Making Pre-trained Language Models Better Continual Few-Shot Relation Extractors

Shengkun Ma, Jiale Han, Yi Liang, Bo Cheng
State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications
{mashengkun, hanjl, liangyi, chengbo}@bupt.edu.cn

Abstract

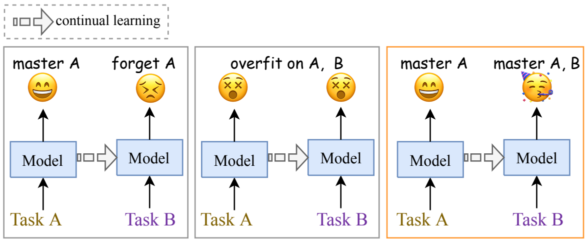
Continual Few-shot Relation Extraction (CFRE) is a practical problem that requires the model to continuously learn novel relations while avoiding forgetting old ones with few labeled training data. The primary challenges are catastrophic forgetting and overfitting. This paper harnesses prompt learning to explore the implicit capabilities of pre-trained language models to address the above two challenges, thereby making language models better continual few-shot relation extractors. Specifically, we propose a Contrastive Prompt Learning framework, which designs prompt representation to acquire more generalized knowledge that can be easily adapted to old and new categories, and margin-based contrastive learning to focus more on hard samples, therefore alleviating catastrophic forgetting and overfitting issues. To further remedy overfitting in low-resource scenarios, we introduce an effective memory augmentation strategy that employs well-crafted prompts to guide ChatGPT in generating diverse samples. Extensive experiments demonstrate that our method outperforms state-of-the-art methods by a large margin and significantly mitigates catastrophic forgetting and overfitting in low-resource scenarios.

Keywords: Continual few-shot Learning, Prompt Learning, Contrastive Learning, Data Augmentation

1. Introduction

Relation Extraction (RE) is a fundamental and important task in the field of natural language processing, which aims to extract the underlying relation between entities in a sentence or document. Traditional RE methods (Peng et al., 2020; Chen et al., 2022) train models on a large number of labeled samples and subsequently test them on data with the same label space. However, in real-life scenarios where new relations emerge all the time, these models may experience a substantial performance decline when adapting to novel relations. In addition, these models heavily rely on the massive labeled data, which demands considerable time and effort to collect.

Therefore, Continual Few-Shot Relation Extraction (CFRE) (Qin and Joty, 2022) has been proposed, which aims to continually learn new relations while retaining knowledge of previously learned relations, all within the constraints of limited labeled data. This practical task brings forth two significant challenges: catastrophic forgetting and overfitting, as shown in Figure 1. Catastrophic forgetting refers to the phenomenon where the model abruptly forgets the knowledge gained from previous tasks when learning new tasks. It's worth noting that some latest studies (Luo et al., 2023; Zhai et al., 2023) point out that catastrophic forgetting even exists in large language models, which makes this issue well worth studying. Overfitting occurs when a model learns to perform exceptionally well on the training data but fails to generalize effectively to unseen data due to fitting noise or irrelevant patterns, which tends to be more pronounced in low-resource scenarios with sparse training data.



(a) Catastrophic Forgetting (b) Overfitting (c) Our Method

Figure 1: Overview of catastrophic forgetting and overfitting in CFRE.

There are several methods have been proposed to address CFRE. Qin and Joty (2022) tackle both issues by introducing embedding space regularization and data augmentation from external data sources. Wang et al. (2023) introduce serial contrastive knowledge distillation to preserve the prior knowledge, thus addressing catastrophic forgetting. Chen et al. (2023) design a consistent prototype learning method to mitigate catastrophic forgetting. While these methods have demonstrated remarkable performance, they do not fully explore the rich inherent knowledge embedded within Pre-trained Language Models (PLMs) to effectively combat catastrophic forgetting and overfitting.

Recently, prompt learning is popular for its magic ability to unlock the potential of PLMs (Lester et al., 2021) and has been widely proven to be simple and effective, especially in the few-shot setting (Gao et al., 2021; Gu et al., 2022). In this work,

LREC-COLING 2024, pages 10970–10983
20-25 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

we delve into the implicit capabilities of PLMs by leveraging prompt learning to address catastrophic forgetting and overfitting, thereby making language models better continuous few-shot relation extractors. To the best of our knowledge, this is the first exploration of prompt technologies in the CFRE task. Specifically, we propose a **C**ontrastive **P**rompt **L**earning framework (CPL), comprising a prompt representation module and a margin-based contrastive learning module. Prompt Representation introduces an effective semi-automated template for RE, which converts RE from classification to text infilling based on context. In this way, PLM tends to learn general task knowledge rather than specific relation categories, becoming adept at identifying previous and novel relations, thereby mitigating catastrophic forgetting. Margin-based contrastive learning makes the model focus more on hard samples and thereby takes a more uniform feature distribution, which effectively alleviates overfitting. To further remedy overfitting in low-resource scenarios, we leverage the power of large language models to bolster smaller PLMs and introduce an effective memory augmentation strategy, which employs well-crafted prompts to guide ChatGPT (OpenAI, 2022) in generating diverse samples. During relation prediction phase, Nearest-Class-Mean classifier is adopted instead of softmax classifier, which is more appropriate for incremental-class classification tasks.

Extensive experiments on two popular RE datasets demonstrate our proposed approach significantly outperforms the baselines, for example, improving accuracy by 6.28% compared to the state-of-the-art method in TACRED dataset. We also conduct a series of ablation studies to prove the effectiveness of each module. We release our code[1] to the community for future research. To summarize, our main contributions include:

- We leverage prompt learning to explore the implicit capabilities of PLMs and propose CPL framework, combining it with a novel margin-based contrastive learning objective for CFRL, which alleviates catastrophic forgetting and overfitting issues simultaneously.
- We introduce a memory augmentation strategy by exploiting the power of LLMs to boost smaller PLMs, which employs well-crafted prompts to guide ChatGPT in generating samples and thus better combat overfitting.
- Extensive experiments on two RE benchmarks show that our method outperforms SOTA models, proving the effectiveness of mitigating catastrophic forgetting and overfitting.

