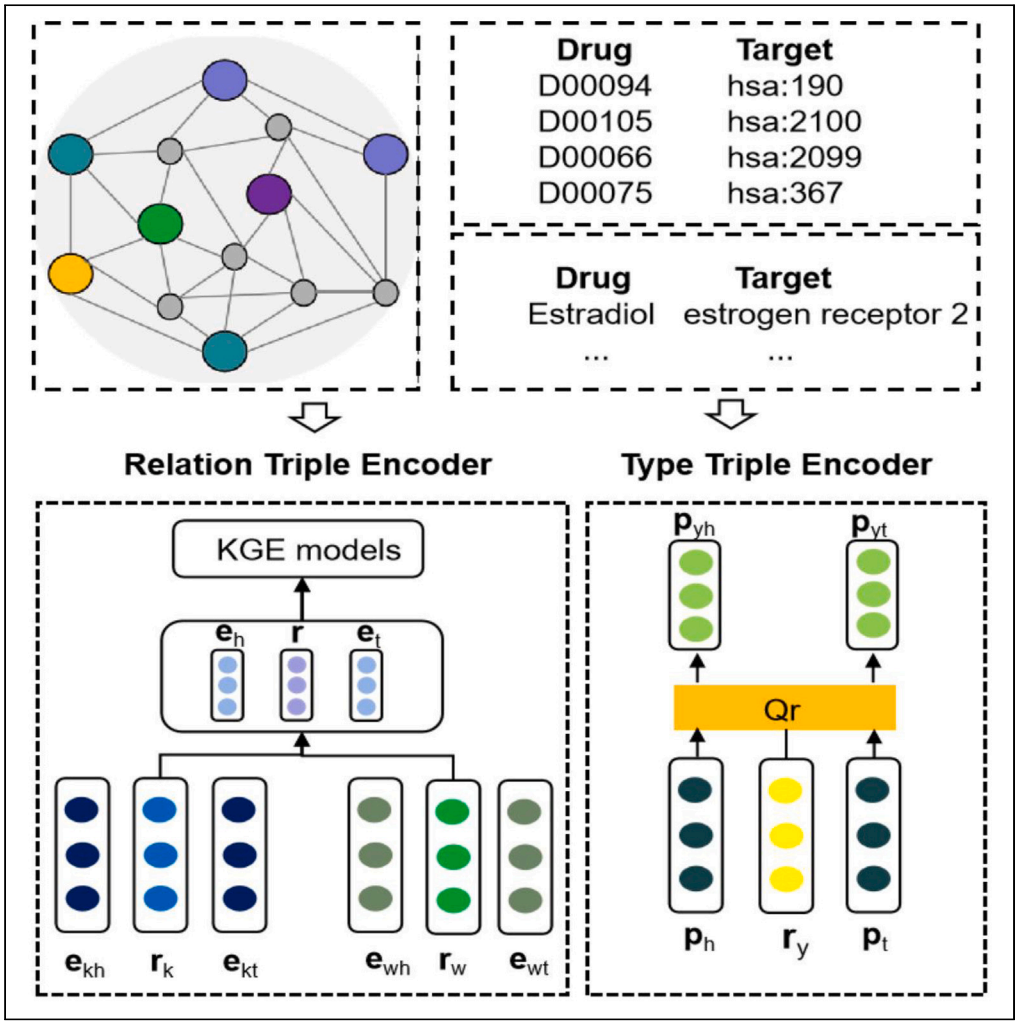
iScience

CellPress
OPEN ACCESS

Article

Drug–target interaction prediction using knowledge graph embedding



Nan Li, Zhihao Yang, Jian Wang, Hongfei Lin

yangzh@dlut.edu.cn

Highlights

Developed a knowledge graph embedding method for drug-target interaction prediction

Modeled text and type information to enrich the features of entities and relations

A pluggable module that can be incorporated into any knowledge graph embedding model

Li et al., iScience 27, 109393
June 21, 2024 © 2024
https://doi.org/10.1016/j.isci.2024.109393

iScience

CellPress
OPEN ACCESS

Article

Drug–target interaction prediction using knowledge graph embedding

Nan Li,[1] Zhihao Yang,[1,2,*] Jian Wang,[1] and Hongfei Lin[1]

SUMMARY

The prediction of drug-target interactions (DTIs) is a critical phase in the sustainable drug development process, especially when the research focus is to capitalize on the repositioning of existing drugs. Computational approaches to predicting DTIs can provide important insights into drug mechanisms of action. However, current methods for predicting DTIs based on the structural information of the knowledge graph may suffer from the sparseness and incompleteness of the knowledge graph and neglect the latent type information of the knowledge graph. In this paper, we propose TTModel, a knowledge graph embedding model for DTI prediction. By exploiting biomedical text and type information, TTModel can learn latent text semantics and type information to improve the performance of representation learning. Comprehensive experiments on two public datasets demonstrate that our model outperforms the state-of-the-art methods significantly on the task of DTI prediction.

INTRODUCTION

The process of identifying potential beneficial treatment effects or medical uses of a new drug candidate is known as drug discovery. Identifying drug targets is a crucial step in the drug discovery process. Drugs function through interaction with various molecular targets such as proteins. This interaction is called drug-target interaction (DTI).[1] Proteins are one useful group of such targets. Through binding, drugs can either enhance or inhibit functions carried out by proteins and thus affect the disease conditions.[2] However, there is a limited number of experimentally identified and validated DTI pairs. Thus, DTI prediction is an essential task in the early stage evaluation of potential novel drugs and the search for novel uses of existing drugs. Several approaches for predicting DTIs have been proposed so far, such as chemical-genetic[3] and proteomic methods.[4] Nevertheless, due to their reliance on laboratory experiments and physical resources, these approaches can only process a limited number of probable medicines and targets. As a result, computational prediction approaches have attracted a lot of attention recently since they can considerably speed up the assessments of potential DTIs.

Recently, Yamanishi et al. proposed a method to predict drug targets computationally.[4] This method employs a statistical model that predicts drug targets based on a bipartite graph of chemical and genomic data. Sleno L. et al. introduced a method to improve the performance by employing neighbor-based interaction-profile inference for both drugs and targets.[5] Furthermore, Cheng et al. suggested a method for predicting DTIs by combining drug similarity, target similarity, and network-based inference.[6,7] Liu et al. proposed a model to leverage drug-drug and target-target similarity measures to infer potential drug targets.[8] However, these methods only utilize a single measure to model components' similarities. Nascimento A et al. adapted to a linear combination of multiple similarity measures to model the overall similarity between drugs and targets.[9] Olayan et al. proposed an approach which used a multi-phase procedure to predict drug targets from relevant heterogeneous graphs.[1] This approach's idea is to use nonlinear fusion to integrate several similarity indices and random walk features obtained from the input graphs. Despite this model achieving better performance, it needs time-consuming training and prediction procedures as they need to compute the similarity features for each drug and target pair during both training and prediction. In addition, most of these methods have a high false-positive rate, particularly when large DTI datasets are used. SHGCL-DTI[10] is designed to supplement the classical semi-supervised DTI prediction task with an auxiliary graph contrastive learning module. Recently, Mohamed S et al. presented an approach, named TriModel, that uses prior information about drugs and targets to overcome the restrictions described above by approaching the problem as link prediction in knowledge graphs (KGs).[11] This approach has the advantage that they consider the structural information in the KG. However, due to the biomedical KGs being usually spare, noisy, and incomplete, TriModel which only relies on the structure information of the KG may suffer from the sparseness and incompleteness of the KG. For instance, when the number of occurrences of an entity in the KG is less, the model learns less information about this entity, resulting in inaccurate representation of this entity. In addition, the latent entity type information can provide KG with supplemental information, enhancing the model's understanding of entities and triples. However, the majority of existing approaches neglect type information.

