Attention Guided Filter for Jointly Extracting Entities and Classifying Relations

Shaoze Chen[1], Su Wang[2], Wenxin Hu*[2]
[1]*School of Computer Science and Technology, East China Normal University, Shanghai, China*
[2]*School of Data Science Engineering, East China Normal University, Shanghai, China*

***Abstract*—Jointly extracting entities and classifying relations aims to detect all possible triples from unstructured text with a single model. Tagging-based method effectively improves the performance of jointly relation extraction. However, some tagging-based approaches ignored that one entity pair may exist multiple relations and others set an empirical threshold value for selecting one or more relevant relations, which becomes the bottlenecks of the model. As a solution, we propose the attention guided filter, namely, AGFRel, which introduces transformer blocks to learn the number of relations for every entity pair to filter out irrelevant relations. Moreover, each module of the model has a multi-head attention guided layer to highlight valuable information. Extensive experimental results show that AGFRel is capable of gaining better performance on various tasks including overlapping triples extraction and multiple triples extraction. On NYT and WebNLG public datasets, our model obtains F1 score 90.8 and 91.9 respectively and achieves a new state-of-the-art performance.**

***Index Terms*—transformer, attention mechanism, joint extraction model, NYT, WebNLG**

I. Introduction

Relation Extraction aims to extract relational triples from unstructured text. It plays a crucial role in many applications of natural language processing, such as biomedical knowledge discovery [1] and knowledge base construction [2].

Pipeline-based approach is an intuitive method to extract triples. It first recognizes all entity mentions in the text and then classifies relations for each entity pair. This method mechanically decomposes the relation extraction task into two independent sub-tasks: named entity recognition (NER) and relation classification (RC), which ignores the relevance between them and results in error propagation [3]. The joint extraction models are proposed to tackle this problem. It can share information and simultaneously extract entities and relations in a single model. The first joint tagging-based model is proposed by [4], which sets a new label for each token containing entity position information and relation type information. This tagging-based method achieves a significant improvement but cannot solve the overlapping problem: two relational triples share one or two entities. [5] further divides the overlapping problem into three scenarios (see Figure 1) : Normal, SingleEntiyOverlap (SEO) and EntityPairOverlap (EPO). [6, 7] propose different decomposition strategies and tagging schemes to handle SEO cases. Although these models can achieve better performance, they ignore the EPO cases that an entity pair has multiple relations. [8] makes an attempt to handle EPO cases, which artificially sets a threshold to determine the number of relations in entity pairs. Such an approach results in a situation where manual adjustment requires extra workload to achieve good performance on a particular dataset. Meanwhile, a fixed threshold affects the performance of relation classification and generalization of the model. On the other hand, these models cannot handle the nested entity problem since their tagging scheme merely serves as a yes or no decision when they detect the span of entities. The problem of missing nested entities misleads the model to extract wrong triples.

Normal	[Shanghai] is one of the most prosperous cities in [China] .
	<Shanghai, country, China>
SPO	[Andrew Mark Cuomo] was born in the [Queens] borough of [New York City].
	<Andrew Mark Cuomo, Born_in, Queens> <Andrew Mark Cuomo, Born_in, New York City>
EPO	[Chaplin] played a factory worker in his directed film [Modern City].
	<Chaplin, Direct, Modern City> <Chaplin, Act_in, Modern City>

Fig. 1. Examples of Normal, SingleEntiyOverlap (SEO) and EntityPairOverlap (EPO) classes.

In this paper, we propose a novel method to extract overlapping triples and handle the nested entity problem. Our main idea is to predict the number of entities and relations firstly, and then derive the corresponding triples. More precisely, our model is composed of two modules: subject extractor (SE) and relation extractor (RE). Each module of our model has a multi-head guided layer to filter out useless information and highlight valuable information. SE aims to extract all possible subjects. RE is comprised of two components. The transformer component is used to recognize the number of relations under a specific subject, and the other component predicts the probability distribution of relations. We propose a novel tagging scheme that annotates the relation numbers at the start and end positions. In the tagging process, if the number of relations is greater than 0, the token is regarded as an object candidate. We then use the nearest strategy to detect the span of entities. Finally, one or more relations in entity pairs are derived from mapping the probability distribution of relations. We evaluate our model on NYT [9] and WebNLG

*Corresponding author: wxhu@cc.ecnu.edu.cn
The source code of this paper: https://github.com/almbreeder/AGFRel
DOI reference number: 10.18293/SEKE2021-153

[10] public datasets and experiments show that our model has better performance than previous models.

