Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 4, No. 1, 2023

Entity Linking based on RoFormer-Sim for Chinese Short Texts

Weiting Xie

Southwest Minzu University, Chengdu 610200, China

Abstract: Entity linking is an important means to identify named entities in text and a key technology for constructing knowledge graphs, playing an important role in fields such as intelligent question answering and information retrieval. However, existing entity linking methods for short texts have low accuracy due to the lack of rich contextual information, informal expression, and incomplete grammar structures. Therefore, this paper proposes a short-text entity linking model based on the RoFormer-Sim pre-training model. Firstly, entity context features are extracted by the RoFormer-Sim pre-training model, and then text similarity calculation and sorting are performed with candidate entity description texts to obtain the corresponding entity in the knowledge base with the disambiguated entity. The experimental results show that the RoFormer-Sim model can provide prior knowledge for entity linking, and the proposed model in this paper has an F1 value of 0.8851, which is better than other entity linking models based on other pre-training models.

Keywords: Entity Linking; Candidate Entity; Text Similarity.

1. Introduction

Entity Linking (EL) is an important task in the field of natural language processing. The task typically involves identifying entity mentions in a piece of text and linking them to corresponding target entities in a knowledge base. Entity linking can be seen as consisting of two subtasks: candidate entity retrieval and candidate entity ranking. Candidate entity retrieval involves querying the knowledge base for entities that match the entity mentions to be linked, while candidate entity ranking is a key focus of the entity linking process. This involves calculating and ranking the relevance of all candidate entities in the candidate entity set to the entity mention, and selecting the candidate entity with the highest relevance score as the entity linked to the entity mention [1].

Currently, in entity linking tasks, linking entities in short texts is more challenging than in long texts because long texts provide rich semantic information that can provide effective entity context information to improve entity linking accuracy [2]. For short texts, the lack of rich contextual information makes entity disambiguation difficult. Especially in Chinese texts, due to the variability of grammar and richness of word meanings, it is necessary to have a precise understanding of the entity context. Currently, there are various methods to achieve entity linking, such as rule-based methods, machine learning-based methods, and deep learning methods. Rule-based methods typically rely on manually defined rules and rule libraries to match and link entities. Machine learning-based methods learn entity linking models from large-scale annotated data by constructing features, selecting models, and training them. Deep learning methods learn features and train models through neural network structures to link entities. Traditional rule-based and machine learning-based methods may not accurately capture the semantic information of text and entities, and their generalization ability needs to be improved because they require manually designed features or shallow models for feature extraction, as well as hand-designed rules or the selection of specific algorithms and models. Deep learning methods automatically learn deep feature representations of text and entities, making semantic understanding of text and entities more accurate and comprehensive. They also have stronger generalization abilities and can adapt to more complex and varied entity linking tasks through large-scale training data and powerful model expression abilities [3-5].

With the continuous development of deep learning and natural language processing technologies, significant progress has been made in the field of Chinese entity linking. For example, researchers have proposed models based on convolutional neural networks (CNN) [6] and recurrent neural networks (RNN) [7], as well as methods for entity linking using pre-trained language models such as BERT [8] and ELMo [9], which have achieved high linking accuracy and robustness in entity linking tasks. Zhou Pengcheng [10] and others used N-grams combined with part-of-speech tagging and multiple mention-entity dictionaries to obtain candidate entities, calculated the relevance of entity sequences through multiple knowledge bases (Wikipedia and Freebase) and entity categories, and selected the entity with the highest relevance as the linking result. Zeng et al. [11] proposed a short-text entity linking method based on a bi-attentional LSTM network, but the LSTM model cannot capture backward semantic information and has more model parameters. Hu et al. [12] then constructed a symmetric BiLSTM model with dual attention mechanisms, which extracts entity semantic features more comprehensively by utilizing structural information and attention mechanisms, and BiLSTM can better capture bidirectional semantic dependencies. Devlin et al. [13] proposed the pre-trained model BERT, which performs well in various natural language processing tasks and has been used by many researchers in entity linking tasks. Chen et al. [14] modeled the context information of entities in knowledge bases using BERT, and the entity embedding based on BERT can better capture entity type information. Cheng et al. [15] proposed the BERT-ENE model based on BERT and treated the short-text entity linking problem as a binary classification task. Zhao et al. [16] proposed a method combining BERT with graph models, which can better capture the relevance between entity mentions and candidate entities.

