Dual Gated Graph Attention Networks with Dynamic Iterative Training for Cross-Lingual Entity Alignment

ZHIWEN XIE, Wuhan University
RUNJIE ZHU, York University and AI Singapore
KUNSONG ZHAO and JIN LIU, Wuhan University
GUANGYOU ZHOU, Central China Normal University
JIMMY XIANGJI HUANG, York University

44

Cross-lingual entity alignment has attracted considerable attention in recent years. Past studies using conventional approaches to match entities share the common problem of missing important structural information beyond entities in the modeling process. This allows graph neural network models to step in. Most existing graph neural network approaches model individual knowledge graphs (KGs) separately with a small amount of pre-aligned entities served as anchors to connect different KG embedding spaces. However, this characteristic can cause several major problems, including performance restraint due to the insufficiency of available seed alignments and ignorance of pre-aligned links that are useful in contextual information in-between nodes. In this article, we propose DuGa-DIT, a dual gated graph attention network with dynamic iterative training, to address these problems in a unified model. The DuGa-DIT model captures neighborhood and cross-KG alignment features by using intra-KG attention and cross-KG attention layers. With the dynamic iterative process, we can dynamically update the cross-KG attention score matrices, which enables our model to capture more cross-KG information. We conduct extensive experiments on two benchmark datasets and a case study in cross-lingual personalized search. Our experimental results demonstrate that DuGa-DIT outperforms state-of-the-art methods.

CCS Concepts: • **Information systems → Entity resolution**; Information retrieval; • **Computing methodologies → Knowledge representation and reasoning**; Natural language processing;

Additional Key Words and Phrases: Knowledge graph, cross graph attention, entity alignment, iterative

This work was supported by the National Natural Science Foundation of China under grants 61972290 and 61972173, and supported by the National Key R&D Program of China under grant 2018YFC1604000. This research was also supported in part by a research grant from the Natural Sciences and Engineering Research Council (NSERC) of Canada and York Research Chairs (YRC) program. In addition, this research was supported by the National Research Foundation, Singapore under its AI Singapore program.
Authors' addresses: Z. Xie, K. Zhao, and J. Liu (corresponding author), School of Computer Science, Wuhan University, Wuhan, China; emails: {xiezhiwen, kszhao, jinliu}@whu.edu.cn; R. Zhu, Information Retrieval and Knowledge Management Research Lab, Lassonde School of Engineering, York University, Toronto, Canada and AI Singapore, Singapore; email: sherryzh@cse.yorku.ca, sherryzhu@aisingapore.org; G. Zhou (corresponding author), School of Computer Science, Central China Normal University, Wuhan, China; email: gyzhou@mail.ccnu.edu.cn; J. X. Huang, Information Retrieval and Knowledge Management Research Lab, School of Information Technology, York University, Toronto, Canada; email: jhuang@yorku.ca.

1046-8188/2021/11-ART44 $15.00
https://doi.org/10.1145/3471165

ACM Reference format:
Zhiwen Xie, Runjie Zhu, Kunsong Zhao, Jin Liu, Guangyou Zhou, and Jimmy Xiangji Huang. 2021. Dual Gated Graph Attention Networks with Dynamic Iterative Training for Cross-Lingual Entity Alignment. *ACM Trans. Inf. Syst.* 40, 3, Article 44 (November 2021), 30 pages.
https://doi.org/10.1145/3471165

1 INTRODUCTION

The amount of data we produce daily and the amount of information available for internet users today are astounding and ever-expanding. According to Raconteur's statistics,[1] it is estimated that around 463 exabytes of data will be produced globally on a daily basis by 2025.[2] In addition, more than 1.8 billion web pages are currently active and interlinked through the World Wide Web and search engines.[3] Among most of these successful social network sites, Internet of Things (IoT), and search engines, the most significant denominator is the use of **knowledge graphs (KGs)** to structurally achieve effective data storage and information retrieval. Each individual, group, or enterprise organizes and refines information into their own form of knowledge with a confined scope of complexity. However, in a broader perspective, we commonly agree that KG refers to a collection of interlinked entities with their relationships, represented in Resource Description Framework (RDF), where information can be acquired, integrated, and discovered with ontologies. From the mathematical perspective, we define KGs as directed heterogeneous multigraphs whose nodes represent entities and the edges stand for the semantic relationships between the nodes.

In the natural language processing and information retrieval fields of research, KGs have successfully supported many downstream tasks such as language modeling [96], question answering [79], and personalized search and recommendation [13, 43, 75, 78]. So far, those open-sourced large-scale KGs such as Freebase [3], DBpedia [1], WordNet [44], COVID-19 KG,[4] and Google KG [62] have also been widely implemented to many specific real-world applications. For example, according to the publicly available data, Google KGs currently contain more than 500 billion facts in the form of triples to support its Google web search. And COVID-19 KG has supported research in drugs and vaccines, disease detection, virus traceability, virus transmission, epidemic prevention, and so on. Although these KGs have been widely adopted, one of the big challenges is that there are still a lot of missing facts within these existing KGs. Due to the high degree of incompleteness among these KGs, the performance and effectiveness of KGs in real-world scenarios are greatly affected. In some critical applications, such as clinical decision support, the lack of key nodes or information would even lead to serious consequences.

To deal with the issues stated previously, one direct solution is to effectively learn the KG embedding and then to predict the missing links using the embedded representations, known as the task of **knowledge graph embedding (KGE)**. Examples of these KGE models mainly include translational based models of TransE [5] and its various extensions TransH [77], TransD [22], and TransR [37]; complex models of DistMult [91] and ComplEx [69]; multi-layered **convolutional neural network (CNN)** based models of ConvE [11] and ConvKB [47]; **graph neural network (GNN)** based models of A2N [2], R-GCN [58], and KBGAT [45], and some other different approaches, such as RotatE [65], QuatE [94], CapsE [49], and ReInceptionE [85]. Although these models are good at delivering promising experimental results, these methods are still limited to the KGE task on a single KG.

[1] https://www.raconteur.net/infographics/a-day-in-data/.
[2] https://www.weforum.org/agenda/2019/04/how-much-data-is-generated-each-day-cf4bddf29f/.
[3] https://www.internetlivestats.com/total-number-of-websites/.
[4] https://covid-19.aminer.cn/kg?lang=en.

Recently, given the crowd-sourcing strategy and the human involvement in the KG construction, many KGs written in different languages have emerged, such as YAGO [63], DBPedia [31], and BabelNet [46]. Even though these multilingual KGs were constructed in various ways, there are certain connections in between them. If we could exercise effective alignment in between these distributed multilingual KGs, and form an even more vast in scale and higher in-coverage KG, that would be one viable solution to the problems existing among these incomplete single KGs [24]. Meanwhile, it could also be another exciting research direction in general. Moreover, these highly integrated multilingual KGs can significantly benefit many cross-lingual natural language processing and information retrieval tasks. In this article, we mainly take the perspective of KG alignment to solve the incompleteness issue of a single KG.

Multilingual KG alignment aims to match entities with their counterpart in KGs of different languages. In general, conventional models for entity alignment rely heavily on the quality of extracted features and machine translations, which restrict their model performances to be robust on heterogeneous data scenarios and on multilingual entity alignment tasks. For example, for conventional models such as translational-based models and complex models, the expressiveness of these models and the diversity of feature extraction have always been a problem due to their shallow structures unless the embedding size increases. Later on, although the CNN methods have shown better results, they still lack modeling important structural information beyond the entities. This opens up an avenue for the powerful GNN-based approaches to learn entity representations with the entity itself and its neighboring features [6, 41, 68, 76, 89, 92]. However, most of these existing GNN-based methods model each individual KG separately with a small amount of pre-aligned entities served as anchors to connect different KG embedding spaces. This characteristic leads to two major problems. First, those approaches are usually restrained by the insufficiency of seed alignments available from the KGs. Second, these existing methods tend to ignore the useful and pre-aligned links in contextual information in-between nodes. Intuitively, equivalent entities across different KGs should share as many pre-aligned neighbors as possible. For example, in Figure 1, even though the Chinese entity "*哥威迅语(Gothic language)*" and the English entity "*Gothic language*" share the same formality after machine translation, the actual entity alignment pair should be "*哥威迅语(Gothic language)*" and "*Gwich'in language*." With the help of other contextual information and alignments existing in the KGs, we could infer the correct alignment from examples of existed alignment pairs of "*美国(United States)*" and "*United States*," "*加拿大(Canada)*" and "*Canada*," "*纳—德内语系(Na de nene)*" and "*Na-Dene languages*," and "*德内语支(German language branch)*" and "*Athabaskan languages*," and so forth.

Given the problem stated earlier, GM-EHD-JEA [88] proposed an easy-to-hard method that aims to predict model-confident alignment as additional input first, and then to predict the remaining hard alignments. Even though the rationale is good, their model depends on a two-stage reasoning and a joint entity alignment, which requires high computational cost and large search space. Most recently, we proposed a **contextual alignment enhanced cross graph attention network (CAECGAT)** [86] to address the issue of the existing GNN-based approaches stated earlier. In this preliminary study, we design the system to jointly learn embeddings in different KGs by using pre-aligned seeds to propagate the information across KGs. Compared to GM-EHD-JEA, our proposed CAECGAT is more efficient. However, we find that this preliminary study still faces certain challenges and issues in propagating information across KGs due to the lack of pre-aligned entity pairs. Taking the example in Figure 1, if there is no neighborhood alignments surrounding the Chinese entity "*哥威迅语(Gothic language)*" and the English entity "*Gothic language*," the preliminary CAECGAT model is not able to propagate cross-KG information. To tackle the preceding challenges, we propose DuGa-DIT, a dual gated **graph attention network (GAT)** with dynamic iterative training, for cross-lingual entity alignment in this study, which is able to mitigate all of

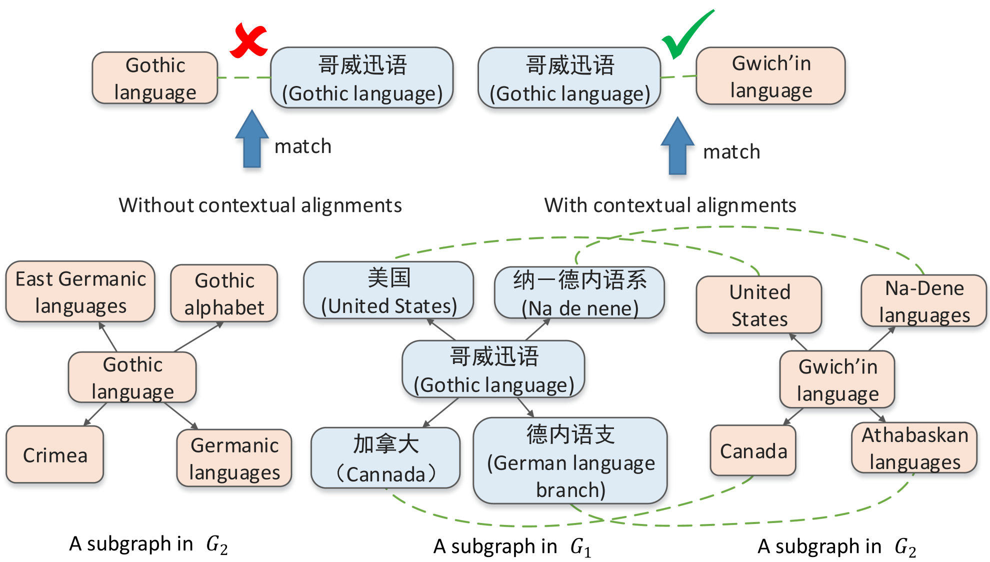


Fig. 1. An example of cross-lingual entity alignment in the DBP15K$_{ZH-EN}$ dataset. The translated surface strings for Chinese entities are provided in the original DBP15K$_{ZH-EN}$ dataset [66]. The dashed lines represent the pre-aligned neighbors.

these problems in a unified framework. Specifically, the dual KG attention layer is composed of one intra-KG attention layer and one cross-KG attention layer. The intra-KG attention layer serves the purpose of learning neighborhood features for each KG, and the cross-KG attention layer serves to collect alignment information across the two KGs. By stacking multiple dual KG attention layers, the local neighborhood and cross-KG alignment information can be propagated to multi-hop neighbors. Our proposed DuGa-DIT model is able to take full use of the pre-aligned entity pairs in both training and prediction procedures because of the cross-KG attention aggregation layer. Furthermore, by applying the dynamic iterative training process, we can dynamically update the cross-KG attention score matrices, which allow more cross-KG information transfer across different KGs. We conduct extensive experiments on two benchmark datasets and a case study in cross-lingual personalized search and recommendation.

The main contributions of this study are thus summarized as follows:

- We propose DuGa-DIT for cross-lingual entity alignment, which can learn cross-KG embeddings to bridge the semantic gap between different KGs by leveraging both neighborhood features and cross-KG alignment information.
- We develop a dynamic iterative training process to dynamically update the cross-KG attention score matrices by iteratively adding new seed alignments, which allow more cross-KG information transfer across different KGs.
- We conduct extensive experiments on two benchmark datasets for cross-lingual entity alignment. The experimental results demonstrate that our proposed method is effective and robust for entity alignment task.

The rest of the article is structured as follows. Section 2 gives a background to our study. Section 3 elaborates on our approach of the proposed DuGa-DIT model. Section 4 describes the datasets and implementation details we applied in this study as well as all experimental results and analyses. Last, Section 5 concludes the article with some suggestions for future work.