2. Related Work

2.1. Continual Learning

Continual Learning (CL) aims to continually learn new knowledge from a sequence of tasks while avoiding forgetting old knowledge. The main challenge in CL is catastrophic forgetting (McCloskey and Cohen, 1989). Existing CL methods are divided into three categories: 1) regularization methods (Li and Hoiem, 2017; Ritter et al., 2018) use extra constraints to restrict the update of parameters, so that the models can remember more old knowledge. 2) dynamic architecture methods (Fernando et al., 2017; Mallya et al., 2018) extend model architecture dynamically to store new knowledge when sequence tasks keep coming. 3) memory-based methods (Rebuffi et al., 2017; Shin et al., 2017) store a few typical samples of current task to the memory, and replay the memory after learning sequence task to review old knowledge. Among these methods, memory-based methods are the most effective in NLP tasks (Wang et al., 2019; Han et al., 2020). However, data for new tasks is not always sufficient and also getting high-quality data tends to be expensive and time-consuming. Qin and Joty (2022) first introduce Continual Relation Extraction (CRE) in the few-shot setting, and Wang et al. (2023); Chen et al. (2023) propose solutions based on memory methods. We also adopt memory-based strategy, but we concentrate more on how to better leverage PLMs to solve CFRE.

2.2. Prompt Learning

Prompt learning emerged with the birth of GPT-3 series (Brown et al., 2020) and has achieved remarkable performance in NLP tasks, especially in few-shot scenarios (Gao et al., 2021; Gu et al., 2022). It reformulates downstream tasks into pre-training tasks by adding prompted tokens and guides PLMs to understand diverse tasks. Previous prompt learning methods can be divided into three categories: 1) hard prompt (Schick and Schütze, 2021) is to add handcrafted prompt tokens to the sentences and convert them into mask language modeling problem. Though effective, it requires sophisticated expert knowledge for different tasks which is cumbersome and time-consuming. 2) soft prompt (Lester et al., 2021) is instead to add continuous trainable vectors contained in the sentences that can be automatically learned by the model. However, models can not always learn the appropriate prompts without any prior expert knowledge, especially in low-resource scenarios. 3) hybrid prompt (Han et al., 2022) combines untunable hard prompts and tuneable soft prompts, allowing the models to easily learn suitable templates with small manual intervention. It is verified to be the most

[1] https://github.com/mashengkun/CPL

effective prompt method according to recent studies (Gu et al., 2022). Zhang et al. (2022) introduce a prompt-based framework to enhance continual learning process in dealing with CRE. Different from them, we focus on the few-shot setting and adopt hybrid prompts to help with PLMs to alleviate catastrophic forgetting and overfitting.

3. Task Formalization

Continual relation extraction aims to continually learn new relations from a series of n tasks $\mathbb{T} = (\mathcal{T}^1, \mathcal{T}^2, \ldots, \mathcal{T}^n)$. where the k-th task $\mathcal{T}^k$ has its own relation set R^k, and each relation $r \in R^k$ owns several instances $\boldsymbol{x}$, $\boldsymbol{x} = [x_1, \ldots, e_h, \ldots, e_t, \ldots, x_{|\boldsymbol{x}||}]$ which represents a natural language sentence with head entity e_h and tail entity e_t. The samples from $\mathcal{T}^k$ are divided into training set D^k_{train}, validation set D^k_{valid}, and test set D^k_{test}. The relations from different tasks are disjoint $\cap^n_{i=1} R^i = \emptyset$. Note that, once finishing task $\mathcal{T}^k$, the samples of $\mathcal{T}^k$ are no longer available for future training. The model after training $\mathcal{T}^k$ will be evaluated on the test set of all seen relations.

To align with the real situation where labeled samples of each relation are often scarce, we focus on CRE in the few-shot setting following the work of Qin and Joty (2022). For the first task $\mathcal{T}^1$, samples are sufficient, and the subsequent tasks $\mathcal{T}^2, \mathcal{T}^3, \ldots, \mathcal{T}^n$ are all the few-shot. Following (Chen et al., 2023), we set N as the relation number of each few-shot task and K as the sample number of each relation, so the continual few-shot task can be called *continual N-way K-shot* task.

Following previous works (Han et al., 2020; Qin and Joty, 2022; Wang et al., 2023), we set a memory $\mathcal{M} = \{\mathcal{M}^1, \mathcal{M}^2, \ldots, \mathcal{M}^n\}$ with size L, which stores L typical samples for each relation from previous tasks. When training on $\mathcal{T}^k$, the memory $\hat{\mathcal{M}}^{k-1} = \cup^{k-1}_{i=1} \mathcal{M}^i$ can be accessed by the model. In the few-shot setting, we just store one sample per relation ($L = 1$) in the memory.

4. Method

In this section, we detail the components of the proposed method and the training procedure.

4.1. Framework Overview

As shown in figure 2, our framework can be divided into three modules: (1) **prompt representation**, (2) **contrastive learning**, and (3) **memory augmentation**. Prompt representation module introduces the prompt template and the encoding method, which makes downstream tasks more similar to pre-training tasks of PLMs, so that the model is better adapted to new tasks, detailed in Section 4.2. Contrastive learning module proposes a margin-based contrastive learning objective and introduces how to train the model during current task and memory replay. After two contrastive training, model tends to learn a more uniform feature distribution, thus reducing overfitting, detailed in Section 4.3. Memory augmentation module selects typical samples from current tasks and then generates extra samples by ChatGPT for data augmentation, it further mitigates overfitting caused by sparse data, see Section 4.4.

4.2. Prompt Representation

Vanilla prompt engineering consists of two parts, namely template and verbalizer. The template reformulates the original input as a cloze-style phrase by adding a set of prompt tokens and one `[MASK]` token, and the verbalizer maps each task label to the corresponding textual token. By predicting the `[MASK]` token as label verbalization, the model can determine the corresponding label of the input example. Different from vanilla prompt engineering, our prompt framework only includes template without verbalizer. By adding a prompt template to input sentences, the downstream tasks are more similar to the pre-training task of PLMs, therefore PLMs can understand well what to do and perform better on downstream tasks.

