Towards Realistic Low-resource Relation Extraction: A Benchmark with Empirical Baseline Study

Xin Xu1,2∗, Xiang Chen1,2, Ningyu Zhang1,2†, Xin Xie1,2, Xi Chen3, Huajun Chen1,2
1Zhejiang University & AZFT Joint Lab for Knowledge Engine
2Hangzhou Innovation Center, Zhejiang University, 3Tencent
{xxucs@zju.edu.cn, xiang_chen, xx2020, huajunsir, zhangningyu}@zju.edu.cn, jasonxchen@tencent.com
https://zjunlp.github.io/project/LREBench

Abstract

This paper presents an empirical study to build relation extraction systems in low-resource settings. Based upon recent pre-trained language models, we comprehensively investigate three schemes to evaluate the performance in low-resource settings: (i) different types of prompt-based methods with few-shot labeled data; (ii) diverse balancing methods to address the long-tailed distribution issue; (iii) data augmentation technologies and self-training to generate more labeled in-domain data. We create a benchmark with 8 relation extraction (RE) datasets covering different languages, domains and contexts and perform extensive comparisons over the proposed schemes with combinations. Our experiments illustrate: (i) Though prompt-based tuning is beneficial in low-resource RE, there is still much potential for improvement, especially in extracting relations from cross-sentence contexts with multiple relational triples; (ii) Balancing methods are not always helpful for RE with long-tailed distribution; (iii) Data augmentation complements existing baselines and can bring much performance gain, while self-training may not consistently achieve advancement to low-resource RE1.

1 Introduction

Relation Extraction (RE) aims to extract relational facts from the text and plays an essential role in information extraction (Zhang et al., 2022b). The success of neural networks for RE has been witnessed in recent years; however, open issues remain as they still depend on the number of labeled data in practice. For example, Han et al. (2018) found that the model performance drops dramatically as the number of instances for one relation decreases, e.g., for long-tail. An extreme scenario is few-shot RE, where only a few support examples are given. This motivates a Low-resource RE (LRE) task where annotations are scarce (Brody et al., 2021).

Many efforts are devoted to improving the generalization ability beyond learning directly from limited labeled data. Early, Mintz et al. (2009) proposes distant supervision for RE, which leverages facts in KG as weak supervision to obtain annotated instances. Rosenberg et al. (2005); Liu et al. (2021a); Hu et al. (2021) try to assign pseudo labels to unlabeled data and leverage both pseudo-labeled data and gold-labeled data to improve the generalization capability of models iteratively. Some studies apply meta-learning strategies to endow a new model with the ability to optimize rapidly or leverage transfer learning to alleviate the data-hungry issue (Gao et al., 2019; Yu et al., 2020b; Li et al., 2020a; Deng et al., 2021). Other studies (Zhang et al., 2019) focus on the long-tailed class distribution, especially in tail classes that only allow learning with a few instances. With the prosperity of the pre-trained language models (PLMs), the pre-train – fine-tune paradigm has become standard for natural language processing (NLP), leading to a tremendous increase in LRE performance. More recently, a new methodology named prompt learning has made waves in the community by demonstrating astounding few-shot capabilities on LRE (Han et al., 2021; Chen et al., 2022).

In this work, we benchmark more realistic scenarios on diverse datasets for low-resource RE, in which models have to handle both extreme few-shot...
shot instances and long-tailed distribution, and can also make use of data augmentation or unlabeled in-domain data without cross-validation (Perez et al., 2021). These settings are appealing as: (i) Such models mirror deployment in applied settings; (ii) Few-shot settings are realistic with long-tailed distribution; (iii) Diverse datasets cover different languages (Chinese and English), domains (general, scientific), and contexts (one or more sentences with single or multiple relational triples).

Specifically, we focus on improving the generalization ability from three directions shown in Figure 1. Instead of using limited few-shot data, we create different types of prompts for RE and empirically analyze low-resource performance. We further implement many popular balancing methods for long-tailed distribution, which can mitigate performance decay in instance-scarce (tail) classes. We also leverage more generated training instances by data augmentation and self-training in conjunction with the limited labeled data.

Our contributions include: (i) We present the first systematic study for low-resource RE, an important problem in information extraction, by investigating three distinctive schemes with combinations. (ii) We conduct extensive comparisons with in-depth analysis on 8 RE datasets and report empirical results with insightful findings. (iii) We release both the data and the source code of these baselines as an open-sourced testbed for future research purposes.

