Applied Mathematics, Modeling and Computer Simulation
C.-H. Chen et al. (Eds.)

doi:10.3233/ATDE230978

A Deep Learning-Based Model for Relation Discovery in the Knowledge Graph of the Power Industry

Xiaoxue LUO[a], Jia WU[b], Zhifei NAN[a,1], Zhenhua LIU[a] and Jianxin WANG[a]
[a] *State Grid Langfang Power Supply Co., Langfang (065014), Hebei, China*
[b] *State Grid Jibei Electric Power Co., Ltd., Beijing (100761), China*

Abstract. In the field of the electric power industry, improving the entity relation model can increase the efficiency of knowledge graph completion, build a smart grid, and provide strong support for relevant regulatory decisions. Therefore, we propose a deep learning-based entity relation discovery model called BERT-SRR for the electric power domain. This model includes constructing an electric power domain knowledge base using professional domain knowledge and adding entities and their relationships to it. It utilizes a BERT model trained incrementally to extract Chinese text features and conducts sequence labeling learning with entity and relationship information from the knowledge graph. Furthermore, combining a stacked convolutional neural network and a student reordering network enables high-accuracy knowledge graph completion and prediction of entity relationships. The comparative experimental results demonstrate that the proposed model outperforms the benchmark model, showing a significant improvement in F1 score ranging from 3.66% to 9.78%.

Keywords. Power industry; entity relation discovery; knowledge graph.

1. Introduction

With the continuous development of science and technology and the deepening of enterprise digital transformation, the construction of digital grids and smart grids has become an important focus in China's power energy field. Against this background, it is particularly important to use artificial intelligence technology for technical support and theoretical research. The automation of public opinion monitoring can help relevant institutions timely understand the public opinion information and development trends in the power industry, so as to adjust production work and warn against risks. In order to condense wisdom and acquire more relevant domain resources, building a smart grid requires turning unstructured data into knowledge. Traditional knowledge graphs have difficulty meeting the rapidly growing demand for information, and Entity Relationship Discovery (ERD), as one of the basic and core tasks of knowledge graph completion technology, is crucial for transforming information into knowledge. In terms of news reporting in the power industry, improving the entity relationship model

1 Corresponding Author, Zhifei Nan, State Grid Langfang Power Supply Co., Langfang (065014), Heibei, China; E-mail: 106271107@qq.com.

can enhance the integrity of knowledge graph, assist in the smart grid's construction, and offer strong support for relevant regulatory decisions, which has great significance.

Therefore, this article proposes a deep learning-based model for entity relationship discovery in the field of electric power. The model first uses a BERT model trained incrementally to extract Chinese text features and then combines stacked convolutional neural networks and student reordering networks to obtain the final scores for candidate entities. This approach facilitates accurate knowledge graph completion and enables effective prediction of relationships between entities.

2. Related Work

The task of knowledge graph completion involves predicting missing entities or relations. To accomplish this, entities and relations are encoded into low-dimensional representations, as well as predicted results are then incorporated into the graph, thereby improving its overall completeness. Existing models for knowledge graph completion can be categorized into translation-based models and neural network-based models. The former can be further classified as translation-based (e.g., TransE [1]) and rotation-based (e.g., RotatE [2]) models. However, these simple models are limited in handling complex relationships like one-to-many or many-to-many. To address this limitation, TransH [3] introduces hyperplanes for each relationship, mapping entities onto these hyperplanes for computation. On the other hand, TransR [4] models entities and relationships in both entity space and multiple relationship spaces. Rotation-based translation models such as RotatE deal with more intricate relationships by considering each relationship as a rotation in complex space. To combine and improve upon these existing models, researchers have proposed methods like STransE [5]. The STransE model merges the SE and TransE models, representing each entity as a low-dimensional vector, and defining each relationship using two matrices and a translation vector. Although STransE performs better than TransE, it is also more prone to data sparsity. Some people proposed a novel temporal rotation model, ChronoR [6], inspired by RotatE, which captures rich interaction information by learning a high-dimensional rotational transformation operator parameterized by both relations and time. Some people encoded quadruples into the real space and implement automatic learning through entity type embedding, effectively enriching the general characteristics of entities [7].