To solve the above problems, we propose a method, TTModel, that utilizes biomedical text and type information to enhance the performance of knowledge graph embedding (KGE) model for DTI prediction. Specifically, unstructured text contains rich and complementary

[1]College of Computer Science and Technology, Dalian University of Technology, Dalian, China
[2]Lead contact
*Correspondence: yangzh@dlut.edu.cn
https://doi.org/10.1016/j.isci.2024.109393

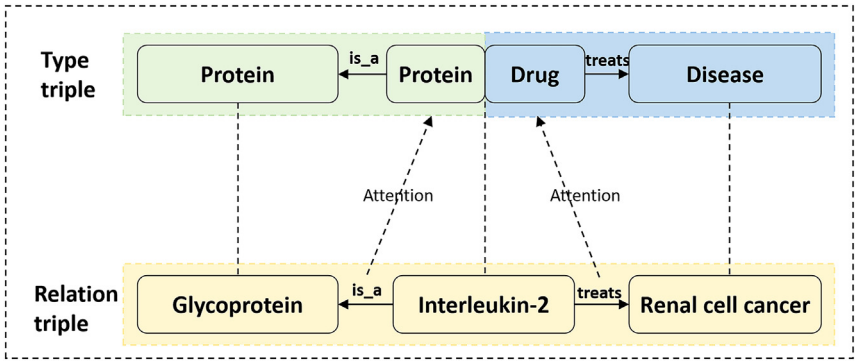


Figure 1. An example of the relation triples and the type triples

information about entities and relationships, providing semantic information for KGE models and effectively alleviating the data sparsity issue in knowledge graphs. Furthermore, the type information of entities can also provide important information for the KGE model, which can be regarded as relatively accurate prior knowledge. Entities with the same type tend to have similar representations.[12,13] Compared with different types of entities, the entity vector representations with the same type are closer to each other in vector space.[14,15] For example, the entities *aspirin* and *genistein* belong to the type of *Drug*, and *Diabetes* belongs to the type of *Disease*. The vector representations of *aspirin* and *genistein* should be closer to each other than to *Diabetes*. However, each entity has multiple latent types, and diverse latent type vector representations should be focused on different specific relations. For example, in Figure 1, given two sentences *"BACKGROUND: Interleukin-2 (IL-2) recently was approved by the Food and Drug Administration for the treatment of renal cell cancer."* and *"Interleukin-2 is a glycoprotein physiologically produced by human lymphocytes which is capable of mediating some still unknown immunologic reactions."*, the triples *treats(Interleukin-2, renal cell cancer)* and *is_a(Interleukin-2, glycoprotein)* are extracted from SemMedDB.[16] The triples in entity level could be extended to triples in the type level. Specifically, in triple *is_a(Interleukin-2, glycoprotein)*, the *Interleukin-2* is considered as a protein. In triple *treats (Interleukin-2, renal cell cancer)*, the *Interleukin-2* is considered as a drug. The entity *Interleukin-2* has different type vector representations when focused on different relations. In addition, entities with the same explicit type often have different fine-grained property information. For example, given two triples *treats* (*SGLT2 inhibitors*, *Diabetes*) and *treats* (*Doxorubicin*, *Tumor*) in the KG, both *SGLT2 inhibitors* and *Doxorubicin* belong to the type *Drug*. However, *SGLT2 inhibitors* is used to treat *Diabetes*, while *Doxorubicin* is used to treat *Tumor*. As a result, *SGLT2 inhibitors* and *Doxorubicin* will have different representations. Therefore, we propose the TTModel, which can capture type information and textual semantic information to enhance the performance of the KGE model. Particularly, this method can effectively alleviate the problem of the long-tail scenario in the biomedical field. As shown in Figures 2 and 3, we train our model to learn the effective representations of drugs and targets in the KG. These representations are then used to score possible drug target pairs. We compare TTModel with other state-of-the-art models using experimental evaluation on standard benchmarks. Experimental results show that TTModel outperforms all baseline methods.

The main contributions of our work can be summarized as follows.

(1) We model biomedical text and type information to enrich the general features of entities and relations and endow the model with the ability to deal with long-tail circumstances.
(2) Our automated text and type representation learning mechanism is a pluggable module that can be easily incorporated into different KGE models.
(3) To verify the performance in the long-tail scenario, we construct a long-tail dataset. The evaluation results demonstrate the superiority of our proposed model over other state-of-the-art methods.

RESULTS

To explore the effectiveness of the TTmodel, we conducted experiments on different datasets to show the model's performance.

Datasets

Yamanishi_08[4] and DrugBank_FDA[1] as the gold standard datasets currently used in the field of DTI prediction,[17,18] which have been widely used in several studies, such as,[1,11] and.[19] Therefore, we utilize these two datasets, i.e., Yamanishi_08[4] and DrugBank_FDA,[1] for the experiments. The Yamanishi_08 dataset is a collection of known DTIs gathered from different sources, including KEGG BRITE,[20] BRENDA,[21] SuperTarget[22] and DrugBank.[23] It consists of four groups of DTIs corresponding to four different target protein classes: (1) enzymes (E), (2) ion-channels (IC), (3) G-protein-coupled receptors (GPCR), and (4) nuclear receptors (NR).[4] The DrugBank_FDA dataset consists of a collection of DTIs of FDA approved that are gathered from DrugBank database https://www.drugbank.ca. We adopt the supplement KG provided by TriModel[11] which extracted from Uniprot,[24] KEGG,[25] InterPro,[26] and DrugBank.[23] In order to facilitate comparative experiments, we utilize