With the rapid development of the Internet and digital information, a large amount of Chinese textual data has emerged, making it a key problem in applications such as information extraction, information retrieval, and semantic search to efficiently identify and link entities from these texts. Given the current issues of colloquialization and lack of rich context in Chinese short texts, this paper proposes a Chinese short text entity linking method based on the RoFormer-Sim pre-training model, which improves the accuracy of entity linking.

The main contributions of this paper are as follows:

(1) Proposing a Chinese short-text entity linking method based on the RoFormer-Sim pre-training model. The RoFormer-Sim pre-training model is used to extract contextual features, which are then used to calculate text similarity with candidate entity description text. The similarity scores are then sorted, and the entity with the highest score is selected as the correct linked entity.

(2) By using the RoFormer-Sim pre-training model, prior knowledge is provided for the entity linking task, achieving a significant improvement with an F1 score of 0.8851 on the CCKS2020 dataset.

2. Methodology

2.1. Short Text Entity Linking Process

Entity linking aims to map the entity mentions mentioned in the text to the corresponding entities in the given knowledge base. The purpose is to solve the problem of ambiguity caused by entity mentions in the text corresponding to entities in different contexts. This article divides the short text entity linking task into two stages: candidate entity generation and candidate entity ranking, as shown in Figure 1.

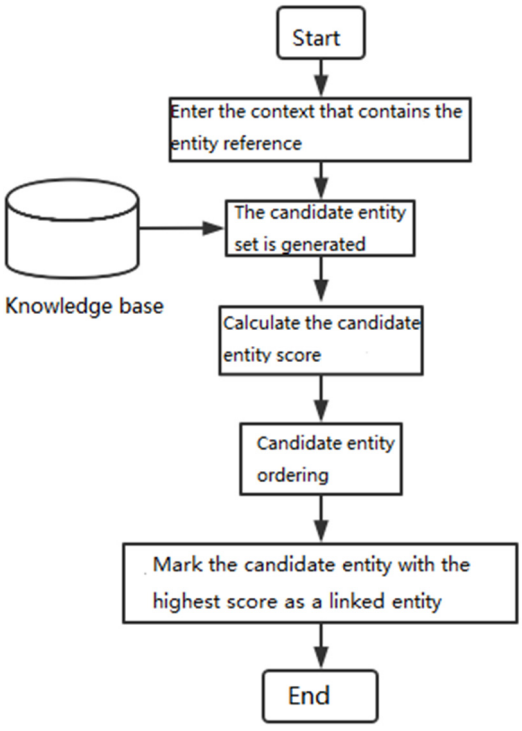

Figure 1. Short Text Entity Linking Workflow

The essence of entity linking is essentially to calculate the text similarity between the short text containing the entity mention and the candidate entity description text, and to sort the similarity scores of the candidate entities, so as to link the entities to the corresponding entities in the knowledge base. That is, the short text containing the entity mention and the entity description text are input to the model as a text pair, then the scores of the candidate entities are calculated and sorted, and finally the candidate entity with the highest score is selected as the linked entity.

2.2. Pre-trained Language Model RoFormer-Sim

RoFormer-Sim model combines retrieval and generation and was released by the team led by Jianlin Su in June 2021. Before RoFormer-Sim, the authors also proposed a similar model called SimBERT. RoFormer-Sim is a further integration and optimization of SimBERT-related techniques and can be used to automatically generate similar questions and as an auxiliary data augmentation tool.

