matching [6], namely matching classes and properties in the schema level of knowledge graphs. More recently, with the fast development of knowledge graphs, the number of entities has increased rapidly. As a result, the entity matching between different knowledge graphs is becoming more and more important to overcome the heterogeneity problem. The process of entity matching decides whether two entities from different knowledge graphs are the same object or not. Owing to the growing number of multilingual knowledge graphs

due to information globalization, entity matching across multilingual knowledge graphs (namely, cross-lingual entity matching) is becoming an urgent sub-problem to be solved for multilingual intelligent applications.

As part of the technologies on knowledge graph representation learning, cross-lingual entity matching has become much easier since we only need to train vector representations of entities from the knowledge graphs of different languages and then decide whether two entities are matched by computing vector similarities. Most of existing studies on cross-lingual entity matching with knowledge graph representation learning only use the structural information of the knowledge graph, i.e, the relation triples in the form of entity, relation, entity to learn the structural embedding of entities. For example, MTransE [7] uses TransE to map two knowledge graphs into their respective vector spaces and then learns the mapping relationship between knowledge graphs. IPTransE [8] and BootEA [9] use iterative methods to continuously discover new matched entities with updated embeddings. In addition to relation triples, some works also consider attribute triples, such as JAPE [10], etc. In addition, KDCoE [11] and HMAN [12] use the text descriptions of entities as the supplement of relation triples to realize entity matching. EVA [13] incorporates images as the complement to align entities across different knowledge graphs.

Although the existing entity matching methods based on representation learning have made remarkable achievements, they still have the following two problems. Firstly, the existing entity matching methods cannot make full use of knowledge graph relevant muti-modal information. Most of them only use the relation triples in knowledge graphs but seldom use other modal information, such as attributes, texts, images, etc. Such information has its own characteristics and can provide useful information for entity representations from different perspectives. For example, text description can provide rich textual context information for entities, and images can provide concrete visual information for entities. Secondly, it is difficult for the existing entity matching methods to integrate the representation learning methods from different modalities. Due to the heterogeneity of cross-lingual knowledge graphs, the representations of entities, relationships, attributes and images with the same meaning in different knowledge graphs may be completely different. That means the entity representation learning from each single modality is insufficient and complementary. Therefore, an effective entity matching method requires dealing with this insufficiency and integrating the information from different modalities.

To address the above problems, we propose a new cross-lingual entity matching method (called CLEM) using knowledge graph representation learning which integrates the rich multi-modal information. The motivation is that the representation of one entity from each single modality (called view) can only capture the partial entity information, while different views describe different aspects of the entity and may share some common redundant information. Therefore, the entity representations can be learned from each particular view and jointly optimized to improve the representation learning (i.e., entity embedding learning) performance. We first attempt to perform representation learning on relation triples, attribute triples, entity text descriptions and images. Then we apply multi-view intact space learning (MISL) [14] to combine multi-modal information of entities and conduct entity matching on cross-lingual datasets. The main contributions are summarized as follows:

- We leverage four views including relations, attributes, text descriptions and images to learn entity representations, and each view corresponds to an independent learning model;
- We apply multi-view intact space learning to solve the insufficiency in each individual view and integrate the multi-modal view information to obtain the entity representation in the intact space;
- We perform experiments on cross-lingual datasets and evaluate our CLEM with different evaluation metrics. The experimental results show that the proposed method outperforms the state-of-the-art methods in most evaluation criteria.

2. Related Work

We discuss two lines of works that are relevant to this paper.

Knowledge Graph Representation Learning. In recent years, knowledge graph representation learning has attracted the attention of researchers. The current representation learning models can be roughly divided into three categories, which are translation-based models, semantic matching models and neural-network-based models. The typical methods of translation-based models are TransE [15] and its improved models TransH [16] and TransR [17]. The core idea of TransE is to regard the relationship as the translation from the head entity to the tail entity. For any relation triples in the knowledge graph, TransE expects that the head entity vector plus the relation vector equals the tail entity vector. The semantic matching models use similarity functions to infer relational facts, e.g., the Hadamard product in ComplEx [18] and the circular correlation in HolE [19]. The neural-network-based models exploit deep learning techniques for knowledge graph embedding. For example, ConvE [20] is a multi-layer convolution neural network which learns the representation through deep network structure and convolution operations; R-GCN [21] is a relational graph convolution neural network which generates the representation by convolving semantic information on the local graph structure. All the above models focus on relational facts and are mostly evaluated by the task of link prediction in a single knowledge graph.

Entity Matching. With the emergence of various knowledge graphs in different domains, entity matching, especially cross-lingual entity matching, is becoming more and more important to solve the problem of heterogeneity. Entity matching decides whether two entities of different knowledge graphs refer to the same object. Ref. [22] transforms the entity matching problem based on attribute similarity scores into a multi-classification problem, which is divided into matching, possible matching and mismatching, and establishes the probability model of the entity matching problem. Although the above method is simple and intuitive, it requires sufficient labeled matching entity pairs and the corresponding annotation costs. Additionally, the calculation of feature similarities is often interfered with by semantic heterogeneity among different knowledge graphs. Recently, the use of the embedding models to match the entities across cross-lingual knowledge graphs has attracted extensive attention. JE [23] uses the model TransE to embed two independent knowledge graphs into the same vector space; MTransE [7] jointly trains a translation embedding model to encode language-specific knowledge graphs in separate embedding spaces and align counterpart entities across embeddings. Some studies also consider other aspects of the entity for the matching task. Some of them (e.g., MultiKE [24] and DAEA [25]) consider the name and structure of entities; others (e.g., JAPE [10], GCN-Align [26], AttrE [27] and RAKA [28]) consider entity attributes; some works (e.g., KDCoE [11] and HMAN [12]) consider the text descriptions of entities. Moreover, EVA [13] incorporates images along with structures, relations and attributes to align entities across different knowledge graphs.