In general, our contributions are as follows:

- We propose a novel model, Attention Guided Filter (AGFRel), which can learn the exact number of relations in the sentences. It means that the model can solve the overlapping problem in different scenarios.
- We adopt a novel tagging scheme to handle the nested entity problem. Our tagging scheme counts the number of occurrences of the entities and marks these values on the corresponding entities' first and last tokens.
- We embed the self-attention mechanism into our model to reduce the effect of irrelevant entities and highlight the crucial features.
- Our model achieves the most effective results on the NYT and WebNLG datasets. We further conduct various experiments on our model, including overlapping triples extraction and multiple triples extraction, and results show that our model exceeds baseline.

The rest of this paper is organized as follows. In section II, there is a brief overview of related work. In section III, we display our network architecture. Experiments and discussions are conducted in section IV. Finally, we conclude with a summary of our main contributions and results.

II. Related Work

Most researchers treated relation extraction as two sub-tasks: NER and RC. The NER task aims to extract all entity mentions in the context. The RC task aims to recognize the relation between entities in a given text. Early works used the pipeline approach that makes two sub-tasks work independently. Such an approach suffers from error propagation since it disregards the correlation between NER and RC tasks. [3, 11, 12] proposed traditional joint models to mitigate propagation error, which need complex feature engineering and heavily rely on manual effort. The joint relation extraction model based on neural networks was studied to solve this problem and achieved state-of-the-art performance. Most neural models like [13] combined entity extraction and relation classification in a network through sharing encoder parameters. [4] first introduced a united tagging schema that can represent entity type and relation simultaneously, which converted the relation extraction task to a sequence tagging problem without identifying entity and relation separately. Previous works cannot handle the overlapping problem. To handle the overlapping problem, lots of models have been proposed and can be categorized into three classes.

The first class of works used the sequence-to-sequence (seq2seq) method to extract triples. [5] divided triples into three classes and proposed a neural model CopyRE that utilizes copy mechanism to extract triples. [14] employed a neural encoder-decoder model for extracting relations that encoder predicts one word at a time like translation machine. [15] applied reinforcement learning into the sequence-to-sequence model to handle the triples extraction order problem. However, the proposed seq2seq models hardly decode the whole span of entities.

The second class designed Multi-task learning (MTL) strategy to extract relation facts. Among these works, [16] introduced multi-task learning based on CopyRE to deal with its drawback that cannot recognize multi-token entity. [17] encoded given texts with convolutional neural networks (CNN) to capture the feature of relation facts. [18] gained considerable improvement through building relation-weighted graph convolutional networks (GCN). [19] designed a novel multi-task learning architecture that enables dynamic interaction and mutual learning between NER and RC, which improves the ability to extract triples. Although effective, they lack the elegance to handle complex scenarios, such as EPO cases.

The third class method converted relation extraction to a sequence labeling problem. [6] proposed a tagging scheme based position-attention mechanism, which can solve SEO cases. [7] presented a novel decomposition strategy that hierarchically decomposes the extraction task into two sequence labeling problems but lacks the elegance to solve EPO cases. Unlike previous works, [8] proposed a new framework that maps subject to object and achieved reasonable performance in EPO cases.

Actually, these models that set an empirical value to select multiple relations inevitably lead to performance degradation. Besides, the sentences contain numerous triples in most cases. When extracting a specific triple, the model will be interfered with by other triples' feature information. As a result, these models cannot adapt to complex scenarios. Our model is based on a unified tagging scheme to extract triples. We utilize attention mechanism [20] to predict the number of relations for a given entity pair and then extract relations in a given triple.

