Fusing Structural and Textual Features for Knowledge Graph Relation Prediction (Post-print)

Authors: Lin Zefei, Ou Shiyan

Date: 2023-04-01T16:16:03+00:00

Abstract

[Purpose/Significance] This paper proposes a novel knowledge graph relation prediction method that fuses internal structural features and external textual features, aiming to predict the types of missing relations between two entities in knowledge graphs. [Method/Process] Relation paths and text reflecting inter-entity relationships are matrixized, structural and textual pattern features associated with specific relation types are learned via convolutional neural networks, and a model is trained on this basis to achieve relation prediction. [Results/Conclusions] Experimental results demonstrate that the method's performance on benchmark datasets surpasses baseline methods, effectively improving knowledge graph relation prediction performance. Practical application reveals that the method possesses good application value in knowledge services.

Full Text

Preamble

Research on Knowledge Graph Relation Prediction by Fusing Structure and Text Features

Lin Zefei[1],[2], Ou Shiyan[1]
[1]School of Information Management, Nanjing University, Nanjing 210093
[2]College of Social Development, Fujian Normal University, Fuzhou 350007

Abstract: [Purpose/Significance] This paper proposes a novel knowledge graph relation prediction method that integrates internal structural features and external textual features, aiming to predict the type of missing relations between two entities in knowledge graphs. [Method/Process] The method transforms relation paths and texts reflecting entity relationships into matrices, learns structure and text pattern features related to specific relation types through

convolutional neural networks, and trains a model for relation prediction. [Result/Conclusion] Experimental results demonstrate that the proposed method outperforms comparison methods on evaluation datasets, effectively improving knowledge graph relation prediction performance. Practical application reveals that the method holds significant value for knowledge services.

Keywords: knowledge graph; relation prediction; feature fusion; deep learning
Classification Numbers: G254.29; TP391.1
DOI: 10.13266/j.issn.0252-3116.2020.21.013

In the era of big data, online information resources are growing explosively. These resources contain substantial valuable knowledge, and extracting and organizing this knowledge can form large-scale knowledge bases that provide users with intelligent and efficient knowledge services. Knowledge graph technology has emerged in this context. A knowledge graph (KG) aims to describe entities and their relationships in the objective world in a structured form, representing knowledge as triples in the form <h, r, t>, where h, t, and r denote head entity, tail entity, and their relationship, respectively. When entities are represented as nodes and relationships as edges, a graph-structured knowledge network can be constructed. Currently, knowledge graphs have been widely applied in semantic search, question answering, and intelligent recommendation, demonstrating broad prospects. In recent years, large-scale knowledge graphs such as DBpedia, Freebase, and CN-DBpedia have emerged. Although these contain numerous triples, their knowledge remains far from complete. For instance, 66% and 71% of person entities in DBpedia and Freebase lack birthplace information. Against this backdrop, knowledge graph completion research holds significant importance.

Knowledge graph completion (KGC) aims to reduce knowledge deficiency and improve completeness. For example, Figure 1 shows a local entity-relationship view constructed from triples in the CN-DBpedia knowledge graph. Using explicit knowledge in this graph, we can infer that “Ma Huateng” and “Liu Chiping” likely have a colleague relationship, and “Shenzhen” is highly probable as their workplace. However, such implicit knowledge is not explicitly represented in the knowledge graph. Therefore, completing this implicit knowledge would significantly enhance semantic search precision and improve service quality in question answering and recommendation systems.

Specifically, knowledge graph completion includes two typical subtasks: entity prediction and relation prediction. Entity prediction aims to predict the missing tail entity given <h, r, ?> or the missing head entity given <?, r, t>. Relation prediction, the focus of this study, predicts the missing relation type between two entities given <h, ?, t>.

Both internal structure and external text of knowledge graphs can serve as bases for relation prediction. For instance, based on the path structure from “Ma Huateng” to “Liu Chiping” in Figure 1, we can infer they may have a “colleague” relationship. If text mentions “Liu Chiping assists Ma Huateng in

managing daily operations," similar reasoning can be made. Combining these two types of information helps computers more accurately identify entity relationships. Thus, internal structure and external text are complementary for relation prediction, yet current research integrating them remains scarce. To address this gap, this study explores a novel knowledge graph relation prediction method that designs a deep learning model to learn relation path structural features and integrates external text features, constructing an ensemble model that fuses internal structure and external text features.