3. Method

3.1. Problem Definition

In this paper, we study the cross-lingual entity matching using knowledge graph representation learning, which aims to learn the multi-view entity embeddings based on different modal perspectives (i.e., views). We consider four views for the entity embedding, including relation view, attribute view, description view and image view. Based on these views, we formalize a knowledge graph as a 6-tuple $G = \{E, R, A, L, X, I\}$, where E, R, A, L, X and I are respectively employed to represent entities, relations, attributes, attribute values, entity text descriptions and images of knowledge graph G.

The problem of entity matching based on knowledge graph representation learning can be defined as follows: Assuming G_1 and G_2 are, respectively, employed to represent two different knowledge graphs.

$$G_1 = \{E_1, R_1, A_1, L_1, X_1, P_1\} \tag{1}$$

$$G_2 = \{E_2, R_2, A_2, L_2, X_2, P_2\} \tag{2}$$

S denotes the set of known equivalent entity pairs from G_1 and G_2, and $\equiv$ denotes the equivalent relation.

$$S = \{(e_1, e_2)|e_1 \in E_1, e_2 \in E_2, e_1 \equiv e_2\} \tag{3}$$

We obtain the embedded representation of each entity from G_1 and G_2 by using the representation learning model of different views and integrate the multi-modal information from each view to generate the final entity representation. Assuming $\mathbf{E_1}$ and $\mathbf{E_2}$ respectively represent the embedding matrix for the set of all entities E_1 and E_2, then, the function $\Im$ computes the similarities between the entity representations to find the equivalent entity set M.

$$M = \Im(\mathbf{E_1}, \mathbf{E_2}) \tag{4}$$

The purpose is to find a set M that contains all matched entity pairs which do not belong to the known set S.

$$M = \{(e_1, e_2)|e_1 \in E_1, e_2 \in E_2, e_1 \equiv e_2, (e_1, e_2) \notin S\} \tag{5}$$

3.2. Overall Framework

Our framework is shown in Figure 1 and consists of two parts: entity representation learning in different views and multi-view entity matching.

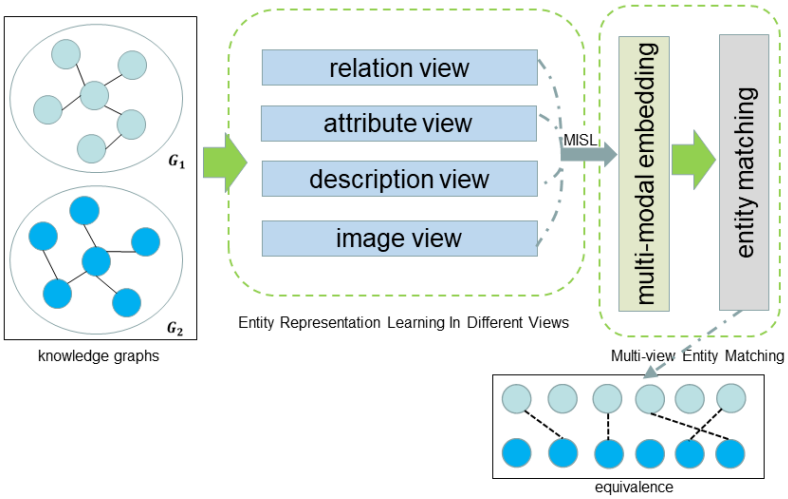

Figure 1. The framework of entity matching. G_1 and G_2 denote different knowledge graphs, *MISL* means multi-view intact space learning, and we use *MISL* to integrate entity information from different modalities.

Entity Representation Learning in Different Views. We make full use of the knowledge graph, extracting the entity information from the relation view, the attribute view, the description view and the image view.

Multi-view Entity Matching. Since the embedded representations of entities from a single view only capture partial entity information, we employ multi-view intact space learning to integrate the complementary information from each view, which obtains the multi-view embedded representations. We utilize the K-nearest neighbor algorithm [29] to find out all matched entity pairs.

3.3. Entity Representation Learning in Different Views

This part consists of the entity embedded representations from four views. We study the embedding models for different views, respectively.

3.3.1. Relation View

Relations are an important part of a knowledge graph, since the relation view describes the structures among different entities. The translation-based knowledge graph embedding model shows its power in characterizing such relational structures [30]. Therefore, we employ the TransE [15] model, which has good generalization capability, to embed knowledge graphs from the relation view. Assuming G_1 and G_2 denote the two knowledge graphs to match, the translation score could be obtained as below:

$$f_{rel}(h, r, t) = ||\mathbf{h} + \mathbf{r} - \mathbf{t}||_2^2 \tag{6}$$

where $T = (h, r, t)$ denotes a relation triple in $G \in \{G_1, G_2\}$ such that h means the head entity, t means the tail entity, and r means the relation. To learn the common embeddings of entities and relations, we embed two different knowledge graphs into the same vector space. The learning objective of the model can be achieved by minimizing the following margin-based [31] loss function :

$$Loss_{rel} = \sum_{G \in \{G_1, G_2\}} \sum_{(h,r,t) \in G} ||f_{rel}(h, r, t) - f_{rel}(\tilde{h}, r, \tilde{t}) + \gamma|| + \sum_{(e_1, e_2) \in S} ||\mathbf{e}_1 - \mathbf{e}_2|| \tag{7}$$

where $(\tilde{h}, r, \tilde{t})$ is the negative example obtained by randomly replacing the head entity or tail entity in (h, r, t) with another entity, S means the known equivalent entity set, which is defined in Section 3.1, and γ is the margin parameter describing the boundary between positive and negative examples.