III. Methodology

This section describes our proposed method. We first introduce the architecture of our model, where the shared encoder captures semantic features of a sentence, subject extractor recognizes subjects, and relation extractor predicts triples under a given subject. Then, we detail the novel tagging scheme of our method that converts the extraction task to the sequence labeling problem. Finally, we define the training objective.

A. Model Architecture

As shown in figure 2, our model consists of three parts: shared encoder, subject extractor (SE), and relation extractor (RE). We use the BERT [21] as a backbone to encode contextual features. The SE recognizes subjects and the RE predicts relations and objects according to these features. Formally, we extract a triple T_j in the sentence S and we model this process as:

$$P(T_j|S) = P(s_j|S)P(r_j|S, s_j)P(o_j|S, s_j, r_j) \quad (1)$$

where s_j, r_j and o_j represent subject, relation and object respectively in the triple T_j.

Eq.(1) illustrates the process of extracting triples. The first step of extracting triples in our model is to identify the

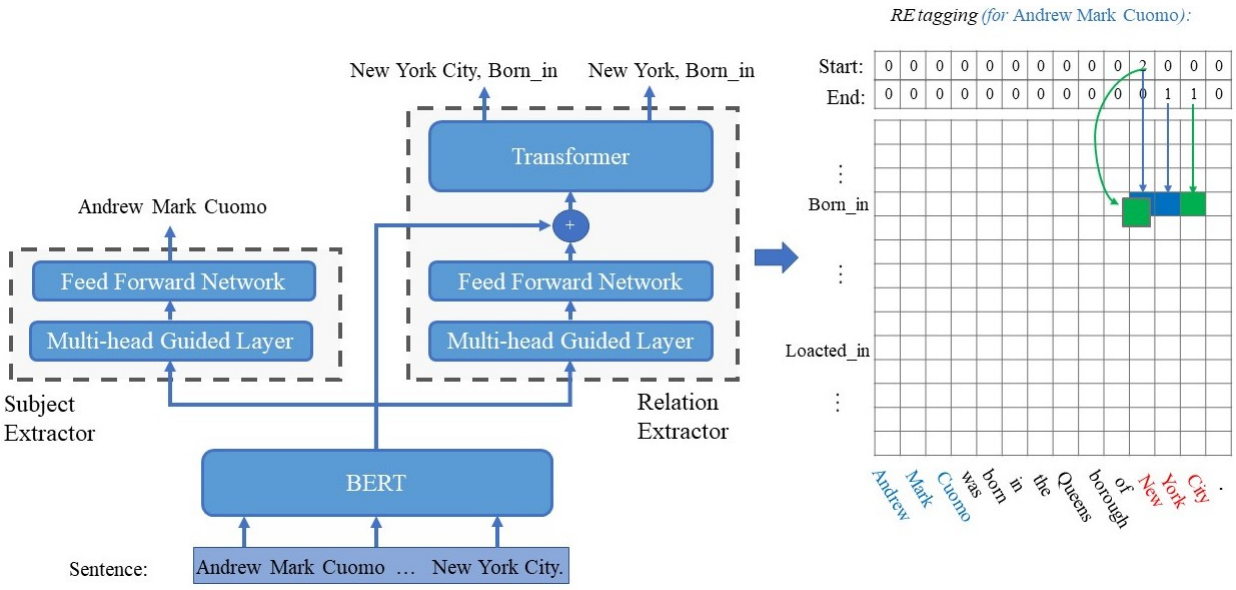

Fig. 2. The architecture of AGFRel.

subjects according to the semantic information of sentences. The difference is that our model does not directly identify the corresponding objects or relations. In order to solve the overlapping problem, we predict a relation probability distribution table for each subject and utilize transformer encoder to predict the number of relations. Finally, our model adopts the nearest match strategy to decide the span of objects. RE takes the concatenation of the sentence representation and the hidden representation of relation probability distribution as input to improve the accuracy of prediction. Meanwhile, an attention guided layer is embedded in each module of model. Such a mechanism ensures that the module filters out noises from other irrelevant triple features.