2 RELATED WORK

To address the problem of multilingual entity alignment that our proposed method is solving, we hereby discuss mainly two subfields of related work: KGE and multilingual KG alignment. The detailed discussions and analyses of these two groups of existing literature are presented as follows.

2.1 Knowledge Graph Embedding

KGE aims to embed entities and relations of KGs into continuous vector spaces. The purpose of embedding is to inherit the original structure of KG entities and simplify the process of manipulating them in the future. As the number and scale of KGs grow rapidly, KGE becomes more important in tasks of KG analyses and semantic data modeling.

2.1.1 Translation-Based Methods. Translation-based models are considered to be one of the major approaches to the KGE problem. Bordes et al. [5] proposed the most representative translational distance method, named *TransE*. TransE is powerful and promising to predict links, classify triples, and model large-scale KGs on 1 to 1 relations. It is also a pioneer to show the feasibility of KG modeling. The proposed TransE model follows an assumption that $h + r \approx t$, for all triples (h, r, t). Since the structure and the assumption of TransE are simple, the model is not capable of modeling complex relations between entities, such as 1 to N, N to 1, or N to N relations. Since then, many variations of TransE have been introduced [14, 20, 22, 37, 57, 77, 84, 93]. However, it is worth noting that there is a trade-off in between model performance and model complexity among these latest approaches. These new approaches of representations of relational translation achieve more promising results than TransE, although due to the complexity nature of the models, computational costs are comparatively higher. To increase the expressiveness of embeddings from labeled and unlabeled entities from KGs, as well as to address the heterogeneity and imbalance nature of KGs, TranSparse [23] was proposed to deal with the two issues. In TranSparse, conventional transfer matrices are changed to adaptive sparse matrices. The sparse degrees of these matrices are generated by the number of entities linking various relations.

2.1.2 Semantic Matching Methods. Semantic matching methods refer to the group of models that calculate the semantic similarities using vectors or matrices. RESCAL [51] is the first knowledge embedding model that incorporates the three-way tensor factorization. Bilinear [21] is the representatives where relations are modeled as matrices while entities are associated with vectors. DistMult [91] captures the best relational semantics when embeddings are learned from the bilinear objective. Moreover, the composition of relations is characterized by the multiplication of matrices. Similar to DistMult, as RESCAL's extension, ComplEx [69] proposes to solve binary relations problems, symmetric or asymmetric, by using complex valued embeddings. Within this category, there have also been some semantic matching models enhanced by neural networks. SME [4] is first proposed to encode the interactions of entities and relations by using the linear and bilinear matching blocks in neural networks. NTN [8] serves as a foundation of the improved model of Klein et al. [28] with multilingual exposures. KrompaB [30] incorporates type-constraint prior knowledge into various state-of-the-art latent variable approaches to enhance the experimental results of link prediction. KR-EAR [36] distinguishes attributes and relations in a single knowledge base for knowledge representation to model correlations and predictions on entities, relations, and attributes. Other related works include but are not limited to HOLE [50] and ProjE [60].

2.1.3 CNN-Based Methods. Another major approach to KGE is applying multi-layer CNN-based models to generate more expressive embeddings due to its high parameter efficiency. ConvE [11] uses 2D convolution operations to model local interactions between the head entities and their relations. ConvE is able to express semantic information by reshaping the head entities and the respective relations into 2D matrices using multiple non-linear feature learning. ConvKB proposed by Nguyen et al. [47] represents triples in a three-column matrix format and feeds them into the CNN to filter out different feature maps. The feature vector is then generated from consolidating the preceding feature maps to show the representation of triples before multiplications with a weighting vector to generate final scores. As a result, ConvKB is capable of capturing both global relationships and translational features between nodes and relations. InteractE [70] improves the former model performance further by increasing interactions between entities and relations embeddings. Even though these methods show significant improvement in learning more expressive features, they share the drawback of considering each triple independently and thus lose plenty of latent information from the KGs. Aside from these models, RSN [18] and CTEA [90] are also examples of leveraging CNN-based deep neural network models to tackle KGE problems. In addition, the context-aware CNN [53] is proposed to improve the KGE results with neighborhood information being incorporated. Moreover, CapsE [49], CoKE [74], and ReInceptionE [85] are also CNN-based KGE models belonging to this category.

2.1.4 Methods Using Neighborhood Information. Neighborhood information based models learn embeddings by targeting the neighborhood information of an entity. Recently, many studies taking this approach have shown significant performance gains in the KGE task [2, 45, 48, 58, 59, 73]. A2N [2] uses the attention mechanism to get query-specific neighbors. R-GCN [58] is the first model to use GCNs to model multi-relational data in KGs. KBGAT [45] proposed to use GATs [71] to assign different importance for neighbors. TransE-NMM [48] extended TransE by considering neighboring entities. With the gating mechanism, NKGE [73] takes in both semantic and topological features of a specific entity to complete the KGE task. SACN [59] was constructed on top of GCN and ConvE to learn richer neighborhood information in local aggregations, leading to more accurate representations.

2.1.5 Methods Using Relation Paths/Literals. In addition, several studies take relation path based approaches to learn and improve embeddings from KGs. DeepPath [87] takes the approach of reinforcement learning to capture multi-hop relation paths with global information. RTransE [15] and PTransE [35] use multi-step relational paths to complete KGs, whereas SimplE [25] and RotatE [65] follow the same bilinear model path to tackle the issue. Aside from that, there are a number of studies of proposed models with literals to learn KGE. TransEA [83] utilizes a combination of the attribute embedding model and the translation-based embedding model to jointly learn entity relations and numeric attributes simultaneously. LiteralE [29], built on top of the existing latent feature approaches, extends the link prediction task by stacking an extra simple module to directly incorporate literals into embedding learnings with a parameterized function. KBLRN [16] innovatively integrates different feature types of latent, relational, and numerical data by combining neural representation learning with probabilistic product of expert models.

2.1.6 Summary and Comparisons. In summary, the five main categories of KGE models have their own pros and cons. Translation-based models have strong capability of preserving the structural information while learning the embeddings. However, due to the complexity nature of these models, computational costs are relatively high. Semantic-based models enjoy their shallow structures, which are fast and easy to scale up to the entire KGs, but they also sacrifice the capability of capturing more expressive features. Relation path based models are able to exploit long-term dependency of entity relations spanning over all the relational paths; however, their expressiveness of the features is still limited. CNN-based models are introduced as the deep learning based models are generally more powerful to learn expressive features, but most of the studies consider each triplet individually and lose much valuable latent information from KGs. Although the existing models belonging to the neighborhood-based models category have achieved much more promising results on the KGE task, they also suffer disadvantages in mainly two aspects: (1) most of these models only consider the inbound directional neighbors of a specific entity and tend to ignore the information-rich contexts from outbound directional neighbors, and (2) most of these models only apply the k-th hop output to learn the multi-hop embeddings, which suffers from the loss of a great number of early-stage embedding information at the step of graph attention.

2.2 Multilingual Entity Alignment

Aside from KGE, another line of research related to this study is to align entities across various different languages. Traditional approaches to entity alignment are mainly human centric at the cost of time, labor, and flexibility (e.g., [33, 40, 42, 72]). However, as the size of KGs grows more extensively over time, the previously proposed methods have become more inapplicable due to the upscaling.

To overcome the limitations of those traditional approaches, researchers have developed many advanced models by leveraging deep learning algorithms recently. Based on whether the models take in structural information, neighborhood information, or extra information outside the KGs, we can further classify the existing models to three major categories, namely the embedding-based methods (Group A), the neighborhood information-based methods (Group B), and methods using extra information beyond structures (Group C).

2.2.1 Embedding-Based Methods. Embedding-based models learn KG embeddings from the embedding characteristic, which view the relations between entities as translations between the head and the tail entity.

For multilingual entity alignment tasks specifically, JE [19] and MTransE [10] are the earliest classic models belonging to this category. JE [19] only uses the structure of KGs to learn a unified vector space for different KGs. MTransE [10] built on top of TransE is a significant work that separately embeds the entities and relations of the individual language into an embedding space, then maps them into a multilingual counterparts by learning its transitions and minimizing the distance between embeddings of seed alignments in different KGs. To exploit the unlabeled data to a deeper level, IPTransE [98] optimizes the MTransE model by applying iterative methods to add new alignments as training data. Then it maps the entities and relations of different languages with same parameters into the unified vector space.

As labeled data are very limited among the KGs, there has been a trend leveraging semi-supervised learning, co-training, and bootstrapping methods to exploit the unlabeled data more effectively. SEA [54] proposed a semi-supervised entity alignment method (SEA) to bring both the information from labeled entities and unlabeled entities into analyses for the entity alignment task. BootEA [67] uses the improved KGE methods with limit-based loss and a bootstrapping technique to solve the problem of lacking labeled data in entity alignment tasks. Similar to IPTransE [98], it also applies iterative methods to add new alignments as training data.

Other embedding-based methods include AKE [34], which uses an adversarial knowledge embedding to jointly learn the representation, alignment mapping, and adversarial modules; MMEA [61], which uses a new multiplicative approach to jointly model entities and relations from different KGs; JarKA [7], which leverages the attributes to learn entity embeddings; and OTEA [56], which uses an optimal transport-based method to dually optimize entity-level and group-level losses.

2.2.2 Neighborhood Information Based Methods. More recently, our research community proposed more graph-based methods to aggregate neighborhood information to enrich the embeddings for KGs. GM [89] represents the entities and their neighbors as topic entity graphs and views the entity alignment task as a graph matching problem. And they use a graph attention based method to match all entities in two topic entity graphs. KECG [32] adopts GATs [71] with sharing parameters to embed entities in different KGs into a unified vector space and jointly applies TransE [5] to implicitly complete different KGs.

MuGNN [6] uses a rule-based method to complete KGs and uses multiple channel graph attention to reconcile the structural differences between two KGs. AliNet [68] uses gated and attention mechanisms to aggregate multi-hop neighborhood information. In other words, it takes the heterogeneity and differences existing among counterpart entities neighbors into consideration, and expands the overlaps between entities neighborhood structures to an attention-based distant neighbors. The final entity alignment is achieved with a gating mechanism among distant and direct neighbors, along with a relation loss calculation to improve entity representations. HGCN-JE [81] approximates relation embeddings by using entity embeddings learned by GCN [27] and jointly learns better representations for both entities and relations.

MRAEA [41] uses a relation-aware self-attention mechanism to learn relation-aware representations and adopts a bidirectional iterative strategy to iteratively add new alignments for training. NAEA [99] designs an attention mechanism to gather neighborhood information at the neighborhood subgraph level and introduces an NAEA neighborhood-aware attentional representation method to learn the neighborhood representations. RDGCN [80] uses GCN [27] to incorporate relation information by taking advantage of relation interactions between KG and its dual relation counterpart. GM-EHD-JEA [88] uses an easy-to-hard decoding strategy to iteratively predict new alignments as additional inputs and a joint entity alignment algorithm to address the many-to-one problem. NMN [82] applies GCNs and a neighborhood sampling method to capture informative neighborhood features. REA [55] uses a reinforced training approach to deal with the noise labels in the entity alignment task. SSP [52] uses GCNs and a translation-based model to capture both global and local features. HyperKA [64] extends the GCNs from Euclidean space to hyperbolic space to better model hierarchical structures of KGs. AttrGCN [38] uses an attributed GNN to learn attribute triples and relation triples in a unified framework.

CAECGAT [86] is a contextual alignment enhanced cross-GAT method introduced for cross-lingual entity alignment. Our preliminary study designs the system to jointly learn embeddings in different KGs by using pre-aligned seeds to propagate the information across KGs. However, we find that propagating information for multi-hop neighbors and making full use of the pre-aligned entity pairs in both training and prediction stages are very important. The model performance could be further improved if we conduct further investigation. Thus, we plan to extend our CAECGAT study in the following three ways: (1) we propose DuGa-DIT for cross-lingual entity alignment, which can learn cross-KG embeddings to bridge the semantic gap between different KGs by transferring information cross-KGs and can work well with iterative training method by dynamically updating the cross-KG attention score matrices; (2) we dive deeper to run the experiments on more datasets and on various scenarios; and (3) we analyze the related work from both the KGE tasks perspective and the cross-lingual entity alignment perspective with a case study on personalized search and recommendation. Different from the previous neighborhood matching methods (e.g., [82, 89]) that match the one-hop neighbors between two sub-graphs, our DuGa-DIT is able to propagate multi-hop neighborhood information and cross-KG alignment information. There are other related models that belong to this category (e.g., [2, 45, 53, 58, 95]).

2.2.3 Methods Using Extra Information Beyond Structures. Aside from the two approaches discussed earlier, there have also been some methods designed to take in extra information, such as attribute and entity description, beyond structures. JAPE [66] takes advantage of both structural information and attribute information in different language KGs and learns the entity representations in a joint manner. GCN-Align [76] incorporates entity and attribute information and uses GCN [27] to learn neighborhood information for entities and directly learn cross-KG equivalent relationships between entities. HMAN [92] employs multiple aspects including structure, relations, and attributes to learn entity embeddings. They also use literal descriptions of entities and fine-tune pre-trained multilingual BERT [12] to further improve the performance. However, their method requires extra entity descriptions, and the best results reported in their model are obtained by combining HMAN an BERT models. For fair comparison, we only compare with the results of a single HMAN model without combining the BERT model. KDCoE [9] presents the co-training of two parallel embedding models, namely the cross-lingual KGE model and the cross-lingual description embedding model. Aside from that, JarKA [7] takes in attribute information to perform the multilingual entity alignment task.

