LeKAN: Extracting Long-tail Relations via Layer-Enhanced Knowledge-Aggregation Networks

Xiaokai Liu[1,3], Feng Zhao[1,2](✉), Xiangyu Gui[1,2], and Hai Jin[1,2]

[1] National Engineering Research Center for Big Data Technology and System, Services Computing Technology and System Lab, Cluster and Grid Computing Lab, Wuhan, China

[2] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China

{zhaof,guixy,hjin}@hust.edu.cn

[3] School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan, China

liuxk@hust.edu.cn

Abstract. Long-tailed relation extraction is a crucial task in the information extraction field for extracting the long-tailed, imbalanced relation between two annotated entities based on related context. Although many works have been devoted to distinguishing valid instances from noisy data and have achieved promising performance, such studies still have critical defects: works based on nonhierarchical relations ignore the correlations among the relations, and those based on hierarchical relations neglect the hierarchy of the relation structure, which is unbalanced and causes difficulty in extracting data-poor classes. In this paper, a novel layer-enhanced knowledge aggregation network, named *LeKAN*, is presented to classify the relations between two annotated entities from text, especially long-tailed relations, which are very common in various corpora. Inspired by the election mechanism, we aggregate the ancestors of long-tailed relation classes into new relation representations to prevent the long-tailed relations from being ignored. Specifically, we use GraphSAGE to learn the relational knowledge from an existing knowledge graph via class embedding. Moreover, we aggregate the acquired relational knowledge into the *LeKAN* by layer-enhanced knowledge-aggregating attention mechanism. Comprehensive experimental results demonstrate that the new method yields considerable improvement over other relation extraction methods on a large-scale benchmark dataset with a long-tailed distribution.

Keywords: Natural language processing · Information extraction · Long-tailed relation extraction · Knowledge-aggregation network

1 Introduction

Relation extraction (RE) is an essential task in the NLP field for extracting the relation between two annotated entities based on the context, especially

A. Bhattacharya et al. (Eds.): DASFAA 2022, LNCS 13245, pp. 122–136, 2022.
https://doi.org/10.1007/978-3-031-00123-9_9

long-tailed, imbalanced relations, which are very common in real-world settings. Long-tailed relations cannot be ignored because they contain rich semantic information. However, it is extremely difficult to extract long-tailed relation classes at the tail of the class distribution because few data is available. There are only a few works which have attempted to dig into the problem of long-tail RE, such as the explanation-based approach [1] and the approach utilizing external knowledge (logic rules) [2]. These works have conducted beneficial studies on the extraction of long-tail relations.

As an emerging technology and an effective solution to help improve the ability of machines to understand the human world, *knowledge graphs* (KGs) can provide higher-quality support for quantitative information retrieval, question answering, recommender systems, search engines, and other natural language processing applications [3,4]. However, the construction of a large-scale knowledge graph system containing massive amounts of knowledge relies on large-scale structured training data. RE, with the purpose of extracting the relation between two named entities based on the given context, is a fundamental task in building large-scale KGs. It is also a crucial technique in automatic KG construction. Using RE, we can accumulatively extract new relation facts to expand the built KG. However, RE model performances quickly degrade when extracting long-tailed relations because many long-tailed relations suffer from data insufficiency. These difficulties make the extraction of long-tailed relations a very difficult problem.

Long-tail relations cannot be ignored because they contain rich semantic correlations. Moreover, long-tailed, imbalanced data is very common in reality. In this work, we followed previous work to employ a widely used corpus, the New York Times (NYT-10) dataset[1] [5], to verify the advantages of our method in long tailed relation extraction. To have comprehensive understanding of the long-tailed distribution in this dataset, we analyze the distribution of the relation classes in NYT-10, as shown in Fig. 1. In this figure, long-tailed relation instances account for less than 4% of the total data, while short-headed instances account for more than 96% of the data. Furthermore, the short-headed relation classes account for less than 20% of the dataset, while the long-tailed relation classes account for more than 80%. Therefore, research on long-tailed RE is significant, and methods that can extract long-tailed relations with high accuracy are urgently needed.

The task of extracting relations from long-tailed distribution context is very difficult because few examples are available to train the models, leading to insufficient relation representation and poor classifier learning. Therefore, this situation motivates us to identify methods that can transfer knowledge between relations and alleviate the imbalance inherent in the hierarchical relational structure. To tackle these problems, a layer-enhanced knowledge aggregation network, named *LeKAN*, is proposed. To transfer knowledge between relations, the conventional method considers only the transfer of knowledge between relation instances in the same branch, e.g., $/people/deceased\_person/place\_of\_death$

[1] http://iesl.cs.umass.edu/riedel/ecml/.