B. Shared Encoder

The model AGFRel utilizes a pre-trained BERT model to extract feature information from a given sentence $S = \{x_1, ..., x_n\}$, due to the excellent performance on different natural language processing tasks. BERT model employs transformer networks as the core component to obtain context-sensitive embeddings. We use WordPiece embeddings [22] to represent the words.

$$h_i = BERT_{shared}(x_i) \tag{2}$$

where h_i is hidden state at position i, x_i is a one-hot vector of word indice. In the training process, we fine-tune the parameters of pre-training model to make it better adapt to relation extraction task in different scenarios. The NER and RC task use a shared encoder, which is to pass h_i into corresponding module for prediction.

C. Subject Extractor

SE aims to recognize all candidate subjects. We embed a multi-head layer to decode the vector h_i. The attention mechanism allows SE to capture the interactions between two arbitrary positions and filter out the interference from other triples. Besides, the key component of BERT encoder is also self-attention mechanism, so the encoder and decoder maintain consistency.

$$h_i^{sbj} = MultiHead(h_i) \tag{3}$$

$$P(y_i^{sbj}) = \sigma(W_i^{sbj} * h_i^{sbj} + b_i^{sbj}) \tag{4}$$

$$Tag_{sbj}(x_i) = \arg\max_k P(y_i^{sbj} = k) \tag{5}$$

where h_i^{sbj} denotes hidden representation of word x_i and $P(y_i^{sbj})$ is the probability distribution of the number of subjects. W_i^{sbj} and b_i^{sbj} are learnable parameters of the multi-head layer. σ is the sigmoid activation function.

Subject extractor minimizes the sum of negative log probabilities of extracting subject candidates as below:

$$\mathcal{L}_{sbj} = -\frac{1}{n}\sum_{i=1}^{n} \log P\left(y_i^{sbj} = \hat{y}_i^{sbj}\right) \tag{6}$$

Here, n is the length of the input sentence, $\hat{y}_i^{sbj}$ is the true tag of the i-th word.

D. Relation Extractor

RE attempts to predict relational triples (s, r, o) from a sentence. Different from subject extractor, relation extractor needs to break down task into two steps. Firstly, we predict the probability distribution over the relation type r between each word and a given subject. The architecture of this component is similar to subject extractor. The specific operation is as follows:

$$h_i^{rel} = MultiHead(h_i + h_{s^k}) \tag{7}$$

$$P_r(y_i^{rel}) = \sigma(W_i^{rel} * h_i^{rel} + b_i^{rel}) \tag{8}$$

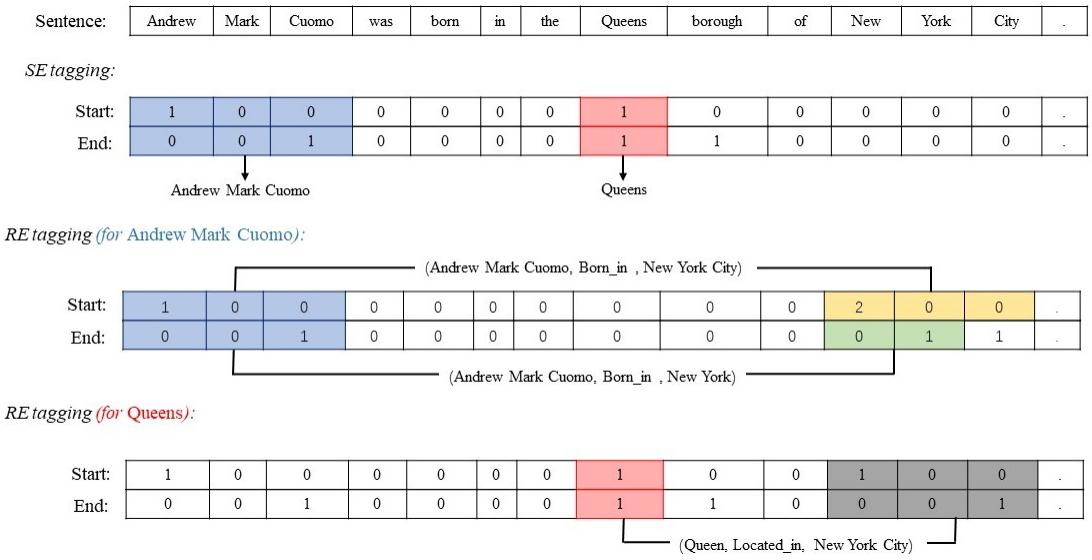

Fig. 3. An example of scheme tagging.

where h_{s^k} is the k-th subject hidden state and $P_r(y_i^{rel})$ is the probability of identifying relation type r between the word x_i and the subject s^k. The context and subject are the main sources to support the prediction [23]. We adopt multi-head mechanism to fuse hidden representation h_i with subject feature h_{s^k} into a single vector h_i^{rel}. It is worth noting that the gold subject is directly used in the training process, but not the subjects extracted by model. Secondly, we predict the number of relations between each word and a given subject as follows:

$$h_i^{num} = Trans(h_i + h_i^{rel}) \tag{9}$$

$$P(y_i^{num}) = \sigma(W_i^{num} * h_i^{num} + b_i^{num}) \tag{10}$$

$$Tag_{rel}(x_i) = arg\max(P(y_i^{num} = \hat{y}_i^{num})) \tag{11}$$

We denote the Transformer block as $Trans(x)$. $P(y_i^{num} = \hat{y}_i^{num})$ is the probability of identifying the number of relations between the word x_i and the subject s^k. The value of $Tag_{rel}(x_i)$ not only represents the number of relations, but also is used to recognize objects.

Relation extractor minimizes the sum of negative log probabilities of extracting relations as below:

$$\begin{aligned}\mathcal{L}_{rel} = &-\frac{1}{n}\sum_{i=1}^{n}\log P\left(y_i^{rel} = \hat{y}_i^{rel}\right) \\ &-\frac{1}{n}\sum_{i=1}^{n}\log P\left(y_i^{num} = \hat{y}_i^{num}\right)\end{aligned} \tag{12}$$

Here, n is the length of the input sentence, $\hat{y}_i^{rel}$ and $\hat{y}_i^{num}$ are the true tag of the i-th word. $P(*)$ is the probability of identifying true tags.

E. Novel Tagging Scheme

Figure 3 shows an example of how the sentences are tagged. Since identifying an entity needs to detect its start and end position, we label these critical positions with a non-zero number and the remain with 0. These labels have different meanings for different extractors. Labels in SE represent the number of subjects at their current position. For RE, labels indicate the number of relations between the object and corresponding subject. Relations can be extracted through the relation probability distribution table as shown in figure 2, in which the value of row i and column j means the probability that the j-th token is relation i.

For example, the word "Andrew" in start tag sequence and the word "Cuomo" in end tag sequence are both labeled 1, thus "Andrew Mark Cuomo" is a subject. When the given subject is "Andrew Mark Cuomo," RE labels the word "New" as 2, "York" as 1 and "City" as 1, which naturally solves the nested entity problem. The word "New" will be used two times to form two different triples in this sentence. The relations of two entity pairs can be inferred from the relation probability table, which are both "$Born\_in$."

Note that our tagging scheme is an improvement of ETL [7], which labels the entity's position with entity type or relation type. This tagging method is unable to express multiple relations and nested entities. Our tagging method uses digital tags to represent these essential information, which helps model solve the nested entity problem and the overlapping problem.