3.3.2. Attribute View

The attribute view characterizes attribute and attribute value information of entities. There are word abbreviations in attributes and cross-lingual differences in attribute values, which make entity matching based on the attribute view quite difficult. To leverage the information of attributes and attribute values to help match entities, we use a graph convolutional network (GCN) [32] to encode the neighbour attributes and attribute values into the the low-dimensional representations of the given entities. GCN can perform feature extraction on arbitrary graphs and will not be affected by the number of neighbour nodes associated with entities.

Firstly, we construct the entity attribute graph. All attribute triples are classified based on the head entity, i.e., attribute triples of the same head entity are grouped together. Then, for an entity e, we collect all its attributes and attribute values. We combine each collected attribute and its corresponding attribute value as a neighbour node of the entity node in the entity attribute graph. Figure 2 shows an example of the entity attribute graph for the entity "Donald Trump".

Secondly, we adopt GCN to encode attributes and attribute values into the representations of entities. We initialize the vector representations of entities, attributes and attribute values using the pre-training model BERT [33]. BERT is a multi-layer bidirectional Transformer encoder, and we use its original implementation, the BERT-base model. To obtain the representation of each neighbour node for the entity node in the built entity attribute graph, we perform element-wise multiplication between the vectors of attributes and attribute values, respectively. Afterwards, GCN is utilized to conduct convolution operations as follows:

$$\mathbf{H}^{(l+1)} = ReLU(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l+1)}) \tag{8}$$

where $\mathbf{H}$ denotes the feature matrix of all nodes, each row in $\mathbf{H}$ is the vector representation of a node, $\mathbf{A}$ is an adjacency matrix showing the connectivity between nodes, $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, $\mathbf{I}$ is the unit matrix, $\hat{\mathbf{D}}$ is the diagonal matrix of $\hat{\mathbf{A}}$, $\mathbf{W}$ is the weight matrix, and l means the l-layer. We choose $ReLU = \max(0, x)$ for the activation function.

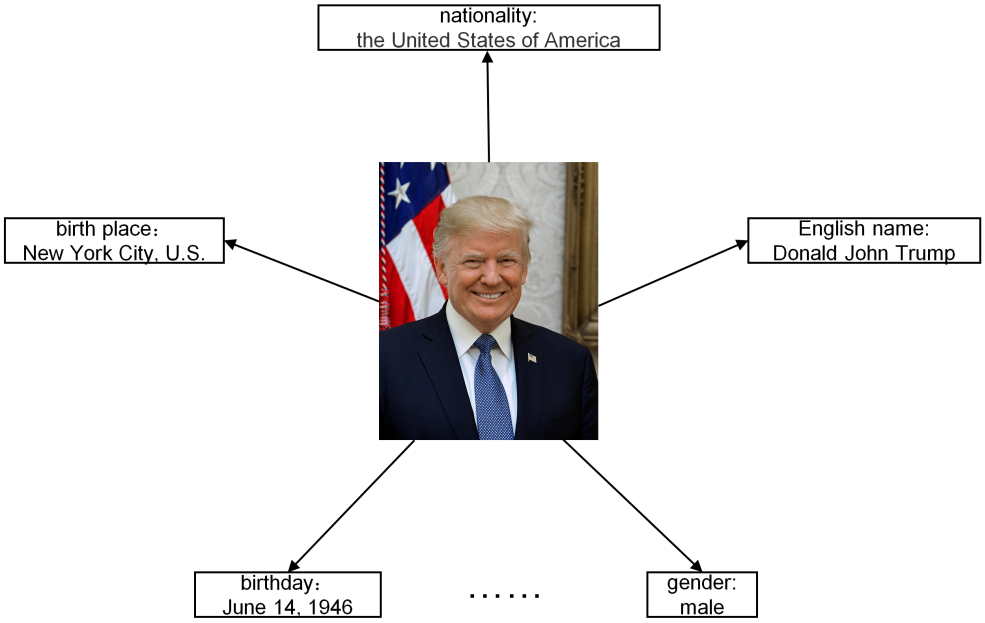

Figure 2. The example of the entity attribute graph for the entity "Donald Trump".

Owing to the single-layer structure of the entity attribute graph, we assign a single-layer GCN for each knowledge graph to encode attributes and attribute values into the representations of entities. Specifically, given two knowledge graphs G_1 and G_2, the objective can be achieved by minimizing the following margin-based loss function:

$$Loss_{attr} = \sum_{(e_1,e_2)\in S} [f(e_1,e_2) - \gamma_1]_+ + \sum_{(e_1{}',e_2{}')\in S'} [\gamma_2 - f(e_1{}',e_2{}')]_+ \quad (9)$$

where $[x]_+$ means $max\{0,x\}$, $f(x,y)$ means the L_2 distance of vector $\mathbf{x}$ and vector $\mathbf{y}$, $(e_1{}',e_2{}')$ is a negative example obtained by randomly replacing one of the two entities in (e_1,e_2) with another entity, and $\gamma_{1/2}$ is the margin parameter describing the boundary between positive and negative examples.

