A Relation Proposal Network for End-to-End Information Extraction

Zhenhua Liu[1][0000−0003−2760−3621], Tianyi Wang[2], Wei Dai[1][0000−0002−0408−1835], Zehui Dai[1][0000−0002−0406−8219], and Guangpeng Zhang[1]

[1] NLP Group, Gridsum, Beijing, China.
{liuzhenhua,daiwei,daizehui,zhangguangpeng}@gridsum.com
[2] wangtianyiftd@gmail.com

Abstract. Information extraction is an important task in natural language processing. In this paper, we introduce our solution on NLPCC 2019 shared task 3 Information Extraction which has provided with the largest industry Schema based Knowledge Extraction(SKE) data-set. Our proposed method is an end-to-end framework which first catches the relation hints in raw text with a relation proposal layer, then follows by an entity tagging design which is targeted to decode the corresponding triplet entities with the given relation proposal. Compared with previous works, our method is efficient and can well handle overlapping and multiple triplets in one sentence. With a simple model ensemble, our solution achieves 0.8903 F1-Score on final leaderborad which ranks forth among all participants.

Keywords: Information Extraction · Relation Proposal · Entity Tagging

1 Introduction

Information extraction is targeted to extract entities and their relations from unstructured natural language text. It is important to many Artificial Intelligence(AI) applications, such as Information Retrieval(IR), Intelligent Question and Answering(QA), and Intelligence Chat-bots(IC). The task involves entity recognition, anaphora resolution and relation classification.

In the NLPCC 2019 Information extraction competition, there are over 173k train data which is firstly annotated with distant supervising, then human corrected on crowd-sourcing platform. It contains over 364k triplets of 49 relations and 28 entity types. The key challenges of this task are summarized as follows:

– **Overlapping Entities**: Overlapping entities of different triplets is a common issue in information extraction that affects the extraction performance. [7] first divide the triplets into three categories, namely No Overlap(Normal), Single Entity Overlap(SEO), and Entity Pair Overlap(EPO). They find that previous methods mainly focus on Normal class and fail to extract relational

triplets precisely. We also observe a plenty of overlapping triplets in train set, so it is of great importance for our solution to handle overlapping entities properly.

- **Multiple Triplets**: There are on average 2.10 triplets per sentence. What's more, 3.72% sentences contain five or more triplets, and these account for 19.83% of total target which can not be ignored in the competition. As mentioned by [6] and also observed in our experiments, methods which are unable to deal with multiple triplets properly will trend to recall fraction of all triplets in one sentence.
- **Categories Imbalance**: Categories of the data in this competition are extremely imbalanced. Among all 49 relation types, the top five number of relation categories account for 45.21% of the total triplets. However, the bottom five relation categories account for only 0.13%.
- **Data-set Inconsistent**: Another serious problem is that the offline train data has lower precision compared with the online test data. The sponsor of the competition has officially announced that the online test data has been double checked with more efforts, which make the offline validation result becomes unreliable. And the defection in train data may also lead to a model bias, which makes the data distill and rule amendment necessary in data pre-processing and post-processing.
- **Entity Position Uncertainty**: Even though the data-set in this competition is of high quality, there still exist problems. 11.53% of the triplets in train data contain entities which repeat more than once in one sentence, but no exact position information is given. As the same entity of different position can contain different semantics, elaborate enumerate or random choose will induce errors into the model.

To solve the aforementioned challenges, we first make a brief review of academic and industrial solutions on information extraction task. There are mainly two pipelines that solve the problem. One recognizes entities firstly and then follows with an entity pair relation classification, these methods suffer from error propagation problems[2]. The other one is to extract entities and relations in an end-to-end way. However, these methods still have problems. [5] propose the HRL which is so far the state-of-the-art, but the reinforcement learning paradigm makes it run extremely slow. [7] propose copyR can only deal with triplets that entities contain one word only. [8] propose the tagging schema that fails to deal with overlapping triplets. [1] transfer the task into a table filling problem, and their method suffers a positive and negative sample imbalance problem.

Our solution is partially inspired by Faster R-CNN[3], a two stage object detection network in computer vision which separates the different target object into parallel pipelines with region proposal layer. It works in similar logic and uses the relation proposal layer which aims to propose the target relation at each token position. With the given proposal relation and token, different triplets are decoded in parallel by entity tagging network which only decodes one triplet for one proposal.

This paper is organized as follows. Section 2 contains the data-set analysis and pre-processing. Section 3 introduces details of our solution. Section 4 presents the experimental results and analysis. Section 5 includes the conclusion and future works.

2 Data-set Analysis and Pre-processing

2.1 Data-set Analysis

The sponsor of competition provides us 173k train data, and 21.7k validation data. The online test data-set has the similar size with offline validation, but is mixed with irrelevant sentences. The Fig. 1 summarizes the triplet types in train data.