Prompt template We design a semi-automated continuous template T, which combines entity information and learnable tokens as prompts rather than hard prompt with only handcrafted tokens or soft prompt with only learnable tokens (Lester et al., 2021). Hard prompt likes "$\boldsymbol{x}$. *The relation between* e_h *and* e_t *is* `[MASK]`.", which needs different sophisticated expert knowledge for specific tasks, and soft prompt might be hard to converge with sparse data. Specifically, for an input sentence $\boldsymbol{x}$ with two entities e_h and e_t, the template is as follows:

$$T(\boldsymbol{x}) = \boldsymbol{x}.\,[v_{0:n_0-1}]\,\boldsymbol{e_h}\,[v_{n_0:n_1-1}]\,[\mathtt{MASK}]\\ [v_{n_1:n_2-1}]\,\boldsymbol{e_t}\,[v_{n_2:n_3-1}]\,. \quad (1)$$

where $[v_i]$ refers to the i-th learnable continuous token, and n_i is the length of token phrases. We add a special token $[\mathtt{MASK}]$ in the template as the representation of the relation between $\boldsymbol{e_h}$ and $\boldsymbol{e_t}$.

Encoding Given a encoding model E and a templated sentence $T(\boldsymbol{x})$ as input, we map the $T(\boldsymbol{x})$ into a sequence of continuous vectors:

$$E_{mb}(T(\boldsymbol{x})) = e(\boldsymbol{x}), h_0, \ldots, h_{n_0-1}, e(\boldsymbol{e_h}),\\ h_{n_0}, \ldots, h_{n_1-1}, e([\mathtt{MASK}]), h_{n_1},\\ \ldots, h_{n_2-1}, e(\boldsymbol{e_t}), h_{n_2}, \ldots, h_{n_3-1} \quad (2)$$

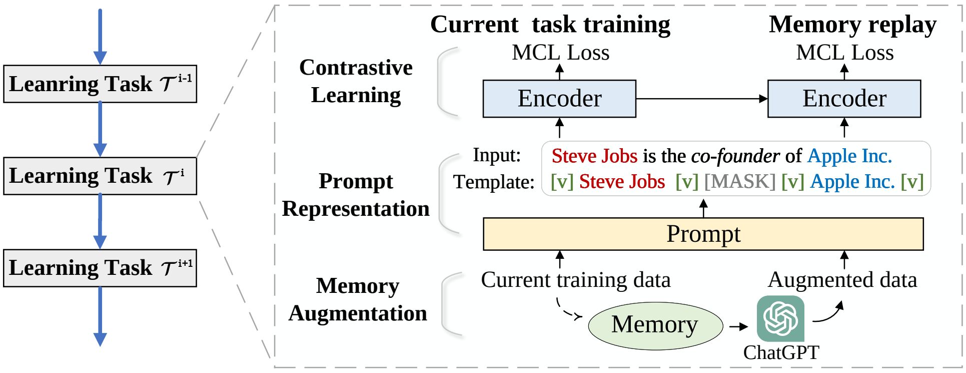


Figure 2: Framework of the proposed CPL.

where $E_{mb}\left(\cdot\right)$ is the embedding function, $e\left(\right)$ is the embedding layer of encoding model E, $h_i \in \mathbb{R}^{d_1}$ are learnable vectors, d_1 is the embedding dimension of E, and $0 \leq i < n_3$. Then the embeddings are fed into the encoding model E and get the hidden representations $\boldsymbol{m}$ of the sentence:

$$\boldsymbol{m} = E_{nc}\left(E_{mb}\left(T\left(\boldsymbol{x}\right)\right)\right) \tag{3}$$

where $E_{nc}\left(\cdot\right)$ represents the hidden layers of E. $\boldsymbol{m} \in \mathbb{R}^{d_2}$ refers the hidden representation of $[\mathrm{MASK}]$, and d_2 is the hidden dimension of E.

4.3. Contrastive Learning

We design a novel margin-based contrastive learning to gain discriminative representations and focus more on hard samples, thus alleviating overfitting. It is worth noting that by doing this, the model does not need the verbalizer module in vanilla prompt engineering, which saves the labor required for prompt engineering and makes the method more general. Instead of linear classifier with extra parameters in previous works (Cui et al., 2021; Qin and Joty, 2022), our method predicts relations based on parameter-free metric method, which is more suitable for incremental-class problems.

Margin-based contrastive learning objective Typically, let $\boldsymbol{z_i}$ be PLMs' output hidden vector after normalization and $s_{i,p} = \boldsymbol{z_i} \cdot \boldsymbol{z_p}$ denotes positive pair, $s_{i,n} = \boldsymbol{z_i} \cdot \boldsymbol{z_n}$ denotes negative pair, the MCL loss can be defined as follows:

$$\mathcal{L}_{\mathrm{MCL}}\left(i\right) = \sum_{p \in P(i)} \log \frac{\exp\left(\alpha_{i,p} \cdot s_{i,p}/\tau\right)}{\mathcal{Z}\left(i\right)}, \tag{4}$$

where $P(i)$ denotes the positive set with i, τ is the temperature constant and $\mathcal{Z}\left(i\right)$ is as follows:

$$\mathcal{Z}(i) = \sum_{p \in P(i)} \exp\left(\alpha_{i,p}\frac{s_{i,p}}{\tau}\right) + \sum_{n \in N(i)} \exp\left(\alpha_{i,n}\frac{s_{i,n}}{\tau}\right) \tag{5}$$

where $N(i)$ is the negative set with i, $\alpha_{i,p}$ and $\alpha_{i,n}$ are relaxation factors that control the relaxation of the decision boundaries.

$$\alpha_{i,p} = m + k \cdot s_{i,p}, \ \alpha_{i,n} = 1 - m + k \cdot s_{i,n} \tag{6}$$

where k is normalization constant depends on $s_{i,p}$ and $s_{i,n}$, m is the margin factor which expects $k \cdot s_{i,p} > 1 - m$ and $k \cdot s_{i,n} < m$.

As we can see, MCL objective can make models pay more attention to hard samples and less attention to easy ones, thus mitigating overfitting problem. It can also make learned distribution more uniform and further alleviate catastrophic forgetting.