1 Related Work

Current approaches for predicting missing relations in knowledge graphs fall into two categories: knowledge representation learning-based methods and relation path-based methods.

Knowledge representation learning uses machine learning to learn low-dimensional dense vector representations of entities and relations. Translation models, exemplified by TransE, are mainstream approaches. TransE treats relation vector r as a translation vector between head entity vector h and tail entity vector t, such that h (Tencent) + r (Chairman) t (Ma Huateng), training entity and relation vectors by minimizing the distance between h+r and t. While achieving high prediction accuracy and strong generalizability, these methods primarily focus on local triple-level information rather than the network structure and logical reasoning between relations.

Relation paths are connectivity paths between two entities in knowledge graphs that reflect semantic relationships. Path-based methods use these paths as features to predict relation types. Carnegie Mellon University's N. Lao et al. proposed the Path Ranking Algorithm (PRA), a representative approach. PRA extracts all triples with relation r, performs random walks from each head entity, and considers paths reaching tail entities within three steps as relation paths for training. It learns path weights via logistic regression and calculates transition probabilities through path-constrained random walks. While leveraging network structure and logical reasoning with good interpretability, PRA's learning capability remains limited, yielding inferior performance compared to mainstream models like TransE.

Both approaches rely on structural features but differ in focus: the former on micro triple-level structure, the latter on macro path-level structure. External text also serves as evidence for relation prediction. Recent studies integrate entity description texts (mainly from encyclopedia abstracts) with knowledge representation learning. For instance, J. Xu et al. combine entity description vectors with TransE for joint learning, while T. Long et al. use average word vectors of descriptions to initialize entity vectors. However, for relation prediction, sentences containing both entity mentions more directly reflect their relationships than entity descriptions, yet such texts remain unused in current research.

Deep learning, particularly convolutional neural networks (CNN), represents a breakthrough in machine learning with excellent feature representation capabilities, though rarely applied to knowledge graph tasks. T. Detmers et al.'s ConvE (2018) was the first CNN-based model for knowledge graph completion, reshaping 1D head and relation vectors into 2D matrices for convolution operations. While effective, ConvE doesn't incorporate path or text information. To our knowledge, no deep learning model fuses path structural and text features for relation prediction.

In summary, current relation prediction primarily relies on structural features. Some studies integrate structural and textual features but neglect macro path-level structures and have limited text types. Deep learning applications remain scarce. Given the potential of multi-feature fusion and deep learning, this study proposes a CNN-based method integrating internal structure and external text features.

2 Design of Knowledge Graph Relation Prediction Model

This study proposes ConvF, a CNN-based model fusing internal structural and external textual features for knowledge graph relation prediction. The model treats relation prediction as a multi-label classification problem. Inputs are structural features reflecting relation paths and textual features reflecting entity relationships; outputs are probabilities of different relation types between two entities. The functional structure is shown in Figure 2.

ConvF comprises two sub-models: relation path reasoning and relation extraction. The former learns internal path structural features; the latter learns external text features. This section describes their design and integration.

2.1 Relation Path Reasoning Sub-model

The relation path reasoning sub-model predicts relations by learning path features. The intuition is: entity pairs with specific direct relations often have multiple connecting paths (relation paths) that serve as pattern features for that relation. For example, in Figure 3, the training pair (e_h, e_t) has direct relation Friend with two paths (Enemy, Enemy) and (Friend, Friend), which jointly form Friend's pattern features. A test pair (e_h', e_t') with similar path structures likely also has the Friend relation.

Formally, knowledge graph G = {E, R} is a directed multigraph where E is the entity set and R is the relation set. For entity pair <e_h, e_t> (e_h, e_t E), r_h→t R denotes e_t's relation to e_h. A relation path π_h→t = (e_h, r_1, e_1, r_2, e_2, …, r_n, e_t) can be simplified as π_h→t = (r_1, r_2, …, r_n), where n is the hop count. The set of all n-hop paths from e_h to e_t is denoted Π_h→t^n. Since G is directed, relations have directions: forward (r) points toward e_t, backward (r^{-1}) toward e_h. The direct forward path π_h→t^d = (e_h, r_h→t, e_t) contains the target relation.

