Relation-Aware Joint Entity and Relation Extraction

Negar Mirgati
University of Alberta
mirgati@ualberta.ca

Mohammadreza Samadi
University of Alberta
samadi2@ualberta.ca

Abstract

Over the last few years, many researchers have focused on the problem of entity and relation extraction from raw text. Although considerable progress has been made to solve this problem, many proposed solutions still struggle with handling overlapping relational triples. Therefore, building upon the work of Wei et al. (2019), we present a relation-aware model that benefits from both relation semantics and interactions. Our experiments on the widely-used WebNLG dataset show that our model achieves an absolute gain of 0.96 in terms of f1-score over the original CASREL framework.

1 Introduction

One of the critical and challenging tasks in information extraction is joint entity detection and relation extraction. Formally, given unstructured natural language text, the goal is to extract relational triples in the form of (subject, relation, object). Early works aimed to solve this problem by designing pipeline-based models. These models detect all entities in the first stage and then predict the relations between them in the second stage (Zelenko et al., 2003; Zhou et al., 2005; Chan and Roth, 2011). Since errors made by early steps cannot be fixed later, these approaches lead to error propagation issues. Therefore, subsequent works tried to avoid error propagation by designing joint entity and relation extraction models (Zeng et al., 2018; Fu et al., 2019; Wei et al., 2019).

Despite the significant improvements that these models achieved compared to early works, there are still issues to be resolved in this area. One of the existing challenges, which is the focus of our project, is overlapping triples. We consider two triples to be overlapping if they share entities. Recent works such as Wei et al. (2019) and Fu et al. (2019) took the problem of overlapping entities into account. However, they still struggle with problems of false object detection and missing relational triples. To overcome these issues, we present two sources of additional features and provide some insights into how they can be helpful in the process.

One potential source of information that recent works have not used is relation semantics. Consider the following example sentence from the WebNLG dataset (Gardent et al., 2017): “Alan Bean, who was part of Apollo 12, was born in Wheeler, Texas on March 15th, 1932 and is now retired.” In this sentence, the subject entity is “Alan Bean”. Moreover, “Apollo 12” and “Wheeler” are the other two entities that can potentially be the objects related to the mentioned subject. Now, let’s consider the relation type “birthPlace”. It is easy to see that given the subject and the semantics of this relation, the entity that must be identified as the object is “Wheeler” since it is a location in contrast to “Apollo 12”. Such information from relations’ semantics could be helpful in the relation detection part of the model. If the classifier realizes that given a specific subject and a relation, an entity would be unrelated, it won’t choose it as an object. On the other hand, this extra information may assist the classifier in predicting correct triples more confidently.

Another additional source of information is relation interactions. Figure 1 represents the co-occurrence matrix of the most frequent relation types in the WebNLG dataset. The value of any pair $(R1, R2)$ corresponds to the number of times $R1$ and $R2$ shared the same subject in the dataset. This data analysis reveals that some relation types, such as $(birthPlace, deathPlace)$ tend to appear together frequently. Also, it is worth mentioning that most of these co-occurrence patterns are probably domain-specific, meaning that different datasets may have their own unique relation interactions. Nevertheless, relational triples containing a particular subject are not entirely independent from

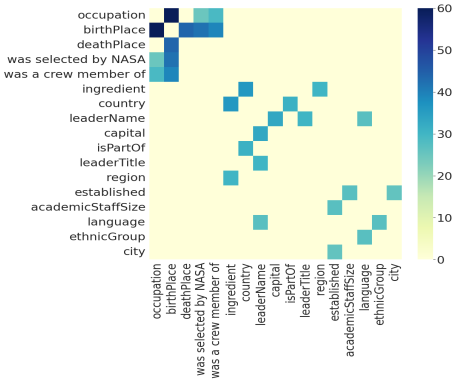

Figure 1: Relation co-occurrence matrix of WebNLG

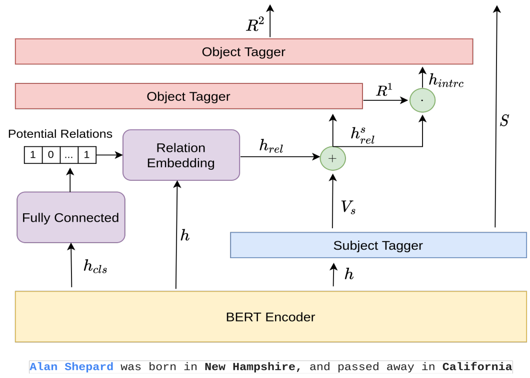

Figure 2: Overview of the Rethinking model

each other, and leveraging the domain-specific relation correlations can also help the model deal with overlapping triples.