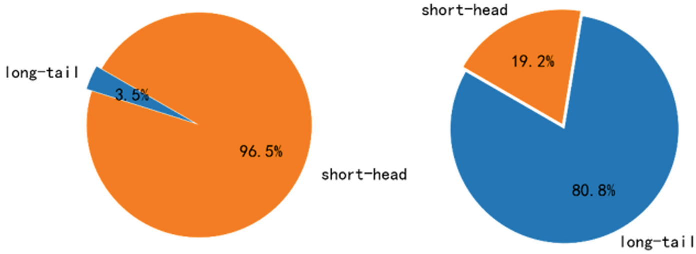


Fig. 1. Proportions of instances and classes of short-headed and long-tailed relations without NA labels in the NYT-10 dataset

and $/people/deceased\_person/place\_of\_birth$, while ignoring the fact that relation instances in different branches may also have similar semantics; e.g., both $/film/film\_festival/location$ and $/broadcast/content/location$ share the base-level relation class $/*/*/location$. *LeKAN* can aggregate the relational knowledge between two relations regardless of whether they are in the same branch, and the extraction of head relations provides evidence for the prediction of long-tailed relations. To alleviate the imbalance of the hierarchical relation structure, we propose a tree-based adjustment strategy to build the distributed relational representation. By pruning the long branches and extending the short branches of the network, all relation nodes are held in the same layer. Moreover, GraphSAGE with embedded KG information can sample the relevant information of the 1-step and 2-step neighbor nodes, which helps alleviate the imbalance problem of the hierarchical relation structure. Various baselines experiments were conducted on NYT-10, which demonstrate that the proposed *LeKAN* achieves best results in extracting the long-tailed relation. Furthermore, by leveraging the aggregated rational knowledge in different branches and levels, our proposed model can transfer relational knowledge more efficiently than existing approaches.

The remainder of the paper is organized as follows. In sect. 2, we discuss the latest progress on the long tail problem in various fields, such as relation extraction, computer vision, that can inspire this work. In Sect. 3, we mainly introduce the theory and interpretability of our proposed *LeKAN* method. Then, our experimental results are reported in Sect. 4. Finally, we conclude the our work and briefly introduce the work to be done in the future in Sect. 5.

2 Related Works

Relation extraction is the cornerstone of automatic construction of large-scale KGs. Early relation extraction mainly depends on the supervision model. Quantities of labeled data is required for relation extraction via conventional supervised models [6,7]. Such a process of tagging large-scale raw datasets is extremely time consuming and difficult to perform. Hence, [8] *proposed the use of distant supervision* (DS) to automatically annotate data. However, DS unavoidably introduces the incorrect labeling problem. To address such an issue caused by DS, [5,9] proposed multi-instance learning mechanisms, [10] proposed a sentence-level framework via negative training and [11] achieved promising performance by adopting DS to construct extensive datasets and alleviate the noisy label problem. Recently, [12] proposed a probabilistic approach to improve the DS relation extraction. However, these works ignored the long-tailed problem or failed to improve the effect of long-tailed RE.

The two intuitive solutions to solve the classification problem of long-tailed distribution are resampling [13–15] and reweighing [16,17]. The essence of these methods is to leverage the dataset with given distribution to violently hack the unknown distribution during the process of model training, i.e., to make change of the point weights, strengthen the tail category learning, and offset the long-tailed effect. Moreover, multi-instance learning [18] and transfer learning [19] can be employed to tackle the long-tail relevance problem. These methods have achieved good results in various computer vision tasks.

Only a few works have attempted to solve the problem of long-tailed RE [1,2,20,21]. The studies by [1,2] treated each class in isolation. Such a way of dealing with different classes of relations naturally ignores the rich semantic correlations between the classes, which are equally important. [20] proposed a hierarchical attention scheme for RE and achieved better performance than nonhierarchical schemes. [21] applied transfer knowledge between instances in the vertical direction (same branches) and leveraged implicit and explicit class embedding from Knowledge Graphs and *Graph Convolutional Networks* (GCNs) instead of learning hyper-parameter spaces using the data-driven mechanism, where similar classes may have different hyper parameters; thus, they impeded the generalization of long-tailed relations. These works conducted beneficial explorations into the long-tailed relation extraction.