2.2.4 Summary and Comparison. In general, the three categories of learning algorithms aforementioned have their own advantages and limitations. Embedding-based models are the simplest in structure, but the shallow structures of these models pose certain restrictions on their model expressiveness. Neighborhood information based models generally show good performance on entity alignment task and link the prediction task by aggregating neighborhood information. However, most of these models are designed to capture neighborhood information from fixed graphs, which largely restricts the capability of capturing relation-specific multi-hop neighboring features for different relations. For those methods using extra information beyond structures, the biggest challenge is that the extra information data might not be available all the time. In most of the real-world applications, extra information is either not available or cannot be used. Therefore, no matter how good the models are designed, the applicability of them is largely affected. Table 1 summarizes the major methods of these three groups involved in this study.

Unlike all the methods described earlier, our proposed DuGa-DIT is able to learn cross-KG embeddings and to bridge the semantic gaps between different KGs by leveraging both neighborhood features and cross-KG alignment information. The dynamic iterative training process in our model allows the cross-KG attention score matrices to update dynamically by iteratively adding new seed alignments. This design makes more cross-KG information able to transfer across different KGs effectively. Furthermore, our model purely relies on neighborhood features and cross-KG alignment information to perform, without requiring additional extra information to boost the performance.

3 APPROACH

In this section, we aim to elaborate on our proposed DuGa-DIT model. We first describe the notations we are going to use in the rest of the article along with an overview of the system architecture. Then, we present the two attention layers, namely the intra-KG attention and the cross-KG attention, and the gated feature update that can yield new entity embeddings from the preceding two layers. Last, we also show the optimization, prediction, and dynamic iterative training processes adopted in our method.

3.1 Problem Formulation

In this section, we describe some preliminaries related to our work and give the problem formulation of our work.

Some basic definitions are presented as follows:

- A *knowledge graph* is a data structure comprised of entities, relations, and triples, and is represented as $G = (E, R, T)$, where E stands for a collection of entities, R stands for a collection of relations, and T stands for a collection of triples.

Table 1. Summary of the Multilingual Entity Alignment Models Involved in This Study

Groups	Methods	Embedding Model	Extra Information	Initialization
Group A	JE	TransE	No	Random
	MTransE	TransE	No	Random
	IPTransE	TransE	No	Random
	BootEA	TransE	No	Random
	MMEA	ComplEx	No	Random
	OTEA	TransE	No	Random
Group B	KECG	GNN+TransE	No	Random
	MuGNN	GNN	No	Random
	AliNet	GNN	No	Random
	HyperKA	GNN	No	Random
	NAEA	TransE	No	Random
	SSP	GNN+TransE	No	Random
	MRAEA	GNN	No	Random
	GM	GNN	No	Pre-trained
	RDGCN	GNN	No	Pre-trained
	HGCN-JE	GNN	No	Pre-trained
	NMN	GNN	No	Pre-trained
	GM-EHD-JEA	GNN	No	Pre-trained
	CAECGAT	GNN	No	Pre-trained
	AttrGCN	GNN	No	Pre-trained
Group C	JAPE	TransE	Attribute	Random
	GCN-Align	GNN	Attribute	Random
	JarKA	TransE	Attribute	Random
	HMAN	GNN	Attribute	Random
	KDCOE	TransE	Description	Random

- *Cross-lingual KG entity alignment* refers to the task of two KGs $G_1 = (E_1, R_1, T_1)$ and $G_2 = (E_2, R_2, T_2)$ with different languages, and entity alignment is the task of deriving a set of high precision and high recall pairs of entities $(e_1, e_2) \in (E_1 \times E_2)$ where these two pairs thus can be referred to the same real-world entities.

3.2 Overview

As depicted in Figure 2, the structure of the proposed DuGa-DIT is composed of multiple dual KG attention layers. Each dual KG attention layer consists of a cross-KG attention layer and an intra-KG attention layer. With two different KGs G_1 and G_2 as input, as well as a collection of seed links between two KGs, we use intra-KG attention to collect neighborhood features and use cross-KG attention to gather useful alignment information from another KG. Thus, we can obtain information for both intra-KG structure and cross-KG alignment. Multiple dual KG attention layers can be stacked to aggregate multi-hop features. Furthermore, we use a dynamic iterative training process to dynamically update cross-KG entity alignments, which allows our model to gather more cross-KG alignment features. The main notations of this article are listed in Table 2.

Formally, given two different KGs $G_1 = (E_1, R_1, T_1)$ and $G_2 = (E_2, R_2, T_2)$, and a set of seed alignments $A = \{(e_1, e_2) | e_1 \in E_1, e_2 \in E_2\}$, in this article, we represent the entities in the two KGs as k-dimensional embedding matrices $\mathbf{E}_1$ and $\mathbf{E}_2$. During training, we split all the seed alignments

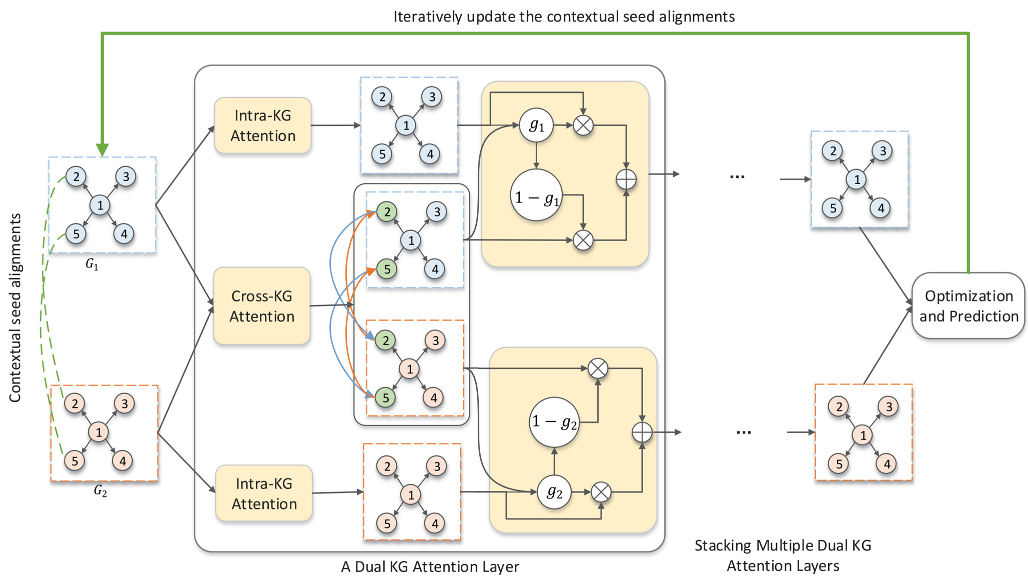


Fig. 2. The architecture of the proposed DuGa-DIT. The dual KG attention layer consists of an intra-KG attention layer and a cross-KG attention layer. The intra-KG attention layer is used to learn neighborhood features for each KG. And the cross-KG attention layer is employed to collect alignment information across two KGs. By stacking multiple dual KG attention layers, the local neighborhood and cross-KG alignment information can be propagated to multi-hop neighbors.

into a set of contextual seed alignments A_{ctx} and a set of objective seed alignments A_{obj}. The contextual seed alignments in A_{ctx} are used as bridge to transfer information across different KGs. Thus, the cross-KG information provided by A_{ctx} can be used as contextual alignment information to predict the matching scores for the entity pairs in objective seed alignments in A_{obj}.

3.3 Intra-KG Attention

Recently, GNNs have shown promising potential in the cross-lingual KG entity alignment task due to their ability of learning local structure in KG data. In this study, we use an attention-based graph attention mechanism to learn the features from each monolingual KG, which is called the *intra-KG attention layer.*

Given two KGs G_1 and G_2, let $\mathbf{M}_1^l$ and $\mathbf{M}_2^l$ denote the normalized attention matrices for G_1 and G_2. The intra-KG attention layer is used to update entity embeddings for each KG by aggregating important neighborhood features, which can be denoted as follows:

$$\begin{aligned} \mathbf{H}_1^{l+1} &= \sigma\left(\mathbf{M}_1^l \mathbf{E}_1^l \mathbf{W}^l\right) \\ \mathbf{H}_2^{l+1} &= \sigma\left(\mathbf{M}_2^l \mathbf{E}_2^l \mathbf{W}^l\right), \end{aligned} \tag{1}$$

where $\mathbf{W}^l$ is the transformation parameters and σ is the activation function (e.g., Relu [17]), and l denotes the l-th layer.

The normalized attention matrices are computed by using GATs, which first compute the score for each neighbor entity and then normalize the scores for all neighbors. Specifically, each element

Table 2. Notation List

Notation	Description
G_1, G_2	The KGs in different languages.
E_1, E_2	The entities in G_1 and G_2.
R_1, R_2	The relations in G_1 and G_2.
T_1, T_2	The triples in G_1 and G_2.
A	The seed alignments.
A_{ctx}, A_{obj}	The contextual and objective seed alignments.
$A_{ctx}^{new}, A_{obj}^{new}$	The newly added contextual and objective seed alignments.
$\mathbf{E}_1, \mathbf{E}_2$	The initial entity embeddings for G_1 and G_2.
$\mathbf{E}_1^L, \mathbf{E}_2^L$	The final entity embeddings for G_1 and G_2.
L	The number of dual KG attention layers.
$\mathbf{e}_1, \mathbf{e}_2$	The initial entity vectors for $e_1 \in G_1$ and $e_2 \in G_2$.
$\bar{\mathbf{e}}_1, \bar{\mathbf{e}}_2$	The final entity vectors for $e_1 \in G_1$ and $e_2 \in G_2$.
$\mathcal{N}_{e_i}$	A set of neighbors for entity e_i.
$\mathbf{M}_1, \mathbf{M}_2$	The intra-KG attention score matrices for G_1 and G_2.
$\mathbf{M}_{12}$	The cross-KG attention score matrices from G_1 to G_2.
$\mathbf{M}_{21}$	The cross-KG attention score matrices from G_2 to G_1.
$\mathbf{H}_1^l, \mathbf{H}_2^l$	The neighborhood features for G_1 and G_2 obtained by the l-th intra-KG attention layer.
$\mathbf{C}_1^l, \mathbf{C}_2^l$	The cross-KG alignment features for G_1 and G_2 in the l-th cross-KG attention layer.
$\mathcal{L}(\phi; A_{obj})$	The loss function of the proposed DuGa-DIT model.
ϕ	The trainable parameters in the model.
$D(\bar{\mathbf{e}}_1, \bar{\mathbf{e}}_2)$	The L_1 distance between entity vectors $\bar{\mathbf{e}}_1$ and $\bar{\mathbf{e}}_2$.

in $\mathbf{M}_1^l$ is computed as follows:

$$\mathbf{M}_1^l[i,j] = \frac{\exp(s_{i,j}^l)}{\sum_{j'} \exp(s_{i,j'}^l)}, \tag{2}$$

where $s_{i,j}^l$ is the attention score between the i-th and j-th entities, which is computed as

$$s_{i,j}^l = \begin{cases} \text{LeakyRelu}\left(\mathbf{v}^T \left[\mathbf{e}_i^l || \mathbf{e}_j^l\right]\right) & e_j \in \mathcal{N}_{e_i} \\ -\infty & otherwise \end{cases}, \tag{3}$$

where LeakyRelu [39] is an activation function that is widely used in the graph attention layer, $\mathbf{v}$ is the attention vector, $\mathcal{N}_{e_i}$ is a set of neighbors for entity e_i, and $\mathbf{e}_i$ and $\mathbf{e}_j$ are the vectors for entities e_i and e_j, respectively. The normalized attention matrix $\mathbf{M}_2^l$ for G_2 can be computed in a similar manner.

3.4 Cross-KG Attention

The intra-KG attention layer can only capture local structure features from mono-lingual KG, which is not sufficient for the entity alignment task. In this section, we present a novel cross-KG attention layer aiming to aggregate cross-KG alignment information by applying graph attention across two KGs. In this study, we take full advantage of the pre-aligned seed alignments and view them as edges linking between two KGs.

Formally, given the contextual alignments A_{ctx}, we can construct cross attention matrices from G_1 to G_2 and from G_2 to G_1 using the cross-KG attention layer. Let $\mathbf{M}_{12}$ denote the attention matrix from G_1 to G_2. Each element in $\mathbf{M}_{12}$ is obtained as follows:

$$\mathbf{M}_{12}^{l}[i,j] = \frac{\exp(c_{i,j}^{l})}{\sum_{j'} \exp(c_{i,j'}^{l}) + \delta}, \tag{4}$$

where $\delta = 1e^{-8}$ is a smoothing factor and $c_{i,j}^{l}$ is the cross-KG attention score between the i-th entity in G_1 and the j-th entity in G_2, which is calculated as follows:

$$c_{i,j}^{l} = \begin{cases} \text{LeakyRelu}\left(\mathbf{v}^{T}\left[\mathbf{e}_{1i}^{l}||\mathbf{e}_{2j}^{l}\right]\right) & (e_{1i}, e_{2j}) \in A_{ctx} \\ -\infty & otherwise \end{cases}, \tag{5}$$

where e_{1i} is the i-th entity in G_1 and e_{2j} is the j-th entity in G_2, and $\mathbf{e}_{1i}^{l}$ and $\mathbf{e}_{2j}^{l}$ are the vector representations of e_{1i} and e_{2j} in the l-th layer, respectively.