The RoFormer model improves the position encoding of the transformer by using Rotary Position Embedding (RoPE) instead of absolute position encoding in the transformer. This is a design that can achieve relative position encoding using the attention mechanism in combination with the "absolute position encoding" method. Generally speaking, absolute position encoding has the advantages of simple implementation and fast computation speed, while relative position encoding directly reflects relative position signals and often has better performance in practice.

The main framework of RoFormer-Sim is the RoFormer model. In addition, RoFormer-Sim uses more training data and is not limited to only question sentences like SimBERT. RoFormer-Sim can be used to generate similar sentences for general sentences, making it more versatile. Other training details include the use of a larger batch size and maxlen in RoFormer-Sim.

The key to both SimBERT and RoFormer-Sim lies in the construction of training corpus. The training corpus of RoFormer-Sim includes two parts: similar question types and general similar types. For similar question types, similar questions are collected from Baidu Knows, and then further cleaned by rules like SimBERT. For general similar types, since there is no existing data to collect, two unsupervised methods are proposed to construct similar sentence pairs.

The first method is based on the idea that "the answers to the same question are similar". Assuming that there is a question-and-answer corpus with multiple answers to the same question, each answer can be divided into sentences, and then a pre-existing similarity function can be used to compare the similarity between answers. Pairs of sentences with similarity scores above a certain threshold are selected as similar sentence pairs.

The second method is based on the idea that "sentences in the same passage are similar". Each passage is divided into sentences, and a pre-existing similarity function is used to calculate the similarity between each pair of sentences. Pairs of sentences with similarity scores above a certain threshold are selected as similar sentence pairs.

The similarity function is a variant of Jaccard similarity that directly uses a rule-based character-level similarity, and semantic relevance is obtained through the intra-chapter correlation and the generalization ability of the pre-trained model itself. Through the first approach, about 4.5 million (pseudo) similar sentence pairs were constructed from several reading comprehension datasets; through the second approach, about 4.7 million (pseudo) similar sentence pairs were constructed from over 30GB of parallel corpus; and

about 30 million groups of similar sentences were crawled (each group can form multiple pairs). It can be seen that the number of question sentences far exceeds that of general sentences, so they were sampled in a 1:1 ratio to ensure balance between each type of sample.

During training, in order to enhance the generation ability of the model, a portion of the input tokens in the input sentence are randomly replaced with [MASK], which is a pre-training method first proposed by BERT. However, unlike BERT which inputs noisy sentences and outputs the original sentence, RoFormer-Sim inputs noisy sentences and outputs a similar sentence to the original one.

Figure 2. Diagram of SimBERT training method.

After training RoFormer-Sim, the retrieval performance of SimBERT is further transferred to RoFormer-Sim through distillation, so that the retrieval performance of RoFormer-Sim is basically on par with or even better than that of SimBERT. The distillation method is simple: if the sentence vectors generated by SimBERT for a batch of sentences are u1, u2, ..., un, and the sentence vectors generated by RoFormer-Sim are v1, v2, ..., vn, then:

$$L = \frac{\lambda}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\cos(u_i,u_j)-\cos(v_i,v_j))^2 \qquad (1)$$

To prevent the model from "forgetting" the generation model, a generation loss is also added during distillation, L=Lsim+Lgen.

RoFormer-Sim is based on the RoFormer architecture and trained using a large number of similar pairs of sentences to perform sentence expansion. There are two tasks during training: generation and retrieval. In the generation task, similar to BERT, a noisy input sentence is provided, and the model outputs a similar sentence to the original sentence. In the retrieval task, the retrieval performance of SimBERT is transferred to RoFormer-Sim through distillation.

2.3. Candidate Entity Generation

The generation of candidate entities is a crucial prerequisite for entity linking. The main goal of candidate entity generation is to obtain a candidate entity set for each entity mention. The selection of candidate entities has a great impact on the accuracy of the entity linking model. The generation of candidate entities first requires a given entity mention, and then a candidate entity list corresponding to the entity mention is obtained based on knowledge, rules, and other information. The quality of the candidate entity set depends on the size of the candidate entity set and whether it contains the target entity [17]. The number of candidate entity sets cannot be too many or too few, as too many can lead to low experimental efficiency and too few may result in the target entity not being included.