The sub-model uses CNN to learn path pattern features. Since CNN requires 2D matrix input, paths must be converted to matrix form. Each relation r is represented as a k-dimensional vector. An n-hop path's vectors are vertically concatenated to form an n×k matrix. Multiple n-hop paths form a matrix by vertical concatenation:

PM_{n,i} = r_{i,1} ↓ r_{i,2} ↓ … ↓ r_{i,n} (1)
M_n = PM_{n,1} ↓ PM_{n,2} ↓ … ↓ PM_{n, _n} (2)

where PM_{n,i} is the matrix for the i-th n-hop path $\pi\{n,i\}$ (π{n,i} ≠ π_h→t^d), ↓ denotes vertical concatenation, M_n is the matrix representation of n-hop path set, and _n is a hyperparameter representing the maximum number of paths.

Figure 4 illustrates this with a family knowledge graph where R = {mother, son, brother, grandson}. To learn the son relation pattern, there are two 1-hop paths (son) and (mother^{-1}), and two 2-hop paths (mother, grandson) and (mother^{-1}, brother). The direct forward path (son) is excluded. Using vector representations, we construct M_1 and M_2 as training samples.

A key issue is relation vector representation. One-hot vectors create sparse matrices for large relation sets, increasing resource consumption. Instead, we use knowledge representation learning (e.g., TransE) to generate pretrained low-dimensional dense vectors with semantic information. Relation direction also affects reasoning. For n-hop paths (n > 1), we prepend a one-hot direction vector: (1,0) for forward, (0,1) for backward. Single-hop paths only contain backward relations and need no direction vector.

The sub-model treats relation prediction as multi-label classification. Input: n-hop path set matrices (n = 1,2,3,…). Output: probabilities of r_h→t belonging to different relation categories. The model comprises five layers (Figure 5):

(1) **Input Layer**: Contains path set matrices M_n. Width w_n depends on vector dimension k: w_1 = k for single-hop; w_n = k+2 for others (adding direction vectors). Height is n× _n. If actual paths exceed _n, excess paths are randomly removed; if fewer, zero vectors pad the matrix.

(2) **Convolutional Layer**: Extracts path features. For each n-hop matrix, F_n convolution kernels of size n×w_n perform convolution with stride n, making each receptive field correspond to a complete path matrix. Each kernel generates a 1D feature map.

(3) **Pooling Layer**: Applies global max pooling to each feature map, producing Σ_i F_i scalar values.

(4) **Fully Connected Layer**: Concatenates all scalars into a vector connected to the output layer. Dropout reduces overfitting.

(5) **Output Layer**: Uses sigmoid activation to compute probabilities for each relation category, with binary cross-entropy loss.

2.2 Relation Extraction Sub-model

The relation extraction sub-model predicts relations by learning external text features. Relation extraction aims to determine the relationship between two entity mentions in text. The sub-model adapts this idea: in external texts (e.g., encyclopedia entries), many sentences contain multiple entity mentions. By aligning triple entities with these mentions, finding sentences containing both head and tail entity mentions, and replacing mentions with masks (e.g.,

,), we can learn text pattern features for specific relations. For example, for <Lu Xun, work, A Madman's Diary>, sentences like "

published the vernacular novel " and " is a vernacular novel created by

" reflect the "work" relation.

The sub-model also treats prediction as multi-label classification using CNN, adopting the TextCNN design (Figure 6). It comprises five layers:

(1) **Input Layer**: Text matrix representation where each row is a word vector. For n words with k-dimensional vectors, the input is n×k.

(2) **Convolutional Layer**: Uses kernels of different heights (hyperparameters) and width k, sliding vertically with stride 1.

(3) **Pooling, Fully Connected, and Output Layers**: Identical to the path reasoning sub-model.

2.3 Fusion Model

While both sub-models can predict independently, relying on only one is often insufficient. For example, path (friend, friend) suggests but doesn't confirm a friend relation, while text mentioning "C takes care of A in life" combined with the path makes the inference more certain.

To integrate both sub-models, we propose ConvF (Figure 7). The model's first part combines both sub-models' input, convolutional, and pooling layers. In the fully connected layer, we concatenate neurons from both sub-models, creating a layer containing both text-based and path-based feature representations. An additional hidden fully connected layer improves feature representation capability.