Thus, we can collect cross-KG alignment features for G_1 by applying graph attention on the attention score matrix $\mathbf{M}_{12}$, which is denoted as follows:

$$\mathbf{C}_{1}^{l+1} = \sigma\left(\mathbf{M}_{12}^{l}\mathbf{E}_{2}^{l}\mathbf{W}_{c}^{l}\right), \tag{6}$$

where $\mathbf{W}_{c}^{l}$ represents the linear transformation parameters in the cross-KG attention layer.

Similarly, the cross-KG alignment features for G_2 is computed as follows:

$$\mathbf{C}_{2}^{l+1} = \sigma\left(\mathbf{M}_{21}^{l}\mathbf{E}_{1}^{l}\mathbf{W}_{c}^{l}\right), \tag{7}$$

where $\mathbf{M}_{21}$ is the attention score matrix from KG G_2 to G_1, which is obtained in a similar way with $\mathbf{M}_{12}$.

By applying the cross-KG attention, we can enable the equivalent entities in two KGs to share some common cross-KG features. And these cross-KG features can be propagated in the two KGs, which will benefit the entity alignment task.

3.5 Gated Feature Update

Intuitively, the neighborhood features are resourceful information to model the structure of a single KG locally, whereas the cross-KG features are resourceful information to transfer alignment features between two KGs globally. In this way, the semantic difference can be largely alleviated. Therefore, both the neighborhood and the cross-KG alignment features are critical for the task of entity alignment. In our model, we incorporate these two kinds of features into a gated feature update mechanism to update the entity embeddings effectively. Given the preceding discussion, the update rule of our DuGa-DIT model can be formulated as follows:

$$\begin{aligned} \mathbf{E}_{1}^{l+1} &= \mathbf{H}_{1}^{l+1} \cdot g_1 + \mathbf{C}_{1}^{l+1} \cdot (1 - g_1) \\ \mathbf{E}_{2}^{l+1} &= \mathbf{H}_{2}^{l+1} \cdot g_2 + \mathbf{C}_{2}^{l+1} \cdot (1 - g_2), \end{aligned} \tag{8}$$

where $\cdot$ is the element-wise multiplication, and g_1 and g_2 are the gate mechanisms for G_1 and G_2, respectively, which are obtained by concatenating the neighborhood features and cross-KG alignment features, and followed by a non-linear transformation:

$$\begin{aligned} g_1 &= \theta(\mathbf{W}_g\left[\mathbf{H}_{1}^{l+1}||\mathbf{C}_{1}^{l+1}\right] + \mathbf{b}_g) \\ g_2 &= \theta(\mathbf{W}_g\left[\mathbf{H}_{2}^{l+1}||\mathbf{C}_{2}^{l+1}\right] + \mathbf{b}_g), \end{aligned} \tag{9}$$

where θ denotes the sigmoid function, and $\mathbf{W}_g$ and $\mathbf{b}_g$ are trainable parameters.

In our study, we stack L dual KG attention layers to propagate both neighborhood features and cross-alignment features. Finally, our model can yield new entity embeddings $\mathbf{E}_{1}^{L}$ and $\mathbf{E}_{2}^{L}$.

3.6 Optimization and Prediction

The cross-lingual entity alignment task aims to learn close representations for equivalent entities in KGs with different languages. Different from those existing methods, we use the contextual seed alignments as input and use the objective seed alignments to optimize the parameters. The loss function in this study is defined as follows:

$$\begin{aligned}\mathcal{L}(\phi; A_{obj}) = &\sum_{(e_1,e_2)\in A_{obj}} [D(\bar{\mathbf{e}}_1, \bar{\mathbf{e}}_2) + \lambda - D(\bar{\mathbf{e}}_1, \bar{\mathbf{e}}_2^-)]_+ \\ &+ \sum_{(e_1,e_2)\in A_{obj}} [D(\bar{\mathbf{e}}_1, \bar{\mathbf{e}}_2) + \lambda - D(\bar{\mathbf{e}}_1^-, \bar{\mathbf{e}}_2)]_+\end{aligned}, \tag{10}$$

where $[x]_+ = max\{0, x\}$, ϕ denotes the parameters in the DuGa-DIT model, (e_1, e_2^-) and (e_1^-, e_2) are the negative entity alignment pairs obtained by randomly replacing the positive entity e_1 or e_2, $\bar{\mathbf{e}}_1 \in \mathbf{E}_1^L$ and $\bar{\mathbf{e}}_2 \in \mathbf{E}_2^L$ are the vectors of e_1 and e_2, $D(\bar{\mathbf{e}}_1, \bar{\mathbf{e}}_2) = ||\bar{\mathbf{e}}_1 - \bar{\mathbf{e}}_2||_1$ denotes the L_1 distance function, and $\lambda > 0$ is the margin hyper-parameter.

3.7 Dynamic Iterative Training

In practice, the seed alignments are always insufficient, making it difficult to train effective alignment models. And the lack of seed alignments makes the cross-KG attention score matrix $\mathbf{M}_{12}$ and $\mathbf{M}_{21}$ sparse, which cannot capture enough cross-KG alignment features. Inspired by existing methods [41, 67, 88, 98] that iteratively add new alignment predicted by the trained model, we propose to use an iterative method to dynamically update the contextual seed alignments and objective seed alignments. We adopt the bidirectional iterative strategy [41] to select newly aligned entity pairs, namely if and only if e_i and e_j are mutual nearest to each other, the entity pair (e_i, e_j) will be considered as a new alignment. Note that a significant difference with previous works is that our model uses the newly added entity alignment to dynamically update both the cross-KG attention score matrices and objective function.

Formally, let A_{ctx}^{new} denote the newly added contextual seed alignments, then each attention score $c_{i,j}^l$ for cross-KG attention score matrix $\mathbf{M}_{12}^l$ can be updated as follows:

$$c_{i,j}^l = \begin{cases}\text{LeakyRelu}\left(\mathbf{v}^T\left[\mathbf{e}_{1i}^l || \mathbf{e}_{2j}^l\right]\right) & (e_{1i}, e_{2j}) \in A_{ctx} \bigcup A_{ctx}^{new} \\ -\infty & otherwise\end{cases}. \tag{11}$$

The cross-KG attention matrix $\mathbf{M}_{21}^l$ can be updated in a similar manner. Thus, our model can learn richer cross-KG alignment information. Furthermore, we add new objective seed alignments A_{obj}^{new} to update the objective function as $\mathcal{L}(\phi, A_{obj} \bigcup A_{obj}^{new})$.

To better understand the proposed DuGa-DIT model, we describe the details of DuGa-DIT model in Algorithm 1. At the first iteration, we set A^{new} as an empty set, which is shown in line 1. Note that we dynamically split the seed alignments into A_{ctx} and A_{obj} during training, which enables us to make full use of the seed alignments to train the model, as shown in lines 5 and 6. Then, we use L dual KG attention layers to update the entity embeddings, which is described from line 7 to line 12. And in lines 13 and 14, we use A_{obj} and A_{obj}^{new} to compute the objective function and optimize the parameters of the DuGa-DIT model. After training the model, we predict new seed alignment to update A^{new} and use the new A^{new} for the next iteration.

During prediction, we set $A_{ctx} = A \bigcup A^{new}$, which is shown in line 18. For a given test entity e_1 (or e_2) in from KG G_1 (or G_2), we rank all entities in another KG G_2 (or G_1) according to the L_1 distance computed by using the entity vectors $\mathbf{E}_1^L$ and $\mathbf{E}_2^L$ by the DuGa-DIT model.

ALGORITHM 1: DuGa-DIT Model

Input: Given two KGs G_1 and G_2, the initial entity embeddings $\mathbf{E}_1$, $\mathbf{E}_2$, and pre-aligned seed alignments A.
Output: The final entity embeddings $\mathbf{E}_1^L$ and $\mathbf{E}_2^L$.

Initialize A^{new} as an empty set.
repeat
  Let $\mathbf{E}_1^0 = \mathbf{E}_1$ and $\mathbf{E}_2^0 = \mathbf{E}_2$.
  repeat
    Randomly choose some seed alignments from A as A_{obj} and the rest as A_{ctx}.
    Randomly choose some seed alignments from A^{new} as A_{obj}^{new} and the rest as A_{ctx}^{new}.
    for $l = 1$ to L **do**
      Perform intra-KG attention and cross-KG attention and obtain H_1^l, H_2^l, C_1^l, and C_2^l
      Update entity embeddings:
      $\mathbf{E}_1^l = \mathbf{H}_1^l \cdot g_1 + \mathbf{C}_1^l \cdot (1 - g_1)$;
      $\mathbf{E}_2^l = \mathbf{H}_2^l \cdot g_2 + \mathbf{C}_2^l \cdot (1 - g_2)$;
    end for
    Compute loss function $\mathcal{L}(\phi; A_{obj} \bigcup A_{obj}^{new})$.
    Optimize parameters ϕ in the DuGa-DIT model.
  until reaching the maximum number of training epochs
  Predict new seed alignments and update A^{new}.
until reaching the maximum number of iterations
Set $A_{ctx} = A \bigcup A^{new}$.
Use the learned DuGa-DIT model to predict the final entity embeddings $\mathbf{E}_1^L$ and $\mathbf{E}_2^L$.

4 EXPERIMENTS

In this section, we describe the datasets used in this study and conduct extensive experiments on the cross-lingual entity alignment task. In the following sections, we show the detailed experimental results.

4.1 Datasets

In our experiments, we use two benchmark datasets to evaluate our proposed DuGa-DIT model: DBP15K [66] and WK31-60K [9, 10]. Both DBP15K and WK31-60K contain abundant node attributes and edges of different relations. We present the detailed statistics of the selected datasets in Table 3 and introduce the two datasets in depth as follows:

- The DBP15K dataset contains four KGs with different languages (e.g., English, Chinese, Japanese, and French), which are extracted from DBpedia [31]. Three cross-lingual subsets are constructed based on these KGs, namely DBP15K$_{ZH-EN}$ (Chinese-English), DBP15K$_{JA-EN}$ (Japanese-English), and DBP15K$_{FR-EN}$ (French-English). There are 15,000 inter-lingual links connecting between two KGs.
- The WK31-60K dataset contains three KGs from DBpedia (e.g., English, French, and German). Each KG contains about 60K entities. Two cross-lingual entity alignment subsets are constructed upon WK31-60K, namely WK-60K$_{FR-EN}$ (French-English) and WK-10k$_{FR-DE}$ (German-English).

4.2 Implementation Details

4.2.1 Hyper-Parameters. For fair comparison, we apply the same data split as various previous works [66, 92], namely 70% for testing and 30% for training. We further sample 10% of the

Table 3. Statistics of the Datasets

Dataset		Entities	Relations	Rel. Triples	Training/Dev/Test
DBP15K$_{ZH-EN}$	Chinese	66,469	2,830	153,929	4,050/450/10,500
	English	98,125	2,317	237,674	
DBP15K$_{JA-EN}$	Japanese	65,744	2,043	164,373	4,050/450/10,500
	English	95,680	2,096	233,319	
DBP15K$_{FR-EN}$	French	66,858	1,379	192,191	4,050/450/10,500
	English	105,889	2,209	278,590	
WK31-60K$_{FR-EN}$	French	45,255	277	258,337	14,095/1,566/36,544
	English	64,539	458	569,393	
WK31-60K$_{FR-DE}$	German	43,503	172	244,647	14,451/1605/37,467
	English	64,539	458	569,393	

Table 4. Hyper-Parameter Setting

Datasets	λ	L	#neg	dropout	lr	batch_size	#iter
DBP15K$_{ZH-EN}$	3	2	30	0.2	0.002	2,000	2
DBP15K$_{JA-EN}$	3	2	30	0.2	0.002	2,000	2
DBP15K$_{FR-EN}$	3	2	30	0.2	0.002	2,000	2
WK31-60K$_{FR-EN}$	3	2	50	0.2	0.004	3,000	2
WK31-60K$_{DE-EN}$	3	2	50	0.2	0.004	3,000	2

"#neg" denotes the number of negative entity pairs, "lr" denotes the learning rate, and "#iter" denotes the number of dynamic iterative training processes.

training data as the development set to select proper hyper-parameters and use the remaining 90% for training. The statistic of the data splitting is shown in Table 3. We apply Adam [26] to optimize the parameters in the model, and we train the model up to 3,000 epochs. The hyper-parameters are selected by using grid search according to the Hits@1 on the development set. We select the number of negative entity pairs from $\{10, 20, 30, 40, 50\}$, the margin parameter λ in Equation (10) from $\{1, 3, 5\}$, the number of dual KG attention layers from $\{1, 2, 3\}$, the dropout rate from $\{0.1, 0.2, 0.3\}$, and the batch size from $\{500, 1000, 1500, 2000, 3000\}$ and the learning rate from $\{0.004, 0.002, 0.001, 0005\}$. Finally, we randomly sample 30 negative alignment entity pairs for DBP15K and 50 negative entity pairs for WK31-60K. The margin parameter in Equation (10) is set to $\lambda = 3$. We stack two dual KG attention layers to propagate multi-hop cross-KG information. The learning rate is set to 0.002 for DBP15K and 0.004 for WK31-60K. The dropout rate is set to 0.2. The batch size is set to 2,000 for DBP15K and 3,000 for WK31-60K. The maximum iteration number of the dynamic iterative training process is 2. The best hyper-parameters of the DuGa-DIT model are listed in Table 4.

