A Sequence Tagging based Framework for Few-Shot Relation Extraction

1st Xukun Luo  
Peking University  
Beijing, China  
luoxukun@pku.edu.cn

2nd Ping Wang*  
Peking University  
Beijing, China  
pwang@pku.edu.cn

Abstract—Relation Extraction (RE) refers to extracting the relation triples in the input text. Existing neural work based systems for RE rely heavily on manually labeled training data, but there are still a lot of domains where sufficient labeled data does not exist. Inspired by the distance-based few-shot named entity recognition methods, we put forward the definition of the few-shot RE task based on the sequence tagging joint extraction approaches, and propose a few-shot RE framework for the task. Besides, we apply two actual sequence tagging models to our framework (called Few-shot TPLinker and Few-shot BiTT), and achieve solid results on two few-shot RE tasks constructed from a public dataset.

Index Terms—relation extraction, text analysis, natural language processing

I. INTRODUCTION

Relation Extraction (RE) is a fundamental task in natural language processing, aiming to extract the relation triples of the form \( (h, r, t) \) in unstructured text. In a relation triple, \( h \) and \( t \) are the entities in the text, called the head entity and the tail entity respectively according to their positions. And \( r \) describes the directed relationship from the head entity to the tail entity. Currently, the mainstream solutions follow the idea of joint extraction, which are mainly divided into two methods: building up a label predicting system with a sequence tagging scheme [19], [20], [26] and constructing a seq2seq system [15], [24], [25]. These systems have achieved solid results on the generic RE datasets with a sufficient amount of labeled instances.

However, one challenge remaining is that these RE systems based on the neural network require a lot of labeled training data, while the tagging effort is labor-intensive. The distant supervision method [12] was proposed to obtain training data more cost-effectively. Based on existing knowledge graphs, the instances in the corresponding domain are automatically annotated. But a serious problem is that the samples labeled by the distant supervision method suffer from a large number of missed and incorrect tags. Thus the models must be reconstructed to accommodate these datasets with noise information. In addition, if there is no available knowledge graph for a particular domain, it is still impossible to complete the annotation for the training data.

As a result, researchers started to explore how to transfer knowledge from the resource-rich domains (such as the news domain) to the resource-poor domains (such as the medical domain). They worked to make the model achieve better results based on a small amount of labeled data in the target domain. Previously, they followed the simple pipeline idea and respectively assigned the knowledge transferred task to RE’s two sub-tasks, i.e. named entity recognition (NER) [6], [9], [22] and relation classification (RC) [1], [7], [8]. These works are all distance-based approaches, drawing on the classical few-shot approaches in the image field. However, the error propagation of the pipeline is self-explanatory. Currently, the methods of few-shot NER and few-shot RC are not effective enough, let alone combining the two for RE.

Besides, the few-shot works in named entity recognition are essential to find the closest label to a particular token of the input text. The idea is also applicable to the sequence label predicting systems in the joint extraction methods. Therefore, it is effective and feasible to apply the few-shot methods to joint extraction models.

In this paper, we propose a method\(^1\) to combine a sequence tagging based model for joint extraction and the few-shot methods to improve the performance with limited target domain’s labeled data. In order to adopt the idea of distance-based few-shot sequence tagging to joint extraction methods, we begin with proposing a definition of the few-shot RE task through the sequence tagging based approaches. Second, we construct a sequence tagging based few-shot RE framework for the few-shot RE task. In our framework, the input instances are encoded by a BERT Encoder and an Adaptive Encoder, and then the distance between the hidden states of tokens and the labels is calculated, finally the predicted label sequence is outputted through distance comparison. Third, we apply two actual sequence tagging RE models [11], [19] to the framework. These two models achieve solid results on two few-shot RE tasks constructed from a public dataset.

The key contributions are summarized as:

- The definition of the few-shot RE task based on the sequence tagging approaches is presented for the first time.

\(^1\)Available at https://github.com/LuoXukun/fewshot-joint.
A few-shot joint RE framework is developed for sequence tagging approaches, and two actual sequence tagging models are applied.

An accelerated approximation scheme is proposed for calculating the distance matrices in the training phrase of Section IV-B.

II. RELATED WORK

A. Relation Extraction

RE, as a sub-task of information extraction, is an important part of building knowledge graphs. Some supervised RE methods followed the pipeline ideas [18], [21], [23], [27], i.e., one model is used to recognize entities in a sentence, and then another model is used to identify the relations between each entity pair. The two models have different structures and parameters. Due to the error propagation problem of the pipeline, other models turned to the joint method. They extracted entities and relations separately through two models that share parameters, or even simultaneously using a single model. For example, [10], [13] extended the relation classification module to share the encoder representation with the entity recognition module; [3], [11], [19], [20], [26] took the extracting tasks as a sequence labeling problem, worked on proposing reliable tagging scheme to reduce out relational triples; [15], [24], [25] directly generated the triples by the seq2seq framework.