Motivated by the above observations, we aim to build upon the work of Wei et al. (2019) and leverage relation semantics and interactions to improve the performance of joint entity and relation extraction.

2 Related Work

Researchers have proposed numerous methods and frameworks to tackle the problem of joint entity and relation extraction. Fu et al. (2019) designed an end-to-end model which uses graph convolutional networks (GCNs) to detect entities and relations. Their method consists of two phases. In phase one, they employ bi-directional RNN and GCN to extract sequential features and regional dependencies. In phase two, they use a relation-weighted GCN to account for the interactions between extracted entities and relations from phase one. Our work is different from them because they did not consider relation semantics, and they also employed a graph-based approach to capture relation interactions.

Wei et al. (2019) revisited the problem of joint entity and relation extraction from a fresh perspective. They proposed a novel architecture called CASREL that addresses the issue of overlapping triples by formulating relations as functions that map subjects to objects. Although CASREL outperformed all baselines and demonstrated strong performance, an in-depth analysis of its results showed that as the number of relations increases, it becomes challenging for CASREL to identify correct relations between entities. In this project, we use CASREL as the base of our project and try to improve its relation-specific object detection module by incorporating relation semantics and dependencies into the model.

Zheng et al. (2021) focused on some of the existing challenges in this task, such as relation prediction redundancy and inefficiency. They decomposed the problem into three subtasks, relation judgment, entity extraction, and subject-object alignment. First, their model predicts a set of potential relations for each sentence. It also generates a global matrix that demonstrates the correspondence between all subjects and objects. Then, the model performs sequence tagging to detect subjects and objects for each potential relation. Finally, it prunes the detected triples according to the global correspondence matrix. Even though we are also detecting potential relations, our work differs from theirs because we aim to use this information to create relational semantic vectors.

3 Method

Figure 2 demonstrates the architecture of our proposed model. We utilized the CASREL (Wei et al., 2019) framework [1] as our base and implemented some new modules to incorporate relation semantics and interactions into the model [2].

3.1 Subject Tagger

To extract subjects, similar to CASREL, we utilized two fully connected layers to extract all subjects' start and end positions. First, the BERT encoder (Devlin et al., 2018) creates the sentence and tokens' representations denoted by $h_{cls} \in R^{d\times 1}$ and $h \in R^{d\times n}$ respectively, where d is the encoding dimension, and n is the number of tokens in the input

[1] We won't entirely explain CASREL due to space limitations, but the reader is encouraged to refer to Wei et al. (2019) for more details.

[2] Our implementation can be found in the following address: https://github.com/MhmDSmdi/Relation-Aware-Joint-Entity-Detection-and-Relation-Extraction

Model	Precision	Recall	F1
GraphRel (Fu et al., 2019)	44.7	41.1	42.9
CASREL (Wei et al., 2019)	87.35	89.12	88.22
PRGC (Zheng et al., 2021)	**94.0**	**92.1**	**93.0**
Rethinking	89.44	88.93	89.18

Table 1: Comparison of the proposed Rethinking method with the baselines.

text. Moreover, V_s is the vector representation of the subject for which the model aims to predict all objects.

3.2 Relation Embedding

We added two new steps to the model architecture to capture relation semantics: one for predicting the potential relations in a sentence and one for generating the relation embeddings. In the following paragraphs, we will explain these steps in detail.

Potential Relation Extraction This part of our model is the fully-connected layer demonstrated as a purple box in Figure 2. Given a sentence, the goal is to predict all relation types that could potentially exist in that sentence. The sentence representation vector is fed into the fully connected layer. The output of this module is a vector $P_{rel} \in R^{1\times n_r}$, where n_r is the number of pre-defined relations. This vector contains a probability for each pre-defined relation type.