This article generates candidate entities by constructing a query dictionary. Specifically, the main process of this method is as follows:

1.Generate query dictionaries. Preprocess the knowledge base and generate two query dictionaries. One dictionary is used to search for the corresponding entity object in the knowledge base through the entity ID, and the other dictionary is used to search for the corresponding entity object through the entity mention name, with the key of the dictionary being the entity name and the value being the set of IDs of the entities with the same name.

2.Generate candidate entity set. After generating the query dictionary, it is necessary to recall the candidate entity set to find candidate entities that may be related to the entities mentioned in the short text. For each entity mention appearing in the short text, the ID set of the corresponding entity in the knowledge base can be extracted through the two query dictionaries in turn, thereby generating the candidate entity set.

2.4. Entity Linking Model Construction

The Chinese short-text entity linking model proposed in this paper based on the RoFormer-Sim pre-training model is shown in Figure 3. The model combines the short text with added entity mention position tags and the candidate entity description text as input and outputs the similarity score between the two. The entity linking model proposed in this paper first uses the RoFormer-Sim pre-training model to extract contextual features and then uses a fully connected layer to obtain the final result.

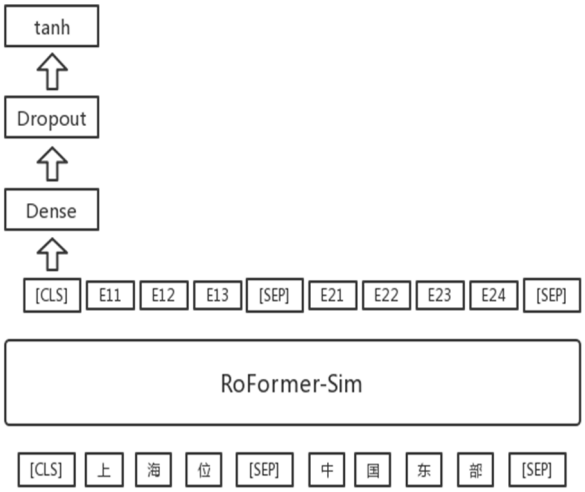

Figure 3. Chinese short-text entity linking model based on RoFormer-Sim pre-training model

For the entity linking task, in order to improve the generalization ability of the model and make it more robust in new datasets and scenarios, this paper treats it as a binary classification problem. This paper adds position markers "#" to each entity mention to indicate the position of the entity to be disambiguated. The short text with entity mentions position markers and the candidate entity description text are combined and input into the model as a text pair. The RoFormer-Sim model is used to extract contextual features, and then the feature vector of the CLS position output by the RoFormer-Sim model is input into a fully connected layer. Then, the candidate entity similarity score is mapped to the range of [-1, 1] using the tanh function after the dropout layer, so that the similarity scores can be normalized and the similarity between different entity mentions and candidate entities can be compared more fairly. The model then outputs the scores of each candidate entity and sorts them, and the candidate entity with the highest score is the object referred to by the entity to be disambiguated.

3. Experimental Results

3.1. Experimental Dataset

This article uses the dataset provided by the CCKS2020 (the 2020 National Knowledge Graph and Semantic Computing Conference) short text entity linking evaluation. The CCKS2020 evaluation dataset includes a standard dataset and a knowledge base. The knowledge base comes from the Baidu Encyclopedia knowledge base. Each entity in the knowledge base contains a subject_id (knowledge base ID), a subject name, entity aliases, corresponding concept types, and a series of binary tuples <predicate, object> (<attribute, attribute value>) information related to the entity.