B. Few-shot Relation Extraction

The above supervised methods rely on a large amount of training data in the target domain. However, the labeling of relational triples in the sentence is labor-intensive, resulting in frequent shortages of the target domain’s training data. Thus, learning from limited new labeled samples, called few-shot relation extraction, is a non-trivial task. To simplify the task, previous researchers followed the pipeline idea, which is different from the few-shot joint RE work in our paper.

a) Few-shot NER: There are already a few approaches for Few-shot NER. An intuitive one is to extract cross-domain generic knowledge based on the source domain’s labeled data, and then utilize the target domain’s limited data for knowledge migration. To carry on, [6] applied the prototypical network [16] for Few-shot NER, utilizing prototypes comparison to tag tokens. [9] put forward the collapsed dependency transfer mechanism, adapting the traditional conditional random field (CRF) to the mutable entity label categories. [22] proposed the nearest neighbor comparison scheme and pre-trained the few-shot NER model by a standard supervised NER method. [17] found and explored the rich semantics in O class. And [5] presented a practical and challenging dataset named FewNERD for few-shot NER.

b) Few-shot RC: Given one entity pair that appears in a sentence, the RC task becomes a multi-classification problem. [8] were the first to formalize the definition of few-shot RC and proposed a new dataset called FewRel. They apply several few-shot learning methods to FewRel, e.g., prototypical network and meta network [14]. [7] built up FewRel2.0 for few-shot RC with two real-world issues, i.e., few-shot domain adaptation and few-shot none-of-the-above detection. [1] employed a simple prototype approach for a CNN-based system. What’s more, a more in-depth analysis [2] is performed on the above works in a more realistic RE setting.

Algorithm 1 N-way K ~ 2K-shot Sampling

Require: N, K, Data X, Relation Category set C
Ensure: A support set S
1: Initialize S ← ∅;
2: for each i ∈ {1, ..., N} do
3: Count_i ← 0;
4: end for
5: while ∃Count_i ≤ K, 1 ≤ i ≤ N do
6: Randomly sample \( \{x, R\} \) ∈ X;
7: Calculate Count_i after update;
8: if Count_i ≤ 2K, ∀i ∈ {1, ..., N} then
9: for each i ∈ {1, ..., N} do
10: \( y_i \leftarrow \mathcal{T}(x, R, i) \);
11: end for
12: S ← S ∪ \{(x, y_1, ..., y_N)\};
13: end if
14: end while

III. Problem Definition

In the sequence tagging based approaches of joint extraction, the label sequence \( y = (y_1, y_2, ..., y_m) \) can be obtained by a specific invertible tagging function \( \mathcal{T}(x, R, r) \), where \( x = (x_1, x_2, ..., x_m) \) is a sequence of tokens. \( R \) indicates the set of relational triples in \( x \) and \( r \) is a specific relation category. Note that the numerical relationship between \( m \) and \( n \) is related to function \( \mathcal{T} \) from previous works, e.g., \( n = m \) in [26] and \( n = m^2 \) in [3].

Thus in few-shot RE, we intend to obtain a function \( \mathcal{F} : (C, S, Q) \rightarrow \{y_1, ..., y_{|C|}\} \). Here \( C \) indicates a set of \( N \) relation categories. And \( S = \{(x^{(i)}, y_1^{(i)}, ..., y_{|C|}^{(i)})\}_{i=1}^N \) is the support set, including \( N \) relation categories (N-way) and \( K \) instances (K-shot) for each category in the target domain. However, there may be multiple categories of triples in a single instance, making it difficult to sample from the source domain when training. To solve the problem, similar to [5], we introduce the N-way \( K \sim 2K \)-shot setting into few-shot RE task, i.e., a sampled \( S \) contains \( K \sim 2K \) instances for each relation category. Besides, \( Q \) is called the query set, which is the collection of the token sequences that need to be heuristically labeled based on \( S \). For simplicity, the number of instances in \( Q \) is assumed to be 1, i.e., \( |Q| = 1 \). And \( \{y_1, ..., y_{|C|}\} \) is the prediction of the single instance \( x \) in \( Q \).

IV. Methodology

In this section, we first present our sequence tagging based few-shot RE framework in general. Then we give two examples of how to apply the previous sequence tagging models for joint extraction to our framework, i.e., Few-shot TPLinker [19] and Few-shot BiTT [11].
A. Few-shot Relation Extraction Framework

As shown in Fig. 1, our few-shot RE framework for sequence tagging approaches consists of the following 4 steps.

First, for each relation category, our framework needs to generate a label sequence by the function $T$ for each instance in $S$. For example, $y^{(1)}_1$ in Fig. 1 is the label sequence derived from the instance $x^{(1)}$ in $S$ and the relation triple set in $x^{(1)}$ with relation category $r_1$.

Second, each token sequence $x$ in $S$ and $Q$ along with the description of a specific relation $r$ are input into a BERT Encoder [4] to get the embedding of each token in $x$, as follow.

$$E = BertEncoder([r; x])$$ (1)

Note that in Eq.(1) we utilize [SEP] as the concatenation token of $r$ and $x$.

Third, for different sequence tagging models, our framework sets up an Adaptive Encoder, allowing models to obtain the hidden states for each position to be labeled that meet their own requirements.

$$H = AdaptiveEncoder(E)$$ (2)

Here the dimension of $H$ is $n \times N_h$, where $n$ is related to the tagging strategy $T$ and $N_h$ is the pre-defined hidden size. Note that the parameters of the BERT Encoder and the Adaptive Encoder can be pre-trained in advance with a traditional sequence tagging task in the source domain by appending a linear layer after $H$ and maximizing the probability of every correct label.