Relation Embedding Fusion The relation embedding module is represented in Figure 2 as a purple box. We used an embedding layer to capture relations' representations. This layer is initialized with the BERT representation of pre-defined relations. The module first creates a relation feature vector using potential relations and relations' embeddings and then adds it to the BERT token representation features. Finally, the subject vector is added to the relation-aware features. Detailed operations of this module are as follows:

$$
\begin{aligned}
V_{rel} &= P_{rel} \times E_{rel} \\
h_{rel} &= h + expand(V_{rel}, n) \\
h^s_{rel} &= h_{rel} + expand(V_s, n)
\end{aligned} \tag{1}
$$

where $V_{rel} \in R^{1\times d}$ is the relation feature vector, $E_{rel} \in R^{n_r\times d}$ is the trainable embedding matrix, $h^s_{rel} \in R^{n\times d}$ is the enhanced features including both relations and subject features, and $expand$ is a function that creates a matrix by repeating a vector n times.

3.3 Rethinking

To capture relation interactions, we added an extra phase to the object tagger module, which is shown as the red boxes in Figure 2. Each object tagger contains two fully connected layers to predict the spans of objects (start and end positions). The first object tagger uses the enhanced features to identify potential relation-specific objects for the given subject as the first guess. Then, we concatenate the first-guess output with the enhanced features and feed it to the next object tagger for the final prediction. The following equations elaborate the details of operations for the start position, which is the same as the end position:

$$
\begin{aligned}
R^1_{start} &= \sigma(W^1_{start} h^s_{rel} + b^1_{start}) \\
h_{intrc} &= h^s_{rel} \odot R^1_{start} \\
R^2_{start} &= \sigma(W^2_{start} h_{intrc} + b^2_{start})
\end{aligned} \tag{2}
$$

where $R^1_{start} \in R^{n\times n_r}$ is the first guess output, $h_{intrc} \in R^{n\times(d+n_r)}$ is the enhanced feature with the potential probabilities of objects, $W^1_{start}, W^2_{start} \in R^{n_r\times(d+n_r)}$, $b^1_{start}, b^2_{start} \in R^{1\times n_r}$ are the trainable weights and biases, σ is the sigmoid activation function, and $R^2_{start} \in R^{n\times n_r}$ is the final output that contains the probabilities of $(relation, object)$ pairs given this iteration's subject.

4 Results

Dataset We chose the WebNLG dataset (Gardent et al., 2017) since it has a relatively large set of pre-defined relation types. Furthermore, it has more overlapping triples than the NYT (Riedel et al., 2010) dataset. Therefore, it is an appropriate benchmark based on our project goals.

Evaluation Metrics Following previous works (Wei et al., 2019; Fu et al., 2019), we choose standard micro precision, recall, and f1-score as our evaluation metrics. We consider a triple to be correct if the relation and the heads of both entities are all correct.

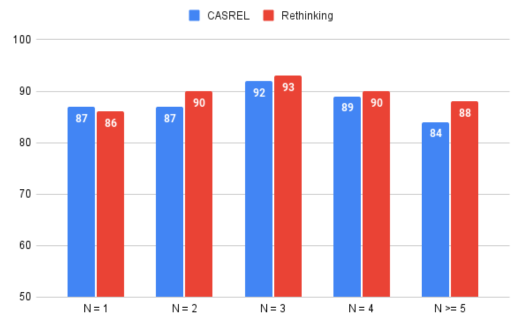

Figure 3: F1-score of extracting relational triples from sentences with different number (denoted as N) of triples.

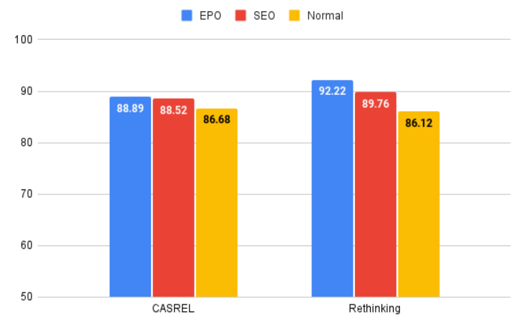

Figure 4: F1-score of extracting relational triples from sentences with different overlapping pattern.