Previous solutions to address the long-tail problem have mainly focused on entity hierarchies and the transfer of relational knowledge between instances in the vertical direction. Unlike them, our methods leverage GraphSAGE to learn knowledge and transfer knowledge in both vertical and horizontal directions using a relational aggregator. To alleviate the imbalance inherent in hierarchical relation structures, we also propose a method to build a layer-enhanced hierarchical relational tree to ensure that all relational branches have identical heights. Compared with the existing RE methods, our models can leverage relation correlations to better classify the given long-tailed instances by transferring knowledge from their related layers.

3 Methodology

In this section, we introduce the methodology of the layer-enhanced knowledge attention network for RE. First, we start with the relevant definitions of RE.

3.1 Framework

We follow the general definition and notations of the knowledge graph by defining the KG as a set of $\mathcal{G}$. Furthermore, $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{F}\}$. The $\mathcal{F}$ indicates triple fact $(h, r, t) \in \mathcal{F}$, the $\mathcal{E}$ indicates entities predefined in KG, and $\mathcal{R}$ indicates

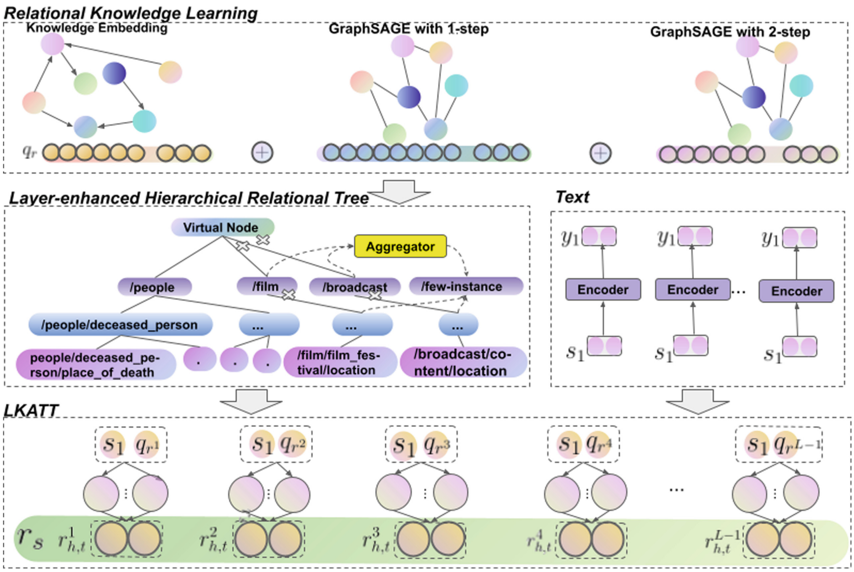


Fig. 2. The architecture of *LeKAN*

relations between such entities. The facts $\mathcal{F}$ indicate that the class of relation $r \in \mathcal{R}$ between two given entities ($h \in \mathcal{E}$ and $t \in \mathcal{E}$) is r. We adopt the multi-instance learning settings and generate multiple entity-pair bags by splitting the instances with identical entity pairs that mention h_i and t_i into the same bags $\mathcal{S}_{h_1,t_1}, \mathcal{S}_{h_2,t_2,...}$. Each instance in entity-pair bags is represented as a word sequence $s = \{w_1, w_2, ...\}$.

In Fig. 2, we demonstrate the overall architecture of the *LeKAN*. There are mainly four parts in *LeKAN* as follows.

Instance Encoder: The instance encoder aims to encode the sentence semantics into a continuous low-dimensional vector. Designated an instance s with the tagged entity pair, we can use the models with neural network architecture to encode it.

Relational Knowledge Learning: Considering the pretrained KG embeddings (e.g., TransE [22]) as nonhierarchical relational knowledge, we use GraphSage to learn hierarchical relational knowledge from the aggregated relational KG. In addition, we combine the GraphSAGE with generic message-passing inference, we can acquire the relational representation for the relation classes. We concatenate the outputs of the GraphSAGE sampling neighbors with different steps and the embeddings learned from knowledge graph to construct the final distributed relational embeddings.

LKATT: Given the hierarchical relation structure of a KG, the relational knowledge aggregator automatically aggregates the parent relations of the long-tailed relation into a new relation. For example, we can aggregate two long-tailed relations under different branches, e.g., /film/film_festival/location and /broadcast/content/location, to a new relation: few_instance_location. Under the guidance of *layer-enhanced knowledge attention* (LKATT), *LeKAN* aims to select the instance with abundant information that exactly matches the relevant relation but to ignore its branch.

3.2 Instance Encoder

Given an instance $s = \{w_1, ..., w_n\}$ containing two entities, we leverage the instance encoder to encode the sentence into a continuous low-dimensional vector. The instance encoder consists of two parts: the embedding layer, which maps the words in the context into vectors, and the encoder layer, which encodes the vectors.