To shed light on future research on low-resource RE, our empirical analysis suggests that: (i) Previous state-of-the-art methods in the low-resource setting still struggle to obtain better performance than that in the fully-supervised setting (Cross-sentence LRE is extremely challenging), which indicates that there is still much room for low-resource RE. (ii) Balancing methods may not always benefit low-resource RE. The long-tailed issue cannot be ignored, and more studies should be focused on model development. (iii) With some simple data augmentation methods, better performance can be achieved, highlighting opportunities for future improvements on low-resource RE.

2 Background on Low-resource RE

2.1 Low-resource RE

RE is a classification task that aims to assign relation labels to entity pairs in given contexts. Formally, in a RE dataset denoted as \( D = \{X, Y\} \), \( X \) is the set of texts and \( Y \) is the set of relation labels. Given a text \( x = \{w_1, w_2, \ldots, w_k, \ldots, w_{|x|}\} \), where \( x \in X \), RE aims to predict the semantic relation \( y_x \in Y \) holding between the subject entity \( w_k \) and the object entity \( w_l \). Conventional RE systems are trained in the standard supervised learning regime, where large amounts of labeled examples are required. Nevertheless, owing to various languages, domains, and the cost of human annotation, there is commonly a very small number of labeled examples in real-world applications. Thus, traditional supervised learning with few-shot labeled data struggle to achieve satisfactory performance (Schick and Schütze, 2021). Consequently, a challenging task, low-resource RE, has emerged.

2.2 Fine-tuning PLMs for RE

A typical baseline method for RE is to fine-tune a PLM \( M \) as shown in Figure 2(a). Firstly, the tokenizer of \( M \) converts the text \( x \) into the input tokens of \( M \), such as \([CLS] x \text{token}[SEP]\), and then encodes tokens into the corresponding hidden vectors, such as \( h = \{h_{[CLS]}, h_1, h_2, \ldots, h_k, \ldots, h_l, \ldots, h_{[SEP]}\} \). Then, a \([CLS]\) head is used to compute the probability distribution over the class set \( Y \) with the softmax \( p(y|x) = \text{Softmax}(Wh_{[CLS]} + b) \), where \( W \) is a set of learnable weight parameters randomly initialized at the start of fine-tuning, \( h_{[CLS]} \) is the hidden vector of \([CLS]\) and \( b \) is the learnable bias. All learnable parameters are fine-tuned by minimizing the cross-entropy loss over \( p(y|x) \) on \( D \). Nevertheless, conventional supervised fine-tuning may over-fit a few training examples and perform poor generalization ability over test sets when encountering the low-resource RE task.

3 Methods for Low-resource RE

In this paper, we conduct a comprehensive empirical study with three distinctive schemes against difficulty in low-resource RE: PLMs-based prompt-based tuning, balancing long-tailed data and leveraging more instances, as shown in Figure 2.
Figure 2: Illustrations of different methods used in our low-resource RE benchmark. (a) A standard RE pipeline of fine-tuning a PLM such as BERT and RoBERTa (§2.2). (b) Prompt-based tuning, which concatenates the original input with the prompt template to predict $[\text{MASK}]$ by an MLM head and then injects the predicted answer words to the corresponding class sets (§3.1). (c) Two balancing methods, re-sampling data and re-weighting losses, to address the long-tailed issue (§3.2). (d) Levering more instances with data augmentation and self-training (§3.3).

swer words, denoted as $V$, through the masked language model (MLM) head. Specifically, $T_{\text{prompt}}$ converts every instance $x$ into a prompt input $x_{\text{prompt}} = T_{\text{prompt}}(x)$, in which there is at least one $[\text{MASK}]$ for $M$ to fill with right answer words $v \in V$. Meanwhile, a verbalizer connects relation labels with answer words via an injective mapping $\gamma : Y \rightarrow V$. With the aforementioned functions, we can formalize the probability distribution over $Y$ with the probability distribution over $V$ at the masked position (Ma et al., 2021):

$$P(y_x| x) = P([\text{MASK}] = \gamma(y_x)| x_{\text{prompt}}) = \text{Softmax}(W_{lm} \cdot h_{[\text{MASK}]})$$

(1)

where $W_{lm}$ is a set of parameters of the PLM head.