ConvF usage involves three phases: (1) **Preprocessing**: Treat known triples <h, r, t> as training samples. For each, generate path set matrices M_n via graph search and text matrices via external texts. (2) **Training**: Input both matrix types with relation r as labels to train the classification model. (3) **Prediction**: For test pairs <h', t'>, generate corresponding matrices. ConvF outputs probabilities for each relation type. Notably, ConvF doesn't require both matrix types; if information is missing, zero matrices substitute for unavailable features.

ConvF's advantages: (1) Integrates structural and textual information; (2) Uses deep learning for more effective path pattern learning. The next section evaluates its performance.

3 Experiments and Results Analysis

3.1 Evaluation Datasets

We evaluate on FB15K and FB15K237 derivative datasets FB15K-T and FB15K237-T. FB15K contains over 500K triples involving 14,951 entities and 1,345 relations. K. Toutanova et al. noted that many FB15K test triples can be mapped to training triples via inverse relations, enabling simple rule-based models to match mainstream performance. Thus, they introduced FB15K237, removing most direct inverse relations to focus on non-trivial reasoning.

FB15K and FB15K237 contain only structural information. To test structure-text fusion, we add external text. Since Freebase knowledge mostly originates from Wikipedia, we map FB15K/FB15K237 entities to Wikipedia entries. For each triple, we extract sentences containing both entity mentions (with mentions masked) from head and tail entity pages, creating quadruples <head entity, tail entity, relation, text>. Approximately one-third of triples match corresponding text. We select these samples to form FB15K-T and FB15K237-T. Table 1 shows dataset statistics.

3.2 Evaluation Metrics

Relation prediction typically uses ranking-based evaluation: given test pair <h, t>, predict a ranked relation list l = [r_1, r_2, ..., r_n]. If the true relation r* ranks high, performance is good. We use MRR and Hits@n:

MRR = (1/N) Σ_{i=1}^N (1/rank_i) (3)
Hits@n = (1/N) Σ_{i=1}^N σ(rank_i $\leq$ n) (4)

where σ is the indicator function: σ(rank_i $\leq$ n) = 1 if rank_i $\leq$ n, else 0.

Both metrics range from 0 to 1, with higher values indicating better performance. Since entity pairs may have multiple correct relations, we compute raw (without removing other correct results) and filtered (removing other correct results) versions, following common practice.

3.3 Model Training

Training requires building a directed multigraph from training triples. We implement this in Neo4j: entities become nodes, triples become edges. Path generation uses Neo4j's DFS algorithm. Multiple paths may share the same relation sequence but differ in intermediate nodes; we merge these as they represent identical relation sequences. To improve efficiency, we limit DFS to the first m randomly matched paths per hop (m = 10, 20, 30 for 1-3 hops).

We generate 1-3 hop path sets for all training pairs, convert paths and texts to matrices, and train ConvF with parameters: batch_{size} = 50; dropout_{rate} = 0.5; learning_{rate} = 0.01; hidden layer neurons = 200; relation vector dimension = 20; text CNN kernels of heights 2, 3, 4, 5; 200 kernels for 2-hop and 3-hop paths; |R| kernels for 1-hop paths (considering only inverse relations). Matrix heights are set to 5, 20, 60 for 1-3 hops, accommodating up to 5, 10, 20 paths respectively. Early stopping based on validation loss reduces overfitting.

3.4 Results and Analysis

We compare ConvF against two translation models (TransE, TransH) and two semantic matching models (DistMult, ComplEx). Since baseline models use only triples, we train them on the first three elements of FB15K-T and FB15K237-T quadruples. ConvF uses complete quadruples. Tables 2 and 3 show results: ConvF significantly outperforms baselines on both datasets across all metrics.

A key question concerns individual feature contributions. We evaluate ConvF with different feature combinations on FB15K-T and FB15K237-T (Table 4). Among single features, 3-hop paths and text perform best. FB15K237-T removes direct inverse relations, so 1-hop features perform poorly. However, ConvF still achieves good performance using 2-hop, 3-hop, and text features. Multi-feature combinations generally outperform single features, with performance increasing as features accumulate. Using 1+2+3 hop path features alone achieves performance comparable to TransE and TransH, while adding text features surpasses mainstream models, demonstrating clear benefits from text integration.