3.3.3. Description View

The description view contains rich textual semantic information, which is difficult to obtain from other views. To extract such semantic information from the description view, we encode the entity text descriptions through two steps. Firstly, the embedded representation of words in the entity text descriptions is obtained by using a pre-training model. After pre-training, the description of each entity will be converted into a vector sequence, which will be input to the description encoder, and then the encoder will output the embedded representation of the entity text description. The details are as follows:

For the pre-training language model, we also use BERT which is proposed by Devlin et al. [33]. As for the encoder, we adopt gated recurrent unit (GRU) [34] since it can model the sequential data well, while it is computationally more efficient than LSTM [35] or other models. GRU is a kind of recurrent neural network (RNN), which is often used to encode natural language text. It consists of two types of gating units, namely the reset gate and the update gate, which are used to track the sequence status without using a separate storage unit. In this paper, two stacked GRU layers are used to model entity descriptions in different knowledge graphs. The input is the embeddings of an entity and its descriptions obtained by BERT. We follow the standard method where the tokens are surrounded by [CLS] and [SEP] on the left and right, respectively. For example, if the input entity and description are "*Trump*" and "*President Trump has insisted on a full-scale convention.*", the input to our encoder would be: [CLS]+ "*Trump*" +[SEP]+ "*President Trump*

has insisted on a full-scale convention." + [SEP]. The output of the second GRU layer can be obtained as follows:

$$\mathbf{t}_e = ReLU(\mathbf{W}_o[s_1; s_2; \ldots; s_l] + \mathbf{b}_o) \tag{10}$$

where $\mathbf{W}_o$ denotes the mapping matrix, $\mathbf{b}_o$ is the bias, and l is the length of the entity and description sequence. We also choose ReLU as the activation function. Then, each $\mathbf{t}_e$ is normalized to $||\mathbf{t}_e||_2^2 = 1$. Finally, we can maximize the similarity (i.e., the dot product) between the embeddings of descriptions for matched entities and minimize the dot product between the embeddings of descriptions for irrelevant entities. This objective can be achieved by minimizing the following loss function:

$$Loss_{text} = -\sum_{(e_1,e_2)\in S} \{log(ReLU(\mathbf{t}_{e_1}^T \mathbf{t}_{e_2})) + \sum_{i=1}^{k} E_{e^{i\sim U(e^i \in E_2)}}[log(ReLU(-\mathbf{t}_{e_1}^T \mathbf{t}_{e^i}))]\} \tag{11}$$

where S means the known equivalent entity set, k is the number of negative entity examples, and U is the distribution of the entities in the entity set E_2.

3.3.4. Image View

The image view provides concrete images of entities, characterizing the visual information of entities more intuitively and vividly. We can directly distinguish the president "Donald Trump" from the physicist "John G. Trump" because the facial features are totally different.

To extract image features, we obtain the embeddings of the entity images according to the VGG16 [36] model. The VGG16 model was pre-trained on the ILSVRC 2012 dataset which is derived from ImageNet [37]. The model we introduce has thirteen convolutional layers, which are followed by three fully connected layers. To obtain the embeddings of the entity images, we remove the softmax layer. Owing to the fact that the image vectors generated from the VGG16 model do not share the same vector space with entity embeddings, we use a *map* function to make them in the same space. Specifically, given a pair of an entity and its corresponding image $\{(e,i)|e \in E, i \in I\}$, where E and I denote the entity and image set in the given two knowledge graphs, respectively, we extract image features as follows:

$$f_{im}(e,i) = -||\mathbf{e} - ReLU(map(\mathbf{i}))||_2^2 \tag{12}$$

where $\mathbf{i}$ denotes the embedding of the entity image i, and each $\mathbf{i}$ is normalized to $||\mathbf{i}||_2^2 = 1$. ReLU is the activation function. Given the known equivalent entity set S, we can maximize the similarity (i.e., the dot product) between the embeddings of images for matched entities and minimize the dot product between the embeddings of images for irrelevant entities. Thus, we define the following loss function:

$$\begin{aligned} Loss_{im} = \sum_{(e,i)\in\{(e,i)|e\in E, i\in I\}} log(1 + exp(-f_{im}(e,i))) - \sum_{(e_1,e_2)\in S} \{log(ReLU(\mathbf{i}_{e_1}^T \mathbf{i}_{e_2})) \\ + \sum_{i=1}^{k} E_{e^{i\sim U(e^i \in E_2)}}[log(ReLU(-\mathbf{i}_{e_1}^T \mathbf{i}_{e^i}))]\} \end{aligned} \tag{13}$$

where k is the number of negative entity examples, and U is the distribution of the entities in the entity set E_2.

3.4. Multi-View Entity Matching

In Section 3.3, we introduce four representation learning models for modeling different views. Each model can be trained to obtain the embedded representations of entities, and the entity matching task can be completed by calculating similarities between entity embeddings. However, the performance of entity matching cannot be maximized by only using one of the models [14]. Therefore, it is crucial to combine the four representations for

each entity, i.e., integrate the multi-modal information from each view to generate the final entity representations.

3.4.1. Multi-View Intact Space Learning

The information from different views can provide different aspects of one entity. For example, the attribute view can provide abstract attribute information of entities themselves, and the relation view can provide the relational structures among different entities. However, each view only captures the entity information from one aspect, and all views may share the common redundant information. We adopt the multi-view intact space learning (MISL) [14] to integrate multi-view entity representations to solve the insufficiency problem when using each individual view .

We assume a multi-view training data set $D = \{\mathbf{z}_i^v | 1 \leq i \leq n, 1 \leq v \leq m\}$, where $\mathbf{z}_i^v$ denotes the v-th view of the i-th entity embedding, n is the number of the entity, and m is the number of the entity embedding views. Suppose that $\mathbf{x}_i$ represents the embedding of an entity in the intact space, so the loss function of the m-view space being reconstructed by the intact space can be defined as follows:

$$Loss_{MISL} = \frac{1}{mn} \sum_{i=1}^{n} \sum_{v=1}^{m} log(1 + \frac{||\mathbf{z}_i^v - \mathbf{W}_v \mathbf{x}_i||^2}{c^2}) + C_1 \sum_{v=1}^{m} ||\mathbf{W}_v||_F^2 + C_2 \sum_{i=1}^{n} ||\mathbf{x}_i||_2^2 \quad (14)$$

where c is a constant scale parameter, C_1 and C_2 are non-negative constants, and $\mathbf{W}_v$ is the v-th view generation matrix. The optimization problem can be decomposed into two sub-problems over the view generation function $\mathbf{W}$ and the variable on entity embedding $\mathbf{x}$ with the alternating optimization method. Given fixed view generation functions $\{\mathbf{W}_v\}_{v=1}^m$, the loss function can be reduced as follows:

$$min_x(Loss_{MISL}) = \frac{1}{m} \sum_{v=1}^{m} log(1 + \frac{||\mathbf{z}^v - \mathbf{W}_v \mathbf{x}||^2}{c^2}) + C_2 ||\mathbf{x}||_2^2 \quad (15)$$