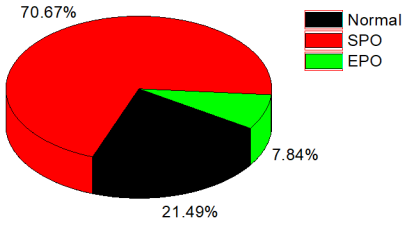

Fig. 1. The triplet types in train data, and we follow the definition by [7]

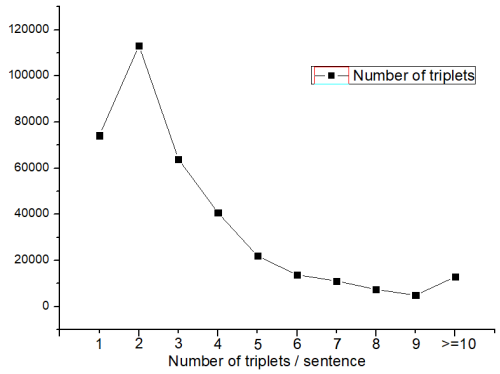

Fig. 2. Number of triplets in one sentence

Obviously, Normal triplet consists 21.49% of the total triplets which indicates the importance to deal with overlapping triplet in this task. The SPO dominates train data-set with 70.67% which is very high. And there are also a fraction of SEO triplets.

From the Fig. 2, we can find most of the sentence contains multiple triplets. There are 19.83% triplets source from the sentence which contains no less than 5 different triplets. And the number of triplets in one sentence can grow up to 25! The multiple triplets case is so common in this task which requires us to pay a special attention to ensure the recall.

The relation distribution is extremely imbalanced. Of all 49 relation types, the *actor*, *writer*, and *singer* are most highest three relations, account for 32.77% triplets of all. However, the bottom three relation types *length of schooling*, *postcode*, and *specialityid* accounts for only 0.02% of total sample.

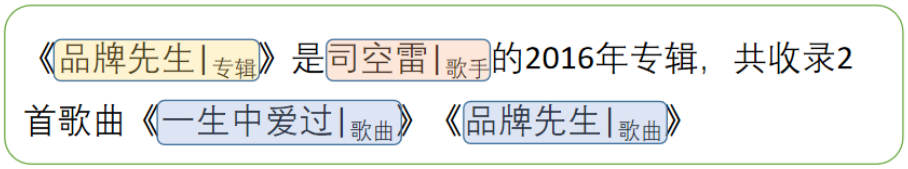

Fig. 3. Case of one entity matches multiple positions in one sentence.

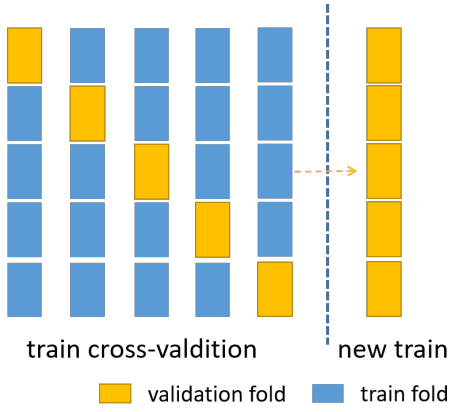

Fig. 4. Data distill. We first split the train data into 5 folds randomly. Each time we train with 4 folds and valid with the rest one fold. Finally, we get the validation on all train samples and use them as new train data for further experiments.

Besides, the data-set do not contain the entity position information, and we can not infer their right position through a regular expression matching when multiple positions are matched. As shown in Fig. 4, the *album* (专辑) entity 品牌先生 at the beginning of sentence can also match the *song* (歌曲) entity 品牌先生 at the end. Missing triplets entity position introduces noise into the model. We solve it within training process by dynamic matching.

2.2 Pre-processing

The pre-processing we use is primarily targeted to solve categories imbalance, data-set inconsistent, and the entity position uncertainty problems. Because the number of sample in different relation types is extremely imbalanced and some relation types have only a fraction of labeled data, we randomly over-sample each relation categories to 2500 sample size if it has number of labeled data below it.

As to the data-set inconsistent problem, we find the mislabeling in train-set does affect the model performance when testing online. We reduce the gaps between offline train and online test by removing errors and adding unmarked triplets in the train data. It can be done by annotating the train-set manually, however the cost would be huge. So, we distill the data with cross validations. See detail in Fig. 4.