4 Application Case

This section demonstrates ConvF's application and value using a person knowledge graph. Baidu Baike person entries contain some relationship information in infoboxes (Figure 8), but these are incomplete. For example, from Figure 8 we can infer a grandparent-grandchild relationship between "Li Xian" and "Li Zhi," though it's not explicitly annotated.

EncycloGraph is a large-scale person knowledge graph we built from Baidu Baike. We use ConvF to complete its person relations: (1) Crawl explicitly annotated person relationships from entries (43,682 pairs); (2) Generate path matrices from these pairs and text matrices from sentences containing both persons; (3) Train ConvF; (4) Predict relations for highly related but unlabeled pairs. If predicted probability exceeds a threshold, we annotate the relation.

This process predicted 106,770 person relations, enriching the knowledge graph with diverse relationship types. This enables dynamic generation of specific relation networks (e.g., kinship, collaboration, academic networks). Figure 9 shows the kinship network of important figures during the Southern Dynas-

ties (420-589 CE). ConvF effectively completes knowledge graphs and enhances knowledge service quality.

Conclusion

This study proposes a novel method fusing structural and textual features for knowledge graph relation prediction. Innovations include: (1) A CNN-based relation path pattern learning algorithm enabling more effective structural feature learning via deep learning; (2) ConvF, a feature fusion model integrating path structural and external textual features.

Experiments show ConvF outperforms state-of-the-art models on evaluation datasets, confirming its effectiveness. Feature analysis demonstrates that multi-feature fusion, especially combining structural and textual features, significantly improves performance. Applying ConvF to a person knowledge graph yielded good predictions, showing practical value. Limitations include dependence on path pattern learning (unsuitable for overly sparse graphs) and difficulty finding corresponding texts for some knowledge graphs. Future work will address these issues.

References

[1] Kejriwal M. Domain-specific knowledge graph construction [M]. Berlin: Springer, 2019.
[2] Sun Yusheng, Chang Kaiyue, Zhu Lijun. Research on large-scale knowledge graphs and their applications [J]. Information Studies: Theory & Application, 2018, 41(11): 138-143.
[3] Krompaß D, Baier S, Tresp V. Type-constrained representation learning in knowledge graphs [C]//Proceedings of the 14th International Semantic Web Conference. Berlin: Springer, 2015: 640-655.
[4] Fan M, Zhou Q, Zheng TF, et al. Distributed representation learning for knowledge graphs with entity descriptions [J]. Pattern Recognition Letters, 2017, 93(7): 31-37.
[5] Fudan University Knowledge Works Laboratory. Chinese General Encyclopedia Knowledge Graph (CN-DBpedia) [EB/OL]. [2020-07-21]. http://openkg.cn/dataset/cndbpedia.
[6] Nguyen DQ, Sirts K, Qu L, et al. Neighborhood mixture model for knowledge base completion [C]//Proceedings of the 28th AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2015: 2181-2187.
[7] Xu Zenglin, Sheng Yongpan, He Lirong, et al. A survey on knowledge graph technology [J]. Journal of University of Electronic Science and Technology of China, 2016, 45(4): 589-606.
[8] Liu Zhiyuan, Sun Maosong, Lin Yankai, et al. Research progress of knowledge representation learning [J]. Journal of Computer Research and Development, 2016, 53(2): 247-261.
[9] Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings

