Collective Knowledge Graph Completion with Mutual Knowledge Distillation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Knowledge graph completion (KGC), the task that aims at predicting missing information based on the already existing relational data inside a knowledge graph(KG), has drawn significant attention in the recent years. However, predictive power of KGC methods is often limited by the completeness of the existing knowledge graphs. In monolingual and multilingual settings, KGs from different sources and languages are potentially complementary to each other. In this paper, we study the problem of multi-KG completion, where we focus on maximizing the collective knowledge from different KGs to alleviate the incompleteness on individual KGs. Specifically, we propose a novel method called CKGC-MKD that uses augmented CompGCN-based encoder models on both individual KGs and a large fused KG in which seed alignments between KGs are regarded as edges for message propagation. Additional mutual knowledge distillation are employed to maximize the knowledge transfer between the models of "global" fused KG and the "local" individual KGs. Experimental results on multilingual datasets has shown that our method outperforms all state-of-the-art models.

1 Introduction

Knowledge graphs (KGs) have been widely adopted in many industry applications because they capture the multi-relational nature between real-world entities well. KGC, along with many other KG-based applications, are usually based on knowledge representation learning (KRL), in which entities and relations in a KG are encoded into low-dimensional vectors. With recent advances in Graph Neural Network(GNN) (Scarselli et al., 2009), many recently published methods like R-GCN (Schlichtkrull et al., 2018) and CompGCN (Vashishth et al., 2020) all employed an encoder-decoder mechanism to tackle the KGC problem: variations of Graph Convolutional Networks (GCN) (Kipf and Welling, 2017) are used as encoders to generate embeddings for entities and relations in a KG, and traditional KG embedding methods like TransE (Bordes et al., 2013) and DistMult (Yang et al., 2015) are used as decoders for the KGC task. With the additional message propagation and aggregation mechanism of graph convolution in the encoding stage, these methods have shown more promising results on the KGC task comparing to the traditional knowledge graph embedding methods. However, even with better encoding mechanism of GCNs, expressiveness and quality of trained models can still be limited by the sparseness of the individual KG the model is trained on. At the same time, real-world entities are usually captured in more than one KGs from either different sources or different languages. The common entities in the disjoint real-world KGs can potentially serve as bridges to better connect them and transfer additional knowledge to one another to alleviate the sparseness problem faced by almost all of the real-world KGs. The common entities across different KGs are known as *seed alignments*, which usually originates from the manual annotation of human annotators. Because of the scale and size of KGs, seed alignments are usually relatively scarce.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

In this paper, we focus on the multi-KG completion problem, where we aim to collectively utilize multiple KGs and seed alignments between them to maximize the the KGC task performance on each individual KG. Concretely, we propose a novel method that concurrently trains CompGCN-based encoders on each individual KGs as well as a fused KG where seed alignments are regarded as edges for connecting KGs together and for augmented message propagation for "knowledge transfer". During the concurrent training, we also employ the mutual knowledge distillation mechanism, in which CompGCN-based encoders on individual KGs and the fused KG are trained to learn potentially complementary features from each other. The intuition behind the mutual knowledge distillation process is that the small encoders trained on individual KGs capture local semantic relationships better, while the large encoder trained on the large fused KG captures the global semantic relationships better because of the intra-KG message propagation. In the mutual knowledge distillation process, the small and large encoders take turns to become "teacher" in the knowledge distillation, to encourage mutual knowledge transfer between them. Lastly, we use ensemble to combine the predictions from the individual KG and fused KG to produce the KGC predictions on test set for each individual KG.

The main contribution of this paper can be summarized as follows: 1) we propose a novel augmented CompGCN encoder to facilitate intra-KG knowledge transfer and tackle the multi-KG completion task; 2) we propose a novel mutual knowledge distillation mechanism to encourage collaborative knowledge transfer between the models trained on individual KGs and globally fused KG. Experimental results on popular multilingual datasets show that our proposed method outperforms all of the state-of-the-art models. Extensive ablation studies are conducted on both monolingual and multilingual datasets to demonstrate the contribution of each component in the proposed method.