Current task training Given a new task $\mathcal{T}^k$, we perform training process with MCL loss on each batch B. According to (Khosla et al., 2020), supervised contrastive loss performs better with big batch size, but it's costly to have a large memory, so we set an additional memory S for contrastive training. Specifically, the sentences of current training set D_{train}^k are encoding as features $\boldsymbol{z}$ and stored in a bucket $\mathcal{C}^k$. Then, for i-th instance $\boldsymbol{x}$ in B, we randomly select partial features from $\mathcal{C}^k$ to form a temporary contrastive features set S_i. Finally, the contrastive loss of i-th instance $\boldsymbol{x}$ can be calculated in S_i. The batch MCL loss is as follows:

$$\mathcal{L}_{\mathrm{MCL}} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left(\alpha_{i,p} \cdot s_{i,p}/\tau\right)}{\mathcal{Z}\left(i\right)}, \tag{7}$$

where $I = \{1, 2, \ldots, |B|\}$, and positive set $P(i) = \{p \in S_i : y_p = y_i\}$, negative set $N(i) = \{n \in S_i : y_n \neq y_i\}$. After backpropagating the gradient of loss on each batch, we update the corresponding features in the bucket $\mathcal{C}^k$:

$$\mathcal{C}^k[\hat{I}] \leftarrow \{\boldsymbol{z}_i\}_{i=1}^{|B|}, \tag{8}$$

where $\hat{I}$ is the corresponding index set of this batch of samples in $\mathcal{C}^k$.

Memory replay The new task training process can make the model concentrate more on new knowledge while forgetting old learned knowledge, especially in low-resource scenarios. To alleviate forgetting, the second training is performed in the memory to consolidate old knowledge. Specifically, we generate additional samples to augment the memory set (refer to section 4.4) and then perform memory training with MCL loss again, which can not only allow the model to review learned knowledge but also prevent the model from overfitting.

4.4. Memory Augmentation

After current task training, We first select typical samples of current training data to store in memory, then design the elaborate prompt to guide ChatGPT to generate diverse samples for memory augmentation.

Representative memory sampling Inspired by previous works (Han et al., 2020), after training for current task $\mathcal{T}^k$, we apply K-means algorithm (Likas et al., 2003) to select typical samples for each relation and store them in the memory $\hat{\mathcal{M}}$. Specifically, we first get features $\boldsymbol{z}_i$ of i-th $\boldsymbol{x}$ on D^k_{train}, and for each relation $r \in R^k$, we use K-means algorithm to cluster features $\{z_i\}_{i=0}^{|R^k|}$ into L clusters. Then, for each cluster, the nearest sample closest to the centroid is chosen as the typical sample which is stored in the memory $\hat{\mathcal{M}}$.

Prompt data augmentation To utilize the rich knowledge of LLMs, we design elaborate prompt to stimulate LLMs' powerful language generation ability to generate relevant examples. We choose GPT-3.5 as our generation model. For every historical relation r in $\hat{R}^k$, we select one typical sample from memory $\hat{\mathcal{M}}^k$ as the instance and construct a prompt input including task instruction, semantic relation explanation (in purple) and demonstrations (in blue) like In-context learning (Dong et al., 2022). Here is an example of prompting ChatGPT to generate samples with the relation *"founded by"*:

> **Prompt:**
> One sample in relation extraction datasets consists of a relation, a context, a head and tail entity. The head entity has the relation with the tail entity.
> *Relation **founded by** means an organization was found by a person.*
> Here is an example:
> Relation: founded by
> Context: Steve Jobs is the co-founder of Apple Inc.
> Head Entity: Steve Jobs
> Tail Entity: Apple Inc.
> Please generate n samples for relation ***founded by***:

Then it is used to ask ChatGPT to generate g diverse samples with relation r. For the output, we parse the text to get structured data as the generated training data $\mathcal{A}$. Finally, memory samples $\hat{\mathcal{M}}^k$ are combined with generated data $\mathcal{A}$ to construct a new training set for subsequent memory replay.

4.5. Training Procedure

First, the parameters of prompt template θ^1 and encoding model θ^2 are initialized with the parameters trained on the last task. Then, there are two main training steps: (1) **Current task training**: current task training set D^k_{train} is encoded with prompt template θ^1, and then fed into the encoding model θ^2 for training by MCL loss $\mathcal{L}_{\mathrm{MCL}}$ (2) **Memory replay**: after training current task, we select typical samples for each relation and store them in the memory $\hat{\mathcal{M}}$. Then, ChatGPT is applied to generate relevant samples with the same relation in memory $\hat{\mathcal{M}}$ to augment training set for subsequent memory replay. Finally, historical memory samples are trained together with generated samples via the MCL loss $\mathcal{L}_{\mathrm{MCL}}$, allowing the model to recall forgotten knowledge without overfitting due to sparse data.

4.6. Relation Prediction

To leverage the discriminative feature distribution after contrastive training, Nearest-Class-Mean (NCM) classifier is adopted to predict the relations in the test phase. Given a historical memory $\hat{\mathcal{M}}^n$, we perform forward propagation with trained encoding model E to get the features of all relations and average features for each relation to calculate prototypes of relations. Then we get the feature of test sample $\boldsymbol{x}$ to compute L2 pairwise distance with all prototypes of seen relations and get the nearest prototype p_r with relation label:

$$\begin{aligned} p_r &= \frac{1}{L}\sum_{i=0}^{L} \boldsymbol{E}\left(\hat{\boldsymbol{x}}_i^r\right), \\ y^* &= \underset{r=1,\ldots,n}{\operatorname{argmin}} \|\boldsymbol{E}\left(\boldsymbol{x}\right) - p_r\|_2 \end{aligned} \tag{9}$$

where L is the memory size per relation, $\hat{\boldsymbol{x}}_i^r$ denotes the samples with label y_r in memory $\hat{\mathcal{M}}^n$. y^* is the predicted relation label.

5. Experiments

In this section, we conduct experiments on two RE datasets and give a further analysis of the results.