4.2.2 Initialization. There are two commonly used methods to initialize the EA models: random initialization and the initialization with pre-trained word vectors of the entity name. The random initialization method randomly initializes the entity embedding without considering the entity name. However, the surface form of the entity name also provides useful information for the cross-lingual entity alignment task, especially for the entity with few neighbors in KG. Therefore, many methods consider the entity name to initialize the entity embedding, which translates the entity name into English form and then uses the sum of word vectors of the entities' surface name as the entity embedding [80–82, 89]. In this work, we use the preceding two ways to initialize the entity embedding. If the words in the entity are out-of-vocabulary in the pre-trained embedding, we initialize them using random vectors.

4.2.3 Evaluation. We report the widely used standard metric Hits@N (H@N) and MRR to measure the performance of our model, where Hits@N denotes the correct alignment proportion ranked in the top N list and MRR denotes the mean reciprocal rank of the entity alignment. Higher Hits@N and MRR mean better performance. To compare with previous studies, we report Hits@1 (H@1), Hits@10 (H@10), and MRR on DBP15K, and report one more metric, Hits@5 (H@5), on WK31-60K.

4.2.4 Implementation. All experiments reported in this study are conducted on NVIDIA GTX 1080Ti GPUs, and the codes are implemented using TensorFlow.[5] To facilitate the reproduction of the results in this article, the datasets and source codes will be available at https://github.com/JuneTse/EntAlignment.

4.3 Comparison Models

To investigate the power of the our proposed DuGa-DIT model, we compare the proposed method with a large amount of baselines described in the literature. According to the initialization methods, we separate results by baselines with and without pre-trained word vectors into two completely different evaluation settings. These related methods for multilingual entity alignment can roughly be divided into three categories: embedding-based methods (Group A), neighborhood information based methods (Group B), and methods using extra information beyond structures (Group C).

4.3.1 Embedding-Based Methods (Group A). The basic idea of embedding-based methods for entity alignment is to learn low-dimensional vectors for entities in different KGs and match entities by computing similarities between entity vectors. We compare our method with the existing embedding-based methods listed as follows: JE [19], MTransE [10], MMEA [61], and OTEA [56]. For methods in this category but leveraging bootstrapping methods to exploit the unlabeled data, these methods include BootEA [67] and IPTransE [98]. Other embedding-based methods for comparison include MMEA [61] and OTEA [56].

4.3.2 Neighborhood Information Based Methods (Group B). Recently, approaches using neighborhood information have achieved powerful feature learning ability for graph data and have demonstrated the superiority over embedding-based methods. Therefore, we compare our DuGa-DIT method with various related neighborhood information based baselines: KECG [32], MuGNN [6], AliNet [68], HyperKA [64], NAEA [99], SSP [52], GM [89], GM-EHD-JEA [88], RDGCN [80], HGCN-JE [81], NMN [82], MRAEA [41], AttrGCN [38], and CAECGAT [86].

4.3.3 Methods Using Extra Information Beyond Structures (Group C). Aside from the two approaches discussed earlier, we have also compared our model performance to those methods taking in extra information beyond structures: JAPE [66], GCN-Align [76], JarKA [7], and HMAN [92] and KDCoE [9].

We include the aforementioned models for comparison, except for AKE [34] and REA [55], since both models use different datasets and experimental settings.

4.4 Main Results

The cross-lingual entity alignment results on DBP15K are summarized in Table 5. We compare our method with various strong baselines. We analyze the methods within each group and compare them with our proposed DuGa-DIT model.

[5] https://tensorflow.org.

Table 5. Entity Alignment Results on the DBP15K Dataset

		DBP15K$_{ZH-EN}$			DBP15K$_{JA-EN}$			DBP15K$_{FR-EN}$		
Groups	Methods	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
Group A	JE	21.27	42.77	–	18.92	39.97	–	15.38	38.84	–
	MTransE	30.83	61.41	0.364	27.86	57.45	0.349	24.41	55.55	0.335
	IPTransE	40.60	73.50	0.516	36.70	69.30	0.474	33.30	68.50	0.451
	BootEA	62.94	84.75	0.703	62.23	85.39	0.701	65.30	87.44	0.731
	MMEA	68.07	86.74	–	65.53	85.90	–	67.70	89.01	–
Group B	KECG	47.77	83.50	0.598	48.97	84.40	0.610	48.64	85.06	0.610
	MuGNN	49.40	84.40	0.611	50.10	85.70	0.621	49.50	87.00	0.621
	AliNet	53.90	82.60	0.628	54.90	83.10	0.654	55.20	85.20	0.657
	HyperKA	57.20	86.50	0.678	56.40	86.50	0.673	59.70	89.10	0.704
	NAEA	65.01	86.73	0.720	64.14	87.27	0.718	67.32	89.43	0.752
	SSP	73.90	92.50	0.808	72.10	93.50	0.800	73.90	94.70	0.818
	MRAEA	75.70	<u>92.98</u>	0.827	75.78	93.38	0.826	78.04	94.81	0.849
	DuGa-DIT (rand)	75.62	85.78	0.805	78.11	91.36	0.829	83.73	91.97	0.867
	GM‡	67.93	78.48	–	73.97	87.15	–	89.38	95.24	–
	GM-EHD-JEA‡	73.58	–	–	79.15	–	–	92.43	–	–
	RDGCN‡	70.75	84.55	–	76.74	89.54	–	88.64	95.72	–
	HGCN-JE‡	72.03	85.7	–	76.62	89.73	–	89.16	<u>96.11</u>	–
	NMN‡	73.30	86.90	–	78.50	91.20	–	90.20	96.70	–
	CAECGAT‡	75.55	**93.38**	0.818	<u>83.58</u>	**95.59**	<u>0.881</u>	<u>94.68</u>	99.18	<u>0.965</u>
	AttrGCN‡	<u>79.60</u>	92.93	**0.845**	78.33	92.08	0.834	91.85	97.77	0.910
	DuGa-DIT ‡	**80.73**	88.17	<u>0.832</u>	**91.44**	<u>95.21</u>	**0.928**	**98.17**	**99.22**	**0.985**
Group C	JAPE	41.18	74.46	0.490	36.25	68.5	0.476	32.39	66.68	0.430
	GCN-Align	41.25	74.38	0.549	39.91	74.46	0.546	37.29	74.49	0.532
	HMAN	56.20	85.10	–	56.70	86.90	–	54.00	87.10	–
	JarKA	70.58	87.81	0.766	64.58	85.50	0.708	70.41	88.81	0.768

The results of the baselines are directly taken from the original papers. ‡ denotes the models using pre-trained word vectors as the initialization method. DuGa-DIT (rand) is the model using the random initialization. The best results are in **bold**, and the second best results are in <u>underline</u>.

Group A. As shown in Table 5 (Group A), among the various embedding-based methods, MMEA achieves the best results on H@1 and H@10 metrics. Compare to other embedding-based methods, MMEA defines a multiplication-based scoring function to deal with the 1-N, N-1, and N-N relations. This enables it to perform better than other embedding-based methods. Both IPTransE and BootEA models also obtain good results by using the bootstrapping strategy to iteratively add all possible entity pairs to train.

Group B. From the results (Group B) reported in Table 5, We evaluate these models using two completely different settings (random initialization vs. pre-trained word vectors as initialization). From the results (Group B) reported in Table 5, we can see that those GNN-based models achieve the promising results (SSP, MRAEA, GM-EHD-JEA, HGCN-JE, CAECGAT, etc.). The reason lies in that GNN-based models can effectively leverage neighborhood features. We also note that using bootstrapping strategies to iteratively train the model can further improve the performance (e.g., CAECGAT vs. DuGa-DIT). Furthermore, we find that our proposed DuGa-DIT (rand) obtains the comparable performance compared to the best reported MRAEA under the random initialization setting. When using the pre-trained word vector initialization, our DuGa-DIT obtains the best results (H@1, H@10, and MRR) on DBP15K$_{FR-EN}$, H@1 on DBP15K$_{ZH-EN}$, and H@1 and MRR on DBP15K$_{JA-EN}$. Regarding the comparison between these two settings, we observe that most models using the pre-trained initialization have better performance than the models using the random initialization.

Group C. The methods in Group C enhance the performance of entity alignment by leveraging extra information, such as attribute and entity description. Among these methods listed in Group C, JAPE, JarKA, and KDCoE use embedding-based methods to encode the KG structure, whereas others use GNN-based methods. Surprisingly, we find that all of these methods in this category cannot obtain very large performance gains. The reason is that such extra information may be very noisy, which can hurt the performance to some extent.

Comparison across groups. As shown in Table 5, we have three interesting findings. First, neighborhood information based methods, which are able to leverage rich neighboring features, perform better than the simple embedding-based method (e.g., Group B vs. Group A). Second, the surface names of entities play a more important role than the extra attribute information (e.g., Group B vs. Group C). Third, the models with an iterative method obtain large performance gains than those without using an iterative method (e.g., BootEA vs. MTransE in Group A and GM-EHD-JEA vs. GM in Group B).

Analysis of our DuGa-DIT model. Since we use GNN-based methods to aggregate neighborhood features, our proposed DUGa-DIT model can be classified into Group B. For fair comparison, we also provide a version of our DuGa-DIT model with the random initialization method denoted as DuGa-DIT (rand). As shown in Table 5, compared to other state-of-the-art methods using random initialization, such as SSP and MRAEA, DuGa-DIT (rand) obtains comparative results on these datasets. Furthermore, by using the pre-trained word vectors as initialization, the proposed DuGa-DIT model performs much better than previous state-of-the-art methods on all the three cross-lingual entity alignment datasets except the H@10 and MRR on DBP15K$_{ZH-EN}$ and the H@10 on DBP15K$_{JA-EN}$. For example, the proposed DuGa-DIT outperforms the CAECGAT by a margin of 7.68% on H@1 for DBP15K$_{JA-EN}$ and an improvement of 3.49% on H@1 for DBP15K$_{FR-EN}$. We also conduct the statistical significant using the t-test, and the results show that DuGa-DIT significantly outperforms CAECGAT and DuGa-DIT (rand) under the evaluation metric H@1 with confidence of $p < 0.5$.

Compared to existing methods, our DuGa-DIT model not only takes advantage of neighboring features from intra-KG but also makes full use of the cross-KG alignment information using cross-KG attention. Furthermore, by utilizing the dynamic iterative training process, we can dynamically update the cross-KG attention score matrices in our model, which can provide more cross-KG alignment information. These advantages enable our model to learn better entity representations for the entity alignment task.

Impact of the initialization. Various studies in the literature have verified that the surface forms of the entity name can provide useful information for entity alignment [80–82, 86, 89]. Therefore, many state-of-the-art methods leverage the word vectors of entities' surface forms to initialize the entity embedding, including GM, GM-EHD-JEA, RDGCN, HGCN-JE, NMN, AttrGCN, and CAECGAT. Following these previous methods, we also initialize the entity embedding using pre-trained word vectors of entity names (e.g., DuGa-DIT). To investigate the impact of different initialization methods, we report the results using a random initialization method (e.g., DuGa-DIT (rand)). As shown in Table 5, the models using entity names achieve better results than most of the methods without considering the entity name (e.g., HGCN-JE vs. AliNet, DuGA-DIT vs. DuGa-DIT (rand)).

Table 6. Entity Alignment Results on WK31-60K Dataset

		WK31-60K$_{FR-EN}$				WK31-60K$_{DE-EN}$			
Groups	Methods	H@1	H@5	H@10	MRR	H@1	H@5	H@10	MRR
Group A	MTransE	13.95	20.25	–	0.177	3.37	10.07	–	0.072
	JAPE	16.85	35.41	–	0.271	14.71	23.86	–	0.192
	BoostEA	33.31	51.14	–	0.425	23.28	39.29	–	0.316
	OTEA	36.07	54.08	–	0.447	26.97	43.97	–	0.352
Group B	GCN-Align	21.47	37.81	–	0.293	13.8	24.55	–	0.190
	MRAEA	54.34	76.03	–	0.646	42.76	62.77	–	0.524
	SSP	62.67	76.64	81.11	0.693	60.55	77.01	80.46	0.684
	DuGa-DIT (rand)	66.12	78.32	80.84	0.718	59.67	75.12	77.64	0.669
	RDGCN‡	77.91	91.72	93.31	0.843	70.87	85.37	86.74	0.776
	HGCN-JE‡	76.59	90.26	93.85	0.834	71.91	85.67	86.71	0.781
	NMN‡	77.63	91.25	92.66	0.841	70.91	85.59	86.75	0.778
	CAECGAT‡	78.24	92.23	95.23	0.852	71.41	**87.20**	**89.26**	0.788
	DuGa-DIT‡	**84.69**	**95.06**	**95.47**	**0.897**	**73.67**	85.99	86.84	**0.796**
Group C	KDCoE	48.32	–	56.95	0.496	33.52	–	45.47	0.349

‡ denotes the models using pre-trained word vectors as initialization methods. DuGa-DIT (rand) is the model using random initialization.

Since the equivalent entities in a different language always share some common words in their translated surface forms, these surface forms can provide useful information to align some easy entity pairs. Compared to these methods that also use entity names, our DuGa-DIT achieves better results (CAECGAT vs. DuGa-DIT and AttrGCN vs. DuGa-DIT regarding the evaluation H@1).