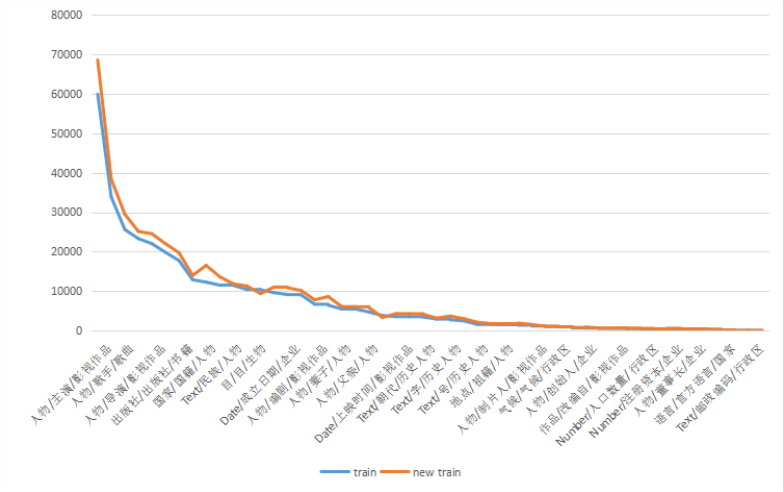

Fig. 5. Difference comparison before and after data distill. The figure contains all relation types from left to right with descending in number of labeled data.

The data distill process has changed the train data-set a lot. And the number of marked triplets increases 13.7% from 364k to 414k. The online test shows that

this change has improved the recall a lot and the precision is less affected. The detail that number of changes in each relation types is presented in Fig. 5.

Furthermore, we find it hard to align the triplets with right position when facing with multiple occurrences situation. So we first enumerate all possible combination of multiple position entities, and then choose the best fitted one during the training process. This is done because we can infer the entity token probability in relation proposal layer and triplet CRF log-likelihood in decoding stage during the train. And we will always choose the position with highest proposal probability or triplet decoding log-likelihood.

3 Our Solution

Our solution is basically based on a relation proposal network for end-to-end information extraction. We use model ensemble and apply post-processing to get the final results.

3.1 The Model

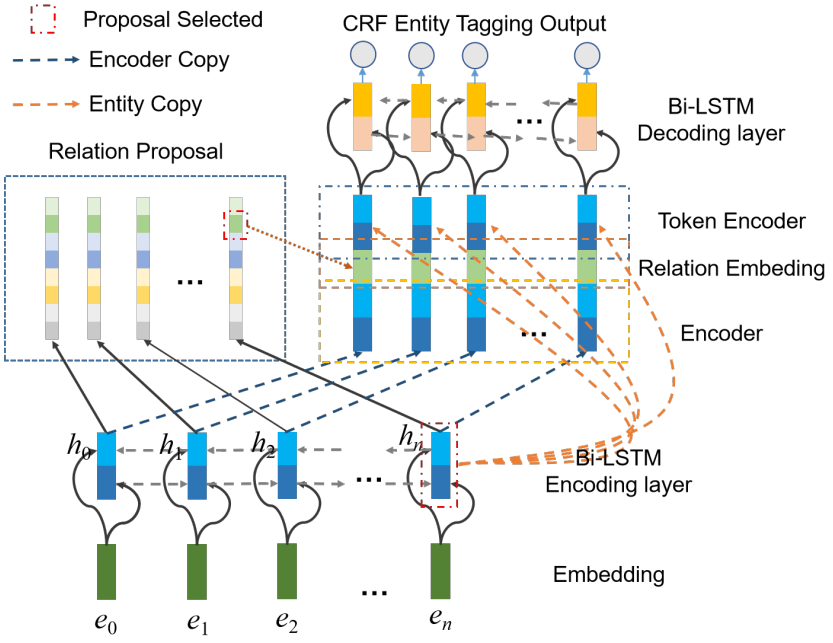

Fig. 6. The relation proposal based end-to-end neural network structure. An encoder(like Bi-LSTM) is used to encode source sentence embedding, then followed by the relation proposal. With the given relation and token, a copy mechanism is used to construct the input for decoder which is used to decode the triplet entities.

The model we use contains three parts: 1) An encoder to encode the input sentences. It is a Bi-LSTM layer at first when we test the model structure, then we replace it with BERT encoder, and it can also be any other predefined generalized language models. 2) A relation proposal layer to propose the relation at each token position. It is a fully connected layer that output the relation proposal probability $p_{i,r}$, where i is the token position index, and r is the r^{th} relation type. 3) A decoder to output the triplets. It is a typical sequence tagging network, and the decoder is designed to decode one triplet for one relation proposal. The model structure is presented in Fig. 6.

The key to solve the overlapping problem is the one-triplet-one-sequence decoding design. During the train, the relation proposal layer is targeted to propose the relation for each triplet at the entity token position. Except for a very special case that two SEO triplets are of the same relation type, each valid proposal must belong to one target triplet only. Thus the decoding layer can only decode one-triplet for one-proposal. As to the special case where one proposal can match multiple triplets for the entity that two triplets share, we choose to discard it during the train. It does not matter much since we can decode the triplet from entity which two triplets do not share. The decoder layer works in one-triplet-one-sequence manner, so there is not existing triplets overlapping problem.