2 Methods

2.1 Preliminaries

The framework of multi-KG completion task involves two or more KGs. Without loss of generality, we assume there are a total of m KGs in the problem setting. We formalize the i-th heterogeneous KG in the task as $KG_i = \{E_i, R_i, T_i\}$, where E_i, R_i, T_i respectively represent the entity set, relation set, and fact triple set of KG_i. A small set of seed alignments between KGs, known before training, is denoted by $S_{KG_i,KG_j} = \{(e_i, \sim, e_j) : (e_i, e_j) \in E_i \times E_j\}$, where $\sim$ denotes the equivalence relation. The full set of seed alignments can then be denoted by $S_{align} = \cup_{i=1}^{m} \cup_{j=i+1}^{m} S_{KG_i,KG_j}$. We can then formalize the large fused KG connected by seed alignments as $KG_f = \{E_f, R_f, T_f | E_f = \cup_{i=1}^{m} E_i, R_f = \cup_{i=1}^{m} R_i, T_f = (\cup_{i=1}^{m} T_i) \cup S_{align}\}$. Let M_i and M_f denote the encoder models for the i-th individual KG and the fused KG respectively.

2.2 Augmented CompGCN Message Propagation

We decide to use CompGCN (Vashishth et al., 2020) as our encoders for the knowledge graph embeddings. In the method, CompGCN encoders are trained on each individual KGs and the fused KG concurrently. The update equation of CompGCN node embeddings can be written as:

$$h_v^t = f(\Sigma_{(u,r) \in N(v)} Me(u, r)), \tag{1}$$

$$Me(u, r) = W_{\lambda(r)} \phi(h_u^{t-1}, h_r^{t-1}), \tag{2}$$

where h_v^t denotes the updated embedding for node v at t-th layer, $N(v)$ denotes the set of neighboring entities and relations of node v, h_u^{t-1} and h_r^{t-1} denotes the embeddings for node u and relation r at (t-1)-th layer respectively, ϕ denotes the non-parametric composition operation and $W_{\lambda(r)}$ denotes the direction specific transformation matrix where λ denotes the direction of relation. In our method, the vanilla CompGCN encoder is used without modification on individual KGs, while we decide to use an augmented CompGCN encoder for better knowledge transfer on the fused KG_f. Specifically, although seed alignments are viewed as relations in the fused KG, we remove the composition operator for message propagation between the KGs and instead use the standard non-relation-specific message passing. The augmented message function in the fused KG can then be written as:

$$Me(u, r) = \begin{cases} W_{align} h_u^{t-1}, & \text{if} (u, \sim, v) \in S_{align} \\ W_{\lambda(r)} \phi(h_u^{t-1}, h_r^{t-1}), & \text{otherwise} \end{cases} \tag{3}$$

where W_{align} denotes the transformation matrix for seed alignments. The composition operation is removed because we view the cross-KG equivalence as a different type of bi-directional relationship comparing to the triples inside KGs. Additionally, many existing methods (Wang et al., 2021; Singh et al., 2021) use a loss regularization to ensure the equivalent entities in each KG to have similar embeddings with or without transformation. However, instead of imposing the regularization directly on the training loss term, we impose a softer regularization in the message passing augmentation, where the contextualized node embeddings of entities in each knowledge graph are passed to their counterparts in other KGs during encoding. As a result, contextualized embedding of entities in each KG can be shared across the KGs by the augmented message propagation in the encoding phase, and optimized during the training of KGC task on fact triples.

The encoded entity and relation embeddings are then passed to the decoder, which performs the link prediction task on triples in KG, and computes the knowledge representation loss. The margin-based knowledge representation loss can be written as:

$$L_T = \Sigma_{t_i \in T_i, t'_i \in T'_i} f(t_i) - f(t'_i) + \gamma, \tag{4}$$

where T'_i denotes the negative samples created from corrupting head or tail entity in triple t_i; $f(t_i)$ denotes the scoring function of traditional knowledge embedding model; and γ denotes the margin, a hyperparameter describing the ideal distance between the positive triples and negative triples.