Note that the main difference between various prompt-based tuning methods lies in the design of the prompt template and verbalizer. Thus, we benchmark different kinds of prompting methods in low-resource RE to empirically investigate their performance. For the prompt template, given the input $x$, the first choice is manually designing the template. We utilize the natural language or task schema to formulate different prompt templates. Formally, we have:

**Template Prompt:**

$[\text{CLS} \ x \ [SEP]]$ The relation between $[\text{sub}]$ and $[\text{obj}]$ is $[\text{MASK}]$. $[SEP]$

**Schema Prompt:**

$[\text{CLS} \ x \ [SEP]]$ $[\text{sub}]$ | $[\text{obj}]$ relation: $[\text{MASK}]$. $[SEP]$

where $<$sub$>$ is the head entity mention and $<$obj$>$ is the tail entity mention. Since there exists rich semantic knowledge within relation labels and structural knowledge implications among relational triples, we also benchmark previous studies such as $\text{PTR}$ (Han et al., 2021) and $\text{KnowPrompt}$ (Chen et al., 2022d) which incorporates relational knowledge into prompt-based tuning as shown in Figure 2(b).
3.2 Balancing for Long-tailed Distribution

Learning with long-tailed data, where the number of instances in each class highly varies, is a common challenge in low-resource RE because instance-rich (head) classes predominate the training procedure. Note that the learnable parameters of the trained model prefer to perform better in these head classes and worse in less frequent (tail) classes (Kang et al., 2020a). To address this issue, we explore two balancing methods: re-sampling data and re-weighting losses for low-resource RE.

Re-sampling Data We re-sample RE datasets to balance the data distribution. For example, the tail classes can be over-sampled by adding copies of data, and the head classes can be under-sampled by removing data, as shown in Figure 2(c). Specifically, we use a toolkit\(^2\), which can estimate the sampling weights automatically when sampling from imbalanced data to obtain datasets with the nearly balanced distribution.

Re-weighting Loss We utilize various re-weighting losses, assigning different weights to different training instances for each class. For instance, DSC Loss (Li et al., 2020b) attaches similar importance to false positives and false negatives. Focal Loss (Lin et al., 2020a) balances the sample-wise classification loss for model training by down-weighing easy samples. GHM Loss (Li et al., 2019a) applies a gradient harmonizing mechanism, making the model ignore outliers to conquer the disharmony in classification. LDAM Loss (Cao et al., 2019) expands the decision boundaries of few-shot classes.

3.3 Leveraging More Instances via Data Augmentation and Self-training

It is also beneficial to leverage more instances to address the low-resource issue. We conduct data augmentation and also leverage unlabeled in-domain data via self-training, as shown in Figure 2(d).

Data augmentation (DA) automatically generates more labeled instances based on only a few labeled instances. For example, we utilize token-level augmentation, which changes or inserts words and phrases in a sentence to generate augmented text remaining with the same labels as the original text. In this work, we apply three DA methods for English RE datasets to substitute words in training sets based on WordNet’s synonyms, TF-IDF similarity and the contextual word embedding implemented by nlpaug\(^3\). And we replace words with their synonyms via nlpcda\(^4\) to augment Chinese RE samples. We further analyze different types of augmentation objects in RE regarding contexts, entities, and both of them.

Since substantial easily-collected unlabeled data are also leveraged in this work for low-resource RE, we conduct self-training, a classical, intuitive and straightforward semi-supervised learning method. Specifically, we train a model with labeled data and then expand the labeled set according to the most confident predictions (a.k.a. pseudo labels) on unlabeled data. We combine the data with gold and pseudo labels to obtain the final RE model. The details of the whole self-training pipeline are described in Appendix A.5.

4 Benchmark Design

In this paper, we provide a comprehensive empirical study for low-resource RE and design the LREBench (Low-resource Relation Extraction Benchmark) to evaluate various methods. In the following section, we will detail the datasets chosen for experiments and the reproducibility of all baselines mentioned above.

4.1 Datasets Selection

As shown in Table 1, we select 8 RE datasets to evaluate baselines in low-resource settings, covering various domains: SemEval 2010 Task 8\(^5\) (Hendrickx et al., 2009), TACREV\(^6\) (Alt et al., 2020), DialogRE\(^7\) (Yu et al., 2020a) and DuIE2.0\(^8\) (Li et al., 2019b) on the general domain, Wiki80\(^9\) (Han et al., 2019) on the encyclopedic domain, ChemProt\(^10\) (Peng et al., 2019) on the biochemical domain, SciERC\(^11\) (Luan et al., 2018) on the scientific domain, and CMeIE\(^12\) (Zhang et al., 2022a) on the medical domain. Except for frequently-used English datasets, we select Chinese datasets.

