Variable-Size Relation Extraction and Table Information Extraction

Shaodong Hou[1], Yiqing Zhou[1], Xianming Tong[1]

[1] Minerva Technologies Co. Ltd. (苏州美能华智能科技有限公司）,
No. 88 Jinjihu Blvd., Suite G1-902, Industrial Park, Suzhou, Jiangsu, China

Abstract. In the Financial Report subtask of the CCKS Task 5, the challenge is that a file may have tens of pages and a table may be across multiple physical pages, we aim at optimizing the performance of detecting the table regions and cells and joining the table parts separated by page boundaries. In the Personnel Change Report subtask of the CCKS Task 5, seven entities are defined in the personnel change relation. Technically relations with more than two entities can be separated into triples[1], but in this case labeling the files requires extra efforts. Identifying multi-entities relation directly is non-trivial since multiple relations of the same type are difficult to be separated, especially when they share some entities. Given that no entity in the personnel relation is mandatory, it's infeasible to retrieve relations by seeking for head entities. Clustering relations with the correlation matrix [2] cannot handle the sharing-entities case. We introduce a two-stage neural architecture to identify variable-size relations. In the first stage, entity spans are retrieved. In the second stage, we link the entities by decoding the relation sequences. Unlike conventional decoders, the number of relations is dynamic during decoding. We introduce a pointer network to track and manage the relations.

1 Introduction

Relation extraction (RE) is a task of identifying the semantic relationship between pairs of entities mentioned in the input sentence. Some neural-based approaches encode linguistic and syntactic properties of long word sequences, making them preferable for sequence-related tasks and significantly better than pattern-based approaches and machine learning approaches based on engineered features[3]. Convolutional Neural networks (CNNs) [4] and Long Short-Term Memory (LSTM) networks [5] are highly efficient and effective as they require less feature engineering in relation extraction tasks. Recent research shows that we can combine CNNs with RNNs to build a more powerful feature extraction network [6]. In addition, by feeding LSTM outputs to reinforcement learning (RL) and generative adversarial network (GAN) remarkable results can also be achieved[7].

Traditionally, Relation extraction task is approached as two separate subtasks: entity extraction and relation extraction. In the phase of entity extraction, there may be a large number of duplicate or wrong entities that are extracted from the text. Introducing a sentence-level selective attention [8] can make the model concentrate on the entities which play a key role in relation construction. Some researches that focus on a general purpose of taking into account dependency tree information [1] to model the relations between the entities and others introduce a quadratic scoring layer [9] to model the two subtasks. However, due to the lack of support from external corpus information, the general ability of the model is insufficient and the deep semantic logic cannot be perceived.

Pretrained language models are able to capture the meaning dynamically according to the context. Recently, some pretrained general-purposed language encoders have been proposed, including GPT [10], GPT2 [11], BERT [12], roBERTa [13], ERNIE [14], ERNIE2.0 [15] and XLNet [16]. These models were trained on large amounts of unlabeled text to embed generalizable words and can be fine-tuned to accommodate supervisory tasks. Among them, the respective Chinese pretrained models published by BERT and ERNIE are most widely used in Chinese Natural Language Processing (NLP) tasks.

BERT (Bidirectional Encoder Representations from Transformers) is a new method of pre-training language representations which can obtain state-of-the-art results on a wide range of NLP tasks. Some recent attempts based on BERT pretrained model have brought forth encouraging results in the relational extraction models, including one-stage relation extraction task combining BERT and CRF [17] and two-stage task that relies on the feature representation of BERT[18]. Most researches focus on using triplet relationships to correlate information between entities and relationships[2]. However, the construction of triples will limit the performance of the model in the face of complex multi-relation entities.[2, 17, 18]

In this paper, we propose a novel two-stage neural network for relation extraction. At stage one, we use an entity model to predict the entities for each relation type. At stage two, a pointer network model [19] is introduced to predict the relations among the entities. With the aid of BERT-base-Chinese pretrained model and pointer network architecture, our model achieves the state-of-the-art performance on the CCKS2019 Task 5.