2.3 Mutual Knowledge Distillation

We employ the mutual knowledge distillation mechanism between each model on individual KGs M_i and the model on the fused KG M_f. At each training step, each M_i pair with M_f to conduct mutual knowledge distillation, where M_i and M_f learns simultaneously from each other via a mimicry loss that measures the difference between the posterior predictions of each other on KGC task on triples T_i in the corresponding KG_i. Three different KGC tasks are used for mutual knowledge distillation: for a triple (s, r, o), the task is to predict the missing component given the other two in the triple, i.e., head prediction, tail prediction and relation prediction. The distillation loss can be written as:

$$L_D^i = \sum_{(s_i, r_i, o_i) \in T_i} \sum_{\beta \in Task} D_{KL}(P_i^\beta(s_i, r_i, o_i), P_f^\beta(s_i, r_i, o_i)), \tag{5}$$

where $Task$ denotes the three KGC tasks, D_{KL} denotes the Kullback Leibler (KL) Divergence, and P denotes the categorical distribution predicted by the knowledge graph embedding scoring function on task β. As an example, for tail prediction, the categorical distribution can be written as softmax of tail prediction across all candidates: $P_i(s_i, r_i, o_i) = \frac{exp(f(M_i(s_i), M_i(r_i), M_i(o_i)))}{\sum_{o_j \in E_i} exp(f(M_i(s_i), M_i(r_i), M_i(o_j)))}$, where $M_i()$ denotes the embedding look up operation for entities and relations from the output of encoder model M_i. In practice, predicting across all candidates E_i and comparing the categorical distribution across all entities can be inefficient due the the size of KG. Therefore, we employ the top-k sampling technique used in the work of Sourty et al. (2020) to use the "teacher" model to select top-k most confident candidates for the categorical distribution comparison.

2.4 Training and ensemble prediction

The overall loss term combines the knowledge representation and mutual knowledge distillation loss: $L = L_T + \alpha L_D$, where α is a hyperparameter controlling the trade-off between two loss terms in the overall loss term. The models M_i and M_f are trained concurrently on KGC tasks while learning from the best-performing model of each other via the mutual distillation process. In practice, for better convergence and faster training, the training process is separated into two stages. In the first stage, both individual models and the fused model are trained independently with only knowledge representation loss; while in the second stage, knowledge distillation losses are introduced so that models can mutually learn from each other.

In the end, the output for KGC tasks are generated by combining predictions from models M_i and M_f using ensemble. Concretely, the for triple $t_i \in T_i$, the final scoring function becomes: $f(M_i(t_i)) + f(M_f(t_i))$. The ensemble scores are then used for further ranking and evaluation.

Table 1: Results on DBP-5L dataset.

	EL	JA	ES	FR	EN
	H@1/H@10/MRR	H@1/H@10/MRR	H@1/H@10/MRR	H@1/H@10/MRR	H@1/H@10/MRR
KenS	28.1 / 56.9 / -	32.1 / 65.3 / -	23.6 / 60.1 / -	25.5 / 62.9 / -	15.1 / 39.8 / -
CG-MuA	21.5 / 44.8 / 32.8	27.3 / 61.1 / 40.1	22.3 / 55.4 / 34.3	24.2 / 57.1 / 36.1	13.1 / 33.5 / 22.2
AlignKGC	27.6 / 56.3 / 33.8	31.6 / 64.3 / 41.6	24.2 / 60.9 / 35.1	24.1 / 62.3 / 37.4	15.5 / 39.2 / 22.3
SS-AGA	30.8 / 58.6 / 35.3	34.6 / 66.9 / 42.9	25.5 / 61.9 / 36.6	27.1 / 65.5 / 38.4	16.3 / 41.3 / 23.1
KGC-I	28.9 / 66.8 / 41.6	30.3 / 61.7 / 41.4	24.8 / 61.2 / 37.5	25.8 / 64.1 / 39.1	20.5 / 58.6 / 33.5
KGC-A	40.4 / 85.3 / 55.3	38.7 / 80.4 / 52.5	31.5 / 75.3 / 46.3	34.3 / **78.9** / 49.3	24.4 / 65.5 / 38.2
CKGC-MKD	**45.1 / 86.0 / 59.8**	**43.6 / 82.1 / 57.0**	**34.8 / 75.9 / 49.3**	**38.1** / 78.1 / **52.3**	**27.8 / 66.4 / 41.3**