Setting the gradient of $Loss_{MISL}$ with respect to $\mathbf{x}$ to 0, we can obtain the following equations:

$$Q^v = \frac{1}{||\mathbf{z}^v - \mathbf{W}_v \mathbf{x}||^2 + c^2} \quad (16)$$

$$\mathbf{x} = (\sum_{v=1}^{m} \mathbf{W}_v^T Q^v \mathbf{W}_v + mC_2)^{-1} \sum_{v=1}^{m} \mathbf{W}_v^T Q^v \mathbf{z}^v \quad (17)$$

Given fixed entity embeddings $\{\mathbf{x}_i\}_{i=1}^n$, the loss function can be reduced as follows:

$$min_W(Loss_{MISL}) = \frac{1}{n} \sum_{i=1}^{n} log(1 + \frac{||\mathbf{z}_i - \mathbf{W}\mathbf{x}_i||^2}{c^2}) + C_1 ||\mathbf{W}||_2^2 \quad (18)$$

We can set the gradient of $Loss_{MISL}$ with respect to $\mathbf{W}$ to 0 and obtain the following equations:

$$Q_i = \frac{1}{||\mathbf{z}_i - \mathbf{W}\mathbf{x}_i||^2 + c^2} \quad (19)$$

$$\mathbf{W} = \sum_{i=1}^{n} \mathbf{z}_i Q_i \mathbf{x}_i^T (\sum_{i=1}^{n} \mathbf{z}_i Q_i \mathbf{x}_i^T + nC_1)^{-1} \quad (20)$$

By constantly iterating the above formulas until the entity embedding $\mathbf{x}$ converges, we can obtain the final integrated embedding for each entity (see Algorithm 1).

3.4.2. Entity Matching Algorithm

This section introduces the whole entity matching process in Algorithm 1. All random initialization for the embeddings from the relation view is provided by the *Xavier*

initializer [38], and the pre-training initialization for the attribute and description view is provided by the BERT. The input of the algorithm is two knowledge graphs G_1 and G_2 to be matched, the known equivalent entity set S and the number of iterations Q. We first separately train the entity embeddings from the relation view, attribute view, description view and image view until Q is reached. Then, we obtain the final multi-view entity representations by MISL. Finally, we find matched entities by the K-nearest neighbor algorithm [39] on the final entity representations. The output of the algorithm is the predicted pairs of matched entities M across two knowledge graphs in different languages.

Algorithm 1 The Entity Matching Algorithm in CLEM.

Input: G_1, G_2, the known equivalent entity set S, and the number of iterations Q.
Output: The predicted matched entity pairs M.

for q = 1, 2, 3,. . . ,Q **do**
    Minimize $Loss_{rel}$ under the relation view;
    Minimize $Loss_{attr}$ under the attribute view;
    Minimize $Loss_{text}$ under the description view;
    Minimize $Loss_{im}$ under the image view;
end for
repeat
    Iterate the Equations (16), (17), (19) and (20);
until The entity embedding **x** in the intact space converges
Find matched entities by the K-Nearest Neighbor algorithm on the final entity representations;
return The predicted matched entity pairs M

4. Experiments

To verify the effectiveness of our proposed method, we used Python to implement our approach with the aid of PyTorch (https://pytorch.org/) All reported experiments were performed on one Linux server with Xeon CPU (2.10GHz) processor with 64 GB RAM and one NVIDIA TITAN Xp GPU (8 GB).

4.1. Dataset

The experiments were performed on DBP15K [10], which is a subset of DBpedia and a classic benchmark dataset for entity matching. DBpedia is a large multilingual knowledge graph which extracts structured content from Wikipedia. We obtained entity descriptions and entity images, respectively, from the "dbpedia-owl:abstract" and "dbpedia-owl:thumbnail". As for the missing entity images, we obtained the supplement from Google Images (https://images.google.com) by querying entity names. DBP15K contains three sub-datasets, i.e., Chinese–English (zh-en), Japanese–English (ja-en), and French–English (fr-en) subsets, which contain language links between different language versions. Table 1 shows the details of DBP15K. Each subset contains the knowledge in two languages and 15,000 pairs of matched entities from DBpedia. In the experiment, the known matched entity pairs were used for model training and testing.

Table 1. The details of the DBP15K dataset.

Sub-Dataset	#Entity	#Relation	#Relation Triple	#Attribute Triple
Chinese	66,469	2830	153,929	379,684
English	98,125	2317	237,674	567,755
Japanese	65,744	2043	164,373	354,619
English	95,680	2096	233,319	497,230
French	66,858	1379	192,191	528,665
English	105,889	2209	278,590	576,543

4.2. Evaluation Measures

Link prediction is used to predict possible links in knowledge graphs or compute link losses caused by incomplete data. Similar to MTransE [7] and subsequent related works, we used link prediction as the evaluation method for entity matching. We applied mean reciprocal rank (MRR) and Hits@n as the evaluation criteria. Given query samples Q, the MRR and Hits@n are defined as follows:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{21}$$

where $rank_i$ is the rank of the correct answer in the response list for the i-th query.

$$Hits@n = \frac{1}{|Q|} \sum_{i=1}^{|Q|} Hit(q_i, L_i, n) \tag{22}$$

where $Hit(q_i, L_i, n)$ means that if the correct answer of the query q_i is at the first n items for the ranked list L_i, then its value is one, otherwise, the value is zero.