2 Relation Extraction

2.1 Related Works

Previous works typically leverage the correlation matrix to predict candidate relations in a whole sentence. Approaches in this type have two limitations. Firstly, an entity is formed by consecutive tokens, predicting relations between entities either requires bipartite-graph-like connections between the tokens or specifying a head/tail token [2] in each entity. Both strategies require the model to learn the constraints, which could be programmed instead. Secondly, to predict a relation with more than two entities, it needs to cluster the relations with connected graph. However, if an entity is reused in more than one relation of the same type, multiple relations are mixed together. It is tricky to separate the relations.

2.2 Model Architecture

We propose a two-stage approach to reduce the prediction space. In stage one, we introduce an entity model to extract the entities for each relation type. In stage two, a pointer network [19] is used to predict the relations among the entities. To manage the models for each stage, we build a framework model to hold the entity models and relation models. Each entity or relation model is dedicated to handle one relation type. That means, the framework model holds two dictionaries of models with relation type as the keys.

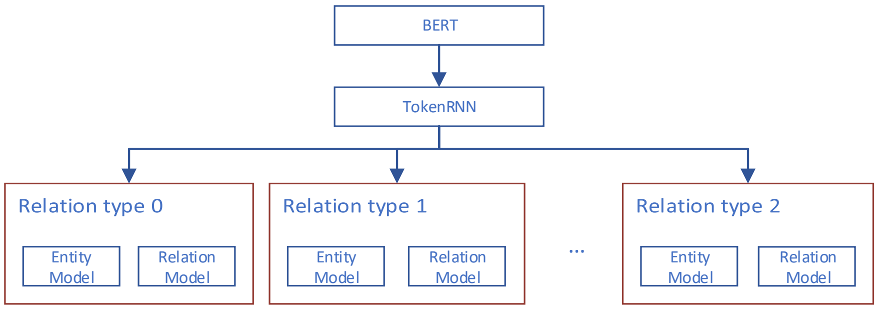


Fig. 1. Model architecture.

2.2.1 The Framework Model

In order to share parameters and reduce memory, we put the token encoding module in the framework model. Since the token encoding shall contain enough unbiased information for models of all relation types to fulfill their subtasks, we introduce a pretrained BERT-base-Chinese [12] as the embedding module and put a RNN layer on top of it.

2.2.2 The Entity Model

The entity model computes the fixed-length encodings of each entity, which takes two steps to get. The entity encoding not only contains the information derived from BERT contextualized embeddings, but also encodes the entity type, positions and occurrence indices, which are useful for differentiating multiple relations of the same type in one paragraph.

Step 1, predict the entity spans with BIO tagging. This step simply projects token encodings into output dimension regarding the number of entities of current relation type. Predicted BIO tags are converted to entity spans before computing entity encodings.

Step 2, compute the entity encodings. Given the entity spans and token encodings, we compute entity encodings by adding following additional information to semantic encodings:

1. Positional embedding

$$P_e = E_p(p_e^s; \theta_p), \quad (\mathbf{1})$$

where p_e^s denotes the start token index of each entity. $E_p(\cdot; \theta_p)$ is the embedding module with θ_p as parameters, which is equivalent to $\text{onehot}(p_e^s)^{\mathrm{T}}\theta_p$. In the remaining of this article, we neglect the parameter θ_p for simplicity.

Since the position could be larger than the maximal position size, we take the remainder of p_e^s/m_s instead, where m_s denotes the maximal position size.

2. Occurrence index embedding

$$O_e = [E_i(O_e^i), E_e(O_e^e)], \quad (\mathbf{2})$$

where O_e^i denotes the entity occurrence index regarding the current entity sequence and O_e^e denotes the occurrence index of the same entity type in the current entity sequence. E_i and E_p are the embedding modules.

3. Entity type embedding

$$T_e = E_t(t_e), \quad (3)$$

where t_e denotes the entity type id and E_t is the related embedding module.

Now we concatenate the embedding to get the entity state

$$S_e = [P_e, O_e, T_e]. \quad (\mathbf{4})$$

The final entity encoding is the RNN final state of the token sequence with S_e as the initial state

$$Y_e = \text{SpanRNN}(X_e|S_e), \quad (\mathbf{5})$$

where X_e denotes the token sequence of each entity and SpanRNN is a single-layer LSTM module.