3 Experiments

3.1 Basic settings

We perform experiments and compare the performance of proposed CKGC-MKD method with the state-of-the-art models on the existing multilingual dataset **DBP-5L** (Chen et al., 2020). The dataset contains five KGs from different languages: English (EN), French (FR), Spanish (ES), Japanese (JA) and Greek (EL). In this work, we follow the evaluation scheme of previous works (Chen et al., 2020; Singh et al., 2021; Huang et al., 2022): for a test triple (h, r, t), rank all possible answers to tail prediction query $(h, r, ?)$; and apply the MRR(mean reciprocal ranks), Hit@1 and Hit@10 metrics under filtered settings (Wang et al., 2014; Yang et al., 2015) to evaluate the performance. The reported CKGC-MKD uses 1-layer encoder, with TransE as knowledge embedding decoder and embedding dimension of 100. However, CKGC-MKD can be easily extended to use other decoders.

3.2 Results

In table 1 we present the experiment results on the **DBP-5L** dataset [1]. In the table, performances of two extra baseline models are reported: KGC-I refers to the standard CompGCN encoder model trained on individual KG, KGC-A refers to the augmented message propagation encoder trained on the fused KG. It can be observed that the proposed CKGC-MKD method outperforms all baseline and state-of-the-art models on the DBP-5L dataset. Comparing to the previous models, the individually trained KGC-I model on each language can already achieve similar performance on most of the languages, which indicates the effectiveness of the CompGCN encoder. The KGC-A model trained on the fused KG provided a large margin over the KGC-I and the previous models. This implies that the inclusion of multiple KGs truly helps the KGC task of each other and also verifies the benefit of the augmented cross-KG message propagation. In the end, with mutual knowledge distillation between KGC-I and KGC-A enabled, the CKGC-MKD model use the ensemble predictions from both distilled models. This achieves the best performances in the table across almost all of the metrics. Complexity wise, additional cross-KG connections in KGC-A model introduced approximately 25% more additions in the message propagation of the encoders. Most of the additional complexities in the proposed method are introduced in the mutual knowledge distillation, in which two more forward passes are required on each individual model while the distillation loss terms also add extra computation complexities during training. At the cost of extra complexity, the proposed model achieves state-of-the-art performances on the multilingual dataset and demonstrated benefits of incorporating knowledge distillation.

4 Conclusions

In this paper, we proposed a novel method CKGC-MKD that focuses on the KGC task across multiple KGs. The proposed method uses an augmented CompGCN encoder for message propagation across different KGs via seed alignments in a fused KG. Additional mutual knowledge distillations between individual KGs and the fused KG are employed by the proposed model to maximize knowledge transfer. CKGC-MKD beats the state-of-the-art models by a significant margin on KGC

[1]We directly report the benchmarking results from the work of Huang et al. (2022) for the first four rows in the table. For fairness of comparison, results we report in the table all adopted the filtered setting by Huang et al. (2022) instead of the traditional setting: Huang et al. (2022) assumes the candidate space during testing excludes all positive triples from training set, while traditional filtered setting also excludes validation and test set.

task on multilingual dataset DBP-5L. We also demonstrate the performance gains provided by each component of the proposed method. Further experiments have been planned to extend CKGC-MKD method to 1) include a fine-tuning stage for the low-resource KG in extreme cases and 2) include probabilistic seed alignments predicted by algorithms. We believe the planned works would greatly enhance the generalizability of our proposed model to tackle more real-world datasets.