We also conduct the experiments on WK31-60K, which is larger and more challenging than DBP15K, as shown in Table 6. Since RDGCN, HGCN-JE, NMN, SSP, and CAECGAT in the original works do not provide the results on WK31-60K, we report the results of these methods by running the source codes. From Table 6, we can see that the proposed DuGa-DIT model also achieves better performance on WK31-60K compared to most existing baselines. In particular, compared to the baseline CAECGAT model, our DuGa-DIT model obtains an improvement of 6.45% on H@1 for WK31-60K$_{FR-EN}$ dataset and an improvement of 2.26% on H@1 for WK31-60K$_{DE-EN}$. The improvements of our DuGa-DIT over the best reported model CAECGAT are statistically significant ($p < 0.5$ using the t-test). These experimental results reconfirm that our DuGa-DIT model is effective and robust for the entity alignment task. In addition, compared to the random initialization, we also find that using the pre-trained entity embedding can further improve the performance on WK31-60K. In the rest of this article, we mainly report the results of our DuGa-DIT using the pre-trained entity embedding.

4.5 Efficiency Analysis

To evaluate the efficiency of our model, we compare the performance and prediction time with some representative baseline models on both datasets, DBP15K$_{FR-EN}$ and WK31-60K$_{FR-EN}$. The results are illustrated in Table 7. It is noted that difference parameter settings and the running environment might influence the time costs. However, we aim to try the best to provide a general picture of some representative methods by adopting the settings reported in their original works. From Table 7, we can see that GM-EHD-JEA is more time consuming than other models on DBP15K$_{FR-EN}$. The reason lies in that GM-EHD-JEA depends on a joint entity alignment method for prediction, which is time consuming. On the contrary, our GuGa-DIT is much more efficient

Table 7. Performance and Prediction Time (in Seconds) for Different Models on DBP15K$_{FR-EN}$ and WK31-60K$_{FR-EN}$

Methods	DBP15K$_{FR-EN}$		WK31-60K$_{FR-EN}$	
	Time	H@1	Time	H@1
GM-EHD-JEA	1,474	92.43	–	–
RDGCN	45	88.64	366	77.91
HGCN-JE	50	89.16	392	76.59
NMN	84	90.20	407	77.63
CAECGAT	42	94.68	382	78.24
DuGa-DIT	44	98.17	385	84.69

Table 8. Ablation Study

Methods	DBP15K$_{ZH-EN}$			DBP15K$_{JA-EN}$			DBP15K$_{FR-EN}$		
	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
BASELINE	56.09	71.55	0.614	67.10	78.25	0.712	85.48	91.92	0.878
DuGa	69.17	89.96	0.765	81.29	94.58	0.861	94.42	98.96	0.961
DuGa w/o C	63.86	**91.92**	0.709	72.35	94.64	0.815	86.24	98.70	0.895
DuGa w/o I	57.98	76.62	0.644	76.78	86.39	0.812	91.29	96.30	0.935
DuGa-DIT$_{iter=1}$	78.48	89.54	0.822	89.34	95.14	0.913	97.61	99.21	0.982
DuGa-DIT$_{iter=2}$	**80.73**	88.17	**0.832**	**91.44**	**95.21**	**0.928**	**98.17**	**99.22**	**0.985**

than GM-EHD-JEA. As shown in Table 7, GM-EHD-JEA spends 1,474 seconds,[6] whereas our DuGa-DIT model only needs 44 seconds on DBP15K$_{FR-EN}$. And compared to our preliminary study of the CAECGAT model and other baseline models, the proposed DuGa-DIT not only achieves comparative efficiency but also obtains better performance. On the large dataset WK31-60K$_{FR-EN}$, the time costs of all the models increase dramatically due to the larger number of parameters and testing samples. Our DuGa-DIT model still achieves good efficiency and performance on WK31-60K$_{FR-EN}$. This experimental results show the superiority of our DuGa-DIT model.

4.6 Ablation Study

The proposed DuGa-DIT model consists of some import modules, including the intra-KG attention layer, cross-KG attention layer, and dynamic iterative training process. To investigate the impact of different components, we conduct some ablation studies by removing some modules in our DuGa-DIT model. As shown in Table 8, "BASELINE" is a simple model that finds the equivalent counterparts using the sum of word vectors within the surface names of entities as embeddings. DuGa is the DuGa-DIT model without using the dynamic iterative training method, "DuGa w/o C" is the DuGa model without using the cross-KG attention layer, "DuGa w/o I" is the DuGa model without using the intra-KG attention layer, and "DuGa-DIT$_{iter=1}$" and "DuGa-DIT$_{iter=2}$" denote the DuGa-DIT model with the dynamic iterative training process for one iteration and two iterations, respectively.

As we can observe in Table 8, the BASELINE model achieves good performance by considering surface names, which has also been proved in some existing works [88, 89]. The DuGa model without using the iterative method obtains great performance gains compared with the BASELINE, which indicates that the intra-KG and cross-KG attention mechanism play important roles

[6]The results are reported in their original work. Since GM-EHD-JEA does not release the code, we do not report the results on WK31-60K$_{FR-EN}$.

in our model. By further removing the cross-KG attention, we can see that the performance of "DuGa w/o C" is inferior to the DuGa model, which demonstrates the importance of the cross-KG attention layer. Removing the intra-KG attention layer also can hurt the performance ("DuGa w/o I" vs. DuGa). When we apply the dynamic iterative training process, the model DuGa-DIT$_{iter=1}$ achieves large performance gains compared to DuGa. And with more iterations, the performance can be further improved (DuGa-DIT$_{iter=2}$ vs. DuGa-DIT$_{iter=1}$). These experimental results in the ablation study clearly show that all the modules in our DuGa-DIT model make contributions to the performance.

4.7 Impact of the Number of Layers

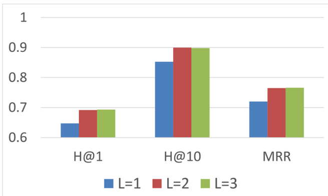
In this section, we conduct experiments to investigate how the number of dual KG attention layers impacts the performance in our model. This experiment is conducted based on the DuGa model without using dynamic iterative training process. Figure 3 illustrates the results of the DuGa models with different layer numbers from $L = 1$ to $L = 3$. We can see that the DuGa model with two layers (e.g., $L = 2$) performs better than the models with one layer (e.g., $L = 1$) on all of these three datasets. This indicates that adding more layers can improve the performance by propagating multi-hop neighborhood and cross-KG alignment features. However, the DuGa model with three layers (e.g., $L = 3$) cannot further improve the performance, which indicates that farther neighbors cannot further provide more useful information for the entity alignment task. And stacking too many layers will increase the parameters in the model, which will hurt the efficiency of the model. Therefore, we set the number of dual KG attention layers to 2.

4.8 Impact of Different Proportions of Seed Alignments for Training

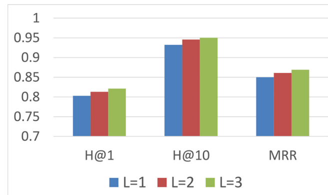
To understand the impact of different proportions of seed alignments on our proposed DuGa-DIT model, we further conduct studies to investigate and evaluate the performance by selecting the proportions of training at 10%, 20%, 30%, 40%, and 50%. Respectively, the testing sets would be the remaining entity alignments of 90%, 80%, 70%, 60%, and 50%. Figure 4 shows the comparison results between our proposed model, DuGa-DIT, and the strong baseline model of DuGa. In contrast to our Duga-DIT, the baseline DuGa eliminates the aggregation layer of the cross-lingual KG in the DuGa-DIT model. As shown in the diagram, the model performances for all datasets follow a gradually uprising trend as the number of seed alignments increases. As DuGa-DIT uses an iterative approach to dynamically add new seed alignments, the performance and robustness of the proposed model is better, and the trendy lines indicated in blue color are smoother. This indicates that our DuGa-DIT model is robust to obtain good performance with limited seed alignments.

4.9 Results for Entities with Different Degrees

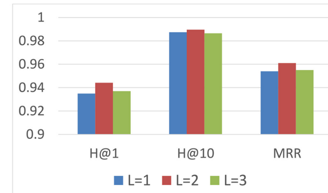
For GNN-based models, the degree of entities is an important factor that may impact the performance. In our DuGa-DIT model, we use the intra-KG attention layer to aggregate neighborhood features. The richness of neighborhood features is related to the degree of entities. Figure 5 illustrates the Hits@1 and MRR results on DBP15K and WK31-60K for entities with different degrees, aiming to better analyze the impact of degree. For DBP15K$_{ZH-EN}$ and DBP15K$_{JA-EN}$, both the H@1 and MRR results gradually improve as the degrees of entities increase. For DBP15K$_{FR-EN}$, the performance grows rapidly when the degree increases to 10, and then the performance is stable at the level of 0.96 to 0.98. Intuitively, the entities with large degrees will have more neighbors and obtain richer neighborhood features by applying intra-KG attention layer. These neighborhood features are beneficial to the entity alignment task. For the DBP15K$_{FR-EN}$, the entity alignment is easy due to the similar surface forms of entities between English and French. Therefore, the



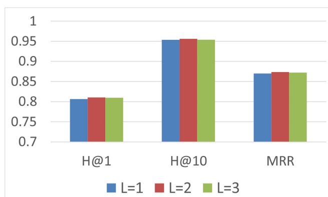
(a) DBP15K$_{ZH-EN}$



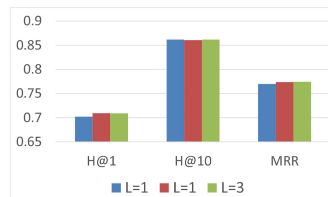
(b) DBP15K$_{JA-EN}$



(c) DBP15K$_{FR-EN}$

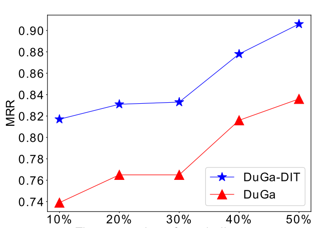


(d) WK31-60K$_{FR-EN}$

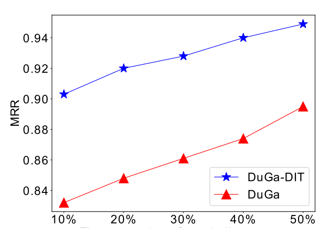


(e) WK31-60K$_{DE-EN}$

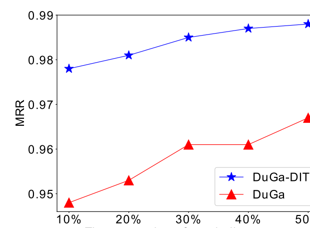
Fig. 3. The results on DBP15K and WK31-60K with different numbers of layers.



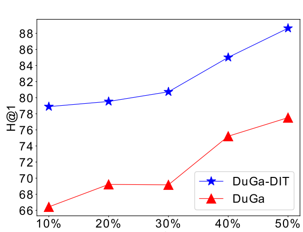
(a) MRR on DBP15K$_{ZH-EN}$



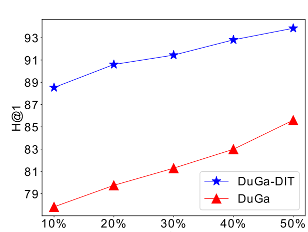
(b) MRR on DBP15K$_{JA-EN}$



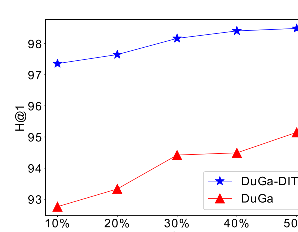
(c) MRR on DBP15K$_{FR-EN}$



(d) H@1 on DBP15K$_{ZH-EN}$



(e) H@1 on DBP15K$_{JA-EN}$



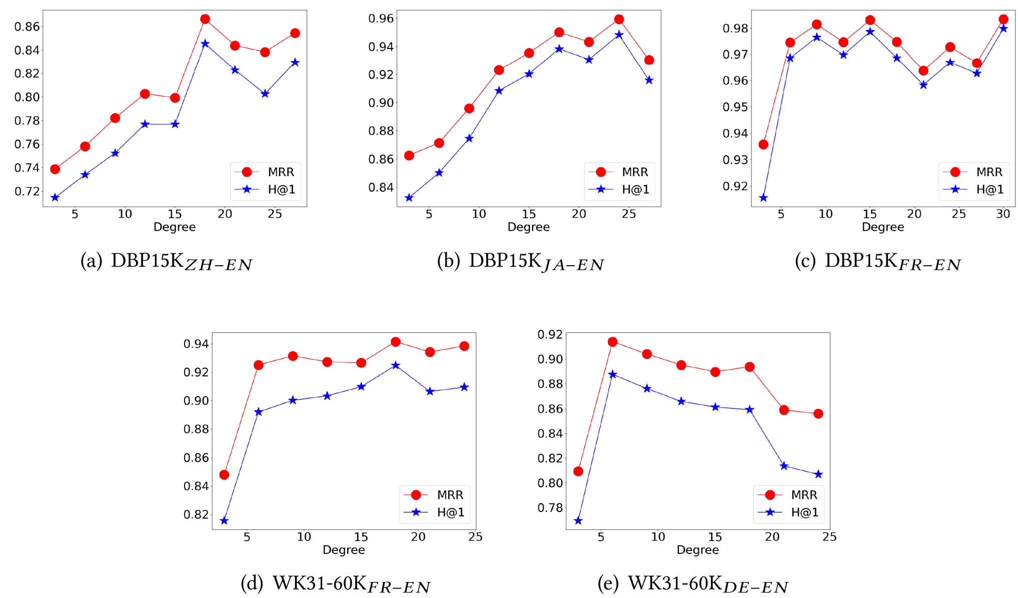
(f) H@1 on DBP15K$_{FR-EN}$

Fig. 4. Performance on DBP15K using different proportions of seed alignments.

performance for DBP15K$_{FR-EN}$ can be improved to a high level with a few degrees. For the WK31-60K dataset, we can observe a similar tendency.