2.2.3 The Relation Model

After encoding the entities, all non-entity tokens are filtered out. Since the entity encodings have already encoded context information, it is possible to predict the relations among the entities by feeding the entity encodings only. To make things clearer, we focus on a single relation type and depict how relations are predicted given the entity encodings.

As addressed previously, we opted out correlation matrix methodology because it is hard to differentiate multiple relations when they share entities. We could work around by specifying a subject entity in each relation type, but we decided not to make any assumption on the relations so that maximal flexibility could be retained. With this point of view, we designed a pointer-network-based relation model to predict all possible relation paths. A relation path is an ordered entity sequence which is equivalent to a relation. Ordering is an extra output which is not part of the orthodox definition of a relation, however it is required by pointer network so that state transition procedure can be defined.

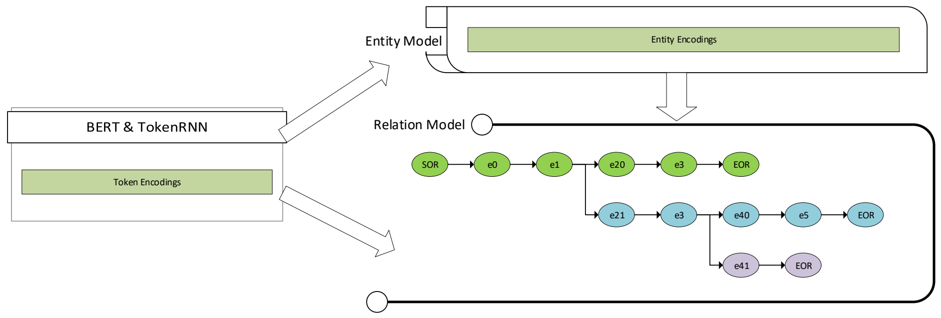


Fig. 2. Workflow of the neural architecture (SOR/EOR denotes the start and end token respectively, e_k denotes the kth entity of a relation)

Given the current decoding state of a relation as s_i, the conditional probability of the next entity position c_t is computed by the following equations in sequence.

$$p(c_t|Y_e, s_i) = \sigma(w^T h_{i,t}) \tag{6}$$

$$h_{i,t} = \text{BLSTM}\left(Y_e^{s_i}\right)_t \tag{7}$$

$$Y_e^s = [[Y_{e,0} s_i], [Y_{e,1} s_i], \dots, [Y_{e,T-1}, s_i]], \tag{8}$$

where $[Y_{e,t} s_i]$ denotes the concatenation of the tth entity encoding and s_i, w is a one-dimension weight vector. $\text{BLSTM}(\cdot)_t$ denotes the t th hidden state produced by a bidirectional LSTM.

In each step, we collect $C = \{c_t, p(c_t|Y_e, s_i) > 0.5\}$ as the set of possible next entities. If C is an empty set, the decoding procedure outputs -1 indicating the end of it. If $|C| > 1$, the decoding procedure duplicates the ongoing relation sequence and its state s_i. For each c_t,

$$s_{i+1}^t = h_{i,c_t}. \tag{9}$$

In the training procedure, c_t is sampled from ground truth relations. The cost function is the cross entropy between the output probabilities and the golden label $\mathrm{y}(c_t)$.

$$\mathcal{L}_i = \sum_{t=1}^{T-1} -\mathrm{y}(c_t) \ln p(c_t) \tag{20}$$

In the evaluation epoch, we evaluate the relation accuracy in non-ordered way.