5.1. Experiments Setup

Datasets We conduct experiments on two RE datasets: FewRel (Han et al., 2018) and TACRED

FewRel (*10-way 5-shot*)

Method	$\mathcal{T}^1$	$\mathcal{T}^2$	$\mathcal{T}^3$	$\mathcal{T}^4$	$\mathcal{T}^5$	$\mathcal{T}^6$	$\mathcal{T}^7$	$\mathcal{T}^8$
Finetune	94.58	41.67	26.42	20.79	16.63	13.09	11.27	9.78
Joint-train	**95.07**	**88.16**	**83.08**	**79.67**	**77.85**	**75.11**	**72.62**	**70.33**
EMAR+ACA‡ (Wang et al., 2022)	94.75	78.33	69.01	67.17	65.55	61.77	60.04	58.48
InfoCL‡ (Song et al., 2023)	95.38	78.92	72.63	69.05	66.75	63.36	60.65	58.90
RP-CRE† (Cui et al., 2021)	93.97	76.05	71.36	69.32	64.95	61.99	60.59	59.57
CRL† (Zhao et al., 2022)	94.68	80.73	73.82	70.26	66.62	63.28	60.96	59.27
CRECL† (Hu et al., 2022)	93.93	82.55	74.13	69.33	66.51	64.60	62.97	59.99
ERDA† (Qin and Joty, 2022)	92.43	64.52	50.31	44.92	39.75	36.36	34.34	31.96
SCKD‡ (Wang et al., 2023)	94.75	82.56	75.98	72.04	70.53	67.02	64.73	62.87
CPL (ours)	94.87	85.14	78.80	75.10	72.57	69.57	66.85	64.50

TACRED (*5-way 5-shot*)

Method	$\mathcal{T}^1$	$\mathcal{T}^2$	$\mathcal{T}^3$	$\mathcal{T}^4$	$\mathcal{T}^5$	$\mathcal{T}^6$	$\mathcal{T}^7$	$\mathcal{T}^8$
Finetune	88.12	23.26	17.23	16.30	14.19	9.64	9.47	7.90
Joint-train	**88.40**	**83.65**	**74.40**	**71.36**	**65.61**	**64.01**	**61.13**	**58.26**
EMAR+ACA‡ (Wang et al., 2022)	88.06	71.55	63.57	51.84	46.03	46.31	42.55	39.78
InfoCL‡ (Song et al., 2023)	87.45	75.35	65.94	59.10	52.24	49.42	44.52	40.37
RP-CRE† (Cui et al., 2021)	87.32	74.90	67.88	60.02	53.26	50.72	46.21	44.48
CRL† (Zhao et al., 2022)	88.32	76.30	69.76	61.93	54.68	50.92	47.00	44.27
CRECL† (Hu et al., 2022)	87.09	78.09	61.93	55.60	53.42	51.91	47.55	45.53
ERDA† (Qin and Joty, 2022)	81.88	53.68	40.36	36.17	30.14	22.61	22.29	19.42
SCKD‡ (Wang et al., 2023)	88.16	78.89	71.68	64.99	61.13	57.92	53.18	51.11
CPL (ours)	86.27	81.55	73.52	68.96	63.96	62.66	59.96	57.39

Table 1: Main results on FewRel and TACRED in 5-shot setting. † are reported in (Wang et al., 2023), ‡ are we re-running the origin code. The **best results** are in bold, and the second-highest are underlined.

(Zhang et al., 2017). For **FewRel**, we follow Wang et al. (2023), and adopt the version of 80 publicly released relations, and split them equally into 8 tasks. For the first task $\mathcal{T}^1$, we select 100 instances per relation, while the subsequent tasks $\{\mathcal{T}^i\}_{i=2}^8$ are few-shot tasks with only 5 and 10 instances per relation for training which called *10-way 5-shot* and *10-way 10-shot*, respectively. For **TACRED**, we filter out *"no_relation"* and divide the remaining 41 relations into 8 tasks. The first task $\mathcal{T}^1$ has 6 relations and 100 instances per relation, and subsequent tasks $\{\mathcal{T}^i\}_{i=2}^8$ have 5 relations with 5 or 10 instances per relation.

Evaluation metrics We evaluate the model's performance by the average accuracy of all tasks. After training on task $\mathcal{T}^k$, the model is evaluated on $\hat{D}_{test}^k = \cup_{i=0}^k D_{test}^i$ to get the overall accuracy of all seen relations. We measure the average performance across six rounds of experiments.

Baselines We compare with some strong CRE models published in recent years: **RP-CRE** (Cui et al., 2021) introduces a novel attention-based memory module to refine subsequent sample embeddings. **CRL** (Zhao et al., 2022) and **CRECL** (Hu et al., 2022) adopt contrastive learning to ensure the feature space is more distinguishable. **EMAR+ACA** (Wang et al., 2022) propose an adversarial class augmentation mechanism using **EMAR** (Han et al., 2020) as the backbone. **InfoCL** (Song et al., 2023) focus on distinguishing analogous classes. **ERDA** (Qin and Joty, 2022) and **SCKD** (Wang et al., 2023) mainly focus on CRE in few-shot, they use additional memory to alleviate forgetting and data augmentation to mitigate overfitting.

We set two borderlines: **Finetune** trains models sequentially without memory set. It faces serious catastrophic forgetting and serves as the lower bound. **Joint-training** stores all samples of previous tasks in memory, and trains models with all data for every new task, which can be regarded as the upper bound.

Implementation details We implement our framework based on PyTorch 1.7.0 (Paszke et al., 2019) and Huggingface's Transformers 4.10.0 (Wolf et al., 2020). For a fair comparison, BERT-base-uncased (Devlin et al., 2019) is adopted as our encoding model. We choose GPT-3.5-turbo (OpenAI, 2023a) for sample generation. We set the random seeds identical to Wang et al. (2023); Qin and Joty (2022) so that the task order is exactly the same. Following Wang et al. (2023), we employ the *"strict"* evaluation rule to put all seen relation labels as candidates when predicting, rather than *"loose"* rule which only picks 10 labels as candidates (Qin and Joty, 2022). For memory size, we set $L = 1$, i.e. there is only one sample per relation. See Appendix A.3 for more details.

FewRel	$\mathcal{T}^1$	$\mathcal{T}^2$	$\mathcal{T}^3$	$\mathcal{T}^4$	$\mathcal{T}^5$	$\mathcal{T}^6$	$\mathcal{T}^7$	$\mathcal{T}^8$
CPL	94.87	85.14	78.80	75.10	72.57	69.57	66.85	64.50
w.o. pro.	93.78	78.86	69.65	63.97	60.89	56.73	53.83	51.09
w.o. MCL	94.97	84.05	77.64	73.32	70.21	66.76	63.71	61.78
w.o. gen	94.73	82.62	76.16	73.38	70.51	66.98	65.30	63.78
w.o. all	93.20	76.46	66.14	61.49	57.97	53.72	50.09	48.29
TACRED	$\mathcal{T}^1$	$\mathcal{T}^2$	$\mathcal{T}^3$	$\mathcal{T}^4$	$\mathcal{T}^5$	$\mathcal{T}^6$	$\mathcal{T}^7$	$\mathcal{T}^8$
CPL	86.27	81.55	73.52	68.96	63.96	62.66	59.96	57.39
w.o. pro.	86.30	75.14	64.27	59.44	54.11	48.87	44.74	42.61
w.o. MCL	86.02	81.37	75.46	71.46	64.60	63.26	57.53	54.75
w.o. gen	86.14	79.61	70.01	65.24	60.16	57.98	52.64	50.63
w.o. all	86.77	76.10	61.05	55.01	50.83	44.71	39.76	38.06

Table 2: Ablation study on components of CPL.