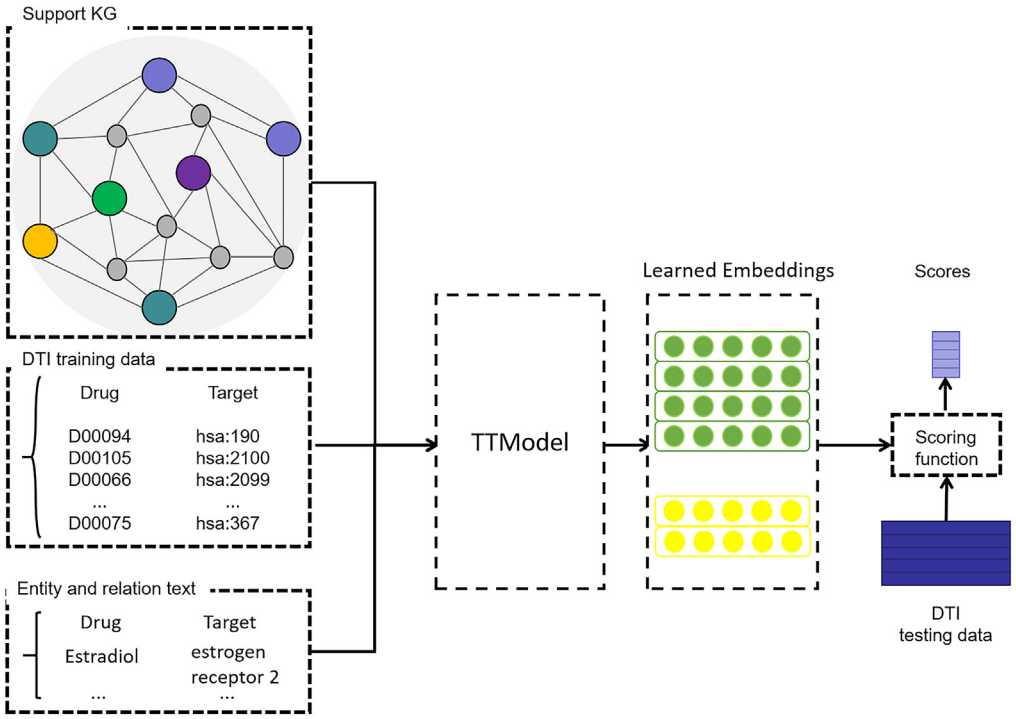


Figure 2. A diagram of the training pipeline of the TTModel model

the common data processing for these datasets following the set of TriModel. The details of the benchmarking datasets are shown in Table 1. In addition, we obtain the entity and relation text data from Uniprot, KEGG, InterPro, DrugBank and PubMed.

Evaluation protocol and experimental setup

Following the same setting of the experiments in TriModel,[11] 10-fold cross-validation (CV) is used to evaluate the model on the Yamanishi_08[27] and DrugBank_FDA[1] datasets. Specifically, for both datasets, we divide all DTI data into 10 splits, and train on the 9 splits. We

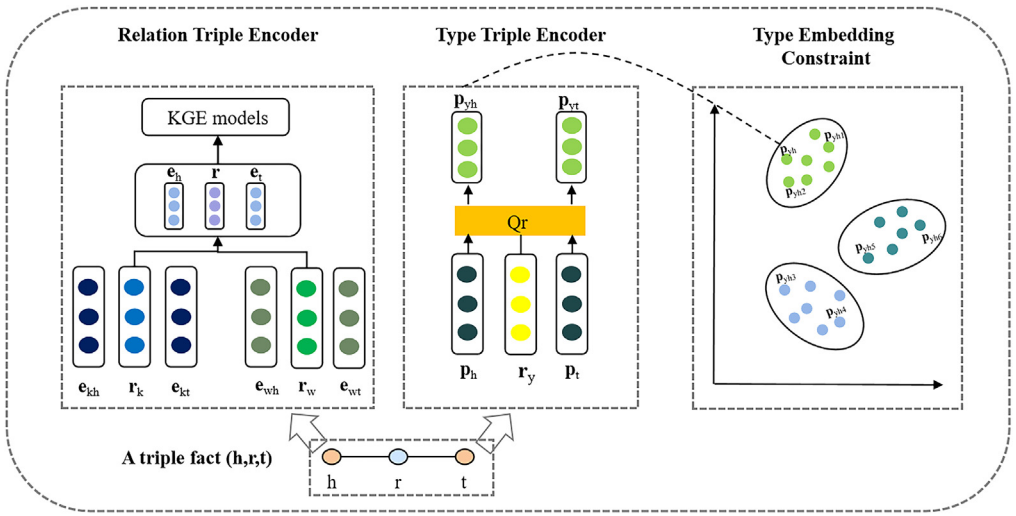


Figure 3. The framework of the TTModel

Given a triple, we randomly initialize the entity and relation representations as input for the relation triple encoder. Simultaneously, we randomly initialize the entity and relation type representations as input for the type triple encoder. Subsequently, the relation triple encoder and type triple encoder are used to generate representations for entities, relations, and types. Text information is only used in the relation triple encoder. In cases where the entity or relation lacks textual information, we solely rely on the structural information from the KG.

Table 1. The details of the benchmarking datasets used in this work

Dataset	Group	Drug	Protein	DTIs
Yamanishi_08	E	445	664	2926
	IC	210	204	1476
	GPCR	223	95	635
	NR	54	26	90
	All	791	989	5127
DrugBank_FDA	–	1482	1408	9881

evaluate the model 10 times on each split, and repeat it 5 times. The results are reported through average results across these runs. We consider the interactions from the known DTIs as positives, and all the other possible combinations between the investigated dataset as negatives. We evaluate the prediction performance using the term of AU-PR[28] and AUC. To determine the AU-PR, we calculate the recall, and precision, based on true positive (TP), false positive (FP), and false negative (FN) values, respectively. AU-PR is calculated based on different precision and recall values at different cut-offs that are used to construct the curve, and then the area under this curve is calculated. The closer the value of AU-PR is to 1, the better the performance is.

$$Recall = \frac{TP}{TP+FN} \quad \text{(Equation 1)}$$

$$Precision = \frac{TP}{TP+FP} \quad \text{(Equation 2)}$$

We use the supporting KGs to perform a grid search to learn the model's best hyper-parameters. In all of experiments, we initialize our model embedding using the uniform random generator and the model is optimized using the adaptive gradient, where the learning rate is among {0.01, 0.02, 0.03}, the dimension of the entity and relation embeddings in relation triple are among {50, 100, 150, 200}, the dimension of the type and relation embeddings in type triples are among {50, 100, 150, 200}, the batch size is among {128, 256, 512, 1024}. Table 2 lists the detailed hyper-parameters of TTModel.

Baseline

Since TTModel mainly utilizes the structured and text information of the KG, methods that additionally introduce pharmacological features of drugs and proteins, such as SMILES strings and protein sequences, are not selected for comparison. We compare our method against a variety of baselines which can be categorized as follows.

(1) BLM-NII[27]: this method is developed to improve the previous approach by using neighbor-based interaction-profile inference for both drugs and targets.
(2) KRONRLS-MKL[9]: this model has used linear combinations of multiple similarity measures to model the overall similarity between drugs and targets.
(3) NRLMF[9]: this model introduces the exclusive use of drug-drug and target-target similarity measures to infer possible drug targets.
(4) DNILMF[29]: this model leverages matrix factorization to predict drug targets over drug information networks.
(5) DDR[1]: this method utilizes a heterogeneous graph that contains known DTIs with multiple similarities between drugs and multiple similarities between proteins.
(6) TriModel[11]: this method is based on formulating the problem as a link predicting drug target proteins.
(7) DTiGEMS+[19]: this method integrates different techniques from machine learning, graph embedding, and similarity-based methods.
(8) SHGCL-DTI[10]: this approach aims to supplement classical semi-supervised DTI prediction tasks through an auxiliary graph-contrastive learning module.