However, with the development of neural networks, there have been many studies applying neural network models to knowledge graph completion. Compared with linear completion models, non-linear neural network models can obtain better features. The ConvE [8] was the first to use convolutional neural networks to complete graphs by reconstructing head entities and relationships as two-dimensional matrices. It then calculates with matrix W and tail entities to determine the credibility of the current factual triple. The ConvKB [9] improved on ConvE by treating triples as a whole and concatenating them into three columns of matrices as convolution inputs, rather than just concatenating individual relationships and entities, to achieve interaction between entities and relationships. With the development of deep learning, graph convolutional network models have overcome the limitations of convolutional neural networks and can aggregate data in the knowledge graph through mapping functions to capture self and neighborhood feature information. Some scholars have defined neighborhood encoders to aggregate direct neighbor information of entities with equal weight and then use long-short-term memory networks for multi-step entity-to-entity matching to achieve relationship prediction [10]. Other researchers have proposed dynamic attribute concepts under few-sample situations and learned dynamic representations using the adaptive attention network FAAN to help solve the problem of only obtaining static entity representations under different relations [11].

In recent years, entity relation extraction techniques have been widely applied in the field of knowledge graph completion. However, there are still issues and limitations in practical applications. Accurate identification and localization of entities and their relationships often require auxiliary information, such as context and relevant knowledge base. This can help algorithms better understand entities and relationships and improve the accuracy of entity relation extraction. However, errors or failures to recognize may occur due to the lack of necessary information, which leads to the inability to build a knowledge graph. Furthermore, entity relations typically involve multiple attributes such as time, location, and causality, and may also span multiple sentences and paragraphs in the electricity industry, which makes entity relation extraction more difficult and increases algorithm complexity, affecting accuracy. In the construction of knowledge graphs in the electricity industry, the accuracy of entity relation extraction is crucial because errors in entity relations will gradually accumulate, ultimately leading to inaccurate and incomplete knowledge graphs. Based on these challenges, this paper proposes using incremental training to extract text features to address the problem of large vocabularies in specialized domain corpora. It calculates candidate entity scores by combining stacked convolutional neural networks with student reordering networks to extract entity relations more accurately, thereby achieving accurate, efficient, and automated mining of entity relationships and enabling the rapid and accurate maintenance and updating of knowledge graphs.

3. A Deep Learning-Based Entity Relation Discovery Model for the Power Industry

The entity relation discovery model utilizes BERT to compute entity embeddings at the beginning. These pre-computed entity embeddings are then combined with learned relation embeddings and projected onto a two-dimensional spatial feature map. The feature map is processed using 2D convolution sequences. Afterward, the resulting feature map is subjected to average pooling, merging, and projection operations to obtain a query vector. This query vector is utilized to rank and sort candidate entities. To generate the final candidate ranking, the model selects the top ten candidates with the highest scores. Additionally, the re-ranking model utilizes the insights and information obtained according to the initial result to improve the overall ranking performance. Refer to figure 1 for a visual representation of this process. The combination of the candidate entity ranking and the student re-ranking model produces the ultimate candidate ranking.

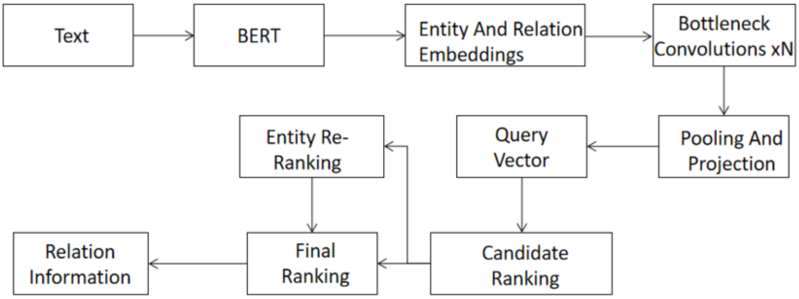

Figure 1. Entity relation discovery model.

To complete a knowledge graph by predicting missing triples, the first step involves representing entities and relations in a low-dimensional space. Then, their features are extracted using a stacked convolutional neural network. In the context of an incomplete triple (e_i, r_j, ?), the embedding of the entity and relation $e \in R^d$, $r \in R^d$ are concatenated to create a feature vector $q \in R^{2d}$ with two channels, each having a length of d. To reduce redundancy, a one-dimensional convolutional kernel with a width of 1 is utilized to perform projections along the feature vector's length. This projection maps each position to a 2-d spatial feature map, represented as $g_i \in R^{f \times f}$, where the convolution entails $f \times f$ filters. Consequently, model generates a spatial feature map G $\in R^{f \times f \times d}$ with d channels, which is utilized to represent partial query triple (e_i, r_j, ?).