Finally, we compute the distance of the hidden state between each position of $y$ in $S$ and $Q$.

$$D = D(H^{(Q)}, H^{(S)})$$ (3)

Here $D \in \mathbb{R}^{n \times |S|}$ is the distance matrix, where $|S|$ means the instance number in $S$. $H^{(Q)}$ and $H^{(S)}$ are the hidden states of $Q$ and $S$ respectively from Eq.(2). Besides, $D(\cdot)$ indicates the squared Euclidean distance function. Note that since $y$ is chunked in some specific tagging strategies, the distances between positions across chunks are not calculated to avoid confusion of the labels in different chunks. For example, in TPLinker [19], there are three tagging matrices that contain different tags and represent different information. The labels sequence in a single matrix for all tokens is called a chunk and $y$ generated based on the TPLinker tagging strategy contains three chunks. For the inference stage, we utilize the label of the closest position $p_s$ in $S$ to a specific position $p_q$ in $Q$ as $p_q$’s label, as follow.

$$y^*(p_q) = t(p_s)$$
$$p_s = \arg\min_{1 \leq i \leq n|S|} D[p_q, i]$$ (4)

Here $y^*$ is the predicted label sequence of $Q$ for the specific relation $r$ and $t(\cdot)$ is the mapping function from each position in $S$ to its label. And for the training stage, we assume that $y$ is divided into $\lambda$ chunks of equal length. In a specific chunk with index $c \in \{0, \cdots, \lambda - 1\}$, the distance between a position $p$ in $Q$ and a label $l$ can be calculated by:

$$D^l_c[p] = \min_{p^* \in \mathcal{W}^l_c} D[p, p^*]$$ (5)

where $\mathcal{W}^l_c$ indicates the set of all positions labeled with $l$ in the chunk $c$ of $S$. Note that the chunk $c$ contains $N_c$ kinds of labels, i.e., $l \in \{0, \cdots, N_c - 1\}$. And we apply the cross entropy loss function for every chunk to minimize the...
distances between \(p\) and the correct label, as follow.

\[
\mathcal{L}_c = -\frac{\lambda}{nN_C} \sum_{i=1}^{n} \sum_{l=1}^{N_c} (\hat{y}_{m+i}^{\neq} (l) \mathcal{P}(\mathbf{D}_c[i]^{\neq}))
\]  

(7)

\[
\mathcal{P}(\mathbf{D}_c[i]) = \log\left(\frac{e^{-\mathbf{D}_c[i]}}{\sum_{l'}^{N_c} e^{-\mathbf{D}_c[l']}}\right)
\]  

(8)

Here \(\hat{y}_{m+i}^{\neq} \in \mathbb{R}^{N_C}\) is the one-hot form of \(y_{m+i}\), which means the label index of the \(i\)-th position in chunk \(c\) of \(y\). The final loss \(\mathcal{L}\) is the average of \(\mathcal{L}_c\).

**B. Few-shot TPLinker**

TPLinker [19] is a one-stage joint RE model. It predicts three tagging matrices for the input instance, i.e., EH-to-ET, SH-to-OH, and ST-to-OT. The elements in the matrices belong to \(\{0, 1, 2\}\). Note that the label 2 is proposed to reduce the matrix space by mapping the label 1 in the lower triangular region to the upper triangular region. However, in the few-shot RE task, there may be no label 2 in \(S\), making it hard to calculate the distance from the positions to be annotated to label 2. Therefore, we cancel this improvement when applying TPLinker to our framework. In this case, the length of the label sequence \(y\) for a specific relation category is \(3n^2\), and \(y_i \in \{0, 1\}\) where \(1 \leq i \leq 3n^2\). And the number of chunks in TPLinker is \(\lambda = 3\). The following describes the process for a single chunk \(c\) of applying TPLinker to the sequence tagging based few-shot RE framework.

**a) Adaptive Encoder:** For the optimization of the next distance calculation step, we define the hidden state \(\mathbf{H}[im+j]\) of \((x_i, x_j)\) as a concatenation of two parts, as follow.

\[
\mathbf{H}[im+j] = [\mathbf{H}_h[i]; \mathbf{H}_t[j]]
\]  

(9)

Here \(\mathbf{H}_h[i]\) and \(\mathbf{H}_t[j]\) respectively denote the hidden states of the head token \(x_i\) and the tail token \(x_j\). They are calculated by two linear layers:

\[
\mathbf{H}_h = \mathbf{W}_h \mathbf{E}[i] + \mathbf{b}_h
\]  

(10)

\[
\mathbf{H}_t = \mathbf{W}_t \mathbf{E}[j] + \mathbf{b}_t
\]  

(11)

**b) Distance Matrix:** As shown in Fig. 2, the distance between the token pairs \((x_i, x_j)\) in \(Q\) and \((x_{i'}, x_{j'})\) in \(S\) is divided into two part, as follow.

\[
\mathbf{D}[im+j, ism+j] = \mathbf{D}_h[i, i'] + \mathbf{D}_t[j, j']
\]  

(12)