Table 2. The hyper-parameters of the TTModel

Dataset	Batch size	Relation triple embedding size	Type triple embedding size	Learning rate	β_1	β_2
Enzymes(E)	1024	100	50	0.01	0.1	0.3
Ion Channels(IC)	1024	100	50	0.01	0.1	0.3
G-Protein Coupled Receptors(GPCR)	1024	100	50	0.01	0.1	0.3
Nuclear Receptors(NR)	1024	100	50	0.01	0.2	0.4
DrugBank_FDA(DB)	1024	100	50	0.01	0.1	0.3
KGHC	1024	100	50	0.01	0.4	0.5

Table 3. A comparison with state-of-the-art models on standard datasets

Model	Ft.	E		IC		GPCR		NR		DB	
	Str.	AUC	AU-PR	AUC	AU-PR	AUC	AU-PR	AUC	AU-PR	AUC	AU-PR
BLM-NII		0.96	0.86	0.91	0.83	0.88	0.53	0.91	0.62	0.90	0.12
KRONRLS-MKL		0.93	0.87	0.90	0.86	0.91	0.67	0.87	0.51	0.88	0.35
NRLMF		0.95	0.89	0.98	0.79	0.95	0.69	0.93	0.72	0.93	0.32
DNILMF		0.96	0.85	0.94	0.87	0.96	0.70	0.92	0.66	0.95	0.42
DDR	Ext.	0.97	0.92	0.98	0.92	0.96	0.79	0.92	0.83	0.96	0.61
TriModel		0.99	0.96	0.99	0.95	0.99	0.80	**0.99**	0.84	0.99	0.64
DTiGEMS+		0.99	0.97	0.99	0.96	0.99	0.86	0.97	0.88	–	–
SHGCL-DTI		0.85	0.88	0.85	0.89	0.88	0.90	0.97	**0.97**	–	–
TTModel		0.99	0.98	0.99	0.98	0.99	0.92	0.97	0.93	0.99	0.90
TTModel_RotatE		**0.99**	**0.98**	**0.99**	**0.98**	**0.99**	**0.95**	0.98	0.95	**0.99**	**0.91**

Results and analysis

Table 3 shows the results in terms of the AU-PR and AUC for all compared models. The *Ft*. column represents model's feature type. The *Str*. feature type represents protein and drug structure-based features and *Ext*. denotes extensive prior knowledge feature. Overall best results are in bold. We first can observe that our method outperforms almost baseline algorithms. For each dataset, TTModel performs better than the DTiGEMS+[19] in terms of AU-PR 1%, 2%, 6%, and 5% for the E, IC, GPCR, and NR datasets, respectively. It verifies the effectiveness of our model. Then, most models that use extensive prior knowledge features outperform models that use protein and drug structure-based features. This shows that the prior knowledge feature can improve performance in DTI prediction. Besides, TriModel[11] can perform better than DDR.[1] Because DDR only employs networks with single relation, and TriModel allows for encoding multiple types of associations within the same graph and thus utilizes more complex patterns. It illustrates the multiple types of related information that are beneficial to DTI prediction.

Compared with the TriModel and SHGCL-DTI,[10] our model achieves better performance. We believe that the success of the model can be attributed to the utilization of biomedical text and type information, which were not leveraged in the previous graph-based models. Firstly, we utilize the large-scale biomedical text feature combined with the KG. This allows the model utilizes the semantic feature from biomedical text and KG feature at the same time. Other models, which only use the topology information of graph, may cause the predictive ability to inevitably suffer from the sparseness and incompleteness of the graph data. Secondly, the latent type information can provide significant supplements for entity and relation representation. The type triple encoder can well integrate the latent entity type information into KG representation. Compared with other graph-based methods, TTModel can automatically learn the text semantic and type information to enhance the entity and relation representation.

Additionally, we have supplemented the experiments to explore the impact of using different encoder on the TTModel. TTModel_RotatE represents the model adopting RotatE[30] as the encoder. TTModel_RotatE outperforms TTModel, indicating that RotatE has more advantages over TransE. The results suggest that TransE is not able to fully exploit the structural information of the knowledge graph and might therefore hamper the performance of TTModel in predicting DTI. Consequently, there is great potential for a strategy that fuses type information and text information of the knowledge graph, as it could maximize the value of the structural information present in the KG.

To verify the performance of the long-tail scenario, we construct a dataset, named KGHC, to simulate a long-tail scenario. Specifically, we adopt a new benchmarking dataset (KGHC) which is collected by extracting the abstracts which are related to hepatocellular carcinoma from PubMed.[31] Each triple in KGHC contains an accurate text description for relationships. For example, the triple *associated_with*(*Obesity*, *HCC*) is extracted from the sentence *"Obesity - A number of observational studies have linked excess body fat with a higher risk for HCC."* We think this sentence is the accurate text description for the tripe *associated_with*(*Obesity*, *HCC*) and retain this information as an attribute. The detail of KGHC is introduced in our previous work.[32] Following the method of,[12] we construct the long-tail dataset for KGHC. Specifically, we count the entities with less than 50 occurrences in the dataset to filter the testing set. In total, KGHC contains 5,028 entities, 28 relations, 8,917 triples for training, 147 triples for validating and 145 triples for testing. We evaluate the model 5 times on the KGHC and average results across these runs. We add two common metrics, i.e., Mean Rank (MR) of correct entities and Hits@10 (H@10) which means the proportion of correct entities in the top 10. A low MR score or a high H@10 score is preferred. As shown in Table 4, the improvement is rather significant for KGHC, which is exactly the sparse KG. This finding empirically demonstrates that our model maintains performance for sparse KGs, and this is probably because the semantic and type information can provide a good supplement for KGE model.

Table 4. The results on the KGHC

	AU-PR	MR	H@10
TransE	0.66	58.48	0.33
TTModel	**0.72**	**28.91**	**0.73**

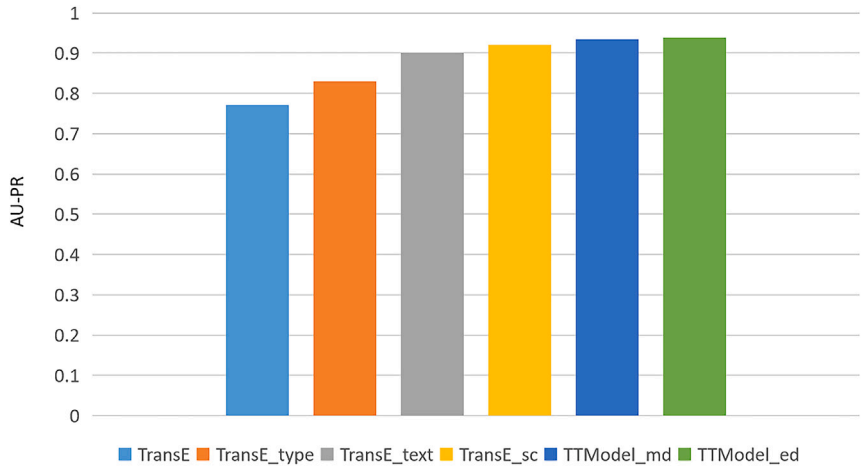


Figure 4. The ablation study on NR

DISCUSSION

Ablation study

In this section, we discuss possible reasons for the improved high performance of our approach when compared to the baseline methods. To test the effectiveness of the central idea, we implement the ablation study of TTModel on NR. As shown in Figure 4, the TransE_type model removes the text information of TTMode and its relation triple encoder uses only the structural information of triples. TransE_text removes the type triple encoder and type similarity constraint of TTModel. Its relation triple encoder simultaneously utilizes both structural and text information. TransE_sc represents that we omit the type embeddings similarity constraint from TTModel. The results show that the metric is improved when the biomedical text feature is added to the model. It verifies that the text information can provide beneficial features for the KGE model. In addition, an improvement is achieved when the type feature and type similarity constraint are added to the model. It illustrates that the type information and type similarity constraint can improve the performance of the model. We also evaluate the effort of different distance methods in type embedding similarity constraint. TTModel_md and TTModel_ed represent the utilization of Manhattan distance and Euclidean distance in type embedding similarity constraint, respectively. We can observe that the performance of the Manhattan distance (AU-PR: 0.934) and Euclidean distance (AU-PR: 0.938) in the TTModel is similar.