References

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. *Advances in Neural Information Processing Systems* 26 (2013). `https://proceedings.neurips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html`

Xuelu Chen, Muhao Chen, Changjun Fan, Ankith Uppunda, Yizhou Sun, and Carlo Zaniolo. 2020. Multilingual Knowledge Graph Completion via Ensemble Knowledge Transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 3227–3238. `https://doi.org/10.18653/v1/2020.findings-emnlp.290`

Zijie Huang, Zheng Li, Haoming Jiang, Tianyu Cao, Hanqing Lu, Bing Yin, Karthik Subbian, Yizhou Sun, and Wei Wang. 2022. Multilingual Knowledge Graph Completion with Self-Supervised Adaptive Graph Alignment. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 474–485. `https://doi.org/10.18653/v1/2022.acl-long.36`

Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv:1609.02907 [cs, stat]* (Feb. 2017). `http://arxiv.org/abs/1609.02907` arXiv: 1609.02907.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*. `https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9571`

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A Three-Way Model for Collective Learning on Multi-Relational Data. (2011), 8.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The Graph Neural Network Model. *IEEE Transactions on Neural Networks* 20, 1 (2009), 61–80. `https://doi.org/10.1109/TNN.2008.2005605` Conference Name: IEEE Transactions on Neural Networks.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *The Semantic Web (Lecture Notes in Computer Science)*, Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam (Eds.). Springer International Publishing, Cham, 593–607. `https://doi.org/10.1007/978-3-319-93417-4_38`

Harkanwar Singh, Prachi Jain, Mausam Mausam, and Soumen Chakrabarti. 2021. *Multilingual Knowledge Graph Completion with Joint Relation and Entity Alignment*.

Raphaël Sourty, Jose G. Moreno, François-Paul Servant, and Lynda Tamine-Lechani. 2020. Knowledge Base Embedding By Cooperative Knowledge Distillation. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 5579–5590. `https://doi.org/10.18653/v1/2020.coling-main.489`

Zequn Sun, Muhao Chen, and Wei Hu. 2021. Knowing the No-match: Entity Alignment with Dangling Cases. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (Online, 2021-08). Association for Computational Linguistics, 3582–3593. `https://doi.org/10.18653/v1/2021.acl-long.278`

Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. 2020. A Benchmarking Study of Embedding-based Entity Alignment for Knowledge Graphs. *Proceedings of the VLDB Endowment* 13, 12 (Aug. 2020), 2326–2340. `https://doi.org/10.14778/3407790.3407828` arXiv: 2003.07743.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2020. COMPOSITION-BASED MULTI-RELATIONAL GRAPH CONVOLUTIONAL NETWORKS. (2020), 16.

Huijuan Wang, Shuangyin Li, and Rong Pan. 2021. An Adversarial Transfer Network for Knowledge Representation Learning. In *Proceedings of the Web Conference 2021*. ACM, Ljubljana Slovenia, 1749–1760. `https://doi.org/10.1145/3442381.3450064`

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*. `https://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8531`

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. EMBEDDING ENTITIES AND RELATIONS FOR LEARN- ING AND INFERENCE IN KNOWLEDGE BASES. (2015), 13.

Donghan Yu, Yiming Yang, Ruohong Zhang, and Yuexin Wu. 2021. Knowledge Embedding Based Graph Convolutional Network. In *Proceedings of the Web Conference 2021 (WWW '21)*. Association for Computing Machinery, New York, NY, USA, 1619–1628. `https://doi.org/10.1145/3442381.3449925`

A Related work

A.1 Knowledge graph embeddings

The research on knowledge graph embeddings has gained significant attention in the recent years. The goal of this task is to encode entities and relations of a KG into low-dimensional vectors. Traditional translation-based methods like TransE (Bordes et al., 2013), TransH (Wang et al., 2014), TransR (Lin et al., 2015), as well as the semantic matching models like RESCAL (Nickel et al., 2011) and DistMult (Yang et al., 2015), all achieved promising results on the KGC task. Another stream of recent works (Schlichtkrull et al., 2018; Vashishth et al., 2020; Yu et al., 2021) all employed the graph structure to propagate information between adjacent entities and encode them into embeddings. Specifically, variants of GCN model are used as encoder to embed entity and relations into vectors, and traditional knowledge graph embedding methods like TransE are then used as decoders for KGC task.