Here \(\mathbf{D}_h, \mathbf{D}_t \in \mathbb{R}^{m \times m(\mathbb{S})}\) are distance matrices between the tokens in \(Q\) and \(S\). \(\mathbf{D}_h\) and \(\mathbf{D}_t\) are generated from \(\mathbf{H}_h\) and \(\mathbf{H}_t\) respectively by Eq.(3). Note that \(x_i, x_j\) belong to the same instance in \(S\). In Few-shot TPLinker, the distances from \((x_i, x_j)\) to label 1 and label 0, called \(\mathbf{D}_p[i, j]\) and \(\mathbf{D}_n[i, j]\), is required to be obtained respectively. If the solution is performed by traversing all token pairs in \(S\), the overall time complexity for working out \(\mathbf{D}_p\) and \(\mathbf{D}_n\) can be \(O(m^4|\mathbb{S}|)\). For example, as shown in the left part of Fig. 2, all distances between \((x_i, x_j)\) and all token pairs in \(S\) should be computed once, for a total of 16 times. Thus, we devise an accelerated approximation scheme in the training phase. First, iterate over the positive samples set \(\mathbb{W}\) to fill \(\mathbf{D}_p\), since \(|\mathbb{W}| \ll |\mathbb{S}|\).

\[
\mathbf{D}_p[i, j] = \min_{(x_i, x_j) \in \mathbb{W}} (\mathbf{D}_h[i, i'] + \mathbf{D}_t[j, j'])
\]  

(13)

Second, assume that there is at least one negative sample among the top-\(\mathbb{E}\) tokens pairs with the shortest distance from \((x_i, x_j)\). These top-\(\mathbb{E}\) distances \(\mathbf{D}[i, j] \in \mathbb{R}^\mathbb{E}\) can be given by:

\[
\mathbf{D}[i, j] = \text{top}_\mathbb{E}(\hat{\mathbf{a}}^s_i(\epsilon_1) + \hat{\mathbf{b}}^s_j(\epsilon_2))
\]  

(14)

where \(1 \leq s \leq |\mathbb{S}|, 1 \leq \epsilon_1, \epsilon_2 \leq \mathbb{E}, \) and \(\mathbb{E} \geq 2\). \(\hat{\mathbf{a}}^s_i \in \mathbb{R}^\mathbb{E}\) and \(\hat{\mathbf{b}}^s_j \in \mathbb{R}^\mathbb{E}\) denote the distances respectively from \(x_i\)'s and \(x_j\)'s top-\(\mathbb{E}\) nearest tokens of the \(s\)-th instance in \(S\).

\[
\hat{\mathbf{a}}^s_i = \text{top}_\mathbb{E}(\mathbf{D}_h[i, sm + a]), 1 \leq a \leq m
\]  

(15)

\[
\hat{\mathbf{b}}^s_j = \text{top}_\mathbb{E}(\mathbf{D}_t[j, sm + b]), 1 \leq b \leq m
\]  

(16)

Last, denote the set of values in \(\mathbf{D}[i, j]\) except for \(\mathbf{D}_p[i, j]\) as \(W\). \(\mathbf{D}_n\) is filled with the minimum (Eq.(17)) or average (Eq.(18)) value in \(W\). Thus the time complexity of our accelerated approximation scheme for \(\mathbf{D}_p\) and \(\mathbf{D}_n\) is \(O(m^2(|\mathbb{W}| + \mathbb{E}))\).

\[
\mathbf{D}_n[i, j] = \begin{cases} 
\mathbf{D}[i, j], & \mathbf{D}_p[i, j] \neq \mathbf{D}[i, j] \\
\mathbf{D}[i, j], & \mathbf{D}_p[i, j] = \mathbf{D}[i, j]
\end{cases}
\]  

(17)

\[
\mathbf{D}_n[i, j] = \frac{\sum_{\mathbf{D}[i, j] \in \mathbb{W}} \mathbf{D}[i, j]}{|\mathbb{W}|}
\]  

(18)

**C. Few-shot BiTT**

BiTT [11] is an end-to-end RE framework with the bidirectional tree tagging scheme. It predicts a label sequence containing the tree structures of the relational triples for an input sentence, and then reconstructs the relational graph to extract triples. In the BiTT scheme, the chunk number \(\lambda\) is 8, and the length of \(y\) is \(\lambda m\). The following introduces the procedure in a single chunk \(c\) for adopting BiTT into our framework.
We define that the hidden state of $y_{t+m+1}$ is related to the embedding of $x_i$ by the following equation:

$$H_c[i] = W_c E[i] + b_c$$  \hspace{1cm} (19)

where $W_c$ is the weighted matrix and $b_c$ is the bias.

b) Distance Matrix: As shown in Fig. 3, we calculate the distances from $x_i$ to the tokens in $S$ and work out the minimum distance for each label. Since there may not be any token with label $l$, we calculate the distance between $x_i$ and a prototype $P_c[l]$ inherited from previous training steps.

$$P_c[l] = (1 - \gamma)P_c[l] + \frac{\gamma}{W_c} \sum_{x_i \in W_c} H_c[i]$$  \hspace{1cm} (20)

Here $W_c$ is the set of all tokens labeled with $l$ in the chunk $c$ of $S$. Note that the update operation is executed only if $W_c \neq \emptyset$.