$$P_i = \frac{n_{corr}^i}{\left|C_{pred}\right|}, R_i = \frac{n_{corr}}{\left|C_{truth}^i\right|} \tag{11}$$

$$h_{i,t} = \mathrm{BLSTM}\left(Y_e^{s_i}\right)_t \tag{12}$$

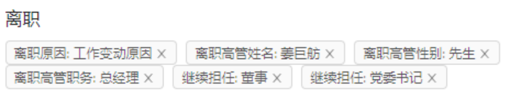
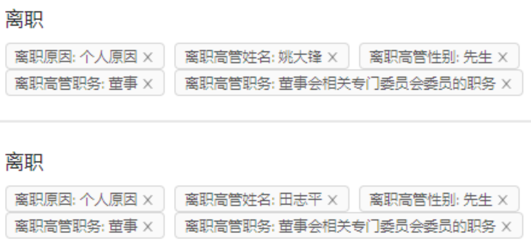
$$F1 = \max_i \left(\frac{2P_iR_i}{P_i+R_i}\right), \tag{13}$$

where n_{corr}^i, P_i,R_i denotes the correct number, precision and recall by comparing the ith relation label with the predicted relation. We find the best-matched relation and compute F1 as the score of the current prediction.

2.2.4 Evaluation

We evaluate our system in CCKS2019 task 5, Personnel Changes dataset. We divided the dataset (totally 616 files) into 501 files as training set and 115 files as validation set.

Table 1. Examples from dataset

Case	Prediction
2018 年 1 月 8 日，姜巨舫先生因工作变动原因，辞去公司总经理职务。姜巨舫先生辞去总经理职务后，仍担任公司董事、党委书记职务。……	 离职 离职原因: 工作变动原因 ×　离职高管姓名: 姜巨舫 ×　离职高管性别: 先生 × 离职高管职务: 总经理 ×　继续担任: 董事 ×　继续担任: 党委书记 ×
中国民生银行股份有限公司（以下简称“本公司”）董事会于 2018 年 7 月 3 日收到董事姚大锋先生和田志平先生的辞职报告。 由于个人原因，姚大锋先生和田志平先生申请辞去本公司董事及董事会相关专门委员会委员的职务。……	 离职 离职原因: 个人原因 ×　离职高管姓名: 姚大锋 ×　离职高管性别: 先生 × 离职高管职务: 董事 ×　离职高管职务: 董事会相关专门委员会委员的职务 × 离职 离职原因: 个人原因 ×　离职高管姓名: 田志平 ×　离职高管性别: 先生 × 离职高管职务: 董事 ×　离职高管职务: 董事会相关专门委员会委员的职务 ×

In the first case, the remaining positions are decoded as sequential entities, because they don't generate a new relation.

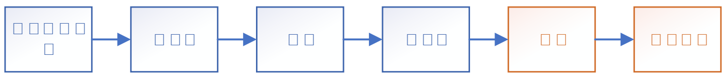


Fig. 3. Sequential entities of the same entity type

In the second case, the two relations share all entities except for the names and genders. In the first step, given a start token, entity "个人原因" is found, only one relation is on track at this moment. In the second step, the relation model infers that "姚大锋" and "田志平" are next parallel entities, so the relation is duplicated. The two relations proceed with different names and genders (genders appear the same, but the positions are different). Although successive entities are reused, relations are not merged.

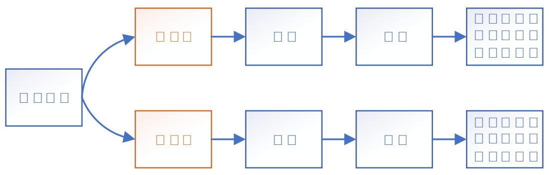
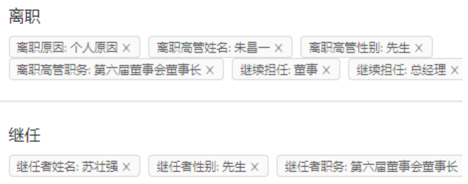


Fig. 4. Parallel entities of the same entity type.

In the following example, resignation and succession events are tagged in different relation types. After the model inference, the two relations are combined by post-process.