Moreover, we also tried to record traditional evaluation metrics for entity matching including precision, recall and F1-score. However, since the embedding-based methods always return a list of candidates for each input entity, recall and F1-score are equal to precision, and the precision actually equals Hits@1 [24].

4.3. Comparison Methods

We compared our method with the following baselines:

- **MTransE [7]**: MTransE is one representative work of multilingual knowledge graph embedding for entity matching. MTransE combines monolingual models with a jointly trained alignment model and achieves good results on the single relation view.
- **JAPE [10]**: JAPE is a joint embedding model on relation triples and attribute triples for entity matching between knowledge graphs.
- **KDCoE [11]**: KDCoE is a semi-supervised entity matching method based on collaborative training which enhances multilingual knowledge graph embedding. It iteratively trains the two parts of the embedding model on the relation triples and entity descriptions in different languages, respectively.
- **MultiKE [24]**: Multi-KE is an entity matching framework based on multi-view knowledge graph embedding. The underlying idea is to divide the various features of knowledge graphs into multiple subsets, which are complementary to each other.
- **EVA [13]**: This research proposes the idea of entity alignment using visual information, which incorporates images along with relations and attributes to align entities in different knowledge graphs.
- **DAEA [25]**: DAEA utilizes graph convolutional networks (GCNs) to integrate the information of entities, relations, attributes and entity name embeddings to learn a unified latent representation to perform cross-lingual entity alignment.

Moreover, we performed the ablation experiment and the multi-view integration experiment to evaluate the effectiveness of each view and our MIST method. For the ablation experiment, we eliminated one of the four views to generate the CLEM variant each time and denoted the CLEM variants without the image view, relation view, the attribute view and the description view by CLEM-RAD, CLEM-ADI, CLEM-RDI and CLEM-RAI, respectively. For the the multi-view integration experiment, we followed three multi-view integration methods of the MultiKE model (i.e., weighted view averaging, shared space learning, and in-training combination) to generate the CLEM variants which are denoted by CLEM-WVA, CLEM-SSL and CLEM-ITC, respectively. In addition, the the multi-view integration experiment included the CLEM variant which employs the vector concatenation strategy to combine the view-specific embeddings. This CLEM variant is denoted by CLEM-CAT.

4.4. Experiment Settings

The following hyper-parameters were used in the experiments. Each training took Q = 3000 epochs, and the proposed model used in our experiments was trained with the Adam as optimizer, the batch size among $\{32, 64, 128, 256, 512\}$, the embedding dimension among $\{50, 75, 100, 125, 150\}$, the margin among among $\{0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$ and the learning rate among $\{0.0001, 0.0005, 0.001, 0.005, 0.01\}$. The best hyperparameters for "(batch size, embedding dimension, margin, learning rate)" vary in different sub-datasets, and the details are given in Table 2. For the baseline models, we used the reported results in their papers or ran their source codes on DBP15K. Specifically, for MTransE and JAPE, we used the results given in [10] (page 639); for EVA and DAEA, we used the results given in [13] (page 4261) and [25] (page 6), respectively; for MultiKE (https://github.com/nju-websoft/MultiKE) and KDCoE (https://github.com/muhaochen/MTransE-tf), we applied their published source codes to DBP15K to obtain evaluation results.

Table 2. The best hyper-parameters for different sub-datasets of DBP15K.

Dataset	Batch Size	Embedding Dimension	Margin	Learning Rate
zh-en	0.005	125	1	256
ja-en	0.005	125	1.5	256
fr-en	0.005	125	1.5	128

4.5. Results and Discussions

Table 3 shows the comparison results between CLEM and other entity matching methods. We found that CLEM achieved the best performance on the three sub-data sets. For example, on DBP15K zh-en, CLEM achieved the Hits@1 score 85.41% with 2.6% improvement compared to the second best method. The effectiveness of CLEM partly comes from the integration of the multi-modal entity information. CLEM fuses the multi-modal information from the relation view, attribute view, description view and the image view. In contrast to that, MTransE, JAPE and KDCoE only use a part of the above four views. Moreover, although MultiKE uses multi-view integration models to deal with each kind of entity information equally, but the performance is not as good as ours (details are presented in Figure 3). Instead, CLEM adopts multi-view intact space learning (MISL) to solve the insufficiency problem of each individual view and integrate the multi-modal information from all given views. As for DAEA and EVA, they perform relatively well. EVA uses the additional visual embedding to characterise each entity in the embedding space, while EVA ignores the importance of the text description. Although DAEA integrates multi-aspect information to achieve good results, it neglects the use of entity images to eliminate the ambiguity caused by text relevant modal information.

The Ablation Experiment. Table 4 shows the results of the ablation experiment. As expected, the performance of different variants without the corresponding modal information decreases obviously. The results show that each modal information is indispensable and plays an important role in the task of entity matching. In addition, we noticed that CLEM-RAD performed the worst among these variants. This phenomenon indicates that CLEM-RAD lacks the most effective view of the four views, i.e., the image view. The effectiveness of the image view is attributed to the fact that we extracted visual features to eliminate the ambiguity in the relation view, attribute view and text view, which is common in heterogeneous knowledge graphs.