V. Experiments

A. Dataset and Evaluation Tasks

Considering that different RE datasets are generated with different methods and relation category naming standards, it may face the issues of inconsistent relation descriptions and contradictory common knowledge when using multiple datasets for few-shot experiments. Moreover, it is practical to utilize a single dataset to perform different few-shot task scenarios, and then effectively validate the performance of our framework. Therefore, we evaluate Few-shot TPLinker and Few-shot BiTT on NYT [23], an English RE dataset with 66,195 instances. NYT is labeled by the distant supervision method and a hierarchical naming standard. It pre-defines 24 fine-grained and 4 coarse-grained relation categories.

Following the few-shot task setting of [5], we split the overall relation triple set into groups on consideration for the different granularity of category names. In this case, every group corresponds to a sub-dataset of instances containing the triples in the group. Besides, to avoid the observation of triples belonging to other groups, we need to mask out these triples first and then perform sequence tagging on the instances. Based on this setting, we develop two few-shot RE tasks adopting different splitting schemes and conduct 2-way 1:2-shot experiments on these tasks.

a) Few-NYT (INTER): In this task, we randomly divide the 24 relation categories in NYT into group A, group B and group C. As shown in Table I, every group is assigned 8 fine-grained relation categories, and the instances number of three sub-sets are 56,546, 14,942 and 8,284 respectively. We do not consider the coarse-grained type to which each fine-grained relation category belongs. For example, /people/person/ethnicity and /people/person/children all belong to the coarse-grained type people, while they are separately partitioned into group A and group C. In the experiments of this task, we take each group as the target domain for evaluating, and the remaining two groups as the source domain for training. In Few-NYT (INTER), our models may be more capable of mining the common knowledge between the source and target domains, since the three groups share some of the coarse information.

b) Few-NYT (INTRA): In NYT, there are 4 predefined coarse-grained relation types, i.e., location, business, sports and people. When dividing these four relation types, we consider about the balance of the amount of data in the training and validation sets, and also expect to mask out as few as possible relation triples. Thus, we simply adopt the instances set of location, business and sports as the training set, and the instances set of people as the validation set. The instances number of the training, validating and testing sets are 45,259, 6,300 and 16,711 respectively. In contrast to Few-NYT (INTER), the source and target domains of Few-NYT (INTRA) do not share coarse information as there is no intersection of coarse-grained relation types. Therefore, our models may be able to explore less inter-domain pervasive information, making this task more difficult.

| Group | Categories                                                                 | Instances Num |
|-------|-----------------------------------------------------------------------------|---------------|
| A     | /people/person/ethnicity, /location/location/contains, /business/company/founders, /people/person/nationality, /business/company/advisors, /business/person/company, /location/country/capital | 56,546        |
| B     | /business/company/shareholder/major_shareholder_of, /people/ethnicity/people, /location/neighbourhood/neighbourhood_of, /business/company/major_shareholders, /people/person/place_of_birth, /business/company/place_founded, /sports/sports_team/location | 14,924        |
| C     | /people/ethnicity/geographic_distribution, /people/person/religion, /people/person/children, /business/company/industry, /people/deceased_person/place_of_death | 8,284         |
TABLE II
MAIN RESULTS ON THE FEW-NYT (INTER) TASK. THE BEST RESULTS ARE IN **BOLD**.

| Model (%)       | Group A          | Group B          | Group C          | Average          |
|-----------------|------------------|------------------|------------------|------------------|
|                 | Prec | Rec | F1   | Prec | Rec | F1   | Prec | Rec | F1   | Prec | Rec | F1   | Prec | Rec | F1   |
| NNshot(NER)+BERTPAIR(RC) | 4.1  | 9.4  | 5.7  | 2.6  | 11.9 | 4.2  | 1.8  | 5.0  | 2.7  | 2.8  | 8.8  | 4.2  |      |      |      |
| NNshot(NER)+Proto(RC)      | 0.5  | 12.8 | 0.9  | 0.8  | 18.3 | 1.5  | 0.5  | 12.1 | 0.9  | 0.6  | 14.4 | 1.1  |      |      |      |
| Proto(NER)+BERTPAIR(RC)    | 6.7  | **21.8** | 10.2 | 3.5  | 18.3 | 5.8  | 2.9  | 9.7  | 4.4  | 4.4  | 16.6 | 6.8  |      |      |      |
| Proto(NER)+Proto(RC)       | 0.4  | 16.3 | 0.8  | 0.7  | 21.6 | 1.4  | 0.6  | 16.2 | 1.2  | 0.57 | 18.0 | 1.1  |      |      |      |
| Few-shot TPLinker (min)    | 21.3 | 18.7 | 19.9 | 50.1 | **33.6** | **40.2** | 37.3 | **24.0** | **29.2** | 36.2 | **25.4** | **29.8** |      |      |      |
| Few-shot TPLinker (avg)    | 43.0 | 17.0 | **24.3** | 48.5 | 31.5 | 38.2 | 25.9 | 20.0 | 22.6 | 39.1 | 22.8 | 28.4 |      |      |      |
| Few-shot BiTT             | **60.5** | 13.4 | 21.9 | **62.2** | 27.2 | 37.9 | **54.7** | 10.3 | 17.3 | **59.1** | 17   | 25.7 |      |      |      |

B. Competitive Models

Based on our few-shot RE framework, we construct three competitive models: 1) **Few-shot TPLinker (min)** is the Few-shot TPLinker model mentioned in Section IV-B, which applies Eq.(17) to the calculation of the negative distance matrix \( D_n \); 2) textbfFew-shot TPLinker (avg) is the Few-shot TPLinker model applying Eq.(18) to \( D_n \); 3) **Few-shot BiTT** is the Few-shot BiTT model mentioned in Section IV-C.