Embedding Layer: To better identify the synaptic and semantic meanings of the sentences. We leverage the neural networks in embedding layer to transform discrete words in specific instance into vector space. Here, we use a pretrained skip-gram model [23] to map each word w_i in the instance to a continuous vector space. Moreover, we adopt position embedding following [11]. Then, we embed the relative distances of every word in the instance from marked entities into two d_p-dimensional continuous vectors. Finally, we gather all input embeddings in the instance and concatenate all of them together. By doing so, we get a sequence of instance embedding, which is ready to be fed into the encoding layer.

Encoding Layer: In encoding layer, we also employ neural networks to encode the outputs of the embedding layer, whose input is a given instance. In this study, we employ vanilla CNNs [11] and PCNNs [24] as the instance encoder.

3.3 Distributed Relational Representation via Transfer Learning

To get distributed relational representations, we need to have pretrained KG embeddings obtained by instanced encoder and define a predefined class relation hierarchy according to the structure of KG. Then, we build the distributed relational representation. First, we use the nonhierarchical relational knowledge from the KGs. Second, we build a layer-enhanced hierarchical relational tree to learn hierarchical relational knowledge. Third, we apply GraphSAGEs with 1-step and 2-step to learn the hierarchical relational knowledge from the layer-enhanced hierarchical relational tree, and obtain a distributed relational representation.

Building a Layer-Enhanced Hierarchical Relational Tree. Given KG $\mathcal{G}$ (e.g., NYT) consisting base-level relations, we extract the set R of it to generate the corresponding layer-enhanced hierarchical relational tree set R^H. The relations in high-level sets (e.g., /location) have the same instances as their child relations (e.g., /people/deceased_person), which indicates that high-level relations are more general and common than low-level relations. The relation hierarchies are separated into tree-structured subgraphs of R^0, which is the set of all relations. The generation of subgraphs can be recursively completed to obtain the relation sets $\{R^0, R^1..., R^i, R^L\}$ and others. Then, we must adjust the hierarchy relation tree to ensure that all leaf nodes have identical heights. We propose two layer-enhanced methods to transform an imbalanced relational tree into a balanced tree: pruning and completion. The pruning method can remove layers from a relation branch, while the completion method can add layers to a relation branch. For long relational branches, we can use the pruning method to reduce their heights; for short relational branches, we can use the completion method to increase their heights. This approach can also prevent overfitting and improve the convergence speed of the network.

Learning Relational Knowledge via GraphSAGE. Because of the missing one-multiple relations in KGs, GraphSAGEs are necessary that they sample 1-step and 2-step neighbors from the hierarchical features. Given the pretrained relation embedding $v_d^{TransE} \in KGs$ via *TransE*, we use the mean aggregator to form a hierarchical representation of the i-th label:

$$h_v^k = \sigma(W^i \cdot Mean(h_v^{k-1} \bigcup h_u^{k-1}, \forall u \in \mathcal{N}(v))) \tag{1}$$

where $W^i \in \mathbb{R}^{q^i}, i = 1, 2, h_u^0 = v_d$. The convolutional aggregator concatenates the parent layer representation h_v^{k-1} of the node with the aggregated neighborhood representation $h_{N(v)}^k$. Finally, we concatenate the pretrained v_i^{TransE} and output vectors $v_i^{GSN_1}, v_i^{GSN_2}$ of the GraphSAGEs to form the hierarchical class embeddings:

$$q_r = v_i^{TransE} || v_i^{GSN_1} || v_i^{GSN_2} \tag{2}$$

where $q_r \in \mathbb{R}^{d+q^1+q^2}$.

3.4 LKATT

Conventional hierarchical RE models treat the top-level relation nodes as independent nodes, which hinders the transfer of knowledge among the base-level relational nodes of different branches and prevents the long-tailed nodes from being selected. We design a relational aggregator to solve these problems. The relational aggregator is guided by the following principles: 1) If two base-level relational nodes are semantically similar, their top-level relational nodes are aggregated. 2) Even if the basic relation nodes of two rare instances have different semantics, their top-level relations can be aggregated. Experiments show that the aggregation of relation nodes can enhance the performance of classifying long-tailed classes, and the decoupling of top-level relation nodes likely has potential effects.

In general, the output layer of the neural network will learn parameters of the specific label optimized by the given loss function. Because, the parameter space of different classes is different, it naturally leads to the fact that long-tail relations can be exposed to only a few training examples during training. Instead, our approach considers more correlations of the long-tailed relations by making the ancestor nodes of semantically similar relation nodes share parameters and concatenating the sentence to the corresponding class embeddings.