	Template
1	$x_0 \ldots [E_0]\ \boldsymbol{e_h}\ [E_1] \ldots [E_2]\ \boldsymbol{e_t}\ [E_3] \ldots x_n$.
2	$\boldsymbol{x}$. *The relation between* $\boldsymbol{e_h}$ *and* $\boldsymbol{e_t}$ *is* [MASK].
3	$\boldsymbol{x}$. $[v_0]$ $[v_1]$ [MASK] $[v_2]$ $[v_3]$.
4	$\boldsymbol{x}$. $[v_0]$ $\boldsymbol{e_h}$ $[v_1]$ [MASK] $[v_2]$ $\boldsymbol{e_t}$ $[v_3]$.

Table 3: Templates of different prompt encoding.

5.2. Results and Analysis

5.2.1. Main Results

Table 1 presents the overall results of the 5-shot setting on FewRel TACRED (See Appendix B.2 for 10-shot results). We have the following findings:

(1) Our CPL outperforms all CRE models and achieves SOTA performance on both RE datasets. Specifically, we outperform the second-best model SCKD by 1.63% and 6.28% on FewRel and TACRED, respectively. Notably, we achieve significantly the best performance on the final task of TACRED, even though the performance on the first task is not very good, which proves that the proposed CPL better handles the catastrophic forgetting and overfitting problems. (2) Observing the performance of two borderlines, we find finetune model leads to rapid drops in average accuracy, which shows catastrophic forgetting and overfitting in CFRE tasks. Particularly, we achieve close performance to the joint-training model on TACRED with only one memory sample per relation. However, there is still a five-point gap behind the joint train on FewRel due to the bias caused by very few memory samples. (3)Compared to data augmentation methods ERDA and SCKD, CPL achieves significant improvements, which demonstrates that our method can generate more diverse samples that are closer to the true distribution. Note that ERDA gains the worst performance in our setting, the reason is that the extra data they import may contain noise and bias, and loose evaluation metrics they used are also ill-considered. (4) Compared to contrastive learning methods CRL and CRECL, CPL outperforms them by a large margin on both datasets, which demonstrates that our proposed margin-based contrastive learning can better alleviate overfitting and is more suitable in low-resource scenarios.

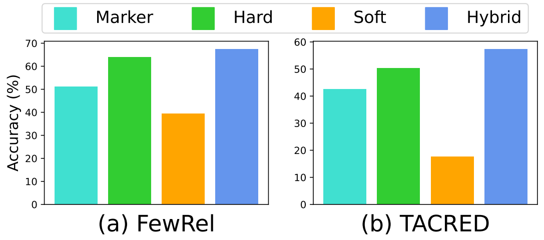
5.2.2. Ablation Study

We conduct ablation experiments on the FewRel 10-way 5-shot and TACRED 5-way 5-shot to verify the effectiveness of each component in our method. (a) w.o. pro. denotes without prompt template. Instead, we use entity markers following previous works (Cui et al., 2021; Zhao et al., 2022; Qin and Joty, 2022; Hu et al., 2022; Wang et al., 2023). (b) w.o. MCL means using supervised contrastive loss (Zhao et al., 2022) instead of our margin-based contrastive loss. (c) w.o. gen represents without generated samples for memory replay. (d) w.o. all means without all of the above components.

As shown in table 2, prompt representation brings a boost of 13.41% and 14.78% after training all tasks, demonstrating the powerful effectiveness of prompt learning for alleviating catastrophic forgetting. MCL loss can also contribute to 2.27% and 2.26% improvement in the 5-shot setting, which shows our margin-based contrastive learning helps models mitigate overfitting caused by sparse data. Additionally, we observe a steady improvement with data augmentation, which represents ChatGPT effectively generates various samples to help the model learn more realistic distributions.

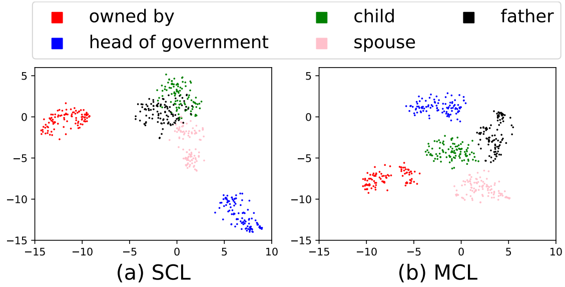
5.2.3. Analysis of Prompt Representation

To verify the effectiveness of proposed prompt representation method, we conduct comparison experiments with four encoding templates in table 3: entity marker, hard prompt, soft prompt, and hybrid prompt template. Figure 3 shows the average accuracy after final task training which represents the degree of forgetting. It can be observed that hybrid prompt outperforms entity marker extensively used in previous works by a big gap, demonstrating the effectiveness of proposed prompt template. The reason is that we combine entity knowledge for RE and make downstream tasks approximate the tasks during pre-training phase of PLMs, hence, the current task training can activate underlying knowledge of PLMs and further alleviate forgetting. Notably, hard prompt is elaborately handcrafted for RE tasks, which contains domain expert knowledge, but even so, our hybrid prompt still outperforms it distinctly, which shows that our hybrid prompt can learn the most suitable template for current tasks with only several samples. The reason for the poor performance of soft prompt is that randomly initialized vectors are difficult to converge with only a few samples, as reported by Gu et al. (2022).



(a) FewRel (b) TACRED

Figure 3: Results of different prompt formats.



(a) SCL (b) MCL

Figure 4: t-SNE plots of instance embeddings trained with SCL and MCL.