The multiple triplet problem is solved with a train strategy for relation proposal layer. We force the proposal layer to propose the relation for every triplet at all token position to which the entities belong. It produces the entities token number of proposal for each triplet. And any one among these potential proposal can successfully decode the target triplet. This train strategy makes the model performance well on multiple triplet cases.

3.2 Ensemble and Post-processing

Besides, we also adopt ensemble and some post-processing methods. We choose all relation proposal probability $p > 0.4$, and CRF triplet decoding log-likelihood $l > -1$ as the valid output of the model. Then we ensemble the model with a simple n *in* m strategy, which means we output triplets which have at least recognized by n times in m models, $n = 7$ and $m = 10$ in our final submission.

We also use rule based post-processing strategies to gain a better result, and these rules are focused to reduce the apparent mistakes resulted by inconsistent annotations, and contradiction triplet outputs. They are:

- **Normalization** Normalization targets to correct the entities with a clear pattern. This is partial resulted by annotation mistakes. For example, the entity *book title* should be all content in 《》 . However, this is not true in annotation data which needs a correction.
- **Deduplication** The model can output two triplets of the same type that partial of the entities are overlapping. It is clear that one output is wrong. Under this circumstance, we will choose the entity with longer length. Fig. 7. shows the example.

Fig. 7. The example of the post-processing.

4 Experimental Results

Our experiment is first conducted with Bi-LSTM as encoder at word level, and we use the Tencent word embedding[4] . This baseline model has achieved a good result at the beginning of the competition. When we stabilize the model structure, the Bi-LSTM encoder is replaced with BERT which leads to a significant improvement on precision and recall. We further apply data distill, which has improved the recall of our model a lot. With the post-processing and ensemble, our solution finally reach 0.8903 F1-score. Table 1. presents the detail.

Table 1. Experiment results.

	Precision	Recall	F1-score
Bi-LSTM Encoder	0.8736	0.8108	0.841
BERT Encoder	0.8752	0.8494	0.8621
BERT + data distill	0.8811	0.8639	0.8724
BERT + data distill + post-processing	0.8803	0.8886	0.8844
4 in 5	0.8955	0.8842	0.8898
7 in 10 Final	0.8948	0.8858	0.8903

we find the most important improvement in our experiments are: 1) A better encoder leads to a better performance. Compared with the origin Bi-LSTM encoder, the BERT based model achieves the biggest improvement of 0.0211 F1-score online. It implicates the importance of a better encoder in information extraction task. 2) Data distill is important in noisy data-set. We apply data distill on train data-set because it is apparently more noisy compared with the validation online. Noisy in train data-set is so common in real world applications as the triplets annotation is extremely hard. The implement of data distill here can also be meaningful to further applications.

5 Conclusion and Future Works

We analyze the data-set of NLPCC 2019 information extraction task, and propose the relation proposal based end-to-end information extraction method. This method successfully solves overlapping and multiple triplets problems, which are typically issues in academic researches and industry applications. Our experiments reveal the importance of better encoder in NLP tasks and data distill in noisy data, and it's meaningful to further study.

During the competition, we have only tested the Bi-LSTM and origin BERT encoder, and the Bi-LSTM decoder in our experiments with limited time. It's meaningful to have other more explorations on different encoder and decoder structures, especially in different language settings.

References

1. Bekoulis, G., Deleu, J., Demeester, T., Develder, C.: Joint entity recognition and relation extraction as a multi-head selection problem. CoRR **abs/1804.07847** (2018), http://arxiv.org/abs/1804.07847
2. Li, Q., Ji, H.: Incremental joint extraction of entity mentions and relations. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 402–412 (2014)
3. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
4. Song, Y., Shi, S., Li, J., Zhang, H.: Directional skip-gram: Explicitly distinguishing left and right context for word embeddings. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). pp. 175–180. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018). https://doi.org/10.18653/v1/N18-2028, https://www.aclweb.org/anthology/N18-2028
5. Takanobu, R., Zhang, T., Liu, J., Huang, M.: A hierarchical framework for relation extraction with reinforcement learning. In: AAAI (2019)
6. Tan, Z., Zhao, X., Wang, W., Xiao, W.: Jointly extracting multiple triplets with multilayer translation constraints (2019)
7. Zeng, X., Zeng, D., He, S., Liu, K., Zhao, J.: Extracting relational facts by an end-to-end neural model with copy mechanism. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 506–514 (2018)
8. Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P., Xu, B.: Joint extraction of entities and relations based on a novel tagging scheme. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1227–1236. Association for Computational Linguistics, Vancouver, Canada (Jul 2017). https://doi.org/10.18653/v1/P17-1113, https://www.aclweb.org/anthology/P17-1113