First, we acquire the instance embeddings $\{s_1, s_2, ..., s_m\}$ using the instance encoder with the entity pair (h, t) and the corresponding bag of instances $S_{h,t} = \{s_1, s_2, ..., s_m\}$. Second, we split the class embeddings into different clusters according to their types (i.e., according to their levels in the layer-enhanced hierarchical relational tree), e.g., $q_r^i, i \in \{0, 1, ..., L\}$. Third, we aggregate the semantically similar relation nodes and adopt $q_r^i, i \neq N$ (we assign an another node N as root node in the tree) as a layer-enhanced attention query vector. Finally, we use the LKATT mechanism to process the vector to get its relation representation $r_{h,t}$. For each relation r, we can build the corresponding hierarchical chain of latent relations $(r^0, ..., r^{(N-1)})$ using a layer-enhanced hierarchical relational tree, where $r^{(i-1)}$ is the subrelation of r^i. We can calculate the attention weight for $s_i \in \mathcal{S}_{h,t} = \{s_1, s_2, ..., s_m\}$ as follows:

$$e_k^i = tanh(W_s[s_k; q_r^i]) + b_s \tag{3}$$

$$a_k^i = \frac{\exp{(e_k^i)}}{\sum_{j=1}^{m} \exp{(e_j)}} \tag{4}$$

where $[x_1; x_2]$ denotes the vertical concatenation of x_1 and x_2, W_s is the weight matrix, and b_s is the bias. The converged nodes share parameters. Then, we can compute the attention scores on each layer of the layer-enhanced hierarchical relational tree to acquire the relational representations.

$$r_{h,t}^i = ATT(q_{r^i}, s_1, s_2, ..., s_m) \tag{5}$$

The global representation is defined as follows:

$$r_{h,r} = Concat(r_{h,t}^0, ..., r_{h,t}^{L-1}) \tag{6}$$

The conditional probability is computed by the global representation $r_{h,t}$: $\mathcal{P}(r|h, t, \mathcal{S}_{h,t})$:

$$\mathcal{P}(r|h, t, \mathcal{S}_{h,t}) = \frac{\exp(o_r)}{\sum_{\hat{r} \in \mathcal{R}} \exp{(o_{\hat{r}})}} \tag{7}$$

where o contains the scores of all relations. And o is calculated via a linear layer:

$$o = Ar_{h,t} \tag{8}$$

where A is the discriminative matrix.

4 Experiments

In this section, we evaluate all models using the proposed evaluation scheme. This method evaluates the models by comparing the relational facts found in the context with those in a large-scale KG, such as *DBpedia*, and adopts an approximate accuracy measure except the manual evaluation. To show the advantages of our methods, we plot the precision-recall curves for all methods for the evaluation. In order to validate whether the effect of our model for the long-tailed RE is superior to other proposed methods, we follow the same evaluation criteria as before [20,21] by reporting the Precision@N. We report the evaluation results in Figure 3 and Figure 4. The dataset and baseline code are from GitHub[2].

4.1 Experimental Setting

Datasets. We evaluate the performance of our method on the most commonly used long-tail RE dataset in recent long-tail RE work: the NYT-10 dataset [21, 25–28]. There are 52 common classes and a NA class in it. The NA relation denotes that the relation between the given instances is not labeled. The dataset contains rich semantic information, which has been split into a training set with 522611 sentences and a testing set with 17448 sentences. There are 281270 entity pairs and 18252 relational facts in the training set and 96678 entity pairs and 1950 relational facts in the testing set. We follow the convention of truncating sentences that contain more than 120 words into 120 words for the dataset.

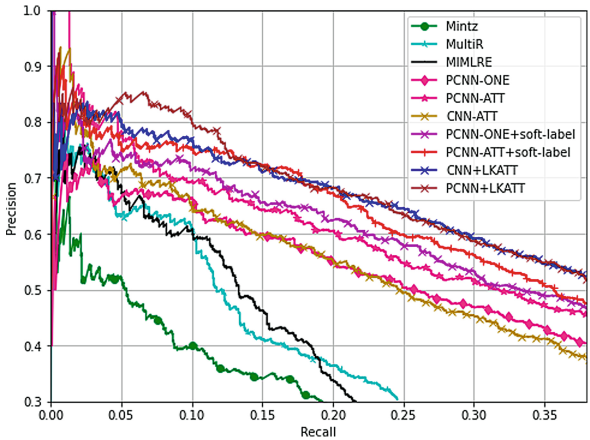


Fig. 3. P-R curves for various models

Comparison Models. For the baseline model comparison, we utilize both neural network models and feature-based models. We report the evaluation results of the neural networks with methods based on various attention schemes: **+LKATT** is our layer-enhanced knowledge-aggregating attention method; **+ONE** is a typical multi instance learning based neural model [24]. The **soft label** is the model with attention schemes using the soft-labeling method to alleviate the effects of the noise problem [27]. In addition, we compare our model