F. Training Objective

The module SE and the module RE jointly extract triples according to our tagging scheme. The training objective of AGFRel is comprised of two components: the loss function for SE $\mathcal{L}_{sbj}$ and the loss function for RE $\mathcal{L}_{rel}$. We try to give different weights to the loss function of two parts and found that slightly increasing the weight of RE could make the model fit faster in the early stage. However, it does not help improve the performance of the model. The joint loss is given by:

$$\mathcal{L} = \mathcal{L}_{sbj} + \mathcal{L}_{rel} \tag{13}$$

TABLE I
MAIN RESULTS ON NYT AND WEBNLG DATASETS. THE BEST PERFORMANCE IS MARKED AS BOLD-TYPE.

model	Exact Match						Partial Match					
	NYT			WebNLG			NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
NovelTagging	32.8	30.6	31.7	52.5	19.3	28.3	-	-	-	-	-	-
CopyRE	-	-	-	-	-	-	61.0	56.6	58.7	37.7	36.4	37.1
GraphRel	-	-	-	-	-	-	63.9	60.0	61.9	44.7	41.1	42.9
CopyMTL	75.7	68.7	72.0	58.0	54.9	56.4	-	-	-	-	-	-
ETL-Span	85.5	71.7	78.0	84.3	82.0	83.1	-	-	-	-	-	-
WDec	-	-	-	-	-	-	94.5	76.2	84.4	-	-	-
RIN	83.9	85.5	84.7	77.3	76.8	77.0	87.2	87.3	87.3	87.6	87.0	87.3
CasRel$_{LSTM}$	-	-	-	-	-	-	84.2	83.0	83.6	86.9	80.6	83.7
CasRel$_{BERT}$	-	-	-	-	-	-	89.7	89.5	89.6	**93.4**	90.1	91.8
AGFRel$_{LSTM}$	86.5	86.7	86.6	84.3	**86.5**	85.4	85.8	85.4	86.0	86.9	83.5	85.2
AGFRel$_{BERT}$	**87.9**	**91.0**	**89.4**	**87.0**	85.8	**86.4**	**90.7**	**91.0**	**90.8**	92.1	**91.7**	**91.9**

The model minimizes the total loss $\mathcal{L}$ over all model parameters with stochastic gradient descent algorithm.

IV. EXPERIMENT

A. Datasets

We evaluate AGFRel on two public datasets NYT and WebNLG. NYT dataset was sampled from 294K articles in New York Times corpus by distant supervision method and consists of 24 predefined relation types. The WebNLG dataset was created by Natural Language Generation (NLG) tasks. We use the datasets preprocessed by [5], which contains 246 predefined relation types. It is worth mentioning that there are two methods for extracting relations. [5, 8] use partial match method simplifying an entity to the last word of an entity. [7, 16] use exact match method to extract the whole of entities. For fair comparisons, we use partial match and exact match to conduct experiments. We use the preprocessed datasets released by [5]. NYT dataset contains 5000 sentences for validation and 5000 sentences for testing. WebNLG dataset contains 500 sentences for validation and 703 sentences for testing. In order to validate the effectiveness of extracting overlapping triples, the test set was divided into three parts: Normal, SEO, and EPO. The statistics are stated in Table II.

TABLE II
STATISTICS OF THE DATASETS IN OUR EXPERIMENTS.

class	Train	Valid	Test	Normal	SEO	EPO
NYT	56195	5000	5000	3266	1297	978
WebNLG	5019	500	703	246	457	26

B. Evaluation Metrics

We adopt the standard micro Precision (Prec.), Recall (Rec.), and F1 score to evaluate results in line with all the baselines. In the exact match task, the triples are considered correct when the whole span of entities and relations are both recognized correctly. The partial match task only needs to recognize the tail of entities and relations.

C. Implementation Details

We use Adam [24] with the learning rete $1e^{-6}$ to optimize the parameters of our model and set the batch size as 32. Dropout is applied to word embeddings and hidden states with a rate of 0.4. In this paper, we propose two models with LSTM and BERT encoder, and keep the network of decoder and encoder uniform respectively. The model with LSTM stacks two-layer BiLSTM as the encoder, one layer BiLSTM as decoder. The initial word embeddings we used are the 300 dimensions Glove [25]. The other one uses the base cased version of BERT as encoder, which contains $110M$ parameters. The maximum length of sentences is limited to 100 words. We implement model using Pytorch [26] on a Linux machine and train the model using Tesla V100 GPU. We choose the model that performed best on the validation set to analyze the test set.