Case study

To evaluate the performance of the model in incomplete scenarios, we conducted the comparative experiments by removing textual information related to biological entities. As shown in Figure 5, *S_d* represents the removal of all drug-related textual information. *S_t* represents the removal of all target-related textual information, and *S_dt* represents the TTModel that does not utilize any textual information. It can be observed that the model's performance decreases when removing textual information related to drugs or targets. The most significant performance drop is observed when the model completely abstains from using textual information. *S_dt* achieves the worst performance, which demonstrates the beneficial impact of textual information on the model's performance.

In addition, we conduct the experiments with different ratios between the number of positive and negative samples to investigate the effects on TTModel. Following the setting of SHGCL-DTI[10] method, we build three experimental datasets, each containing all the 90 positive

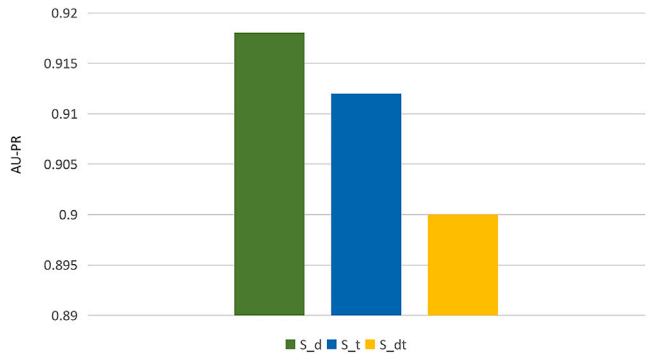


Figure 5. The impact of textual Information on the TTModel on the NR

Table 5. Results of positive and negative examples with different ratios in NR

Ratio	AUC	AU-PR
1:1	0.98	0.97
1:5	0.98	0.95
1:all	0.98	0.95

samples from the NR dataset. Meanwhile, these three experimental sets consist of 90 negative samples (1:1 ratio), 450 negative samples (1:5 ratio), and 1,314 negative samples (1:all ratio), respectively. Then we perform 10-fold cross-validation experiment using these three experimental datasets for TTModel_RotatE. The experimental results are shown in Table 5. We can observe that TTModel_RotatE achieves the best performance on the AUC and AU-PR metric when the ratios are 1:1.

Visualization of clustering entity type representations

To verify the proposed method and learn the entity type information, we conduct the case study in Figure 6. Specifically, we leverage K-Means to cluster the type embeddings on KGHC and then t-SNE to reduce dimensionality for 2d visualization. Figure 6 (a) and (b) show the clustering of entity embedding and the clustering of the entity type embeddings on KGHC dataset. It can be clearly observed that entity type clustering has better compactness than entity clustering, which demonstrates that entity type embeddings could reflect the characteristics of types. We also take the relation "*treats*" as an example and obtain the triples whose relation is "*treats*". Then, we utilize the head entities and head entity type embedding in these triples which learns from the proposed method by relation triple encoder and type triple encoder for clustering. We observe that most of these header entities contain two entity types, i.e., "*Drug*" and "*Protein*". Therefore, we set the *k* value to 2. Figure 6 (c) and (d) show the clustering of entity embedding and entity type embedding, respectively. We can find that most of blue points represent the type of "*Drug*" and most of red points represent the type of "*Protein*". It is evident that some type embeddings representing the type

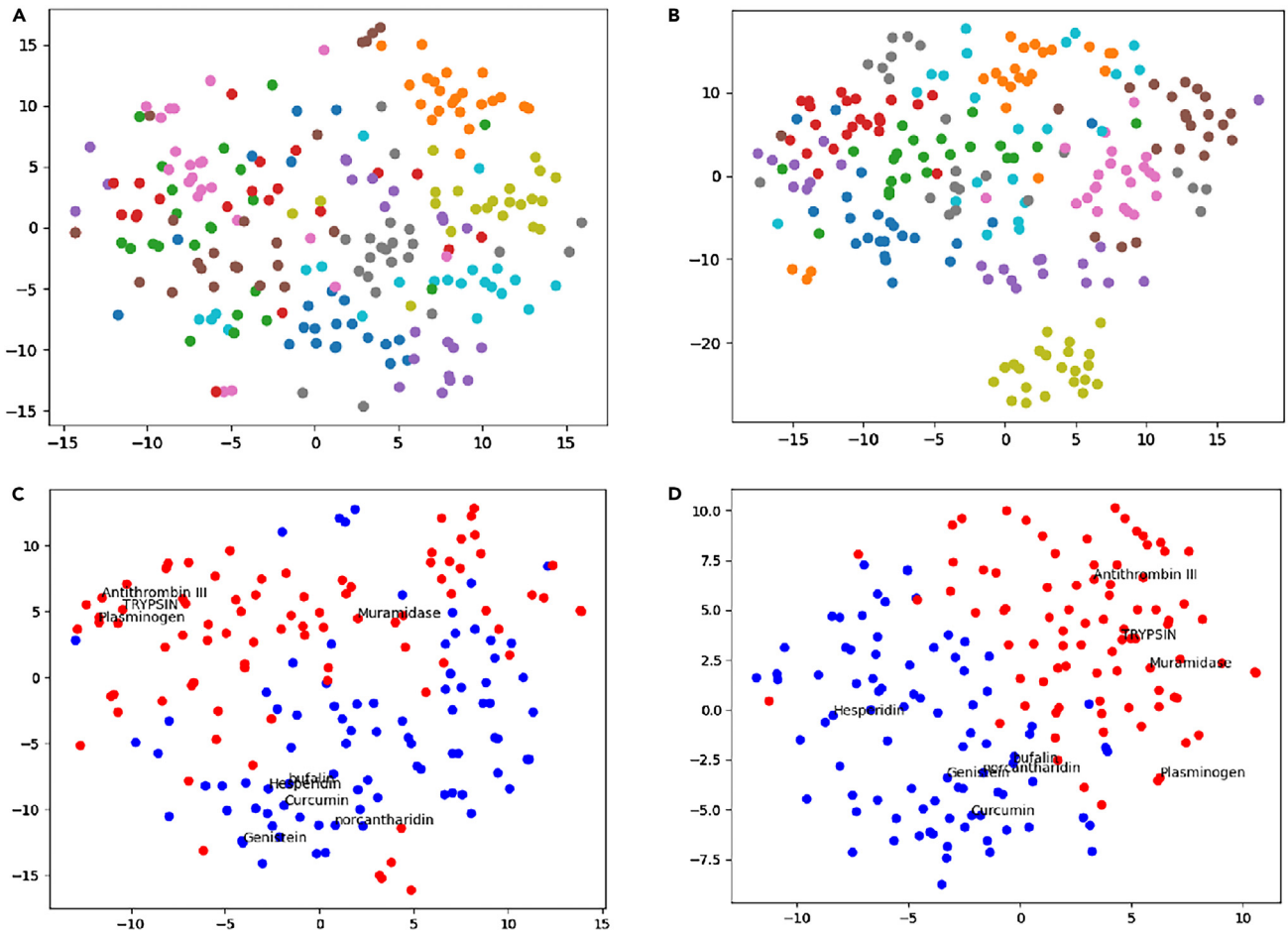


Figure 6. The visualization of entity and entity type embeddings clustering using TTModel on KGHC