[2] https://github.com/thunlp/OpenNRE

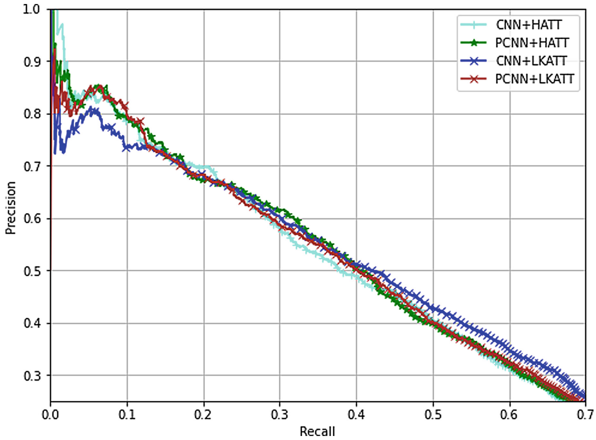


Fig. 4. P-R curves for various models with attention mechanism

with various feature-based models, including **MIML**, **MultiR** [9], and **Mintz** [8] [29]. To effectively evaluate the effect of our method on the long-tailed RE task, we also compare it with **HATT** [20] and **KATT** [21].

Hyperparameter Settings and Reproducibility. In order to prove that our model is superior to other baseline models and fairly compare its performance with that of baseline methods, we keep almost all experimental parameters identical to the previous work and pretrain the sentence encoder of the neural networks [25]. During the training process, a dropout layer is adopted before the output layer to prevent overfitting.

4.2 Overview of the Evaluation Results

As shown in Figs. 3 and 4, our method using a novel denoising scheme and additional auxiliary information achieved the best performance. The results also demonstrate that *LeKAN* can leverage the rich correlation between relations to improve its RE performance. We anticipate to enhance the performance of our model by integrating some novel mechanisms such as meta-learning.

To prove the advantages and performance improvement of the proposed methods for long-tail relation RE, we follow the convention to extract subsets of the test dataset, where the training instances of all relations are less than 100 and 200. We use the Hits@K metric to evaluate the long-tail RE. Then, the RE models will recommend the relations in the first K candidate classes for each entity pair. Since extracting long-tail relations is extremely difficult in existing models, we choose K from the set {10, 15, 20}. We report the macroaverage accuracy of Hits@K for all subsets. The results shown in Table 1 demonstrate that our new method is outperforms the attention mechanism based methods, even the most sophisticated HATT and KATT. Although our LKATT method achieves better results than the ordinary ATT, HATT, and KATT methods on long-tail relations, the results show that the current achievements in long-tail RE remain unsatisfactory. Thus, the RE model may require additional information. We will explore further in our future work.

Table 1. Accuracies (%) in terms of Hits@K on long-tail classes

Number of training instances		<100			<200		
Hits@K(Macro)		*10*	*15*	*20*	*10*	*15*	*20*
CNN	+ATT	<5.0	<5.0	18.5	<5.0	16.2	33.3
	+HATT	5.6	31.5	57.4	22.7	43.9	65.1
	+KATT	9.1	41.3	58.5	23.3	44.1	65.4
	+LKATT	**16.7**	**55.6**	**77.7**	**31.8**	**63.6**	**81.8**
PCNN	+ATT	<5.0	7.4	40.7	17.2	24.2	51.5
	+HATT	29.6	51.9	61.1	41.4	60.6	68.2
	+KATT	35.3	62.4	65.1	43.2	61.3	69.2
	+LKATT	29.6	61.1	**77.8**	42.4	**68.2**	**81.8**

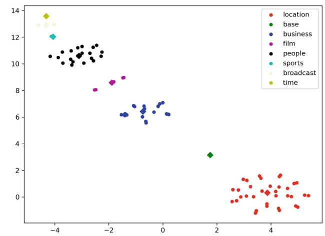
4.3 Ablation Study

To have a comprehensive understanding of the contributions and impact of different techniques in the proposed method, we design ablation tests. We demonstrate the evaluation results of ablation study in Table 2. **+LKATT** is the proposed method; **w/o aggregation** is the method where node aggregation is not implemented; **w/o KG** is the method where the nodes are initialized with random embeddings, so it is natural that there is no relational knowledge obtained from KGs; and **w/o GraphSage** is the method without GraphSage, which denotes no structured relational knowledge. By analyzing the results in Table 2, we can draw the conclusion that the performance of our method to extract long-tailed relations is slightly degraded without KG, and the performance is significantly degraded after node aggregation or GraphSage is removed. This degradation is reasonable because GraphSAGEs consider the distances between neighbors, and node aggregation can prevent relation classes with few examples in the training set from being ignored.