D. Comparison Models and Results

We compare our model with three kinds of models in recent years: (1) seq2seq-based methods, including CopyRE [5] and WDec [14], (2) MLT-based methods, including GraphRel [18], CopyMTL [16] and RIN [19], (3) tagging-based methods, including NovelTagging [4], ETL-Span [7] and CasRel [8]. Table I shows the results of our models and other baseline methods. We note that our model surpasses all the baseline methods and achieves the state-of-the-art F1 score. We propose two versions of AGFRel to conduct experiments. The AGFRel$_{LSTM}$ uses the LSTM as shared encoder and replaces the multi-head guided layers as LSTM.

In the exact match task, our model AGFRel$_{BERT}$ achieves improvements of 4.7% and 4.3% in F1 scores on NYT and WebNLG datasets over the state-of-the-art models. Even with the AGFRel$_{LSTM}$ also has a relative 1.9% F1 score improvement on NYT dataset compared with MLT-based model RIN, and a relative 2.3% F1 score improvement on WebNLG dataset compared with tagging-based model ETL-Span. These results prove the effectiveness of our method.

In the partial match task, the performance of AGFRel$_{BERT}$ is close to CasRel$_{BERT}$ on WebNLG dataset. We deem that it

TABLE III
F1 SCORE ON SENTENCES WITH DIFFERENT OVERLAPPING PATTERN AND DIFFERENT TRIPLE COUNT.

Method	Number of triples					Overlapping pattern		
	N = 1	N = 2	N = 3	N = 4	N = 5	Normal	SEO	EPO
CopyRE	67.1	58.6	52	53.6	30	66	48.6	55
GraphRel	71	61.5	57.4	55.1	41.1	69.6	51.2	58.2
CasRel$_{BERT}$	**88.2**	90.3	91.9	94.2	83.7	87.3	91.4	92
AGFRel$_{BERT}$	88.1	**91.6**	**93.2**	**94.5**	**86.5**	**87.9**	**92.3**	**92.7**

is because (1) the last word of the entity losses key semantic information, i.e., Apollo 12 is regarded as 12, which impedes models from extracting correct triples. (2) The performances are already saturated since the training sentences are too small for the model to learn the way to distinguish 246 relation types properly. The training sentences of NYT dataset are far more than WebNLG dataset. Our model achieves an improvement of 1.2% in F1 score. Besides, we find that CasRel$_{BERT}$ shows an imbalance between precision and recall on WebNLG. We consider that CasRel$_{BERT}$ sets a high threshold value to select relations, which sacrifices recall.

We observe that AGFRel$_{BERT}$ gains better performance on partial match task than exact match task. In the partial match task, model only needs to detect the last token of entities. We speculate that the performance gap is due to increased difficulty for the NER task. Although we mentioned that the last word of entities could not represent the original meaning of entities, entities that contains many words will also affect AGFRel since our tagging scheme is only to label the start and end position of entities.

E. Analysis on Different Sentence Types

To verify the ability of our model in handling the overlapping problem and extracting multiple relations, we conduct further experiments on NYT test set. We firstly divide the test set of NYT into five subclasses, each of which means a sentence contains N triples. The results are shown in table III. We observe that CopyRE and GraphRel present a decreasing trend with the increasing number of triples in a sentence. Our model and CasRel$_{BERT}$ present an upward trend on the whole and are less affected by the number of triples. We attribute the difference to the reason that tagging-based methods simplify the complexity of tasks by converting relation extraction task to a sequence labeling problem. In the most challenging class($N \geq 5$), our model achieves better performance than CasRel$_{BERT}$, since we introduce attention mechanism to eliminate the impact of irrelative triples. We also compare the results under different types of triples. AGFRel$_{BERT}$ outperforms baselines in all scenarios, which proves the validity of our tagging scheme for solving the overlapping problem.