A.2 KGC across multiple Knowledge graphs

Comparing to KGC on single KG, KGC across multiple KGs is a relatively under-explored area. Wang et al. (2021) proposed ATransN, an adversarial embedding transfer network which aims to facilitate the knowledge transfer from a pre-trained embedding of a teacher KG to a student KG with a set of seed alignments. Chen et al. (2020) was the first to propose multilingual KGC problem setting and tackled the problem from a model ensemble perspective. On the same multilingual problem setting, Singh et al. (2021) proposed AlignKGC to jointly trains KGC, entity alignment and relation alignment tasks. Huang et al. (2022) proposed SS-AGA, which models seed alignment as edges to fuse multiple knowledge graphs, while using a generator model to dynamically capture more potential alignments between entities and iteratively add more edges to the graph. Additionally, Sourty et al. (2020) proposed KD-MKB, which assumes the existence of both shared relations and shared entities across individual KGs, and therefore tackles multi-KG completion task from a knowledge distillation perspective.

B Ablation study

In table 2, we report the results of our ablation studies to analyze how each of the components in the proposed method affect the results. We choose to report the ablation study results on the multilingual

Table 2: Ablation study results on DBP-5L and D-W-15K-LP.

		DBP-5L					D-W-15K-LP	
	Metric	EL	JA	ES	FR	EN	DBpedia	Wikidata
KGC-I	H@1	22.0	23.1	19.0	21.1	17.2	29.9	25.5
	H@10	49.1	44.7	44.1	45.3	45.5	54.2	49.2
	MRR	31.3	30.7	27.9	29.5	26.9	38.4	34.2
KGC-C	H@1	31.6	29.6	24.3	25.9	19.2	29.9	26.8
	H@10	66.3	61.4	54.7	58.1	50.3	55.3	50.4
	MRR	43.3	40.3	34.6	36.7	29.5	38.8	35.4
KGC-A	H@1	32.4	30.9	25.6	27.0	20.3	30.7	27.5
	H@10	67.6	62.7	56.9	**59.5**	52.5	55.6	50.8
	MRR	44.1	41.6	36.3	37.9	31.0	39.3	35.8
KGC-I-D	H@1	32.3	30.6	24.5	25.5	20.3	31.2	29.2
	H@10	63.0	57.4	52.8	52.7	49.1	54.9	49.8
	MRR	43.0	40.1	34.4	35.1	30.2	39.4	36.5
KGC-A-D	H@1	35.7	33.3	27.7	28.4	22.3	**31.7**	29.2
	H@10	67.6	63.3	**57.9**	58.7	53.2	**55.8**	**50.7**
	MRR	46.8	43.6	38.0	38.8	**32.7**	**40.0**	36.9
CKGC-MKD	H@1	**37.5**	**34.8**	**28.2**	**29.3**	**22.4**	**31.7**	**29.5**
	H@10	**68.6**	**64.1**	57.8	58.3	52.7	**55.8**	50.6
	MRR	**48.3**	**45.0**	**38.5**	**39.5**	**32.8**	**40.0**	**37.1**

DBP-5L dataset as well as a monolingual self-generated D-W-15K-LP dataset. D-W-15K-LP is a dataset generated from the entity alignment benchmarking datasets D-W-15K (Sun et al., 2020). To mimic a more real-life setting, we employed the sampling strategy proposed in the work of Sun et al. (2021), to create dangling entities (entities without alignment across KGs) in the KGs. In the sampling process, by removing part of the alignments from KGs, triples containing removed entities are also excluded. This results in a more sparse dataset with dangling entities in each individual KG. In addition to the KGC-I and the KGC-A models reported in the section 3, we additionally report the performance of several ablation models: KGC-C refers to the ablation model trained on fused KG without augmented message propagation, KGC-I-D and KGC-A-D respectively represent the ablation models with mutual distillation enabled for KGC-I and KGC-A. Therefore, the reported CKGC-MKD is the ensemble results of KGC-A-D and KGC-I-D. For a more complete and universal comparison, in the ablation study we use the traditional "link prediction" task that includes both head prediction and tail prediction with the traditional filtered setting used in the works of Wang et al. (2014) and Yang et al. (2015).