In addition, we design three training modes for these models to explore the effect of pre-training and fine-tuning on them. *No pretrain, no finetune* indicates that the models are trained directly from scratch based on the loss (Eq.(7)) of our few-shot RE framework. *Pretrain, no finetune* indicates that the models are only pre-trained in the traditional sequence tagging RE task, while not fine-tuned in the few-shot RE task. *Pretrain, finetune* indicates that the models are both pre-trained in the traditional sequence tagging RE task and fine-tuned in the few-shot RE task.

To the best of our knowledge, our few-shot RE framework is the first metric-based few-shot system for joint RE. So we adopt two metric-based few-shot NER models and two metric-based few-shot RC models for combination. The few-shot NER models are: 1) **NNshot(NER)** [22], an effective method with nearest neighbor learning; 2) **Proto(NER)** [6], a few-shot NER system with the prototypical network. The few-shot RC models are: 1) **BERTPAIR(RC)** [7], a few-shot RC method focused on the none-of-the-above (NOTA) relation; 2) **Proto(RC)** [7], a prototypical network with BERT. As a result, we get four pipelined few-shot RE models as baselines.

C. Training Details

a) **Few-shot Sequence Tagging Framework for RE:**

In our framework, we adopt the default BERT-Based-Cased hyper-parameter values provided by Hugging Face\(^2\). Overall, in our experiments, the number of iterations for direct training and fine-tuning are 500,000 and 100,000 separately, and the number of epochs in pre-training is 20. In the process of direct training and fine-tuning, 5,000 iterations of validation are performed for every 10,000 iterations of training. In the pre-training process, we divide the data in the source domain into training set and validation set in the ratio of 4:1, and evaluate on the validation set after two training epochs. Then we save the pre-trained model with the best result. Besides, the batch size in pre-training process is set to 128, and the learning rate of Adam optimizer is 2e-5.

What’s more, there are a few hyper-parameters that need to be set artificially in Section IV. For our few-shot RE framework, in Eq.(1), the maximum length of the relation category description and the input tokens sequence are 10 (including \([CLS]\) and \([SEP]\)) and 50 (including \([SEP]\)) respectively. And the relation category description \( r \) is the word sequence obtained from the corresponding fine-grained relation category name by the tokenization based on the BERT-Based-Cased vocabulary. In Eq.(2), the hidden size of \( H \) is \( N_h = 32 \). For few-shot TPLinker, in Eq.(10) and Eq.(11), the dimension of \( \mathbf{W}_h \) and \( \mathbf{W}_t \) is 768×32. And \( \mathbf{E} \) is set to 3 in Eq.(14), Eq.(15) and Eq.(16). For few-shot BiTT, in Eq.(19), the dimension of \( \mathbf{W}_c \) is 768×32. In Eq.(20), the hidden size \( \mathbf{P}_c \) is 32 and the weighted variable \( \gamma \) is set to 0.9.

b) **Few-shot Pipelined Models for RE:**

The experiments on the few-shot pipelined RE models are divided into three steps\(^3\). 1) The sentences in the training set are tagged with sequence annotations by IO tagging scheme. NNshot(NER) and Proto(NER) are trained with the tagging data. 2) For a sentence in the training set, we enumerate all combinations of entities and label each entity pair with a predefined relation category or a NOTA relationship. BERTPAIR(RC) and Proto(RC) are trained with these samples with relation labels. 3) In the testing phase, we input the text into the NER model to get the entities present in the text and then combine the entities into entity pairs. We put the text and entity pairs into the RC model to get the relationship categories. For an entity pair whose relationship category is not NOTA, we combine the entities and the relation category into a triple. Then the pipelined models can extract the triples in the text.

The settings of BERT parameters, fine-tuning, and testing of our baseline models are kept the same as above. Note that the text input to BERTPAIR(RC) is a two-by-two combination of sentences in the support and query set. Thus the maximum

\(^2\)https://huggingface.co/

\(^3\)The source code of the experiments on few-shot pipelined models is available at https://anonymous/for/review.
length of the input tokens sequence is set to twice the previous length, i.e., 100.

c) Evaluation Metrics: The optimal models after training are evaluated on the target domain by 20,000 iterations of sampling. We report the standard evaluation metrics for RE: Precision (Prec), Recall (Rec) and F1 score (F1). We consider a predicted triple \((h, r, t)\) as a correct one only if \(h, t\) and \(r\) are all correct.