for modeling multi-relational data [C]//Proceedings of Advances in Neural Information Processing Systems. San Diego: Neural Information Processing Systems Foundation, 2013: 2787-2795.
[10] Wang Z, Zhang J, Feng J, et al. Knowledge graph embedding by translating on hyperplanes [C]//Proceedings of the 28th AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2014: 1112-1119.
[11] Lin Y, Liu Z, Sun M, et al. Learning entity and relation embeddings for knowledge graph completion [C]//Proceedings of the 28th AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2015: 2181-2187.
[12] Ji G, He S, Xu L, et al. Knowledge graph embedding via dynamic mapping matrix [C]//Proceedings of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on NLP. Stroudsburg: ACL Press, 2015: 687-696.
[13] Nie Binling. Research on knowledge representation learning methods based on graph structure information [D]. Hangzhou: Zhejiang University, 2019.
[14] Nickel M, Tresp V, Kriegel HP. A three-way model for collective learning on multi-relational data [C]//Proceedings of the 28th International Conference on Machine Learning. New York: ACM Press, 2011: 809-816.
[15] Yang B, Yih SW, He X, et al. Embedding entities and relations for learning and inference in knowledge bases [EB/OL]. [2020-08-16]. https://arxiv.org/pdf/1412.6575.
[16] Trouillon T, Welbl J, Riedel S, et al. Complex embeddings for simple link prediction [C]//Proceedings of the 33rd International Conference on Machine Learning. New York: ACM Press, 2016: 2071-2080.
[17] Lao N, Mitchell T, Cohen WW. Random walk inference and learning in a large scale knowledge base [C]//Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. Stroudsburg: ACL Press, 2011: 529-539.
[18] Liu Qiao, Han Minghao, Jiang Liu, et al. A double-layer random walk based relation inference algorithm [J]. Chinese Journal of Computers, 2017, 40(6): 1275-1290.
[19] Xu J, Chen K, Qiu X, et al. Knowledge graph representation with jointly structural and textual encoding [C]//Proceedings of the 26th International Joint Conference on Artificial Intelligence. San Rafael: Morgan & Claypool Publishers, 2017: 1318-1324.
[20] Long T, Lowe R, Cheung JC, et al. Leveraging lexical resources for learning entity embeddings in multi-relational data [EB/OL]. [2020-08-24]. https://arxiv.org/pdf/1605.05416.
[21] Dettmers T, Minervini P, Stenetorp PP, et al. Convolutional 2D knowledge graph embeddings [C]//Proceedings of the 32nd AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2018: 1811-1818.
[22] Hinton GE, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing co-adaptation of feature detectors [EB/OL]. [2020-05-22]. https://arxiv.org/pdf/1207.0580.
[23] Surdeanu M, Tibshirani J, Nallapati R, et al. Multi-instance multi-label learning for relation extraction [C]//Proceedings of the 2012 Joint Conference

on EMNLP and Computational Natural Language Learning. Stroudsburg: ACL Press, 2012: 455-465.
[24] Kim Y. Convolutional neural networks for sentence classification [C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Stroudsburg: ACL Press, 2014: 1746-1751.
[25] Adnan WA, Yaakob M, Anas R, et al. Artificial neural network for software reliability assessment [C]//Proceedings of Intelligent Systems and Technologies for the New Millennium. Piscataway: IEEE, 2000: 446-451.
[26] Toutanova K, Chen D. Observed versus latent features for knowledge base and text inference [C]//Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality. Stroudsburg: ACL Press, 2015: 57-66.
[27] Microsoft. FB15K-237 knowledge base completion dataset [EB/OL]. [2020-08-10]. https://www.microsoft.com/en-us/download/details.aspx?id=52312.
[28] Bisong E. Building machine learning and deep learning models on Google cloud platform [M]. Ottawa: Apress, 2019.
[29] Han X, Cao S, Lv X, et al. OpenKE: an open toolkit for knowledge embedding [C]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Stroudsburg: ACL Press, 2018: 139-144.
[30] Lin Zefei, Ou Shiyan. Extraction and analysis of large-scale person social networks from online encyclopedias [J]. Journal of Library Science in China, 2019, 45(6): 100-118.

Author Contributions:
Lin Zefei: Research design, data processing, and paper writing;
Ou Shiyan: Paper review, revision, and finalization.

Research on Relation Prediction in Knowledge Graphs by Fusing Structure and Text Features

Lin Zefei[1],[2], Ou Shiyan[1]
[1]School of Information Management, Nanjing University, Nanjing 210093
[2]College of Social Development, Fujian Normal University, Fuzhou 350007

Abstract: [Purpose/significance] Relation prediction is an important task in knowledge graph completion, which plays an important role in improving the completeness of knowledge in knowledge graphs. The paper proposes a new relation prediction method that combines internal structure features and external text features, which aims to predict the missing relations between two entities in knowledge graphs. [Method/process] The method transforms the relation paths in a knowledge graph and the texts that involve entity relationships into matrices, learns the structure features and text pattern features related to a specific relation type through convolutional neural networks, and then trains a model for relation prediction. [Result/conclusion] The results show that the performance of our proposed method on evaluation datasets is superior to the state-of-the-art approaches, and the method can effectively improve the performance of knowledge graph relationship prediction. Through practical application, it was found that this method has high application value in knowledge services.

Keywords: knowledge graph; relation prediction; feature fusion; deep learning

Note: Figure translations are in progress. See original paper for figures.