---

2. https://github.com/makcedward/nlpaug
5. https://github.com/DFKI-NLP/tacrev
7. https://github.com/25776824/nlpcda
8. https://github.com/425776024/nlpcda
11. https://github.com/makcedward/nlpaug

such as DuIE2.0 and CMeIE. Besides, the SciERC, ChemProt, DialogRE, and CMeIE datasets contain the situation where multiple sentences are in one instance, which is for cross-sentence RE and more challenging than single-sentence RE in SemEval, TACREV and Wiki80.

For simplicity, we provide a unified input-output format for all datasets in the low-resource setting. Specifically, each instance in LREBench consists of one text and one relational triple (one head entity and one tail entity in the text and the corresponding relation between them). For those datasets with instances having one text related to multiple relational triples, such as ChemProt, SciERC, DialogRE, DuIE2.0 and CMeIE, we follow Zhong and Chen (2021) to place such a text to multiple instances with only one relational triple. In this way, we can utilize a unified input-output format for widespread models.

We conduct experiments in three settings with different proportions of training data to simulate different resource levels: 8-shot, 10% and 100%. For the 8-shot setting, we sample 8 instances for each relation category in the training and test sets. For the 10% and 100% settings, we sample 10 percent of the training set and use the whole training set, respectively. Since fine-tuning on small datasets can suffer from instability and results may change dramatically given a new split of data (Gao et al., 2021), we sample all training datasets 5 times randomly in 8-shot and 10% settings and measure their average performance in experiments. Also, we follow the same sampling strategy in the re-sampling long-tailed data method and data augmentation methods to obtain a fair comparison.

### 4.2 Reproducibility

**Methods** Throughout our experiments, we employ $M = RoBERTa$-large (Liu et al., 2019) for SemEval, TACREV, Wiki80 and DialogRE, Chinese $RoBERTa$-large (Cui et al., 2020) for DuIE2.0 and CMeIE, and BioBERT-large (Lee et al., 2020) for ChemProt and SciERC from HuggingFace\(^\text{15}\) as the backbone network (detailed in Appendix A.1). For each method, we investigate the following three schemes in different settings for the comparative empirical study, as shown in Table 2: (i) Normal is the general scheme with the PLM for low-resource relation extraction, in which we evaluate with 8-shot, 10% and 100% settings. (ii) Balance refers to balancing methods in §3.2 for long-tailed data distribution with 10% and 100% settings. We list the best performance among all balancing methods for each dataset in Table 2 and detailed results in Table 3. (iii) Data augmentation (DA) methods are applied to 10% training sets. We list the best performance among all DA methods in Table 2 and all performance in Table 4. We also conduct self-training (ST) that firstly trains a teacher $M$ on 10% training data and then tags the rest 90% training data with pseudo labels by $M$. Both gold-labeled and pseudo-labeled data are used to obtain a final student RE model as introduced in §3.3.

**Training and Evaluation** We only train models on training sets without validation on development sets to ensure true few-shot learning with limited labeled data. For all training data sizes, we set the training epoch = 10 following Huang et al. (2021). Except for re-weighting losses for addressing the long-tailed problem, the cross-entropy loss is used in all training processes. Since the performance of head and tail classes varies a lot, we use both Macro F1 and Micro F1 together as the evaluation metrics. Implementation details can be found in Appendix A.

### Table 1: Statistics on the 8 public RE datasets selected for evaluation in LREBench. MS indicates if datasets contain instances with multiple sentences in one text, and MT indicates if one text in these datasets can be related to multiple relational triples. "*" means that we re-sample and convert Wiki80 into long-tailed distribution through an exponential function since its original distribution is exactly balanced. "cn" represents datasets with Chinese.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>SemEval</th>
<th>TACREV</th>
<th>Wiki80*</th>
<th>SciERC</th>
<th>ChemProt</th>
<th>DialogRE</th>
<th>DuIE2.0 (cn)</th>
<th>CMeIE (cn)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Domain</strong></td>
<td>General</td>
<td>General</td>
<td>Encyclopedic</td>
<td>Scientific</td>
<td>Biochemical</td>
<td>Dialogue</td>
<td>General</td>
<td>Medical</td>
</tr>
<tr>
<td># Train</td>
<td>6.5k</td>
<td>68.1k</td>
<td>12.0k</td>
<td>3.2k</td>
<td>19.5k</td>
<td>6.0k</td>
<td>153k</td>
<td>34k</td>
</tr>
<tr>
<td># Test</td>
<td>2.7k</td>
<td>15.5k</td>
<td>5.6k</td>
<td>974</td>
<td>16.9k</td>
<td>1.9k</td>
<td>18k</td>
<td>8.7k</td>
</tr>
<tr>
<td># Relation Class</td>
<td>19</td>
<td>42</td>
<td>80</td>
<td>7</td>
<td>14</td>
<td>37</td>
<td>48</td>
<td>44</td>
</tr>
<tr>
<td>MS / MT</td>
<td>× / ×</td>
<td>× / ×</td>
<td>× / ×</td>
<td>✓ / ✓</td>
<td>✓ / ✓</td>
<td>× / ✓</td>
<td>✓ / ✓</td>
<td>✓ / ✓</td>
</tr>
</tbody>
</table>