Table 2. Example from dataset

Case	Prediction
联美股份有限公司（以下简称 "公司"）董事会于 2016 年 11 月 23 日收到公司董事长朱昌一先生书面辞职报告，朱昌一先生因个人原因辞去公司第六届董事会董事长职务。辞职后，朱昌一先生仍然担任公司董事、总经理职务。在任职董事长期间，朱昌一先生勤勉尽责地履行董事长职责， 公司董事会在此对朱昌一先生为公司健康发展所做的贡献表示衷心感谢！2016 年 11 月 23 日公司第六届董事会第十九次会议 7 名董事一致同意选举苏壮强先生为公司第六届董事会董事长。 苏壮强先生简历附后。	 离职 离职原因: 个人原因 × 离职高管姓名: 朱昌一 × 离职高管性别: 先生 × 离职高管职务: 第六届董事会董事长 × 继续担任: 董事 × 继续担任: 总经理 × 继任 继任者姓名: 苏壮强 × 继任者性别: 先生 × 继任者职务: 第六届董事会董事长

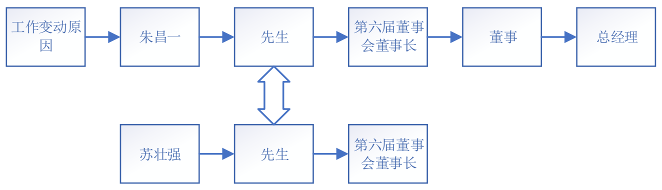


Fig. 5. Relations merged by post-process.

The final output of this example is:

```
{
  "600167-联美控股-董事长辞职及选举新任董事长的公告": {
  "证券代码": "600167",
  "证券简称": "联美控股",
  "人事变动": [
        {
          "离职高管姓名": "朱昌一",
          "离职高管性别": "先生",
          "离职高管职务": "第六届董事会董事长(仍担任公司董事、总经理
职务)",
          "离职原因": "个人原因",
          "继任者姓名": "苏壮强",
          "继任者性别": "先生",
          "继任者职务": "第六届董事会董事长"
        }
      ]
    }
  }
```

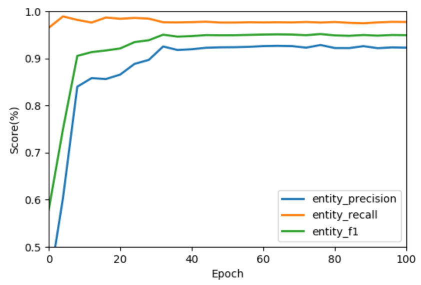
2.3 Analysis and Future Work

In this two-stage model design, the BIO tagging of the entity model conversed to 0.946 after 40 iterations and afterwards keeps stable, as shown is Fig. 6. The reason why the tagging step cannot reach higher accuracy is that some entities occur multiple times but only one occurrence is tagged.

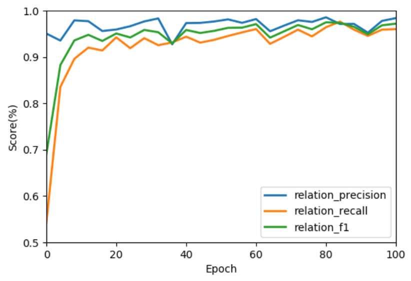
Another shortcoming of this design is that accuracy might decrease as the number of entities in each relation type increases. We predefined two relation types for the test. In CCKS 2019 Task 5 dataset, we leave the combining work to post-process.

For the future work, we will try to control the variation and support larger relations. We plan to explore the following directions:

1. Tagging all matching entities and merging them before relation recognition. The methodology will reduce ambiguity when predicting the entity spans.
2. Concatenating the content-based embeddings to the entity encodings. Entity model tends to output similar entity encodings of the same entity type. But we want the encodings to capture more context such as the name's literal embedding. This is essential for the relation model to connect long distance entities.



(a) Metrics of the entity model.



(b) Metrics of the relation model.

Fig. 6. Metrics during training iterations.

3 Table Information Extraction

All of the target tables in this task have a plane structure. Therefore, although we have models to handle more complex table layouts, a method that simply extract data from the tables row by row is adopted here. The overall workflow is shown in Fig. 5. Among the steps, what dominate the duration and accuracy are *extract table cells* and *handle cross-page situation*, respectively, which will be illustrated later.