V. CONCLUSION

In this paper, we propose an end-to-end sequence labeling model AGFRel for joint extraction of entities and relations based on a novel decomposition strategy. Compared to previous sequence labeling models, our model can learn the exact number of relations for each entity pair to filter out relation distracters in sentences. Our experiments show that our results exceed the baseline and achieve the optimal F1 score. Moreover, it has a good performance in handling the overlapping problem and extracting multiple triples. In the future, we will utilize GCN to encode the document-level text. The overlapping problem and nested entity problem also exist in the biomedical domain. It is beneficial to apply our method to biomedical information extraction tasks.

ACKNOWLEDGMENTS

We thank all viewers who provided thoughtful and constructive comments on this paper. This work is funded by the Fundamental Research Funds for the Central Universities. East China Normal University, Shanghai 200241, China. The experiment is completed with the support of ECNU Multifunctional Platform for Innovation (001).

REFERENCES

[1] C. Quirk and H. Poon, "Distant supervision for relation extraction beyond the sentence boundary," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 2017, pp. 1171–1182.

[2] Y. Zhang, V. Zhong, D. Chen, G. Angeli, and C. D. Manning, "Position-aware attention and supervised data improve slot filling," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 35–45.

[3] Q. Li and H. Ji, "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)*, 2014, pp. 402–412.

[4] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, and B. Xu, "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)*, 2017, pp. 1227–1236.

[5] X. Zeng, D. Zeng, S. He, K. Liu, and J. Zhao, "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)*, 2018, pp. 506–514.

[6] D. Dai, X. Xiao, Y. Lyu, S. Dou, Q. She, and H. Wang, "Joint extraction of entities and overlapping relations using position-attentive sequence labeling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6300–6308.

[7] B. Yu, Z. Zhang, X. Shu, Y. Wang, T. Liu, B. Wang, and S. Li, "Joint extraction of entities and relations based on a novel decomposition strategy," *arXiv preprint arXiv:1909.04273*, 2019.

[8] Z. Wei, J. Su, Y. Wang, Y. Tian, and Y. Chang, "A novel cascade binary tagging framework for relational triple extraction," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 1476–1488.

[9] S. Riedel, L. Yao, and A. McCallum, "Modeling relations and their mentions without labeled text," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2010, pp. 148–163.

[10] C. Gardent, A. Shimorina, S. Narayan, and L. Perez-Beltrachini, "Creating training corpora for nlg micro-planners," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 179–188.

[11] X. Yu and W. Lam, "Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach," in *Coling 2010: Posters*, 2010, pp. 1399–1407.

[12] M. Miwa and Y. Sasaki, "Modeling joint entity and relation extraction with table representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1858–1869.

[13] M. Miwa and M. Bansal, "End-to-end relation extraction using lstms on sequences and tree structures," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1105–1116.

[14] T. Nayak and H. T. Ng, "Effective modeling of encoder-decoder architecture for joint entity and relation extraction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 8528–8535.

[15] X. Zeng, S. He, D. Zeng, K. Liu, S. Liu, and J. Zhao, "Learning the extraction order of multiple relational facts in a sentence with reinforcement learning," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 367–377.

[16] D. Zeng, H. Zhang, and Q. Liu, "Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 9507–9514.

[17] H. Adel and H. Schütze, "Global normalization of convolutional neural networks for joint entity and relation classification," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1723–1729.

[18] T.-J. Fu, P.-H. Li, and W.-Y. Ma, "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*, 2019, pp. 1409–1418.

[19] K. Sun, R. Zhang, S. Mensah, Y. Mao, and X. Liu, "Recurrent interaction network for jointly extracting entities and classifying relations," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 3722–3732.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6000–6010.

[21] J. Devlin, M.-W. Chang, K. Lee, and K. N. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2018, pp. 4171–4186.

[22] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[23] H. Peng, T. Gao, X. Han, Y. Lin, P. Li, Z. Liu, M. Sun, and J. Zhou, "Learning from context or names? an empirical study on neural relation extraction," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 3661–3672.

[24] K. Da, "A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[25] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 8026–8037.