\(^{13}\)We utilize a unified json format for evaluation, and it is straightforward to adapt to other datasets.

\(^{14}\)If there are less than 8 instances in one relation class, we delete all instances of this class.

\(^{15}\)https://huggingface.co/
5 Results and Discussions

5.1 Main Results

We leverage the basic PLM fine-tuning code from OpenNRE\(^1\) (Han et al., 2019) and the state-of-the-art prompt-based RE method KnowPrompt (Chen et al., 2022d) to conduct extensive experiments across 8 datasets in various methods and settings. The results of the main experiments are shown in Table 2, which illustrates the following findings:

**Finding 1:** Prompt-based tuning largely outperforms standard fine-tuning for RE, especially more effective in the low-resource scenario. The comparison between the results of standard fine-tuning and prompt-based tuning indicates that prompts can provide task-specific information and bridge the pre-train—fine-tune gap, thus, empowering PLMs in low-resource RE.

**Finding 2:** Though balancing methods obtain more advancement with long-tailed distribution, they may still fail on challenging RE datasets, such as ChemProt, DialogRE and DuIE2.0. By comparing Macro F1 Scores of the Balance columns and Normal columns, blue (bad) results illustrate that balancing methods are affected by complexity of long contexts with multiple sentences and relational triples.

**Finding 3:** Data augmentation achieves much gain on RE and sometimes even better performance than prompt-based tuning, such as on SemEval, according to the difference between two pairs of DA columns and Normal columns in the 10% setting. More data generated through DA methods are complementary with other baselines, boosting the performance.

**Finding 4:** RE systems struggle against difficulty in obtaining correct relations from cross-sentence contexts and among multiple triples. The extremely low F1 scores for 8-shot ChemProt, and DialogRE datasets in standard fine-tuning demonstrate this finding. One text in ChemProt is related to too many relational triples (there are 347 texts related to 3 triples and 699 texts related to 2 triples in the training set). At the same time, in DialogRE, the input text is extremely long (one text can contain 10 sentences). Even with the powerful prompt-based tuning method, it is non-trivial to address the low-resource issue according to the unexpected drop in F1 scores of ChemProt and SciERC.

**Finding 5:** Self-training with unlabeled in-domain data may not always show an advantage for low-resource RE. There is much noise in those

---

1\(https://github.com/thunlp/OpenNRE\)
generated pseudo labels. Furthermore, for assigning labels in RE, both semantics and positions of entities in a text need to be considered, which is exceedingly challenging. Therefore, the model with self-training cannot always obtain better performance in low-resource settings.

5.2 Comprehensive Empirical Analysis

Different Prompting Methods
To investigate the effects of different prompts, we conduct an empirical analysis on SemEval, TACREV, SciERC and ChemProt as shown in Figure 3. We observe the following insights: (i) **Prompt-based tuning is more beneficial in general domains than specific domains for low-resource RE.** Prompt-based tuning achieves the most gain, 44.85% Micro F1 Score, by comparing fine-tuning and KnowPrompt on 8-shot SemEval, while obtaining the worst drop, 25.65% Micro F1 Score, by comparing fine-tuning and the template prompt on 8-shot SciERC even with the domain-specific PLM. Except for the difficulty of these two datasets, general manual prompts have little domain knowledge related to vertical domains, hindering performance. (ii) **Entity type information in prompts is helpful for low-resource RE.** The head and tail entity types in prompts provide strong constraints between relations and their related entities. Prompting methods with entity type information in KnowPrompt and PTR perform better than the template and schema-based prompt in most datasets, which illustrates that prompts with entity-type information are more appropriate for low-resource RE. The reason for the abnormal phenomenon that KnowPrompt and PTR obtain worse results than the template and schema-based prompts in TACREV is that annotation errors in the training set of TACREV (Stoica et al., 2021) can lead to overestimation of the performance of models depending on the side information of entities such as entity names, spans and types (Zhou and Chen, 2021), and the templates of KnowPrompt and PTR are natural language sentences consisting of the head, and tail entities and their relations, which require high-quality annotated entity mentions, positions, types and relational words, while they are relatively trivial to the template and schema-based prompts.