Next, a series of 2D convolutions are performed on the feature maps, which is a square image with dimensions $f \times f \times d$, where f is the edge length and d is the number of channels. The convolutional network uses convolution layers consisting of three bottleneck blocks, where N is a hyperparameter that controls the depth of the network. To streamline the explanation, the bottleneck block involves 3 CNN layers: A 1×1 CNN layer, a 3×3 CNN layer, and the other 1×1 CNN layer. The initial 1×1 CNN layer decreases dimension of the feature map by a quarter, while the second 1×1 CNN layer restores it to the original dimension. Hence, the bottleneck block compresses and then expands the feature map, effectively reducing its dimensionality. At last, the dimension of the bottleneck block feature is doubled, resulting in a final feature map with a dimension of 4D.

To increase the depth of the network, residual connections are added to each bottleneck block, and batch normalization and ReLU non-linear functions are applied before each convolution layer. To preserve the spatial size of the feature map, 3×3 convolutions, and cyclic padding are employed. The convolutions are performed with a stride of 1 for all layers. The bottleneck block in the model is designed to increase the dimensionality of the feature map two-fold. Within this block, a projection shortcut is employed for the residual connection. By employing the projection shortcut, the feature map from the previous layer is projected to match the increased dimensionality of the feature map in the bottleneck block.

Finally, the network generates a feature map $G \in R^{f \times f \times 4d}$, which undergoes global pooling across the spatial dimensions to obtain the next feature vector $G \in R^{4d}$. Subsequently, we employ a fully connected layer followed by the Rectified Linear Unit (ReLU) activation function to project the feature vector back to its original embedding dimension, denoted as d. The resulting representation, denoted as e, is then used to compute the scores of candidate entities by taking the dot product with their respective embeddings. To efficiently compute scores for all entities simultaneously, the embedding matrix $y = e^T$ is employed.

During the training process, a one-vs-all strategy is utilized, employing binary cross-entropy loss. Each fact (e_i, r_j, e_k) in the dataset, is treated as a training example with (e_i, r_j, ?) serving as the input for BERT-SRR. Using the aforementioned method, scores are computed for all entities, and probabilities are obtained by applying a sigmoid operation. All entities except for e_k are considered negative candidates, and binary cross-entropy loss is calculated.

To improve the prediction accuracy, the most likely candidate entities are extracted and retrained using the knowledge from the original ranking model and re-ranking model. Thus, the ultimate ranking is obtained by these two models. The re-ranking process involves top-m predicted triples from the knowledge graph completion model, which is based on stacked convolution. For the purpose of re-ranking, a student re-ranking model is employed as a three-way classification network. This network incorporates complete candidate fact (e_i, r_j, e_k) and allows the modeling of interactions among all elements of the triple.

To integrate relationship information, a relationship label is assigned to each relation in the knowledge graph. For constructing the text input, the relationship label is inserted before the head entity and tail entity. Subsequently, these two sequences are concatenated to form the input representation. To illustrate, the triple ("head entity name", r_i, "tail entity name") is represented as "[CLS][REL_i]head entity name[SEP][REL_i]tail entity name[SEP]". To obtain the final feature representation for prediction, we calculate the linear combination of the [CLS] embeddings learned from each layer.

For each training query n, the top-m candidate triples $f_T(g_n)_{0:m}$ are normalized using the teacher model $f_T(g_n)$, as shown in Equation (1).

$$s_{nm:(n+1)m} = soft\,max(f_T(g_n)_{0:m}/T) \tag{1}$$

The training objective of the student model $f_S(g_n)$ is to compute the function L_{bce}, utilizing the normalization s from the teacher model and noise-adding training labels y, as shown in Equation (2), to train the student model.

$$\begin{aligned} L_{KD}(y_n, g_n) &= \lambda L_{bce}(s_n, f_S(g_n)) + (\lambda - 1) L_{bce}(y_n, f_S(g_n)) \\ &= L_{bce}((\lambda - 1) y_n + \lambda s_n, f_S(g_n)) \end{aligned} \tag{2}$$

Finally, the student re-ranking and teacher networks are used to compute the final ranking. To obtain the final ranking for each query, we employ the re-ranking model on the top-m=15 triples. In order to calculate the ultimate outcome, we combine the predictions from both the teacher and student networks, as illustrated in Equation (3).

$$\hat{s}_{nm:(n+1)m} = b(soft\,max(f_S(g_{nm:(n+1)m}))) + (1 - b)(soft\,max(f_T(g_n)_{0:m})) \tag{3}$$

Here, the value of b, where $0 \leqslant b \leqslant 1$, influences the student re-ranking network, and we conduct experiments with a step size of 0.05 to incrementally increase b until we find the optimal performance.