D. Results

The experimental results on Few-NYT (INTER) and Few-NYT (INTRA) are shown in Table II, III and IV respectively. Note that the metrics of the pipelined baselines are almost all zeros, so they are not shown in Table III.

a) Comparison with Baselines: As shown in Table II, our Few-shot TPLinker (min) achieves the best average \(\text{Rec} (59.1\%)\) on Few-NYT (INTER), which is 54.7\% higher than that of the best \(\text{Rec} \) and \(\text{F1} \) of pipelined baselines. And our Few-shot BiTT achieves the best average \(\text{Prec} (59.1\%)\) on Few-NYT (INTER), which is 54.7\% higher than that of the best pipelined baseline ProtoNER+BERTPAIR(RC). As shown in Table III, on Few-NYT (INTRA), our Few-shot TPLinker (min) achieves the best \(\text{Rec} (6.2\%)\) and \(\text{F1} (9.2\%)\), and Few-shot BiTT achieves the best \(\text{Prec} (39.8\%)\). These results illustrate that our proposed models are far more effective than the few-shot pipelined models.

b) Comparison between Proposed Models: As shown in Table IV, each of the three models based on our proposed Few-shot RE framework has its own benefits. First, Few-shot TPLinker (min) achieves the best average \(\text{F1} (29.8\%)\) and \(\text{rec} (25.4\%)\) scores on Few-NYT (INTER), outperforming the second best model Few-shot TPLinker (avg) by 1.4\% and 2.6\% separately. This indicates that Few-shot TPLinker (min) works best overall and is able to recognize more valid relation triples in the target domain instances. Third, Few-shot BiTT achieves the best average \(\text{Prec} (60.4\%)\) score on Few-NYT (INTER), outperforming the second model Few-shot TPLinker (min) by 19.7\%. This demonstrates that Few-shot BiTT tends to focus on the validity of its own predictions, rather than locating as many triples in the instance as possible. It is due to the fact that Few-shot BiTT only outputs a predicted triple when all tokens of both the head entity and tail entity are consistently labeled. The results in Table III also support the above conclusion on Few-shot TPLinker (min). Few-shot TPLinker (min) achieves the best \(\text{F1} (9.2\%)\) and \(\text{rec} (6.2\%)\) scores on Few-NYT (INTER), outperforming the second model (Few-shot TPLinker (avg)) by 0.1\% and 1.1\%. Few-shot TPLinker (avg) obtains the best \(\text{Prec} (39.8\%)\) score, outperforming the second model (Few-shot BiTT) by 19.6\%.

c) Effect of Pre-training: In our paper, we pretrain our encoders by appending a linear layer, and then perform softmax to achieve classification for each token. This operation can restrict the hidden state of tokens in the same class to a cluster, making the distance between these tokens small. As demonstrated by comparing the results of No pretrain, no finetune and Pretrain, finetune in Table III and IV, pre-training enables the models to learn more knowledge related to RE, which greatly boosts the performance. And the improvement is reliable in different few-shot task scenarios. On Few-NYT (INTER), Few-shot TPLinker (min), Few-shot TPLinker (avg) and Few-shot BiTT boost their average \(\text{F1} \) score respectively by 5.0\%, 14.8\% and 2.7\% after pre-training. On Few-NYT (INTER), the three models boost their \(\text{F1} \) separately by 6.6\%, 5.6\% and 1.8\% after pre-training. In particular, on Few-NYT (INTER), the performance boost of Few-shot TPLinker (avg) is obvious. It indicates that the few-shot training of Few-shot TPLinker (avg) has a strict requirement on the initialization of the parameters, while pre-training optimizes the parameters by utilizing the knowledge of the sequence tagging RE task.

d) Effect of Fine-tuning: In our framework, the role of the loss in fine-tuning stage is to further reduce the distance of intra-cluster tokens and increase the inter-cluster distance, thus increasing the probability of correctly classifying tokens on cluster boundaries. Fine-tuning on the few-shot task based on pre-training has been proven to be an appropriate method to improve the generalization of our models. Comparing the results of Pretrain, no finetune and Pretrain, finetune in Table IV, Few-shot TPLinker (min), Few-shot TPLinker (avg) and Few-shot BiTT boost their average \(\text{F1} \) respectively by 6.4\%, 4.6\% and 4.6\% after fine-tuning. However, experiments show that fine-tuning may lead to overfitting in the source domain when the common knowledge between source and target domains is insufficient. As shown in Table III, on Few-NYT (INTER), the \(\text{Prec}, \text{Rec} \) and \(\text{F1} \) of Few-shot BiTT separately decrease by 11.7\%, 0.1\% and 0.3\% after fine-tuning.

e) Few-NYT (INTER) vs. Few-NYT (INTRA): Since there is no overlap of coarse-grained relation types between the source and target domains in the Few-NYT (INTER) task, our models are able to obtain less common information between the two domains, making it more difficult to extract triples. From Table III and Table IV, all models show a substantial decrease in performance on the Few-NYT (INTER) task. Among them, Few-BiTT decreases the most, as evidenced by a significant drop in \(\text{Prec} \) and a tiny \(\text{Rec} \). From this situation, it can be seen that the Few-shot BiTT model mainly learns...
In this paper, we put forward the definition of the few-shot RE task based on the sequence tagging approaches, and propose a few-shot RE framework for the task. Based on the few-shot RE framework, we construct three models called Few-TPLinker(min), Few-TPLinker(avg) and Few-BiTt, and achieve solid results on Few-NYT (INTER) and Few-NYT (INTRA) tasks. Our future work aims to improve our framework in two aspects, i.e. combining more category description information and exploring other ways to model cross-domain knowledge.

VI. CONCLUSION

In this paper, we put forward the definition of the few-shot RE task based on the sequence tagging approaches, and propose a few-shot RE framework for the task. Based on the few-shot RE framework, we construct three models called Few-TPLinker(min), Few-TPLinker(avg) and Few-BiTt, and achieve solid results on Few-NYT (INTER) and Few-NYT (INTRA) tasks. Our future work aims to improve our framework in two aspects, i.e. combining more category description information and exploring other ways to model cross-domain knowledge.