Different Balancing Methods
We also conduct experiments to validate the effectiveness of different balancing methods on two long-tailed datasets. We categorize the classes into three splits based on the number of training instances per class, including Few, Medium, and Many, and also report the results on the whole dataset with the Overall setting in Table 3 (split schemes are in Appendix B). We notice that with re-balancing methods (e.g., Focal Loss and LDAM Loss), the tail relations (Few) can yield better performance on both general and domain-specific datasets. However, some technologies, such as GHM-C, fail to contribute to performance gains. Overall, our empirical analysis illustrates that the RE performance can be improved with balancing methods, which indicates that long-tailed RE is a challenging classification task, and it should be paid more attention to developing suitable methodologies.

Different Data Augmentation
To evaluate the low-resource RE performance with more instances, we generate 30% and 100% augmented instances.
Table 3: F1 Scores (%) on SemEval and SciERC datasets of diverse balancing methods via KnowPrompt. MaF1 and MiF1 mean Macro F1 Score (%) and Micro F1 Score (%) respectively. Normal means conducting the experiment without any balancing methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>SemEval</th>
<th>SciERC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Few</td>
<td>Medium</td>
</tr>
<tr>
<td>Normal</td>
<td>50.42</td>
<td>74.58</td>
</tr>
<tr>
<td>Re-sample</td>
<td>38.17</td>
<td>56.18</td>
</tr>
<tr>
<td>DSC</td>
<td>49.80</td>
<td>73.87</td>
</tr>
<tr>
<td>Focal</td>
<td>53.31</td>
<td>77.69</td>
</tr>
<tr>
<td>GHM-C</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>LDAM</td>
<td>53.53</td>
<td>79.66</td>
</tr>
</tbody>
</table>

Table 4: Micro F1 Scores (%) on four datasets generated by different data augmentation methods from 10% training sets. Three DA methods are conducted to substitute words at three positions: only in contexts, only in entities and in both of them. “-” represents non-repeated data generated based on contextual word embedding is not available.

<table>
<thead>
<tr>
<th>Method</th>
<th>SemEval</th>
<th>TACRED-Revisit</th>
<th>SciERC</th>
<th>ChemProt</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Context</td>
<td>Context</td>
<td>Context</td>
<td>Context</td>
</tr>
<tr>
<td>WordNet’s Synonym</td>
<td>75.49</td>
<td>75.50</td>
<td>76.47</td>
<td>83.54</td>
</tr>
<tr>
<td>TF-IDF Similarity</td>
<td>73.93</td>
<td>76.23</td>
<td>74.30</td>
<td>82.92</td>
</tr>
<tr>
<td>Contextual Word Embedding (RoBERTa)</td>
<td>73.84</td>
<td>74.41</td>
<td>81.63</td>
<td>82.56</td>
</tr>
<tr>
<td>KnowPrompt (RoBERTa)</td>
<td>69.90</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>76.54</strong></td>
<td>76.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>74.30</strong></td>
<td>83.50</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>82.33</td>
<td>83.50</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>75.54</td>
<td>75.63</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>76.12</td>
<td>76.59</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>76.37</td>
<td></td>
</tr>
</tbody>
</table>

6 Related Work

General and Low-resource RE Relation extraction is essential in information extraction. Learning algorithms for RE models involve feature-based methods (Kambhatla, 2004), semi-supervised (Chen et al., 2006; Rosenfeld and Feldman, 2007; Sun et al., 2011), graph-based methods (Zhang et al., 2018; Guo et al., 2019, 2020) and applies PLMs as the backbone (Lin et al., 2021; Zheng et al., 2021; Wu et al., 2022; Chen et al., 2022c,b). Since labeled instances may be limited in practice, low-resource RE has appealed to researchers (Sabo et al., 2021).