Drug such as *curcumin* and *genistein* are clustered into the same type while others stay far away. These visualization results can verify that our model can learn the entity type information. Combined with the results of the ablation study, it can be further verified that the type information can improve the performance of the KGE model.

Limitations of the study

We attempt to incorporate the textual and type information into the KGE model to enhance the accuracy of entity and relation representation. However, we only utilize the latent type information in the KG and ignore the deep hierarchical information. In future work, we aim to learn hierarchical information by further improving the type triple encoder.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Materials availability
  - Data and code availability
- METHOD DETAILS
  - Problem Formulation
  - Relation triple encoder
  - Type triple encoder
  - Type embeddings similarity constraint
  - Model training
- QUANTIFICATION AND STATISTICAL ANALYSIS

ACKNOWLEDGMENTS

This work was supported by the grants from the Natural Science Foundation of China (No. 62276043) and the Fundamental Research Funds for the Central Universities (No. DUT22ZD205).

AUTHOR CONTRIBUTIONS

Methodology: N.L. and Z.Y.; Investigation: N.L. and Z.Y.; Writing-original draft: N.L., Z.Y., J.W., and H.L.; Writing-review and editing: N.L., Z.Y., J.W., and H.L.

DECLARATION OF INTERESTS

All authors declare no competing interests.

Received: September 11, 2023
Revised: January 16, 2024
Accepted: February 28, 2024
Published: March 5, 2024

REFERENCES

1. Olayan, R.S., Ashoor, H., and Bajic, V.B. (2018). Ddr: efficient computational method to predict drug–target interactions using graph mining and machine learning approaches. Bioinformatics *34*, 3779.
2. Santos, R., Ursu, O., Gaulton, A., Bento, A.P., Donadi, R.S., Bologa, C.G., Karlsson, A., Al-Lazikani, B., Hersey, A., Oprea, T.I., and Overington, J.P. (2017). A comprehensive map of molecular drug targets. Nat. Rev. Drug Discov. *16*, 19–34.
3. Terstappen, G.C., Schlüpen, C., Raggiaschi, R., and Gaviraghi, G. (2007). Target deconvolution strategies in drug discovery. Nat. Rev. Drug Discov. *6*, 891–903.
4. Yamanishi, Y., Araki, M., Gutteridge, A., Honda, W., and Kanehisa, M. (2008). Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. Bioinformatics *24*, i232–i240.
5. Sleno, L., and Emili, A. (2008). Proteomic methods for drug target discovery. Curr. Opin. Chem. Biol. *12*, 46–54.
6. Cheng, F., Liu, C., Jiang, J., Lu, W., Li, W., Liu, G., Zhou, W., Huang, J., and Tang, Y. (2012a). Prediction of drug-target interactions and drug repositioning via network-based inference. PLoS Comput. Biol. *8*, e1002503.
7. Cheng, F., Zhou, Y., Li, W., Liu, G., and Tang, Y. (2012b). Prediction of chemical-protein interactions network with weighted network-based inference method. PLoS One *7*, e41064.
8. Liu, Y., Wu, M., Miao, C., Zhao, P., and Li, X.L. (2016). Neighborhood regularized logistic matrix factorization for drug-target interaction prediction. PLoS Comput. Biol. *12*, e1004760.
9. Nascimento, A.C., Prudêncio, R.B., and Costa, I.G. (2016). A multiple kernel learning algorithm for drug-target interaction prediction. BMC Bioinf. *17*, 1–16.
10. Li, Y., Qiao, G., Gao, X., and Wang, G. (2022). Supervised graph co-contrastive learning for drug–target interaction prediction. Bioinformatics *38*, 2847–2854.
11. Mohamed, S.K., Nováček, V., and Nounu, A. (2020). Discovering protein drug targets using knowledge graph embeddings. Bioinformatics *36*, 603–610.
12. Xie, R., Liu, Z., Sun, M., et al. (2016). Representation learning of knowledge graphs with hierarchical types. In