ACKNOWLEDGMENT

We sincerely thank the reviewers for their insightful comments and valuable suggestions. This research work is supported by 2022 Interdisciplinary Clinical Research Project of Peking University First Hospital under Grant no. 2022CR50.

REFERENCES

[1] L. Baldini Soares, N. FritzGerald, J. Ling and T. Kwiatkowski, “Matching the blanks: distributional similarity for relation learning,” in Proceedings of ACL, 2019, pp. 2895-2905.
[2] S. Brody, S. Wu and A. Benton, “Towards realistic few-shot relation extraction,” in Proceedings of EMNLP, 2021, pp. 5338-5345.
[3] 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 AAAI, 2019, pp. 6300-6308.
[4] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, “BERT: pretraining of deep bidirectional transformers for language understanding,” in Proceedings of NAACL, 2019, pp. 4171-4186.
[5] N. Ding, G. Xu, Y. Chen, X. Wang, X. Han and P. Xie, et al., “Few-NERD: a few-shot named entity recognition dataset,” in Proceedings of ACL-IJCNLP, 2021, pp. 3198-3213.
[6] A. Fritzler, V. Logacheva and M. Kretov, “Few-shot classification in named entity recognition task,” in Proceedings of ACM/SIGAPP, 2019, pp. 993-1000.
[7] T. Gao, X. Han, M. Zhu, Z. Liu, P. Li and M. Sun, et al., “FewRel 2.0: towards more challenging few-shot relation classification,” in Proceedings of EMNLP-IJCNLP, 2019, pp. 6250-6255.
[8] X. Han, H. Zhu, P. Yu, Z. Wang, Y. Yao and Z. Liu, et al., “FewRel: a large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation,” in Proceedings of EMNLP, 2018, pp. 4803-4809.
[9] Y. Hou, W. Che, Y. Lai, Z. Zhou, Y. Liu and H. Liu, et al., “Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network,” in Proceedings of ACL, 2020, pp. 1381-1393.
[10] A. Katiyar and C. Cardie, “Investigating LSTMs for joint extraction of opinion entities and relations,” in Proceedings of ACL, 2016, pp. 919-929.
[11] X. Luo, W. Liu, M. Ma and P. Wang, “A bidirectional tree tagging scheme for joint medical relation extraction,” CoRR, vol. abs/2008.13139, 2020.
[12] M. Mintz, S. Bills, R. Snow and D. Jurafsky, “Distant supervision for relation extraction without labeled data,” in Proceedings of ACL and AFNLP, 2009, pp. 1003-1011.
[13] M. Miwa and M. Bansal, “End-to-end relation extraction using LSTMs on sequences and tree structures,” in Proceedings of ACL, 2016, pp. 1105-1116.
[14] Munkhdalai, Tsenduresen and Y. Hong, “Meta networks,” in Proceedings of ICML, 2017.
[15] T. Nayak and H. Tou, “Effective modeling of encoder-decoder architecture for joint entity and relation extraction,” in Proceedings of AAAI, 2020, pp. 8528-8535.
[16] J. Snell, K. Swersky and R. Zemel, “Prototypical networks for few-shot learning,” in Proceedings of NeurIPS, 2017.
[17] M. Tong, S. Wang, B. Xu, Y. Cao, M. Liu and L. Hou, et al., “Learning from miscellaneous other-class words for few-shot named entity recognition,” in Proceedings of ACL-IJCNLP, 2021, pp. 6236-6247.
[18] N. T. Yu, H. Adel, P. Gupta and H. Schütze, “Combining recurrent and convolutional neural networks for relation classification,” in Proceedings of ACL, 2016, pp. 534-539.
[19] Y. Wang, B. Yu, Y. Zhang, T. Liu, H. Zhu and L. Sun, “TPLinker: single-stage joint extraction of entities and relations through token pair linking,” in Proceedings of COLING, 2020, pp. 1572-1582.
[20] Z. Wei, J. Su, Y. Wang, Y. Tian and Y. Chang, “A novel cascade binary tagging framework for relational triple extraction,” in Proceedings of ACL, 2020, pp. 1476-1488.
[21] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng and Z. Jin, “Classifying relations via long short term memory networks along shortest dependency paths,” in Proceedings of EMNLP, 2015, pp. 1785-1794.
[22] Y. Yang and A. Katiyar, “Simple and effective few-shot named entity recognition with structured nearest neighbor learning,” in Proceedings of EMNLP, 2020, pp. 6365-6375.
[23] D. Zeng, K. Liu, S. Lai, G. Zhou and J. Zhao, “Relation classification via convolutional deep neural network,” in Proceedings of COLING, 2014, pp. 2333-2344.
[24] D. Zeng, H. Zhang and Q. Liu, “CopyMTL: copy mechanism for joint extraction of entities and relations with multi-task learning,” in Proceedings of AAAI, 2020, pp. 9507-9514.
[25] 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 ACL, 2018, pp. 506-514.
[26] 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 ACL, 2017, pp. 1227-1236.
[27] D. Zhong and D. Chen, “A frustratingly easy approach for entity and relation extraction,” in Proceedings of NAACL, 2021, pp. 50-61.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.