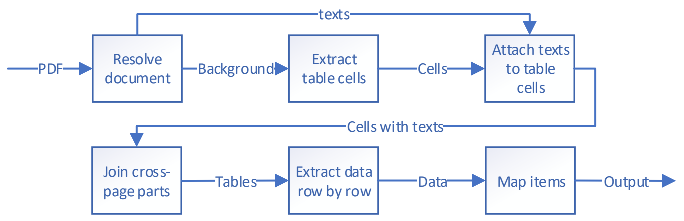


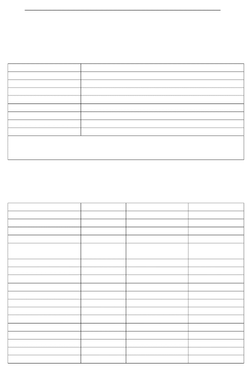
Fig. 7. Workflow of table information extraction.

3.1 Extract Table Cells

With a binarized background image as the input (Fig. 6 (a) shows an example), we extract table cells with BFS algorithm. Extracted cells are represented as (x0, y0, x1, y1), where x0, y0, x1 and y1 are left, top, right and bottom positions (by pixel) of a cell. This method applies two-layers of BFS. The first layer seeks for the outlines of the tables by tracking black pixels. The minimal rectangle containing each connected outline pixels suggests the region of a table. The second layer tracks all white connected graphs inside the table regions. Based on the assumption that all cells are isolated by black boundaries, the outcome of the second BFS will be the region of each cell.

One trick to accelerate the process is to set all-white pixel rows as visited before starting BFS (illustrated in Fig. 8 (b), yellow area). Array operation costs less time than BFS tracking does as it needs not to acquire for dynamic memory.

Replacing BFS with horizontal scanning could make it even faster, at the sacrifice of universality however, so the scanning means is not employed.


(a) Binarized background image.

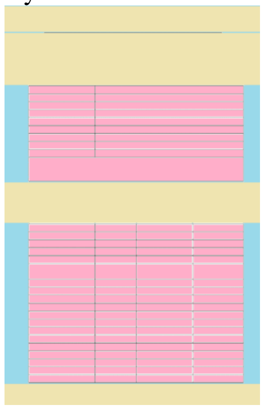
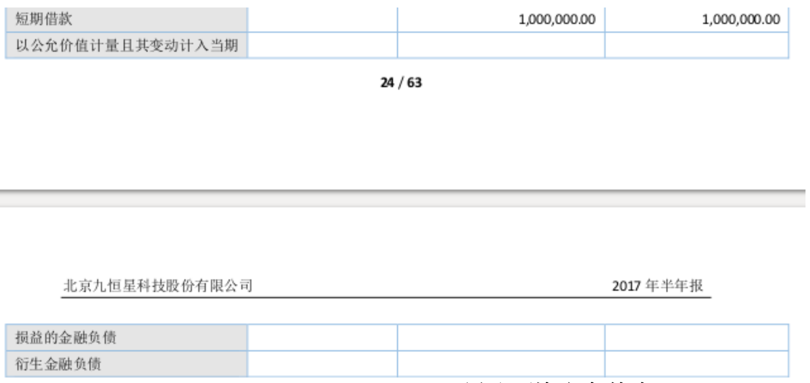

(b) Illustration of regions and cells detection. Yellow area is visited by detecting all-white rows. Pink area is visited by cells detection.

Fig. 8. Extract table cells from a background image.

3.2 Handle Cross-page Situation

A table and cell completeness detection algorithm is adopted to deal with the situation that a table crosses pages. There are two challenges: 1) recognizing and ignoring headers and footers; 2) determining whether we should merge the cross-page cells or not.

For the first challenge, we identify the header by detecting header line and recognize the footer by compositing position, width of text box and text pattern.



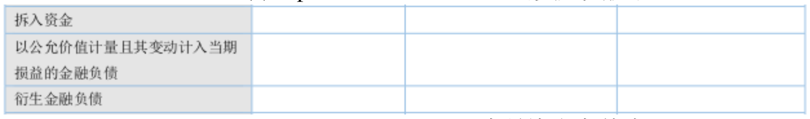
短期借款		1,000,000.00	1,000,000.00
以公允价值计量且其变动计入当期			

24 / 63

北京九恒星科技股份有限公司 2017 年半年报

损益的金融负债			
衍生金融负债			

(a) separated cells in <母公司资产负债表>



拆入资金			
以公允价值计量且其变动计入当期 损益的金融负债			
衍生金融负债			

(b) the integral cell in <合并资产负债表>

Fig. 9. Example of comparing “项目” value sets among tables in the same format.