Proceedings of the 25th International Joint Conferences on Artificial Intelligence (IJCAI), pp. 2965–2971.
13. Niu, G., Li, B., Zhang, Y., Pu, S., and Li, J. (2020). Autoeter: Automated entity type representation for knowledge graph embedding. Preprint at arXiv. https://doi.org/10.48550/arXiv.2009.12030.
14. Kamper, H., Wang, W., and Livescu, K. (2016). Deep convolutional acoustic word embeddings using word-pair side information. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4950–4954.
15. Zhang, L., Zhang, S., and Balog, K. (2019). Table2vec: Neural word and entity embeddings for table population and retrieval. In Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval, pp. 1029–1032.
16. Kilicoglu, H., Shin, D., Fiszman, M., Rosemblat, G., and Rindflesch, T.C. (2012). Semmeddb: a pubmed-scale repository of biomedical semantic predications. Bioinformatics *28*, 3158–3160.
17. Chu, Y., Shan, X., Chen, T., Jiang, M., Wang, Y., Wang, Q., Salahub, D.R., Xiong, Y., and Wei, D.Q. (2021). Dti-mlcd: predicting drug-target interactions using multi-label learning with community detection method. Briefings Bioinf. *22*, bbaa205.
18. Wang, S., Li, J., Wang, Y., and Juan, L. (2022). A neighborhood-based global network model to predict drug-target interactions. IEEE ACM Trans. Comput. Biol. Bioinf *19*, 2017–2025.
19. Thafar, M.A., Olayan, R.S., Ashoor, H., Albaradei, S., Bajic, V.B., Gao, X., Gojobori, T., and Essack, M. (2020). Dtigems+: drug–target interaction prediction using graph embedding, graph mining, and similarity-based techniques. J. Cheminf. *12*, 44.
20. Kanehisa, M., Goto, S., Hattori, M., Aoki-Kinoshita, K.F., Itoh, M., Kawashima, S., Katayama, T., Araki, M., and Hirakawa, M. (2006). From genomics to chemical genomics: new developments in kegg. Nucleic Acids Res. *34*, D354–D357.
21. Schomburg, I., Chang, A., Ebeling, C., Gremse, M., Heldt, C., Huhn, G., and Schomburg, D. (2004). Brenda, the enzyme database: updates and major new developments. Nucleic Acids Res. *32*, D431–D433.
22. Günther, S., Kuhn, M., Dunkel, M., Campillos, M., Senger, C., Petsalaki, E., Ahmed, J., Urdiales, E.G., Gewiess, A., Jensen, L.J., et al. (2007). Supertarget and matador: resources for exploring drug-target relationships. Nucleic Acids Res. *36*, D919–D922.
23. Wishart, D.S., Knox, C., Guo, A.C., Shrivastava, S., Hassanali, M., Stothard, P., Chang, Z., and Woolsey, J. (2006). Drugbank: a comprehensive resource for in silico drug discovery and exploration. Nucleic Acids Res. *34*, D668–D672.
24. Apweiler, R., Bairoch, A., Wu, C.H., Barker, W.C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., et al. (2004). Uniprot: the universal protein knowledgebase. Nucleic Acids Res. *32*, D115–D119.
25. Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., and Morishima, K. (2017). Kegg: new perspectives on genomes, pathways, diseases and drugs. Nucleic Acids Res. *45*, D353–D361.
26. Mitchell, A.L., Attwood, T.K., Babbitt, P.C., Blum, M., Bork, P., Bridge, A., Brown, S.D., Chang, H.Y., El-Gebali, S., Fraser, M.I., et al. (2019). Interpro in 2019: improving coverage, classification and access to protein sequence annotations. Nucleic Acids Res. *47*, D351–D360.
27. Mei, J.P., Kwoh, C.K., Yang, P., Li, X.L., and Zheng, J. (2013). Drug–target interaction prediction by learning from local information and neighbors. Bioinformatics *29*, 238–245.
28. Davis, J., and Goadrich, M. (2006). The relationship between precision-recall and roc curves. In Proceedings of the 23rd international conference on Machine learning, pp. 233–240.
29. Hao, M., Bryant, S.H., and Wang, Y. (2017). Predicting drug-target interactions by dual-network integrated logistic matrix factorization. Sci. Rep. *7*, 40376.
30. Sun, Z., Deng, Z.H., Nie, J.Y., and Tang, J. (2019). Rotate: Knowledge graph embedding by relational rotation in complex space. Preprint at arXiv. https://doi.org/10.48550/arXiv.1902.10197.
31. Canese, K., and Weis, S. (2013). Pubmed: the bibliographic database. The NCBI handbook *2*.
32. Li, N., Yang, Z., Luo, L., Wang, L., Zhang, Y., Lin, H., and Wang, J. (2020). Kghc: a knowledge graph for hepatocellular carcinoma. BMC Med. Inf. Decis. Making *20*, 135–211.
33. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., and Kang, J. (2020). Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics *36*, 1234–1240.
34. Davis, A.P., Grondin, C.J., Johnson, R.J., Sciaky, D., McMorran, R., Wiegers, J., Wiegers, T.C., and Mattingly, C.J. (2019). The comparative toxicogenomics database: update 2019. Nucleic Acids Res. *47*, D948–D954.
35. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In Advances in neural information processing systems (NeurIPS), pp. 2787–2795.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Data and code	TTModel	https://github.com/Nan-ll/TTModel
Other		
Baseline Model	KRONRLS-MKL	www.cin.ufpe.br/~acan/kronrlsmkl/
Baseline Model	DNILMF	https://github.com/minghao2016/DNILMF
Baseline Model	DDR	https://bitbucket.org/RSO24/ddr/
Baseline Model	TriModel	drugtargets.insight-centre.org
Baseline Model	DTiGEMS+	https://github.com/MahaThafar/Drug-Target-Interaction-Prediciton-Method
Baseline Model	SHGCL-DTI	https://github.com/catly/SGCL-DTI
BioBERT: a biomedical language model for biomedical vector extracting	BioBERT	https://github.com/dmis-lab/biobert

RESOURCE AVAILABILITY

Lead contact

Further information and request should be directed to the lead contact, Zhihao Yang (yangzh@dlut.edu.cn).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- The data reported in this paper is publicly available on GitHub (https://github.com/Nan-ll/TTModel).
- All original code has been deposited at GitHub and is publicly available as of the date of publication. The DOI is listed in the key resources table.
- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

METHOD DETAILS

Problem Formulation

We denote KGs as $G = \{E, R, T\}$, where $E, R \in \mathbb{R}^m$ and T indicate sets of entities, relations, and triples, respectively. m represents the dimension of entities and relations in KG. Each triple represents a relation between the head entity and the tail entity. For each entity, relation, and word, we use the boldface to indicate their low-dimensional vectors, respectively.

Relation triple encoder

The biomedical text information can provide the potential textual semantic information for KGE model. Therefore, in this section, we introduce an unsupervised method that leverages biomedical text information to enhance the entities and relations representation. To capture information of the entities from the biomedical textual data, we obtain the representations of entities by training word embedding on the biomedical text. Specifically, the entities are extracted from the biomedical text usually consist of the words. We adopt the BioBERT[33] model, to embed the entity text. BioBERT https://github.com/dmis-lab/biobert is pre-trained on biomedical domain corpora (PubMed abstracts and PMC full-text articles). Compared with the traditional word embedding only learns a global vector representation for a word, BioBERT model can provide different representations for a varying sense of the same word according to the contextualized information. We first obtain the entity name from Uniprot,[24] KEGG,[25] InterPro[26] and DrugBank,[23] Comparative Toxicogenomics Database (CTD)[34] such as *Estradiol*. Then an augmented vector for entity e is defined as:

$$\mathbf{e} = \alpha_1 \cdot \mathbf{e}_k + (1 - \alpha_1) \cdot \mathbf{e}_w \mathbf{M} \quad \text{(Equation 3)}$$

where $\mathbf{e}_k \in \mathbb{R}^m$ denotes entity embedding learned from the KG, $\mathbf{e}_w \in \mathbb{R}^w$ represents the entity embedding obtained from the BioBERT model. w denotes the dimension of word embedding. $\mathbf{M} \in \mathbb{R}^{w \times m}$ represents the global matrix which is utilized to map the text vector space to the KG

vector space. For the entity which cannot obtain the entity text, we only leverage the entity representation learned from KG, and the weight factor α_1 is set to 1.

For each relation r, there are several entity pairs that can form fact triples with the relation r. The same relationship is expressed differently in different entity pairs. To enhance the effectiveness of KGE models, we utilize semantic information extracted from biomedical text to assist relations in fitting the most reasonable entity pairs. Given a sentence containing two entities, and the sentence can accurately represent the semantic information of triple, we think the sentence includes implicit features of the textual relationship between the two entities. We fuse the latent semantic information from biomedical text into the KGE model. Specifically, for each triple, we obtain the relation text from PubMed and utilize the BioBERT model to obtain the embedding of the sentence as the relationship context representation. The enhanced relation representation function is shown as:

$$\mathbf{r} = \alpha_2 \cdot \mathbf{r}_k + (1 - \alpha_2) \cdot \mathbf{r}_w \mathbf{N} \quad \text{(Equation 4)}$$