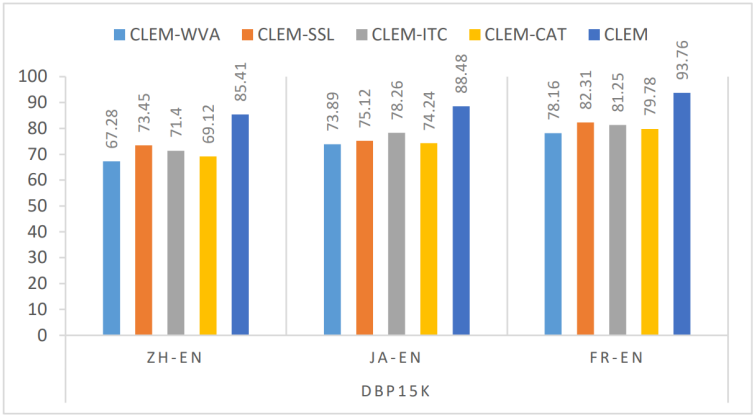

Figure 3. The result of the multi-view integration experiment in terms of Hits@1.

Table 3. The comparison results between CLEM and other entity matching methods.

DBP15K	zh-en			ja-en			fr-en		
Methods	**Hits@1 (Prec.)**	**Hits@10**	**MRR**	**Hits@1 (Prec.)**	**Hits@10**	**MRR**	**Hits@1 (Prec.)**	**Hits@10**	**MRR**
MTransE	30.83	61.41	36.40	27.86	57.45	34.90	24.41	55.55	33.50
JAPE	41.18	74.46	49.00	36.25	68.50	47.60	32.39	66.68	43.00
KDCoE	43.42	75.77	52.70	39.48	70.75	50.50	33.59	69.52	44.70
MultiKE	50.87	57.61	53.20	39.30	48.85	42.60	63.94	71.19	66.50
EVA	76.10	90.70	81.40	76.20	91.30	81.70	79.30	94.20	84.70
DAEA	82.80	92.40	84.30	87.00	95.10	88.20	93.60	97.10	94.80
CLEM	**85.41**	**93.45**	**87.90**	**88.48**	**95.81**	**90.40**	**93.60**	**97.64**	**95.20**

Moreover, although we found that the image view is the most effective view, the entity matching results rely on image quality to a certain extent according to our error analysis. Since some images are missing in the DBpedia dataset, we collected images from Google Images which also introduces some noise due to the color, background and brightness differences. Such noisy images do cause a number of errors in the entity matching results.

Table 4. The comparison results of CLEM and its variants in the ablation experiments.

DBP15K	zh-en			ja-en			fr-en		
Models	**Hits@1**	**Hits@10**	**MRR**	**Hits@1**	**Hits@10**	**MRR**	**Hits@1**	**Hits@10**	**MRR**
CLEM-RAD	53.45	80.02	56.50	51.70	84.19	55.07	57.13	71.61	60.40
CLEM-ADI	64.45	83.41	68.20	74.37	80.38	77.90	74.25	80.40	76.20
CLEM-RDI	56.46	78.35	60.80	62.45	73.62	65.10	67.85	72.24	69.30
CLEM-RAI	61.87	80.29	64.70	69.09	82.64	74.30	71.99	78.20	74.50
CLEM	**85.41**	**93.45**	**87.90**	**88.48**	**95.81**	**90.40**	**93.60**	**97.64**	**95.20**

The Multi-View Integration Experiment. To verify the effectiveness of our multi-view representation learning method, we conducted the multi-view integration experiment in terms of Hits@1. Figure 3 shows the results. Compared with other variants, we observed that the Hits@1 of our proposed method increased at least 10.22%. This is because compared with the multi-view integration methods of MultiKE, our applied MISL handled the relevance and complementarity between different modal entity information in the intact space better. The use of the MISL method successfully integrates the multi-modal information from each view to generate the final entity representation, which effectively

improves the results of the entity matching. In addition, we found that the CLEM-CAT is inferior to the other variants. This may be due to the fact that the concatenation method does not have the training process, and the integration after training cannot fully consider the association between different modalities.

5. Conclusions

In this paper, we proposed a cross-lingual knowledge graph embedding method CLEM for entity matching, which extracts and integrates the rich multi-modal information from different views, including the relation view, attribute view, description view and image view. The multi-view intact space learning was adopted to generate the final multi-view entity representations, which effectively solved the insufficiency problem of the single modal information in cross-lingual entity matching. Our experiments on three real-world datasets show the effectiveness of CLEM and our multi-view learning strategy.

As for future work, we plan to study the few-shot cross-lingual entity matching with knowledge graph embedding when the known matched entities are quite few. Moreover, we will explore a global matching method for align at least three knowledge graphs in different languages simultaneously.

Author Contributions: T.W. and C.G. proposed the idea of this work and designed the method; L.L. finished a part of the experiments; T.W. and C.G. wrote the paper; Y.W. revised this paper and gave a lot of suggestions. All authors read and approved the final manuscript.