For the second challenge, we solve it by comparing the “项目” value sets of tables in the same format. As the example shown in Fig. 9, when we discover “以公允价值计量且其变动计入当期” and “损益的金融负债” in <母公司资产负债表>, it can be noticed that neither of these items appear in <合并资产负债表>, while the concatenated string of them does. Consequently, we should merge these cross-page cells.

4 Conclusions

In this paper, we propose a two-stage neural architecture to predict the BIO tagging of entities spans and the conditional probabilities of next entity index $p(c_t)$ in relation decoding. By using the two-stage architecture, we are able to reduce the prediction space in relation identifying. Pointer network is introduced to transfer the relation state s_{i+1}^t from the last hidden state h_i and the next entity set C. Our method works with the variable size of relations as well as relations-sharing entities. No constraint is imposed during relation labeling and prediction.

In future work, we will try tagging and merging all candidate entities to improve the entity tagging accuracy. Besides, more context will be encoded into entity representation so that long-distance entities can be connected. For table information extraction, we are adding more coverages on corner cases on top of our table detection and reconstruction algorithms for complex layouts.

References

1. M. Miwa and M. J. a. p. a. Bansal, "End-to-end relation extraction using lstms on sequences and tree structures," 2016.
2. G. Bekoulis, J. Deleu, T. Demeester, and C. J. E. S. w. A. Develder, "Joint entity recognition and relation extraction as a multi-head selection problem," vol. 114, pp. 34-45, 2018.
3. T. H. Nguyen and R. Grishman, "Relation extraction: Perspective from convolutional neural networks," in Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, 2015, pp. 39-48.
4. D. Zeng, K. Liu, Y. Chen, and J. Zhao, "Distant supervision for relation extraction via piecewise convolutional neural networks," in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1753-1762.
5. Z. Liu et al., "Entity recognition from clinical texts via recurrent neural network," vol. 17, no. 2, p. 67, 2017.
6. Z. Liu, X. Wang, Q. Chen, and B. Tang, "Chinese clinical entity recognition via attention-based CNN-LSTM-CRF," in 2018 IEEE International Conference on Healthcare Informatics Workshop (ICHI-W), 2018, pp. 68-69: IEEE.
7. T. Zhang, H. Ji, and A. J. D. I. Sil, "Joint entity and event extraction with generative adversarial imitation learning," vol. 1, no. 2, pp. 99-120, 2019.
8. Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances," in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 2124-2133.
9. I. Bekoulis, J. Deleu, T. Demeester, and C. Develder, "Reconstructing the house from the ad: Structured prediction on real estate classifieds," in EACL2017, the 15th Conference on the European Chapter of the Association for Computational Linguistics, 2017, pp. 1-6.
10. A. Radford, K. Narasimhan, T. Salimans, and I. J. U. h. s.-u.-w.-a. c. o.-a. r. l. l. u. p. p. Sutskever, "Improving language understanding by generative pre-training," 2018.
11. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. J. O. B. Sutskever, "Language models are unsupervised multitask learners," vol. 1, no. 8, 2019.
12. J. Devlin, M.-W. Chang, K. Lee, and K. J. a. p. a. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018.
13. Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," 2019.
14. Y. Sun et al., "ERNIE: Enhanced Representation through Knowledge Integration," 2019.
15. Y. Sun et al., "ERNIE 2.0: A Continual Pre-training Framework for Language Understanding," 2019.
16. Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. J. a. p. a. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding," 2019.
17. N. Pang, L. Qian, W. Lyu, and J.-D. J. a. p. a. Yang, "Transfer Learning for Scientific Data Chain Extraction in Small Chemical Corpus with BERT-CRF Model," 2019.
18. S. Yang, D. Feng, L. Qiao, Z. Kan, and D. Li, "Exploring Pre-trained Language Models for Event Extraction and Generation," in Proceedings of the 57th Conference of the Association for Computational Linguistics, 2019, pp. 5284-5294.
19. O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in Advances in Neural Information Processing Systems, 2015, pp. 2692-2700.