where $\mathbf{r}_k \in \mathbb{R}^m$ represents the relation embedding which obtains from the KG. $\mathbf{r}_w \in \mathbb{R}^w$ is the text embedding which obtains from the BioBERT model. Then we adopt the global matrix $\mathbf{N} \in \mathbb{R}^{w \times m}$ to map the text vector space to the KG vector space. Thus, each latent feature contains a contribution from the original relation vector and from the BioBERT vector. We only use the $\mathbf{r}_k$ when the triple cannot obtain the relation text information, and the weight factor α_2 is set to 1. The enhanced relation representations not only contain the semantic information from the text, but also contain the structure information from the KG. We adopt the score function of relation triple encoder which was proposed by TransE.[35] For TransE, given a triple (h, r, t), this method expects $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ when (h, r, t) is hold. This indicates that the tail entity t should be the nearest neighbor of $(\mathbf{h} + \mathbf{r})$. Hence, TransE assumes the score function as:

$$f_r(h, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\| \quad \text{(Equation 5)}$$

$$E_1 = \|\mathbf{e}_h + \mathbf{r} - \mathbf{e}_t\| \quad \text{(Equation 6)}$$

Different from the TransE, the entity and relation vector representations of relation triple encoder utilize not only the structural information of triples, but also the semantic information of text. As shown in Equation 6, $\mathbf{e}_h$ and $\mathbf{e}_t$ represent the head entity vector representation and tail entity vector representation, respectively, which are obtained from Equation 3. $\mathbf{r}$ denotes the relation vector representation which obtains from the Equation 4.

Type triple encoder

Most of the proposed methods merely rely on the structured information in KG, paying less attention to type information. In fact, entity type information can provide important information for KGE and deepen the model's understanding of entities and triples.[12] For the biomedical field, entity type often has a structured system defined manually, which can be regarded as *a priori* knowledge. However, there are also certain differences in the potential type information of entities according to the relationship in different triples. This also sideways effects that the same entity has different representations in different triples. Therefore, in this section, we will detail how to obtain the type information and use them to enhance the entity and relation representation. Specifically, given an entity and its associated relation in a triple, we aim to learn the entity type vector representation with a relation-aware projection mechanism to output the type representation. We obtain the entity type representation:

$$g(e, r) = \mathbf{p}_e \mathbf{Q}_r \quad \text{(Equation 7)}$$

where $\mathbf{p}_e \in \mathbb{R}^d$ represents the type embedding of entity e with dimension d. $\mathbf{Q}_r \in \mathbb{R}^{d \times d}$ is denoted as the projection weight matrix associated with the relation r, which could automatically select the latent information of each type embedding most relevant to the relation r. $\mathbf{p}_h$, $\mathbf{p}_t$ denote the type embedding of head entity and the type embedding of tail entity embedding, respectively. The score function involved in type triples is defined as,

$$E_2 = \left\|\mathbf{p}_{yh} + \mathbf{r}_y - \mathbf{p}_{yt}\right\| \quad \text{(Equation 8)}$$

where $\mathbf{p}_{yh}$ and $\mathbf{p}_{yt}$ which obtain from Equations 9 and 10 are the type embeddings of entities h and t both focusing on the relation r.

$$\mathbf{p}_{yh} = g(h, r) \quad \text{(Equation 9)}$$

$$\mathbf{p}_{yt} = g(t, r) \quad \text{(Equation 10)}$$

$\mathbf{r}_y \in \mathbb{R}^d$ represents the embedding of the relation r in the type triple.

Type embeddings similarity constraint

Due to the head or tail entities sharing the same relation tend to cluster and have similar representation. For example, for the relation *treats*, the head entity type is usually *drug* and the tail entity type is usually *diseases*, *symptoms*, etc. Therefore, we expect triples with the same relationship, the type embedding of head entities involved in the triples are closer to each other, and correspondingly, the type embedding of tail

entities involved in the triples are closer to each other. Specifically, for two triples, the score function for evaluating the dissimilarity of the type embedding is defined as follows,

$$E_3((h_1, r_1, t_1), (h_2, r_2, t_2)) = \frac{1}{2}\left(\left\|\mathbf{p}_{yh1} - \mathbf{p}_{yh2}\right\| + \left\|\mathbf{p}_{yt1} - \mathbf{p}_{yt2}\right\|\right) \quad \text{(Equation 11)}$$

where $\mathbf{p}_{yh1}$ and $\mathbf{p}_{yh2}$ represent two head entity type embedding which obtain from Equation 9 and $\mathbf{p}_{yt1}$ and $\mathbf{p}_{yt2}$ represent two tail entity embedding which obtain from Equation 10, respectively. It is expected to be a low score when the type representation is consistent with the same relation.

Model training

The designed relation triple encoder, type triple encoder, and type embeddings similarity constraint optimized according to a three-component objective function:

$$L = \sum_{(h,r,t) \in S}\left\{\sum_{(h',r,t') \in S'}\{L_1 + \beta_1 L_2\} + \beta_2 L_3\right\} \quad \text{(Equation 12)}$$

where L_1, L_2, and L_3 are the loss functions that correspond to the relation triple encoder, the type triple encoder and type embeddings constraint, respectively, $\beta_1 \in [0, 1]$ and $\beta_2 \in [0, 1]$ denote the weights of L_2 and L_3 for the tradeoff between the relation triple, the type triple and the type similarity constraint. S contains all the triples in the train set, and S' is the corrupted triples set generated by replacing the entities and relations in S. Taking TransE[35] as an example, L_1, L_2, and L_3 are defined as:

$$L_1 = max(0, \gamma_1 + E_1 - E_1') \quad \text{(Equation 13)}$$

$$L_2 = max(0, \gamma_2 + E_2 - E_2') \quad \text{(Equation 14)}$$

$$L_3 = \sum_{(h_p,r,t_p) \in P}\sum_{(h_n,r_n,t_n) \in P'} max(0, \gamma_3 + E_3 - E_3') \quad \text{(Equation 15)}$$

where γ_1, γ_2, and γ_3 are the margin parameters between correct triples and negative triples. As the embeddings of entities and relations are normalized, the margin γ_1, γ_2, and γ_3 can actually regularize the above objective and keep the weights from collapsing or deviating. E_1 is the relation triple encoder score function, E_2 is the type triple encoder score function, and E_3 is the type embeddings similarity constraint score function. E_1', E_2', and E_3' represent the score function with negative instances. $max(0, x)$ maximizes the margin between 0 and x. In L_3, given a relation r, we treat the triples with the same relation r as positive samples, expecting the type embedding of head entities or tail entities to be close to each other. Simultaneously, we consider the triples with different relation as negative, expecting the type embedding of head entities or tail entities to be far from each other. Specifically, given a triple (h, r, t), (h_p, r, t_p) is a positive instance in the set P containing other triples correlated to the same relation r, while (h_n, r_n, t_n) is any negative instance in the set P' containing the other triples without the relation r. To train the parameters of the score function, it needs to make the margin-based score function minimization as its training objective.

In the test phase, the energy function for evaluation is designed as follows:

$$E_p = E_1 + \beta_1 E_2 \quad \text{(Equation 16)}$$

where E_1 is the energy function of relation triple and E_2 is the energy function of type triple. β_1 is the weight which is the same as Equation 12.

QUANTIFICATION AND STATISTICAL ANALYSIS

In the final evaluation of the model's performance, we employed a 10-fold cross-validation approach. The ultimate results consisted of the mean of the 10-folds, with specific evaluation metrics including AUC and AU-PR.