On both datasets we can observe a clear margin that KGC-A model created over the KGC-C model, which verifies the effectiveness of augmented message propagation. Additionally, on both datasets the distillation enabled KGC-I-D and KGC-A-D models have shown superior performance in almost all metrics over the KGC-I and KGC-A model respectively. This has shown that the mutual knowledge distillation process is beneficial for both individual models and the fused model. Lastly, CKGC-MKD achieves the best performances in most of the metrics, which verifies the gains provided by the ensemble technique. An interesting observation is that even after the mutual knowledge distillation, the KGC-I-D models still performs slightly worse than the fused model KGC-A-D; and the difference in performance also varies across different KG. One of the possible reason behind this observation is that we used a constant α for all KGs in one dataset to control the trade-off between knowledge distillation loss and knowledge representation loss. Limited by the hardware resources, we did not explore possibilities of assigning different α for each KG, and decided to leave that for the future work that possibly explores a fine-tuning stage of the model to better reconcile the difference and imbalance of resource in each of the KG.

Algorithm 1 Pseudocode of the training process of CKGC-MKD.

▷ Stage 1: trains each model M_i and M_f with only knowledge representation loss.
for $i \in 1..m+1$ **do**
  while M_i not converged **do**
    $L^i \leftarrow T_i$
    $M_i \leftarrow$ Update w.r.t L^i
  end while
end for
▷ Stage 2: trains each model M_i and M_f with knowledge representation loss and knowledge distillation loss.
while not converged **do**
  $batch_f \leftarrow$ sample from triple set T_f
  $L_T^f \leftarrow$ calculate loss of $batch_f$ base on equation 4
  for $i \in 1..m$ **do**
    $batch_i \leftarrow$ sample from triple set T_i
    $L_T^i \leftarrow$ calculate loss of $batch_i$ base on equation 4
    $L_D^i, L_D^f \leftarrow$ calculate distillation losses between M_i and M_f on $batch_i$ base on equation 5 with top-k sampling to select candidates space of distillation
    $L^i \leftarrow L_T^i + \alpha L_D^i$
    $L^f \leftarrow L_T^f + \alpha L_D^f$
    $M_i \leftarrow$ Update w.r.t L^i
  end for
  $M_f \leftarrow$ Update w.r.t L^f
end while

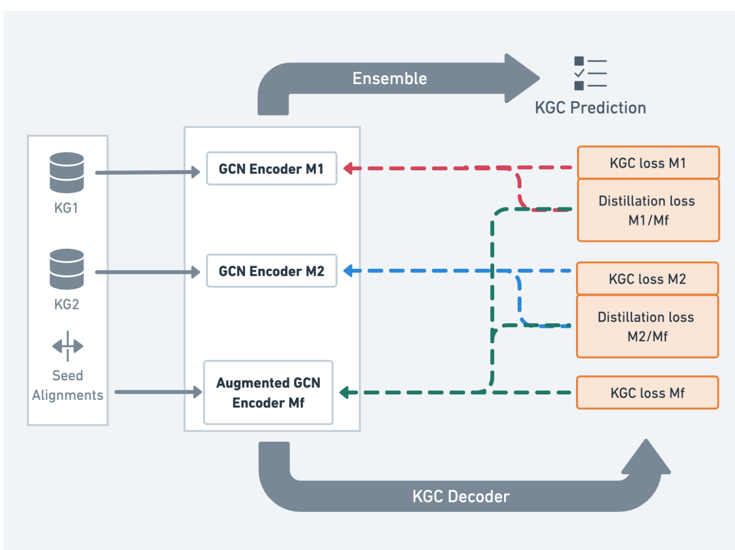

Figure 1: An illustrative figure of the proposed CKGC-MKD with 2 KGs.