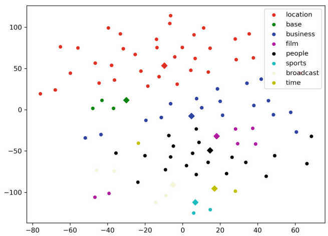
Table 2. Accuracies (%) in terms of Hits@K on relations with fewer than 100/200 training instances

Number of training instances	<100			<200		
Hits@K(Macro)	*10*	*15*	*20*	*10*	*15*	*20*
+LKATT	**29.6**	**61.1**	**78.8**	**42.4**	**68.2**	**81.8**
w/o/ hier	16.7	44.4	44.4	31.8	54.5	54.5
w/o Aggregation	5.6	44.4	50.0	22.7	54.5	59.1
w/o/ KG	24.1	33.3	72.2	37.9	45.6	77.3
w/o/ GraphSage	18.5	44.4	72.2	33.3	54.5	77.3

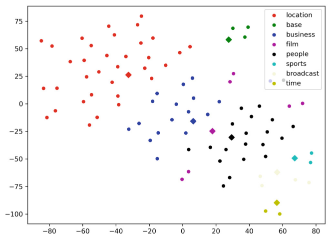
4.4 Visualization of Class Embeddings



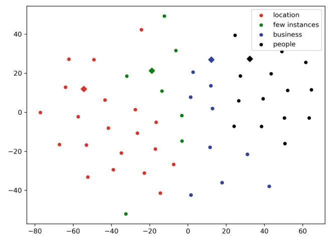
(a) + KATT



(b) + GraphSAGE



(c) + KG & GraphSAGE



(d) + LKATT

Fig. 5. Dimension reduction visualization of relation class embedding

Here, we demonstrate the rationality of our class embedding work through a visualization tool t-SNE [30]. This visualization work on relation embedding deeply shows how KG and GraphSAGE embeddings positively affect the extraction of long tail relations. In Fig. 5, the square points represent the top-level relations of the relation clusters. Figure 5(a) demonstrates that the GCN combines relations that are under the same branch and ignores semantically similar relations on different branches; Fig. 5(b) and Fig. 5(c) show that GraphSAGE can help with knowledge transfer between semantically similar long-tailed relations by aggregating the corresponding knowledge. However, if there is no KG, outliers will occur; Fig. 5(d) shows that the long-tailed relation can be emphasized by aggregating the ancestral relation nodes with few instances to a new relation, which helps to aggregate the ancestral nodes with fewer instances to prevent the corresponding base-level relation from being ignored. However, when we embed the features of ancestral relation nodes into a high-dimensional continuous vector space, the classification of long-tailed relations relies more on the representations of the base-level relations, which is a problem. In the near future, we will tackle this issue by integrating more information, for instance, relation information, or by decoupling the relations with fewer semantic correlations than other relations from their branches.

5 Conclusion

We propose a novel KG- and GraphSAGE-based layer-enhanced knowledge aggregation network to identify the classes of relations between two given entities from a corpus with imbalanced class distribution. This method leverages the relational knowledge from relations at the head of their distribution and uses semantically similar relational instances in different branches to boost the performance of the low-resource RE. Compared to previous works, the new method achieves significant improvements according to evaluations on a large-scale RE dataset. Although we have made a breakthrough in long tail relation extraction, there are still many problems waiting to be solved in the field of construction of the knowledge graph and information extraction. Be aware of these facts, we decided to conduct exploration in the following areas of long-tailed information extraction.: (1) We will evaluate the effect of GraphSAGE on RE tasks across knowledge graphs. (2) We will explore the effect of a more complex short-headed relation decoupling and long-tailed relation aggregation scheme on RE tasks.

Acknowledgment. This work was supported in part by National Key R&D Program of China under Grants No. 2018YFB1404302, National Natural Science Foundation of China under Grants No.62072203.