Prompting Methods for RE Though fine-tuning PLMs has waved the NLP community, there is still a big gap between pre-training and fine-tuning objectives, hindering the few-shot performance. Hence, prompt-based tuning is proposed in GPT-3 (Brown et al., 2020) and drawn much attention. A series of researches have illustrate the decent performance of prompt-based tuning (Shin et al., 2020; Lester et al., 2021; Li and Liang, 2021), especially in few-shot classification tasks (Schick and Schütze, 2021; Liu et al., 2021b; Chen et al., 2022a). Typically, PTR (Han et al., 2021) encodes prior knowledge using logic rules in prompt-based tuning with several sub-prompts for text classification. KnowPrompt (Chen et al., 2022d) incorporates knowledge among relation labels into prompt tuning for RE with synergistic optimization for better performance.

Methods for Long-tailed Distribution Data Many re-balancing methods are proposed to tackle the long-tailed problem (Kang et al., 2020b; Nan et al., 2021). Data distribution re-balancing meth-
ods re-sample the dataset into a more balanced data distribution (Han et al., 2005; Mahajan et al., 2018). Various re-weighing losses (Cui et al., 2019; Li et al., 2019a, 2020b; Lin et al., 2020a; Cao et al., 2019) assign balanced weights to training samples from each class. For RE, Nan et al. (2021) introduces causal inference to mitigate the spurious correlation issues for information extraction.

Data Augmentation for NLP An effective method for NLP in low-resource domains is data augmentation. Token-level DA approaches include replacing tokens with their synonyms (Kolomiyets et al., 2011; Wang and Yang, 2015), deleting tokens (Iyyer et al., 2015), inserting random tokens (Wei and Zou, 2019; Miao et al., 2020) or replacing meaningless tokens with random tokens (Xie et al., 2020; Niu and Bansal, 2018).

7 Conclusion

We provide an empirical study on low-resource RE. Specifically, we analyze the prompt-based tuning for few-shot RE, balancing methods for long-tailed RE datasets, and use data augmentation or unlabeled in-domain data. We systematically evaluate baselines on 8 benchmark datasets in low-resource settings (e.g., 8-shot, 10%) and provide insightful findings. We hope this study can help inspire future research for low-resource RE with more robust models and promote transitioning the technology to real-world industrial scenarios.

8 Limitations

With the fast development of low-resource RE, we cannot compare and evaluate all previous studies due to the settings and non-available open-sourced code. Our motivation is to develop a universal, GLUE-like, and open platform on low-resource RE for the community. We will continue to maintain the benchmark by adding new datasets.

Acknowledgment

We would like to express gratitude to the anonymous reviewers for their kind comments. This work was supported by the National Natural Science Foundation of China (No.62206246, 91846204 and U19B2027), Zhejiang Provincial Natural Science Foundation of China (No. LGG22F030011), Ningbo Natural Science Foundation (2021J190), and Yongjiang Talent Introduction Programme (2021A-156-G).

References


Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples. In CVPR.


Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin


Ang Sun, Ralph Grishman, and Satoshi Sekine. 2011. Semi-supervised relation extraction with large-scale word clustering. In Proceedings of the 49th Annual...
We utilize PyTorch to conduct experiments with one NVIDIA RTX 3090 GPU. All optimization is performed with the


A Implementation Details

A.1 Settings

We detail the training procedures and hyperparameters for each of the datasets. We utilize PyTorch to conduct experiments with one NVIDIA RTX 3090 GPU. All optimization is performed with the
AdamW optimizer (Loshchilov and Hutter, 2019). The training is always continuous for 10 epochs without validation. All pre-trained language models used in this work are downloaded from HuggingFace. The names of PLMs are "hfl/chinese-roberta-wwm-ext-large" for DuIE2.0 and CMeIE, "dmis-lab/biobert-large-cased-v1.1" for SciERC and ChemProtm, and "roberta-large" for other benchmark datasets.

A.2 Prompting Methods

In the prompt-based tuning experiments with KnowPrompt (PyTorch-Lightning), the early stop in the original code is dropped. The learning rate is set as $4 \times 10^{-5}$ for all datasets. Instead of using the original code for multi-labeled DialogRE with BCEloss, we implement experiments with DialogRE the same as the other seven datasets to unify our benchmark.

A.3 Balancing Methods

For re-sampling methods, we firstly use the sampler on all 10% and 100% imbalanced training sets to get nearly balanced training sets and then use them in all methods the same way as imbalanced datasets. We leverage the official code of various re-weighting losses and provide the alternative parsing argument named "useloss" for developers to choose them.