4.10 Impact of the Number of Negative Samples

To train our models, we need to sample some negative entity pairs to compute the objective function. Therefore, the number of negative samples is an important hyper-parameter. We further compare the results on DBP15K with different sizes of negative samples to explore the impact of the number of negative samples. As shown in Table 9, we conduct a series of experiments with



(a) DBP15K$_{ZH-EN}$ (b) DBP15K$_{JA-EN}$ (c) DBP15K$_{FR-EN}$

(d) WK31-60K$_{FR-EN}$ (e) WK31-60K$_{DE-EN}$

Fig. 5. The results on DBP15K and WK31-60K for entities with different degrees.

Table 9. Results on DBP15K with Different Numbers of Negative Samples

	DBP15K$_{ZH-EN}$			DBP15K$_{JA-EN}$			DBP15K$_{FR-EN}$		
Negative Samples	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
10	68.67	88.92	0.762	80.76	94.67	0.858	93.44	98.95	0.957
20	68.85	89.90	0.763	80.74	94.40	0.857	93.93	98.87	0.957
30	**69.17**	**89.96**	**0.765**	81.29	94.58	0.861	**94.42**	98.96	**0.961**
40	68.76	89.85	0.763	81.41	**94.64**	0.862	94.16	98.09	0.959
50	68.72	89.68	0.761	**81.58**	94.57	**0.863**	94.29	**99.07**	0.960

different numbers of negative samples, ranging from 10 to 50, and apply them to the proposed DuGa model. We can see that as the number of negative samples increases, the performance can be slightly improved. However, the degree of improvement is limited. Specifically, when the number of negative samples is set to 30, the model can reach good performance. Therefore, for the purpose of balancing both the effectiveness and efficiency, we select the number of negative samples as 30 in our study.

4.11 Case Study in Personalized Search

Personalized search aims to return relevant user-specific answers for a query submitted by a user. However, most of the personalized search methods focus on monolingual datasets, and very few works have been done to deal with cross-lingual personalized search. In this study, we investigate how to apply the entity alignment model to cross-lingual personalized search. To the best of our knowledge, no dataset for cross-lingual personalized search is available from previous studies, which makes it difficult to do research on cross-lingual personalized search. Inspired by Zhou et al. [97], we collect a cross-lingual personalized search dataset from DBpedia [31]. Specifically, we extract two cross-lingual sub-graphs about songs and artists from English DBpedia and Japanese DBpedia. And we use this dataset to simulate personalized music search, where an artist is viewed as

Table 10. Result on Cross-Lingual Personalized Search

Method	H@1	MRR
BASELINE	21.16	0.409
RDGCN	48.52	0.638
HGCN-JE	44.38	0.608
NMN	38.34	0.556
CAECGAT	58.92	0.757
DuGa	58.70	0.726
DuGa-DIT	64.04	0.761

a query, a genre of the songs is viewed as a user, and songs are viewed as documents to return. This is possible, because genres can represent different points of user interest. When a user searches the songs of a given artist, we should return the songs with a genre in which the user is most interested. The cross-lingual personalized search dataset is created by considering the query and document from different KGs (e.g., the query from English KG and the document from Japanese KG). Formally, let G_1 and G_2 denote the KGs with two different languages. Let $(query_1, user, document_1)$ denote the query and document for a user from G_1, and let $(query_2, user, document_2)$ denote its equivalent query and document for the user from G_2. We call the query and document for a user from a monolingual KG a *monolingual query triple*. Then, we can construct a cross-lingual query and document for the user, which is called a *cross-lingual query triple* and denoted as $(query_1, user, document_2)$. Finally, the cross-lingual personalized search dataset consists of 1,068 monolingual query triples and 1,068 cross-lingual query triples from English to Japanese KG. And we randomly sample 10 negative documents for each of the query triples. To evaluate our entity alignment model, we use the monolingual query triples for training and cross-lingual query triples for testing.

We first use the proposed DuGa-DIT model to learn entity embeddings for entities from two KGs with 30% seed alignments available. Then, the entity embeddings are used as input features for the cross-lingual personalized search. Formally, the feature representations of $query_1$, $user$, $document_1$, and $document_2$ are denoted as $\mathbf{e}_{1,q}$, $\mathbf{e}_u$, $\mathbf{e}_{1,d}$, and $\mathbf{e}_{2,d}$, respectively. Then, we compute the score for a query triples as follows:

$$f(query_1, user, document_2) = \theta(\mathbf{W}_q(\mathbf{e}_{1,q} + \mathbf{e}_u - \mathbf{e}_{2,d})^2 + \mathbf{b}_q), \tag{12}$$

where $\mathbf{W}_q$ and $\mathbf{b}_q$ are the trainable parameters.

We use the embeddings produced by DuGa-DIT as input features and train the scoring model on monolingual query triples and evaluate the model by ranking the cross-lingual query triples according to the scores. Table 10 shows the results on cross-lingual personalized search. The "BASELINE" translates the surface names of entities into the same language and use the sum of word vectors of the surface name as the entity's embedding. We also compare with the state-of-the-art models that also use the pre-trained word vector as the initial entity embedding, including RDGCN, HGCN-JE, NMN, and CAECGAT.

As illustrated in Table 10, the DuGa model performs better than most of the baseline models in the task of cross-lingual personalized search, which shows that our DuGa model is effective to model cross-lingual entities. Furthermore, by applying the dynamic iterative training process, our DuGa-DIT further improves the performance on the cross-lingual personalized search. The results reconfirm the effectiveness of the proposed DuGa-DIT model and demonstrate that our DuGa-DIT model is able to apply to cross-lingual personalized search.

Table 11. Cross-Lingual Personalized Search Examples of the Predicted Results of Our DuGa-DIT Versus the Strong Baseline RDGCN Model

Query and User	Candidates	Rank	
		DuGa-DIT	RDGCN
(*Lennon McCartney*, *Art_rock*, ?)	ア　デイ　イン　ザ　ライフ (*A Day in the Life*)	1st	6th
	ザ　タッチ (*Stan Bush*)	2nd	9th
	アイ　ウォント　トゥ　テル　ユー (*I Want to Tell You*)	3rd	1st
(*The Beatles*, *Folk_music*, ?)	ハー　マジェスティー (*Her Majesty*)	1st	5th
	消えた恋 (*What Goes On*)	2nd	7th
	ロング　ロング　ロング (*Long, Long, Long*)	3rd	2nd
(*Joe Strummer*, New_wave, ?)	スペイン戦争 (*Spanish Bombs*)	1st	4th
	ロッキー　ラクーン (*Rocky Raccoon*)	2nd	8th
	エニイ　タイム　アット　オール (*Any Time at All*)	3rd	5th

Underline indicates the true document.

Table 11 shows some examples of the cross-lingual personalized search. We compare the rank of our DuGa-DIT model with the strong baseline RDGCN model. As illustrated in Table 11, the baseline model RDGCN fails to rank these candidate documents, whereas our DuGa-DIT model is able to correctly rank the true document at the first rank. For example, with the query *Lennon McCartney* and the user *Art_rock*, we can rank the song ア　デイ　イン　ザ　ライフ (*A Day in the Life*), which is a song with a genre of art rock, as the first position. These examples verify the effectiveness of our DuGa-DIT model to apply to cross-lingual personalized search.

5 CONCLUSION AND FUTURE WORK

In this article, we propose a DuGa-DIT network with dynamic iterative training for the cross-lingual KG entity alignment task, which takes full use of seed alignments to alleviate the semantic gap between different KGs. In the DuGa-DIT model, we use a intra-KG attention layer to aggregate local neighborhood features and a cross-KG attention layer to gather cross-KG alignment information, and these two kinds of features are merged to update the embedding of two KGs. A dynamic iterative training process is used to dynamically update the cross-KG attention score matrices, which enable our model to capture more cross-KG information. Experimental results on two benchmark datasets show that our model is effective and robust for the cross-lingual entity alignment task. And a case study on cross-lingual personalized search task demonstrates that our DuGa-DIT model achieves promising results for cross-lingual personalized search.

Even though the DuGa-DIT model performs well on the given datasets, some future work still needs to be done to bring this stream of research further. Details of proposed future directions are presented as follows:

- The proposed method in this article simply exercises alignment between two KGs. The purpose is for us to build more effective, efficient, and robust models on top of the foundation. In the future, it would be helpful to learn the auto alignment task considering more entity information, such as entity properties and descriptions, among three or more KGs.
- How to design a distributed auto alignment method would become an important direction for future research. In real-life scenarios, the size of KGs is usually very large, especially for tasks in personalized search and recommendation. Furthermore, the structural differences of the KGs can be even bigger.
- In addition to applying our model to the KG entity alignment task to English, Chinese, French, and Japanese, we plan to apply this to more resourceful and other specific domains, which include but are not limited to English to Malay and English to Portuguese.

ACKNOWLEDGMENTS

We thank the anonymous reviewers and associate editor for their insightful comments.

REFERENCES

[1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. DBpedia: A nucleus for a web of open data. In *The Semantic Web*. Lecture Notes in Computer Science, Vol. 4825. Springer, 722–735. https://doi.org/10.1007/978-3-540-76298-0_52

[2] Trapit Bansal, Da-Cheng Juan, Sujith Ravi, and Andrew McCallum. 2019. A2N: Attending to neighbors for knowledge graph inference. In *Proceedings of the ACL*. 4387–4392.

[3] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of SIGMOD*. 1247–1250. https://doi.org/10.1145/1376616.1376746

[4] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2014. A semantic matching energy function for learning with multi-relational data—Application to word-sense disambiguation. *Machine Learning* 94, 2 (2014), 233–259.

[5] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of NIPS*. 2787–2795.

[6] Yixin Cao, Zhiyuan Liu, Chengjiang Li, Zhiyuan Liu, Juanzi Li, and Tat-Seng Chua. 2019. Multi-channel graph neural network for entity alignment. In *Proceedings of the ACL*. 1452–1461. https://doi.org/10.18653/v1/p19-1140

[7] Bo Chen, Jing Zhang, Xiaobin Tang, Hong Chen, and Cuiping Li. 2020. JarKA: Modeling attribute interactions for cross-lingual knowledge alignment. In *Proceedings of PAKDD*. 845–856. https://doi.org/10.1007/978-3-030-47426-3_65

[8] Danqi Chen, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2013. Learning new facts from knowledge bases with neural tensor networks and semantic word vectors. arXiv:1301.3618.

[9] Muhao Chen, Yingtao Tian, Kai-Wei Chang, Steven Skiena, and Carlo Zaniolo. 2018. Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. In *Proceedings of IJCAI*. 3998–4004. https://doi.org/10.24963/ijcai.2018/556

[10] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of IJCAI*. 1511–1517.

[11] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D knowledge graph embeddings. In *Proceedings of AAAI*. 1811–1818.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*. 4171–4186. https://doi.org/10.18653/v1/n19-1423

[13] Xiaoyu Du, Xiang Wang, Xiangnan He, Zechao Li, Jinhui Tang, and Tat-Seng Chua. 2020. How to learn item representation for cold-start multimedia recommendation? In *Proceedings of MM*. 3469–3477. https://doi.org/10.1145/3394171.3413628

[14] Jun Feng, Minlie Huang, Mingdong Wang, Mantong Zhou, Yu Hao, and Xiaoyan Zhu. 2016. Knowledge graph embedding by flexible translation. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016*, Chitta Baral, James P. Delgrande, and Frank Wolter (Eds.). AAAI Press, 557–560.

[15] Alberto García-Durán, Antoine Bordes, and Nicolas Usunier. 2015. Composing relationships with translations. In *Proceedings of EMNLP*. 286–290.

[16] Alberto Garcća-Duran and Mathias Niepert. 2018. KBLRN: End-to-end learning of knowledge base representations with latent, relational, and numerical features. arXiv:1709.04676.

[17] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of AISTATS*. 315–323.

[18] Lingbing Guo, Zequn Sun, and Wei Hu. 2019. Learning to exploit long-term relational dependencies in knowledge graphs. In *Proceedings of ICML*. 2505–2514.

[19] Yanchao Hao, Yuanzhe Zhang, Shizhu He, Kang Liu, and Jun Zhao. 2016. A joint embedding method for entity alignment of knowledge bases. In *Proceedings of CCKS*, Vol. 650. 3–14.

[20] Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. 2015. Learning to represent knowledge graphs with Gaussian embedding. In *Proceedings of CIKM*. 623–632.

[21] Rodolphe Jenatton, Nicolas L. Roux, Antoine Bordes, and Guillaume R. Obozinski. 2012. A latent factor model for highly multi-relational data. In *Proceedings of NIPS*. 3176–3184.

[22] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the ACL*. 687–696.

[23] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2016. Knowledge graph completion with adaptive sparse transfer matrix. In *Proceedings of AAAI*. 985–991.