4. Experimental Process and Results Analysis

4.1. Experimental Data

In this paper, we used text from news reports related to the power industry collected by web crawlers and performed data preprocessing and manual annotation to generate a dataset named SRR-DATA. The dataset is divided into three parts: training set, validation set, and testing set, as shown in table 1.

Table 1. Experimental dataset.

Data set	Training set	Test set	Validation set
Dialogue	3204	832	524

4.2. Experimental Conclusions

This paper compares our entity relation discovery model with other classic neural network models, as shown in table 2.

Table 2. Experimental Results of entity relation discovery.

Model	PEMN-DATA		
	P%	R%	F1%
ConvE	84.23	80.37	82.25
BERT-ConvE	86.41	85.90	86.15
BERT-ConvTransE	87.12	89.66	88.37
BERT-SRR	91.79	92.27	92.03

From Table 2, it can be seen that by preprocessing the text of power-related news reports gathered by web crawlers and enriching the existing public domain knowledge base with specialized knowledge of the power industry, we constructed a power industry-specific resource library. We then extracted Chinese text features using BERT neural networks that were trained incrementally and encoded the triples in the knowledge graph using multiple convolutional layers. These convolutional layers utilize different kernel sizes and numbers to capture information in the triples from different perspectives. The output of the convolutional layers is concatenated into a vector and then predicted using a fully connected layer. The student re-ranking network uses the output as input and makes further predictions, leading to more precise identification of relational information in the text. The experimental results show that the entity relation discovery model BERT-SRR used in this paper not only improves precision and recall but also achieves the highest F1 score.

5. Conclusion

This paper proposes a deep learning-based entity relation discovery model, BERT-SRR, for the power industry domain. The BERT neural network, which is trained incrementally, extracts Chinese text features, enabling better representation of specialized vocabulary in the power industry. Multiple convolutional layers are used to encode triples from the knowledge graph. These layers utilize different kernel sizes and quantities to capture information from triples at different perspectives. A student model is obtained through knowledge distillation, and the result ranking is further optimized to achieve more accurate identification of relational information in the text. The proposed entity relation discovery model for the power industry performs better in both efficiency and accuracy compared to other classic models on the self-built dataset.

Acknowledgments

This work was supported by the Project: Adaptive improvement of artificial intelligence model for production application (SGJBLF00FCJS2250073).

References

[1] Bordes A, Usunier N, Garcia-Duran A, et al. Translating Embeddings for Modeling Multi-relational Data [C]. Neural Information Processing Systems, 2013.
[2] Zhen W, Zhang J, Feng J, et al. Knowledge Graph Embedding by Translating on Hyperplanes [C]. National Conference on Artificial Intelligence, 2014.
[3] Lin Y, Liu Z, Sun M, et al. Learning entity and relation embeddings for knowledge graph completion [C]. Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. Austin, Texas: AAAI Press, 2015: 2181-2187.
[4] Sun Z, Deng Z H, Nie J Y, et al. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space [J/OL], arXiv, 2019.
[5] Nguyen D Q, Sirts K, Qu L, et al. STransE: A novel embedding model of entities and relationships in knowledge bases [C]. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego, California: Association for Computational Linguistics, 2016: 460-466.
[6] Sadeghian A, Armandpour M, Colas A, et al. ChronoR: Rotation Based Temporal Knowledge Graph Embedding [C]. Proceedings of the AAAI Conference on Artificial Intelligence: 2021, 35: 6471-6479.
[7] Niu G, Li B, Zhang Y, et al. Findings of the Association for Computational Linguistics: EMNLP 2020 [C/OL]. Online: Association for Computational Linguistics, 2020: 1172-1181.
[8] Dettmers T, Minervini P, Stenetorp P, et al. Convolutional 2D Knowledge Graph Embeddings[C]//Proceedings of the AAAI Conference on Artificial Intelligence: 2018, 32.
[9] Nguyen D Q, Nguyen T D, Nguyen D Q, et al. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network [C]. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). New Orleans, Louisiana: Association for Computational Linguistics, 2018: 327-333.
[10] Xiong W, Mo Y, Chang S, et al. One-Shot Relational Learning for Knowledge Graphs [C/OL]. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: Association for Computational Linguistics, 2018: 1980-1990.
[11] Sheng J, Guo S, Chen Z, et al. Adaptive Attentional Network for Few-Shot Knowledge Graph Completion [C]. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Online: Association for Computational Linguistics, 2020: 1681-1691.