A.4 Data Augmentation

Different DA methods mentioned in §3.3 are utilized on English and Chinese datasets via nlpaug and nlpcda. After generating augmented data, we merge them with original data in order to delete repeated instances that make no sense. Then both original and augmented data are combined and fed into models to evaluate their performance.

A.5 Self-training

Given unlabeled data $D^U$ and a few labeled data $D^L$, we conduct self-training for semi-supervised learning. The scheme is executed as the following steps (Huang et al., 2021):

1. Train a teacher model $\Theta^T$ with gold-labeled data $D^L$ via cross-entropy.

2. Use the trained teacher model $\Theta^T$ to generate soft labels on unlabeled data $D^U$:

$$\tilde{y}_i = f_{\Theta^T}(\tilde{x}_i), \quad \tilde{x}_i \in D^U$$

3. Train a student model $\Theta^S$ via cross-entropy $\mathcal{L}$ on both gold-labeled data $D^L$ and soft-labeled data $D^SU$. The loss function of $\Theta^S$ is:

$$\mathcal{L}_{STU} = \frac{1}{|D^L|} \sum_{x_i \in D^L} \mathcal{L}(f_{\Theta^S}(x_i), y_i) + \frac{\lambda_U}{|D^U|} \sum_{\tilde{x}_i \in D^U} \mathcal{L}(f_{\Theta^S}(\tilde{x}_i), \tilde{y}_i)$$

where $\lambda_U$ is the weighting hyper-parameter, and we set it 0.2 in this work. It is an alternative to iterate from Step 1 to Step 3 multiple times by initializing $\Theta^T$ in Step 1 with newly learned $\Theta^S$ in Step 3. We only perform self-training once in our experiments for simplicity because the result is not good, and it is not sensitive to continue the next iteration.

<table>
<thead>
<tr>
<th>Relation</th>
<th>Number</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other</td>
<td>1145</td>
<td></td>
</tr>
<tr>
<td>Entity-Destination</td>
<td>686</td>
<td></td>
</tr>
<tr>
<td>Cause-Effect</td>
<td>536</td>
<td></td>
</tr>
<tr>
<td>Member-Collection</td>
<td>498</td>
<td></td>
</tr>
<tr>
<td>Entity-Origin</td>
<td>462</td>
<td></td>
</tr>
<tr>
<td>Message-Topic</td>
<td>399</td>
<td></td>
</tr>
<tr>
<td>Component-Whole</td>
<td>383</td>
<td></td>
</tr>
<tr>
<td>Component-Whole</td>
<td>382</td>
<td></td>
</tr>
<tr>
<td>Instrument-Agency</td>
<td>331</td>
<td></td>
</tr>
<tr>
<td>Product-Producer</td>
<td>321</td>
<td></td>
</tr>
<tr>
<td>Content-Container</td>
<td>304</td>
<td></td>
</tr>
<tr>
<td>Cause-Effect</td>
<td>280</td>
<td></td>
</tr>
<tr>
<td>Product-Producer</td>
<td>263</td>
<td></td>
</tr>
<tr>
<td>Content-Container</td>
<td>135</td>
<td></td>
</tr>
<tr>
<td>Entity-Origin</td>
<td>121</td>
<td></td>
</tr>
<tr>
<td>Message-Topic</td>
<td>117</td>
<td></td>
</tr>
<tr>
<td>Instrument-Agency</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>Member-Collection</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>Entity-Destination</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Relation splits on SemEval.

<table>
<thead>
<tr>
<th>Relation</th>
<th>Number</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>USED-FOR</td>
<td>1690</td>
<td>Many</td>
</tr>
<tr>
<td>CONJUNCTION</td>
<td>400</td>
<td>Medium</td>
</tr>
<tr>
<td>EVALUATE-FOR</td>
<td>313</td>
<td>Many</td>
</tr>
<tr>
<td>HYPONYM-OF</td>
<td>298</td>
<td>Medium</td>
</tr>
<tr>
<td>PART-OF</td>
<td>179</td>
<td>Few</td>
</tr>
<tr>
<td>FEATURE-OF</td>
<td>173</td>
<td>Few</td>
</tr>
<tr>
<td>COMPARE</td>
<td>166</td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Relation splits on SciERC.
B  Class Splits in Balancing Methods

Evaluation

The few-level, medium-level and many-level relation splits based on the number of each relation class on SemEVal and SciERC are shown in Table 5 and Tabel 6 for comparative experiments on different re-weighting losses in §5.2.