References

1. Gui, Y., Liu, Q., Zhu, M., Gao, Z.: Exploring long tail data in distantly supervised relation extraction. In: Proceedings of 2016 Natural Language Understanding and Intelligent Applications, pp. 514–522 (2016)
2. Lei, K., Chen, D., et al.: Cooperative denoising for distantly supervised relation extraction. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 426–436 (2018)
3. Huang, Y., Zhao, F., Gui, X., Jin, H.: Path-enhanced explainable recommendation with knowledge graphs. World Wide Web **24**(5), 1769–1789 (2021). https://doi.org/10.1007/s11280-021-00912-4
4. Song, S., Huang, Y., Lu, H.: Rumor detection on social media with out-in-degree graph convolutional networks. In: Proceedings of the 2021 IEEE Conference on Systems, Man, and Cybernetics, pp. 2395–2400 (2021)
5. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases: Part III, pp. 148–163 (2010)
6. Zelenko, D., Aone, C., Richardella, A.: Kernel methods for relation extraction. J. Mach. Learn. Res. **3**(Feb), 1083–1106 (2003)
7. Zhou, G., Su, J., Zhang, J., Zhang, M.: Exploring various knowledge in relation extraction. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, pp. 427–434 (2005)
8. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pp. 1003–1011 (2009)

9. Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., Weld, D.S.: Knowledge-based weak supervision for information extraction of overlapping relations. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 541–550 (2011)
10. Ma, R., Gui, T., Li, L., Zhang, Q., Zhou, Y., Huang, X.: Sent: sentence-level distant relation extraction via negative training. arXiv preprint arXiv:2106.11566 (2021)
11. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers, pp. 2335–2344 (2014)
12. Christopoulou, F., Miwa, M., Ananiadou, S.: Distantly supervised relation extraction with sentence reconstruction and knowledge base priors. arXiv preprint arXiv:2104.08225 (2021)
13. Kang, B., et al.: Decoupling representation and classifier for long-tailed recognition. arXiv preprint arXiv:1910.09217 (2019)
14. Zhou, B., Cui, Q., Wei, X.S., Chen, Z.M.: BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In: Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition, pp. 9719–9728 (2020)
15. Wang, Y., Gan, W., Yang, J., Wu, W., Yan, J.: Dynamic curriculum learning for imbalanced data classification. In: Proceedings of the 2019 IEEE International Conference on Computer Vision, pp. 5017–5026 (2019)
16. Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S.: Class-balanced loss based on effective number of samples. In: Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, pp. 9268–9277 (2019)
17. Jamal, M.A., Brown, M., Yang, M.H., Wang, L., Gong, B.: Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In: Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition, pp. 7610–7619 (2020)
18. Jiang, X., Wang, Q., Li, P., Wang, B.: Relation extraction with multi-instance multi-label convolutional neural networks. In: Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers, pp. 1471–1480 (2016)
19. Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., He, Q.: A comprehensive survey on transfer learning. Proc. IEEE **109**(1), 43–76 (2020)
20. Han, X., Yu, P., Liu, Z., Sun, M., Li, P.: Hierarchical relation extraction with coarse-to-fine grained attention. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2236–2245 (2018)
21. Zhang, N., et al.: Long-tail relation extraction via knowledge graph embeddings and graph convolution networks. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 3016–3025 (2019)
22. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, Vol. 2, pp. 2787–2795 (2013)
23. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
24. Zeng, D., Liu, K., Chen, Y., Zhao, J.: Distant supervision for relation extraction via piecewise convolutional neural networks. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1753–1762 (2015)

25. Lin, Y., Shen, S., Liu, Z., Luan, H., Sun, M.: Neural relation extraction with selective attention over instances. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, (Volume 1: Long Papers), pp. 2124–2133 (2016)
26. Han, X., Liu, Z., Sun, M.: Neural knowledge acquisition via mutual attention between knowledge graph and text. In: Proceedings of the 2018 AAAI Conference on Artificial Intelligence. vol. 32(1), pp. 4832–4839 (2018)
27. Liu, T., Wang, K., Chang, B., Sui, Z.: A soft-label method for noise-tolerant distantly supervised relation extraction. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 1790–1795 (2017)
28. Feng, J., Huang, M., Zhao, L., Yang, Y., Zhu, X.: Reinforcement learning for relation classification from noisy data. In: Proceedings of the 2018 AAAI Conference on Artificial Intelligence, vol. 32, no. 1, pp. 5779–5786 (2018)
29. Surdeanu, M., Tibshirani, J., Nallapati, R., Manning, C.D.: Multi-instance multi-label learning for relation extraction. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 455–465 (2012)
30. Laurens, V.D.M., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(2605), 2579–2605 (2008)