Results Analysis Table 1 presents the evaluation results of our work and the baseline models. Our Rethinking model outperforms CASREL in terms of all the evaluation metrics, demonstrating our solution's effectiveness in detecting more correct triples and making fewer mistakes. We also performed two additional experiments regarding different sentence types to investigate our model's performance more carefully.

For the first experiment, we categorized the sentences based on the number of triples that exist in them. Figure 3 demonstrates that our rethinking model maintains a consistent performance for different degrees of sentence complexity. One interesting observation is that for the $N = 1$ category, our model has a 1% drop in terms of f1-score. We believe the reason for this decline in performance is that the extra complexity of our model makes it more suitable for cases where the sentence has some overlapping triples.

For the second experiment, we tried grouping sentences based on the three overlapping patterns (Normal, Single Entity Overlap (SEO), and Entity-pair Overlap (EPO)). Figure 4 shows the results of this experiment. It can be seen that the most significant performance gap between our model and CASREL is for the EPO class. We believe this is probably due to our two-step mechanism that leverages relation interactions to update the model's initial decisions. Moreover, our model has a slight drop in f1-score of the normal class compared to CASREL. Similar to the observation for the $N = 1$ class, this issue probably stems from the additional complexity of our model, which makes it more suitable for detecting overlapping triples.

5 Discussion

Table 2 represents the case study of CASREL and Rethinking. We denote the additional correct triples that only our model extracted by green and the mistakes made by the models by red.

The first row in table 2 shows an example where our model was able to extract more correct triples than CASREL. We believe this might be due to our two-phase object extraction mechanism, and the fact that "LeaderName", "Capital", and "LeaderTitle" appear together frequently (See figure 1). The second row represents a case where our model was able to prevent a mistake that was made by the original CASREL. Considering the fact that "Malay Penninsula" is a place, not a group of people, we believe the relation semantics have helped in avoiding this mistake and extracting the correct triple. The third row in table 2 shows an example where CASREL performed better than Rethinking. This is probably due to the additional complexity that our model has compared to CASREL, which may sometimes mislead the model in simpler scenarios where the sentence has only a single triple.

6 Conclusion

In this project, we focused on the task of joint entity and relation extraction and tried to improve its performance by taking relation semantics and interactions into account. Based on our evaluation results, we found that leveraging these information sources can improve the overall performance of joint entity and relation extraction. As future work, we suggest trying the model on different datasets to investigate the generalization ability of the solution. Moreover, another interesting idea would be to parallelize the processing of subjects and objects to reduce computational costs.

Sentence	CASREL	Rethinking
The capital of Azerbaijan is Baku and the leader and Prime Minister of Azerbaijan is called Artur Rasizade.	(Azerbaijan, capital, Baku)	(Azerbaijan, capital, Baku) (Azerbaijan, leaderTitle, Azerbaijan) (Azerbaijan, leaderName, Rasizade)
Asam pedas is found in Malaysia and is from the Malay Peninsula region.	(pedas, country, Malaysia) (pedas, region, Peninsula) (Malaysia, ethnicGroup, Malay)	(pedas, country, Malaysia) (pedas, region, Peninsula)
Another dish that is a dessert is Baked Alaska which has Christmas pudding as an ingredient.	(Alaska, Ingredient, Pudding)	None

Table 2: Case study for CASREL and Rethinking.

References

Yee Seng Chan and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 551–560.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. GraphRel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1409–1418, Florence, Italy. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for nlg micro-planning. In *55th annual meeting of the Association for Computational Linguistics (ACL)*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2019. A novel cascade binary tagging framework for relational triple extraction. *arXiv preprint arXiv:1909.03227*.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of machine learning research*, 3(Feb):1083–1106.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. 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)*, pages 506–514, Melbourne, Australia. Association for Computational Linguistics.

Hengyi Zheng, Rui Wen, Xi Chen, Yifan Yang, Yunyan Zhang, Ziheng Zhang, Ningyu Zhang, Bin Qin, Ming Xu, and Yefeng Zheng. 2021. Prgc: potential relation and global correspondence based joint relational triple extraction. *arXiv preprint arXiv:2106.09895*.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting of the association for computational linguistics (acl'05)*, pages 427–434.