The standard dataset consists of training set, validation set, and test set, with a total of about 90,000 annotated data points, distributed in a 7:1:1 ratio. All three sets of data were generated through crowdsourcing by Baidu. The annotated data set mainly comes from real internet web page titles, video titles, and user search queries. Each annotated data point contains a piece of text, an entity mentions in the text, and the target entity corresponding to the entity mention in the given knowledge base. One training data point may contain multiple entity mentions. The specific annotated data is shown in Table 1.

Table 1. Examples of Specific Annotation Data

Number	Text	Mention information
2484	医疗保险占工资的比例	{"kb_id": "17855", "mention": "医疗保险", "offset": "0"} {"kb_id": "55578", "mention": "工资", "offset": "5"} {"kb_id": "117247", "mention": "比例", "offset": "8"}
2519	《刘宽》答案解析及翻译	{"kb_id": "119182", "mention": "刘宽", "offset": "1"}, {"kb_id": "156318", "mention": "翻译", "offset": "9"}
2540	新疆拉面的做法视频	{"kb_id": "48047", "mention": "新疆", "offset": "0"} {"kb_id": "219021", "mention": "拉面", "offset": "2"} {"kb_id": "136981", "mention": "做法", "offset": "5"} {"kb_id": "37477", "mention": "视频", "offset": "7"}

3.2. Evaluation Metrics

Given a short text input Text with N entity mentions: M_{Text}= {m_1, m_2, …, m_N}, where each entity mention is linked to an entity id in the knowledge base E_{Text}= {e_1, e_2, … ,e_N}, and the output of entity linking is E'Text={e'1,e'2,…,e'N}, the precision, recall, and F1-score of entity linking are defined as follows:

$$P = \frac{\sum_{n\in N}\left|E_n \cap E'_n\right|}{\sum_{n\in N}\left|E'_n\right|} \tag{2}$$

$$R = \frac{\sum_{n\in N}\left|E_n \cap E'_n\right|}{\sum_{n\in N}\left|E_n\right|} \tag{3}$$

$$F1 = \frac{2(P\times R)}{P+R} \tag{4}$$

Since the entity mentions in the standard dataset are given, we have =, P = R = F1, and use F1 score as the main evaluation metric for the experimental results.

3.3. Experimental Environment and Parameter Settings

In this experiment, we rely on the Python programming language, which contains a wealth of deep learning libraries and is easy to debug. We use the PyTorch deep learning framework to build the overall network. In addition, we use a single GPU to train the model and operate on the VSCode compilation platform based on the Ubuntu18.04 operating system. The specific experimental hardware and software environment are shown in Table 2.

Table 2. Experimental Hardware and Software Environment

Device	Configuration
Operating System	Ubuntu18.04
GPU	GeForce RTX 2080Ti
Deep Learning Framework	bert4torch
Programming Language	Python3.7
Compilation Platform	VSCode

The detailed parameters of the proposed entity linking model in this paper are as follows: the learning rate is 2×10^{-5}, the maximum sequence length is 512, the training batch size is 8, and the Adam optimization algorithm is used.

3.4. Result Analysis

To demonstrate the superior performance of the RoFormer-Sim model, experiments were conducted using different pre-trained models on the same dataset and their effects were compared. Specifically, the semantic information of the text was first extracted using a pre-trained model, and then the semantic information was fed into a fully connected layer for prediction. The batch_size of the model training was set to 8, the initial learning rate was 2e-5, the maximum sequence length was 512, and a learning rate decay strategy based on exponential decay was used. The F1 values of the models obtained using different pre-trained models are shown in Table 3.

Table 3. F1 scores using different pre-trained models

model	F1
BERT	0.7644
Roberta	0.8139
ERNIE1.0	0.8501
RoFormer-Sim	0.8851

Based on the experimental results, the model performance is in the order of RoFormer-Sim > ERNIE1.0 > Roberta > BERT. The F1 scores of RoFormer-Sim model are 0.1207, 0.0712, and 0.035 higher than those of BERT, Roberta, and ERNIE1.0 models, respectively. The essence of the entity disambiguation task is to calculate text similarity, and RoFormer-Sim model is trained on corpora that include both question-type similar sentences and general-type similar

sentences, which can provide prior knowledge for entity disambiguation tasks. Therefore, RoFormer-Sim can improve the performance of entity linking models.