Funding: This work is supported by the NSFC (Grant No. 62006040, 62072149), the Project for the Doctor of Entrepreneurship and Innovation in Jiangsu Province (Grant No. JSSCBS20210126), the Fundamental Research Funds for the Central Universities, and ZhiShan Young Scholar Program of Southeast University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Wu, T.; Qi, G.; Li, C.; Wang, M. A survey of techniques for constructing Chinese knowledge graphs and their applications. *Sustainability* **2018**, *10*, 3245. [CrossRef]
2. Wu, W.; Li, H.; Wang, H.; Zhu, K.Q. Probase: A Probabilistic Taxonomy for Text Understanding. In Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, Scottsdale, AZ, USA, 20–24 May 2012; pp. 481–492.
3. Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Hruschka, E.R.; Mitchell, T.M. Toward an Architecture for Never-Ending Language Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Atlanta, GA, USA, 11–15 July 2010.
4. Wu, T.; Wang, H.; Li, C.; Qi, G.; Niu, X.; Wang, M.; Li, L.; Shi, C. Knowledge graph construction from multiple online encyclopedias. *World Wide Web* **2020**, *23*, 2671–2698. [CrossRef]
5. Wang, J.; Wang, X.; Ma, C.; Kou, L. A survey on the development status and application prospects of knowledge graph in smart grids. *IET Gener. Transm. Distrib.* **2021**, *15*, 383–407. [CrossRef]
6. Liu, X.; Tong, Q.; Liu, X.; Qin, Z. Ontology Matching: State of the Art, Future Challenges, and Thinking Based on Utilized Information. *IEEE Access* **2021**, *9*, 91235–91243. [CrossRef]
7. Chen, M.; Tian, Y.; Yang, M.; Zaniolo, C. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. *arXiv* **2016**, arXiv:1611.03954.
8. Zhu, H.; Xie, R.; Liu, Z.; Sun, M. Iterative Entity Alignment via Joint Knowledge Embeddings. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Melbourne, Australia, 19–25 August 2017; pp. 4258–4264.
9. Sun, Z.; Hu, W.; Zhang, Q.; Qu, Y. Bootstrapping Entity Alignment with Knowledge Graph Embedding. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, Sweden, 13–19 July 2018; pp. 4396–4402.
10. Sun, Z.; Hu, W.; Li, C. Cross-Lingual Entity Alignment via Joint Attribute-Preserving Embedding. In Proceedings of the International Semantic Web Conference, PART I, Vienna, Austria, 21–25 October 2017; Springer: Cham, Switzerland; pp. 628–644.
11. Chen, M.; Tian, Y.; Chang, K.W.; Skiena, S.; Zaniolo, C.. Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. *arXiv* **2018**, arXiv:1806.06478.
12. Yang, H.W.; Zou, Y.; Shi, P.; Lu, W.; Lin, J.; Sun, X. Aligning cross-lingual entities with multi-aspect information. *arXiv* **2019**, arXiv:1910.06575.

13. Liu, F.; Chen, M.; Roth, D.; Collier, N. Visual Pivoting for (Unsupervised) Entity Alignment. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 2–9 February 2021; pp. 4257–4266.
14. Xu, C.; Tao, D.; Xu, C. Multi-view intact space learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 2531–2544. [CrossRef]
15. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 1–8.
16. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge Graph Embedding by Translating on Hyperplanes. In Proceedings of the AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; pp. 1112–1119.
17. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 2181–2187.
18. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex embeddings for simple link prediction. In Proceedings of the International Conference on Machine Learning, New York City, NY, USA, 19–24 June 2016; pp. 2071–2080.
19. Nickel, M.; Rosasco, L.; Poggio, T. Holographic Embeddings of Knowledge Graphs. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 1955–1961.
20. Dettmers, T.; Minervini, P.; Stenetorp, P.; Riedel, S. Convolutional 2D Knowledge Graph Embeddings. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 1811–1818.
21. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Berg, R.v.d.; Titov, I.; Welling, M. Modeling Relational Data with Graph Convolutional Networks. In Proceedings of the European Semantic Web Conference, Heraklion, Crete, Greece, 3–7 June 2018; Springer: Berlin/Heidelberg, Germany; pp. 593–607.
22. Fellegi, I.P.; Sunter, A.B. A theory for record linkage. *J. Am. Stat. Assoc.* **1969**, *64*, 1183–1210. [CrossRef]
23. Hao, Y.; Zhang, Y.; He, S.; Liu, K.; Zhao, J. A Joint Embedding Method for Entity Alignment of Knowledge Bases. In Proceedings of the China Conference on Knowledge Graph and Semantic Computing, Beijing, China, 19–22 September 2016; Springer: Berlin/Heidelberg, Germany; pp. 3–14.
24. Zhang, Q.; Sun, Z.; Hu, W.; Chen, M.; Guo, L.; Qu, Y. Multi-view knowledge graph embedding for entity alignment. *arXiv* **2019**, arXiv:1906.02390.
25. Zhang, G.; Zhou, Y.; Wu, S.; Zhang, Z.; Dou, D. Cross-lingual entity alignment with adversarial kernel embedding and adversarial knowledge translation. *arXiv* **2021**, arXiv:2104.07837.
26. Wang, Z.; Lv, Q.; Lan, X.; Zhang, Y. Cross-lingual Knowledge Graph Alignment via Graph Convolutional Networks. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 349–357.
27. Trisedya, B.D.; Qi, J.; Zhang, R. Entity Alignment between Knowledge Graphs Using Attribute Embeddings. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 297–304.
28. Chen, B.; Zhang, J.; Tang, X.; Chen, H.; Li, C. RAKA: Co-Training of Relationships and Attributes for Cross-lingual Knowledge Alignment. *arXiv* **2019** arXiv:1910.13105.
29. Fix, E.; Hodges, J.L. Discriminatory analysis. Nonparametric discrimination: Consistency properties. *Int. Stat. Rev. Int. Stat.* **1989**, *57*, 238–247. [CrossRef]
30. Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; Philip, S.Y. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 494–514. [CrossRef]
31. Wu, C.Y.; Manmatha, R.; Smola, A.J.; Krahenbuhl, P. Sampling matters in deep embedding learning. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2840–2848.
32. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2017** arXiv:1609.02907.
33. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
34. Jozefowicz, R.; Zaremba, W.; Sutskever, I. An Empirical Exploration of Recurrent Network Architectures. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 2342–2350.
35. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [CrossRef]
36. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
37. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
38. Kumar, S.K. On weight initialization in deep neural networks. *arXiv* **2017**, arXiv:1704.08863.
39. Peterson, L.E. K-nearest neighbor. *Scholarpedia* **2009**, *4*, 1883. [CrossRef]