[24] Hailong Jin, Chengjiang Li, Jing Zhang, Lei Hou, Juanzi Li, and Peng Zhang. 2019. XLORE2: Large-scale cross-lingual knowledge graph construction and application. *Data Intelligence* 1, 1 (2019), 77–98. https://doi.org/10.1162/dint_a_00003

[25] Seyed Mehran Kazemi and David Poole. 2018. SimplE embedding for link prediction in knowledge graphs. In *Proceedings of NeurIPS*. 4289–4300.

[26] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

[27] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of ICLR*.

[28] Patrick Klein, Simone Paolo Ponzetto, and Goran Glavaš. 2017. Improving neural knowledge base completion with cross-lingual projections. In *Proceedings of EACL*. 516–522.

[29] Agustinus Kristiadi, Mohammad Asif Khan, Denis Lukovnikov, Jens Lehmann, and Asja Fischer. 2019. Incorporating literals into knowledge graph embeddings. arXiv:1802.00934.

[30] D. Krompab, S. Baier, and V. Tresp. 2015. Type-constrained representation learning in knowledge graphs. In *Proceedings of ISWC*. 640–655. https://doi.org/10.1007/978-3-319-25007-6_37

[31] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, et al. 2015. DBpedia—A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* 6, 2 (2015), 167–195. https://doi.org/10.3233/SW-140134

[32] Chengjiang Li, Yixin Cao, Lei Hou, Jiaxin Shi, Juanzi Li, and Tat-Seng Chua. 2019. Semi-supervised entity alignment via joint knowledge embedding model and cross-graph model. In *Proceedings of EMNLP-IJCNLP*. 2723–2732. https://doi.org/10.18653/v1/D19-1274

[33] Juanzi Li, Jie Tang, Yi Li, and Qiong Luo. 2009. RiMOM: A dynamic multistrategy ontology alignment framework. *IEEE Transactions on Knowledge and Data Engineering* 21, 8 (2009), 1218–1232. https://doi.org/10.1109/TKDE.2008.202

[34] Xixun Lin, Hong Yang, Jia Wu, Chuan Zhou, and Bin Wang. 2019. Guiding cross-lingual entity alignment via adversarial knowledge embedding. In *Proceedings of ICDM*. 429–438. https://doi.org/10.1109/ICDM.2019.00053

[35] Yankai Lin, Zhiyuan Liu, Huan-Bo Luan, Maosong Sun, Siwei Rao, and Song Liu. 2015. Modeling relation paths for representation learning of knowledge bases. In *Proceedings of EMNLP*. 705–714.

[36] Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2016. Knowledge representation learning with entities, attributes and relations. In *Proceedings of IJCAI*. 2866–2872.

[37] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of AAAI*. 2181–2187.

[38] Zhiyuan Liu, Yixin Cao, Liangming Pan, Juanzi Li, and Tat-Seng Chua. 2020. Exploring and evaluating attributes, values, and structures for entity alignment. In *Proceedings of EMNLP*. 6355–6364.

[39] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the ICML Workshop*.

[40] Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. 2015. YAGO3: A knowledge base from multilingual Wikipedias. In *Proceedings of CIDR*.

[41] Xin Mao, Wenting Wang, Huimin Xu, Man Lan, and Yuanbin Wu. 2020. MRAEA: An efficient and robust entity alignment approach for cross-lingual knowledge graph. In *Proceedings of WSDM*. 420–428. https://doi.org/10.1145/3336191.3371804

[42] Andrew McCallum and Ben Wellner. 2004. Conditional models of identity uncertainty with application to noun coreference. In *Proceedings of NIPS*. 905–912.

[43] Lei Meng, Fuli Feng, Xiangnan He, Xiaoyan Gao, and Tat-Seng Chua. 2020. Heterogeneous fusion of semantic and collaborative information for visually-aware food recommendation. In *Proceedings of MM*. 3460–3468. https://doi.org/10.1145/3394171.3413598

[44] George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM* 38, 11 (1995), 39–41. https://doi.org/10.1145/219717.219748

[45] Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. Learning attention-based embeddings for relation prediction in knowledge graphs. In *Proceedings of the ACL*. 4710–4723.

[46] Roberto Navigli and Simone Paolo Ponzetto. 2010. BabelNet: Building a very large multilingual semantic network. In *Proceedings of the ACL*. 216–225.

[47] Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2018. A novel embedding model for knowledge base completion based on convolutional neural network. In *Proceedings of NAACL*. 327–333. https://doi.org/10.18653/v1/N18-2053

[48] Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. 2016. Neighborhood mixture model for knowledge base completion. In *Proceedings of SIGNLL*. 40–50. https://doi.org/10.18653/v1/K16-1005

[49] Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Q. Phung. 2019. A capsule network-based embedding model for knowledge graph completion and search personalization. In *Proceedings of NAACL-HLT*. 2180–2189.

[50] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of AAAI*. 1955–1961.

[51] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of ICML*. 809–816.

[52] Hao Nie, Xianpei Han, Le Sun, Chi Man Wong, Qiang Chen, Suhui Wu, and Wei Zhang. 2020. Global structure and local semantics-preserved embeddings for entity alignment. In *Proceedings of IJCAI*. 3658–3664. https://doi.org/10.24963/ijcai.2020/506

[53] Byungkook Oh, Seungmin Seo, and Kyong-Ho Lee. 2018. Knowledge graph completion by context-aware convolutional learning with multi-hop neighborhoods. In *Proceedings of CIKM*. 257–266. https://doi.org/10.1145/3269206.3271769

[54] Shichao Pei, Lu Yu, Robert Hoehndorf, and Xiangliang Zhang. 2019. Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In *Proceedings of WWW*. 3130–3136. https://doi.org/10.1145/3308558.3313646

[55] Shichao Pei, Lu Yu, Guoxian Yu, and Xiangliang Zhang. 2020. REA: Robust cross-lingual entity alignment between knowledge graphs. In *Proceedings of KDD*. 2175–2184.

[56] Shichao Pei, Lu Yu, and Xiangliang Zhang. 2019. Improving cross-lingual entity alignment via optimal transport. In *Proceedings of IJCAI*. 3231–3237. https://doi.org/10.24963/ijcai.2019/448

[57] Wei Qian, Cong Fu, Yu Zhu, Deng Cai, and Xiaofei He. 2018. Translating embeddings for knowledge graph completion with relation attention mechanism. In *Proceedings of IJCAI*. 4286–4292.

[58] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *Proceedings of ESWC*. 593–607.

[59] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of AAAI*. 3060–3067. https://doi.org/10.1609/aaai.v33i01.33013060

[60] Baoxu Shi and Tim Weninger. 2017. ProjE: Embedding projection for knowledge graph completion. In *Proceedings of AAAI*. 1236–1242.

[61] Xiaofei Shi and Yanghua Xiao. 2019. Modeling multi-mapping relations for precise cross-lingual entity alignment. In *Proceedings of EMNLP*. 813–822. https://doi.org/10.18653/v1/D19-1075

[62] Amit Singhal. 2012. Introducing the knowledge graph: Things, not strings. *Google Blog*.

[63] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2008. YAGO: A large ontology from Wikipedia and WordNet. *Journal of Web Semantics* 6, 3 (2008), 203–217. https://doi.org/10.1016/j.websem.2008.06.001

[64] Zequn Sun, Muhao Chen, Wei Hu, Chengming Wang, Jian Dai, and Wei Zhang. 2020. Knowledge association with hyperbolic knowledge graph embeddings. arXiv:2010.02162.

[65] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge graph embedding by relational rotation in complex space. In *Proceedings of ICLR*.

[66] Zequn Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *Proceedings of ISWC*. 628–644.

[67] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *Proceedings of IJCAI*. 4396–4402.

[68] Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. 2020. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *Proceedings of AAAI*. 222–229.

[69] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of ICML*. 2071–2080.

[70] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Nilesh Agrawal, and Partha P. Talukdar. 2020. InteractE: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In *Proceedings of AAAI*. 3009–3016.

[71] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph attention networks. arXiv:1710.10903.

[72] Denny Vrandecic. 2012. Wikidata: A new platform for collaborative data collection. In *Proceedings of WWW*. 1063–1064. https://doi.org/10.1145/2187980.2188242

[73] Kai Wang, Yu Liu, Xiujuan Xu, and Dan Lin. 2018. Knowledge graph embedding with entity neighbors and deep memory network. arXiv:1808.03752. https://doi.org/CoRR,abs/1808.03752.

[74] Quan Wang, Pingping Huang, Haifeng Wang, Songtai Dai, Wenbin Jiang, Jing Liu, Yajuan Lyu, Yong Zhu, and Hua Wu. 2019. CoKE: Contextualized knowledge graph embedding. arXiv:1911.02168.

[75] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of AAAI*. 5329–5336. https://doi.org/10.1609/aaai.v33i01.33015329

[76] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of EMNLP*. 349–357. https://doi.org/10.18653/v1/d18-1032

[77] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of AAAI*. 1112–1119.

[78] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. 2020. Graph-refined convolutional network for multimedia recommendation with implicit feedback. In *Proceedings of MM*. 3541–3549. https://doi.org/10.1145/3394171.3413556

[79] Peng Wu, Shujian Huang, Rongxiang Weng, Zaixiang Zheng, Jianbing Zhang, Xiaohui Yan, and Jiajun Chen. 2019. Learning representation mapping for relation detection in knowledge base question answering. In *Proceedings of the ACL*. 6130–6139. https://doi.org/10.18653/v1/p19-1616

[80] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019. Relation-aware entity alignment for heterogeneous knowledge graphs. In *Proceedings of IJCAI*. 5278–5284.

[81] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. 2019. Jointly learning entity and relation representations for entity alignment. In *Proceedings of EMNLP-IJCNLP*. 240–249.

[82] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. 2020. Neighborhood matching network for entity alignment. In *Proceedings of the ACL*. 6477–6487.

[83] Yanrong Wu and Zhichun Wang. 2018. Knowledge graph embedding with numeric attributes of entities. In *Proceedings of RepL4NLP*. 132–136.

[84] Han Xiao, Minlie Huang, and Xiaoyan Zhu. 2016. TransG: A generative model for knowledge graph embedding. In *Proceedings of the ACL*.

[85] Zhiwen Xie, Guangyou Zhou, Jin Liu, and Jimmy Xiangji Huang. 2020. ReInceptionE: Relation-aware inception network with joint local-global structural information for knowledge graph embedding. In *Proceedings of the ACL*. 5929–5939.

[86] Zhiwen Xie, Runjie Zhu, Kunsong Zhao, Jin Liu, Guangyou Zhou, and Jimmy Xiangji Huang. 2020. A contextual alignment enhanced cross graph attention network for cross-lingual entity alignment. In *Proceedings of COLING*. 5918–5928. https://doi.org/10.18653/v1/2020.coling-main.520

[87] Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. DeepPath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of EMNLP*. 564–573. https://doi.org/10.18653/v1/d17-1060

[88] Kun Xu, Linfeng Song, Yansong Feng, Yan Song, and Dong Yu. 2020. Coordinated reasoning for cross-lingual knowledge graph alignment. In *Proceedings of AAAI*. 9354–9361.

[89] Kun Xu, Liwei Wang, Mo Yu, Yansong Feng, Yan Song, Zhiguo Wang, and Dong Yu. 2019. Cross-lingual knowledge graph alignment via graph matching neural network. In *Proceedings of the ACL*. 3156–3161. https://doi.org/10.18653/v1/p19-1304

[90] Zhihuan Yan, Rong Peng, Yaqian Wang, and Weidong Li. 2020. CTEA: Context and topic enhanced entity alignment for knowledge graphs. *Neurocomputing* 410 (2020), 419–431. https://doi.org/10.1016/j.neucom.2020.06.054

[91] Bishan Yang, Wen-Tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of ICLR*.

[92] Hsiu-Wei Yang, Yanyan Zou, Peng Shi, Wei Lu, Jimmy Lin, and Xu Sun. 2019. Aligning cross-lingual entities with multi-aspect information. In *Proceedings of EMNLP*. 4430–4440. https://doi.org/10.18653/v1/D19-1451

[93] Shihui Yang, Jidong Tian, Honglun Zhang, Junchi Yan, Hao He, and Yaohui Jin. 2019. TransMS: Knowledge graph embedding for complex relations by multidirectional semantics. In *Proceedings of IJCAI*. 1935–1942.

[94] Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embeddings. In *Proceedings of NeurIPS*. 2731–2741.

[95] Youmin Zhang, Li Liu, Shun Fu, and Fujin Zhong. 2018. Entity alignment across knowledge graphs based on representative relations selection. *Proceedings of ICSAI*. 1056–1061. https://doi.org/10.1109/ICSAI.2018.8599288

[96] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Enhanced language representation with informative entities. In *Proceedings of the ACL*. 1441–1451. https://doi.org/10.18653/v1/p19-1139

[97] Dong Zhou, Wenyu Zhao, Xuan Wu, Séamus Lawless, and Jianxun Liu. 2018. An iterative method for personalized results adaptation in cross-language search. *Information Sciences* 430 (2018), 200–215.

[98] Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Iterative entity alignment via joint knowledge embeddings. In *Proceedings of IJCAI*. 4258–4264. https://doi.org/10.24963/ijcai.2017/595

[99] Qiannan Zhu, Xiaofei Zhou, Jia Wu, Jianlong Tan, and Li Guo. 2019. Neighborhood-aware attentional representation for multilingual knowledge graphs. In *Proceedings of IJCAI*. 1943–1949.

Received November 2020; revised April 2021; accepted June 2021