4. Conclusion

This paper aims to construct an entity linking model based on the RoFormer-Sim pre-training model for Chinese short texts and verifies it on the dataset provided by the CCKS2020 (National Knowledge Graph and Semantic Computing Conference 2020) short text entity linking evaluation. The experimental results show that the RoFormer-Sim pre-training model can provide prior knowledge in the process of entity disambiguation, which improves the linking effect. In the future, the effectiveness of the proposed model will be verified on other public entity linking datasets, and the methods for entity recognition and entity disambiguation will be optimized to further improve the accuracy of Chinese short text entity linking.

Acknowledgments

This project is supported by the Southwest University for Nationalities Graduate Innovative Scientific Research Project (Project No. YB2023145).

References

[1] ZHAN Fei, ZHU Yanhui, LIANG Wentong, et al. Multi-task learning-based short text entity linking method. Computer Engineering, 2022, 48(3): 315-320.

[2] Zhang Shengqi, Wang Yuanlong, Li Ru, et al. Chinese short text entity linking based on local attention mechanism [J]. Computer Engineering, 2021, 47(11): 77-83, 92.

[3] Lample G, Ballesteros M, Subramanian S, et al. Neural architectures for named entity recognition[C]. North American Chapter Of The Association For Computational Linguistics, 2016: 260-270.

[4] PETERS M E,NEUMANN M,IYYER M,et al. Deep contextualized word representations. NAACL-HLT[J]. 2018. [J]. ar Xiv preprint ar Xiv:1802.05365,2018.

[5] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.

[6] Lipton, Zachary C., John Berkowitz, and Charles Elkan. "A critical review of recurrent neural networks for sequence learning." arXiv preprint arXiv:1506.00019 (2015).

[7] Devlin J, Chang M W, LeeK, et al. BERT: pre- training of deep bidirectional transformers for language understanding [J]. arXiv preprint, arXiv: 1810. 04805,2018.

[8] Ilić, Suzana, et al. "Deep contextualized word representations for detecting sarcasm and irony." arXiv preprint arXiv: 1809. 09795 (2018).

[9] Zhou Pengcheng, Wu Chuan, Lu Wei. Short Text Entity Linking Based on Multiple Knowledge Bases: A Case Study of Wikipedia and Freebase[J]. New Technology of Library and Information Service, 2016, 32(06): 1-11.

[10] Zeng W, Tang J, Zhao X. Entity linking on Chinese microblogs via deep neural network[J]. IEEE Access, 2018, 6: 25908-25920.

[11] Hu S, Tan Z, Zeng W, et al. Entity linking via symmetrical attention-based neural network and entity structural features[J]. Symmetry, 2019, 11(4): 453.

[12] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. ar Xiv preprint ar Xiv:1810.04805, 2018.

[13] Chen S, Wang J, Jiang F, et al. Improving entity linking by modeling latent entity type information[C]//Proceedings of the Advancement of Artificial Intelligence(AAAI). 2020, 34(05): 7529-7537.

[14] Cheng J, Pan C, Dang J, et al. Entity linking for Chinese short texts based on BERT and entity name embeddings[C]//China Conference on Knowledge Graph and Semantic Computing. 2019, 2: 1-12.

[15] Zhao Y, Wang Y, Yang N. Chinese Short Text Entity Linking Based on Semantic Similarity and Entity Correlation[C]//2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI). IEEE, 2020: 426-431.

[16] BERANT J,LIANG P. Semantic Parsing via Paraphrasing[C]// 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore: Association for Computational Linguistics, 2014: 1415−1425.

[17] Li Fei. Research on entity disambiguation method for knowledge graph based on deep learning [D]. Chang'an University, 2021. DOI: 10.26976/d.cnki.gchau.2021.001439.