5.2.4. Visualization of Contrastive Learning

In order to study the effects of proposed margin-based contrastive objective in low resource scenarios, we use t-SNE (Van der Maaten and Hinton, 2008) to visualize the feature space of test data after current task training. Figure 4 illustrates the results of supervised contrastive loss (SCL) and our MCL loss. As we can see, the feature space of MCL is more uniform than SCL, and samples with different relations are discriminative. As for SCL, though samples with easy-distinguish relations like *"owned by"* and *"head of government"* have obvious boundaries, some samples with similar relations as *"child"* and *"father"* are difficult to distinguish. This is because SCL focuses more on easy pairs and less on hard ones, making it difficult to distinguish similar relations, especially in the few-shot setting.

5.2.5. Analysis of Memory Augmentation through LLMs

To analyze the effect of memory augmentation through LLMs on results, we conduct experiments with different numbers of generated samples in the 5-shot setting, where one sample per relation is stored in memory ($L = 1$). As shown in Figure 5, data generation with LLMs can yield better performance in the few-shot setting. However using more generated data can only boost RE performance to a certain extent, not continuously better, and even cause a performance degradation with on FewRel. This is because the samples generated by LLMs may be noisy and not diverse enough. We further conduct a detailed case study to show the real inputs and corresponding outputs of ChatGPT, as presented in table 4. We combine relation name and description (in purple italics) as prompt input, output 1 *with example* in input while output 2 *without example*. We observe that LLMs can generate the correct sample with the relation *"child"* when with the example, but generate the sample with the wrong relation *"siblings"* between entities when without example. It shows some relations can not be understood well by LLMs only with relation names and descriptions, examples are sometimes necessary for LLMs. More results refer to Appendix C.

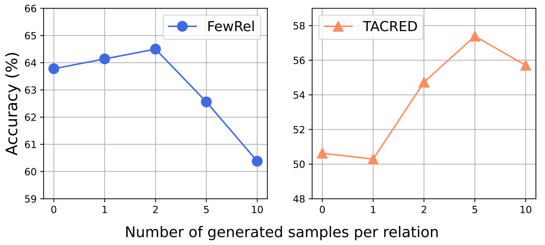


Figure 5: Results with different generated samples.

Prompt: One sample in relation extraction datasets consists of a relation, a context, a head and tail entity. The head entity has the relation with the tail entity. *Relation **child** means subject has object as child*.
Here is an example:
Relation: child
Context: **Pierre-Suzanne-Augustin Cochin** was the father of the Catholic politician **Denys Cochin** and the grandfather of the historian Augustin Cochin.
Head Entity: Pierre-Suzanne-Augustin Cochin
Tail Entity: Denys Cochin
Please generate n samples for relation **child**:

Output 1: (with example in input)
Relation: child
Context: **John Smith** is the father of **Emily Smith** and the husband of Sarah Smith. Emily is a talented musician, while Sarah is a painter.
Head Entity: John Smith
Tail Entity: Emily Smith

Output 2: (without example in input)
Relation: child
Context: **John** and **Mary** are **siblings**. John is the eldest child in the family.
Head Entity: John
Tail Entity: Mary

Table 4: Case of the inputs and outputs of memory augmentation.

6. Conclusion

In this paper, we propose CPL, which leverages prompt learning and contrastive learning to empower pre-trained language models to better continual few-shot relation extractors. To alleviate catastrophic forgetting, we explore using prompt learning to activate generalized knowledge of PLMs that can easily adapt to old and new relations. To mitigate overfitting, we design a novel margin-based contrastive learning objective and apply ChatGPT to generate diverse samples for memory augmentation. Extensive experiments indicate that CPL enables PLMs to better mitigate catastrophic forgetting and overfitting in low-resource scenarios. In the future, we are going to study how to design better prompts to guide LLMs to generate more reliable samples.

7. Limitations

We argue that the main limitations of our work are mainly twofold: (1) Time efficiency. As the model continues to learn new tasks, more different relation samples are held in memory, and more augmented data are generated by the memory samples. Therefore, with the continuous accumulation, the sample of subsequent tasks will be more and more, and the training time will be more and more. In the future, we will explore more time-efficient data augmentation strategies. (2) Stability of results. Due to the unstable content generated by ChatGPT, augmented samples are inconsistent in some experiments with the same input, even when we had set the temperature in OpenAI to 0 for consistency. Therefore, under the same setting, experimental results are not completely consistent each time. We tried our best to ensure the validity of the experimental results by taking the average value of six rounds of experiments.

8. Acknowledgments

We would like to thank the anonymous reviewers, our meta-reviewer, and senior area chairs for their thoughtful comments and support on this work. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFF0902701, in part by the National Natural Science Foundation of China under Grants U21A20468, 62372058, U22A201339, in part by the Fundamental Research Funds for the Central Universities under Grant 2020XD-A07-1.

9. Bibliographical References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In Proceedings of the ACM Web conference 2022, pages 2778–2788.

Xiudi Chen, Hui Wu, and Xiaodong Shi. 2023. Consistent prototype learning for few-shot continual relation extraction. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 7409–7422.

Li Cui, Deqing Yang, Jiaxin Yu, Chengwei Hu, Jiayang Cheng, Jingjie Yi, and Yanghua Xiao. 2021. Refining sample embeddings with relation prototypes to enhance continual relation extraction. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 232–243.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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), pages 4171–4186.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. arXiv preprint arXiv:2301.00234.

Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. 2017. Pathnet: Evolution channels gradient descent in super neural networks. arXiv preprint arXiv:1701.08734.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural

Language Processing, pages 3816–3830. Association for Computational Linguistics.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2022. Ppt: Pre-trained prompt tuning for few-shot learning. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 8410–8423.

Xu Han, Yi Dai, Tianyu Gao, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2020. Continual relation learning via episodic memory activation and reconsolidation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6429–6440.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2022. Ptr: Prompt tuning with rules for text classification. AI Open, 3:182–192.

Chengwei Hu, Deqing Yang, Haoliang Jin, Zhen Chen, and Yanghua Xiao. 2022. Improving continual relation extraction through prototypical contrastive learning. In Proceedings of the 29th International Conference on Computational Linguistics, pages 1885–1895.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. Advances in neural information processing systems, 33:18661–18673.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: A benchmark for question answering research. Transactions of the Association for Computational Linguistics, 7:452–466.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3045–3059.

Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence, 40(12):2935–2947.

Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. 2003. The global k-means clustering algorithm. Pattern recognition, 36(2):451–461.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. arXiv preprint arXiv:2308.08747.

Zheda Mai, Ruiwen Li, Hyunwoo Kim, and Scott Sanner. 2021. Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3589–3599.

Arun Mallya, Dillon Davis, and Svetlana Lazebnik. 2018. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In Proceedings of the European conference on computer vision, pages 67–82.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Introduction to information retrieval.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. Psychology of Learning and Motivation-Advances in Research and Theory, 24(C):109–165.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. In https://openai.com/blog.

OpenAI. 2023a. gpt-3.5-turbo. In https://platform.openai.com/docs/models/gpt-3-5.

OpenAI. 2023b. Gpt-4 technical report. In https://arxiv.org/abs/2303.08774.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32.

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2020. 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, pages 3661–3672.

Chengwei Qin and Shafiq Joty. 2022. Continual few-shot relation learning via embedding space regularization and data augmentation. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2776–2789.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 2001–2010.

Hippolyt Ritter, Aleksandar Botev, and David Barber. 2018. Online structured laplace approximations for overcoming catastrophic forgetting. Advances in Neural Information Processing Systems, 31.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 255–269.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 815–823.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. Advances in neural information processing systems, 30.

Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. Advances in neural information processing systems, 29.

Yifan Song, Peiyi Wang, Weimin Xiong, Dawei Zhu, Tianyu Liu, Zhifang Sui, and Sujian Li. 2023. Infocl: Alleviating catastrophic forgetting in continual text classification from an information theoretic perspective. In Findings of the Association for Computational Linguistics: EMNLP 2023, pages 14557–14570.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. Journal of machine learning research, 9(11).

Hong Wang, Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. 2019. Sentence embedding alignment for lifelong relation extraction. 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), pages 796–806.

Peiyi Wang, Yifan Song, Tianyu Liu, Binghuai Lin, Yunbo Cao, Sujian Li, and Zhifang Sui. 2022. Learning robust representations for continual relation extraction via adversarial class augmentation. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 6264–6278.

Xinyi Wang, Zitao Wang, and Wei Hu. 2023. Serial contrastive knowledge distillation for continual few-shot relation extraction. In Findings of the Association for Computational Linguistics, pages 12693–12706.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems, 35:24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations, pages 38–45.

Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. 2023. Investigating the catastrophic forgetting in multimodal large language models. arXiv preprint arXiv:2309.10313.

Han Zhang, Bin Liang, Min Yang, Hui Wang, and Ruifeng Xu. 2022. Prompt-based prototypical framework for continual relation extraction. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 30:2801–2813.

Kang Zhao, Hua Xu, Jiangong Yang, and Kai Gao. 2022. Consistent representation learning for continual relation extraction. In Findings of the Association for Computational Linguistics, pages 3402–3411.

10. Language Resource References

Han, Xu and Zhu, Hao and Yu, Pengfei and Wang, Ziyun and Yao, Yuan and Liu, Zhiyuan and Sun, Maosong. 2018. FewRel: A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-of-the-Art Evaluation. ISLRN https://thunlp.github.io/1/fewrel1.html.

Zhang, Yuhao and Zhong, Victor and Chen, Danqi and Angeli, Gabor and Manning, Christopher D. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. ISLRN https://nlp.stanford.edu/projects/tacred/.

A. Experimental Details

A.1. Datasets

We conduct experiments on two RE datasets: FewRel (Han et al., 2018) and TACRED (Zhang et al., 2017).

FewRel is a recently popular few-shot relation extraction dataset built from Wikidata. It has high-quality manually annotated data, which contains 100 relations and 700 instances per relation. It is also a dataset with evenly distributed instances for each relation, without long-tail distribution issues.

TACRED is one of the most widely used relation extraction datasets with 106,264 examples and 41 relation types and *no_relation* if on defined relation if held. It is built over newswire and web text from the corpus used in the yearly TAC Knowledge Base Population (TAC KBP) challenges (Zhang et al., 2017). Unlike FewRel, TACRED has an unbalanced distribution of relations, which simulates real-world scenarios. Based on the open relation assumption of CRE (Cui et al., 2021), models continually learn certain relation types, we filter out *no_relation* type as previous works (Han et al., 2020; Cui et al., 2021; Qin and Joty, 2022; Wang et al., 2023).

A.2. Baselines

SCKD (Wang et al., 2023) also focuses on CFRE, but different from our work: First, they focus on using knowledge distillation to preserve the prior knowledge, thus addressing catastrophic forgetting. Although it works, their approach requires extra computation and consumes more computing resources. While we concentrate more on how to better leverage PLMs to learn more generalized knowledge that can be easily adapted to old and new categories, thus activates catastrophic forgetting without extra computation cost. Second, they use bidirectional data augmentation to generate pseudo samples, which essentially replace similar entities to generate new samples. Their augmented samples are unnatural and noisy and also lack diversity. We exploit the power of LLMs to boost smaller PLMs and propose a memory augmentation strategy that can generate more natural and diverse samples. Another work ConPL (Chen et al., 2023) for CFRE, is not adopted for comparison due to its different task settings.

A.3. Implementation Details and Hyperparameters

EMAR+ACA (Wang et al., 2022) and InfoCL (Song et al., 2023) are designed for CRE, not CFRE. We re-implement their origin code in our few-shot setting with the default hyperparameters. For task consistency, we re-implement their data processing using our data to keep the task sequence exactly the same as ours. Note that, we set the temperature parameter in OpenAI API to 0, to ensure the stability of ChatGPT generation as much as possible, so the experimental results can be reproduced as much as possible. However, there is no guarantee that ChatGPT generation will remain the same in the future.

We utilize 1 single NVIDIA Tesla P40 GPU with 24 GB memory on an Intel(R) Xeon(R) Gold 5118 CPU @2.30GHz to run all experiments. For the hyperparameters search, we conduct a grid search to choose the appropriate values. Hyperparameters are illustrated in Table 5.

Hyperparameters	Values
seed	100
batch_size	16
epoch_currenTrain	10
epoch_memTrain	10
learning_rate	$1e-5$
optimizer	Adam
BERT_hidden_size	768
Encoder_output_size	768
BERT_input_max_length	256
margin_m	0.3
normalization_k	0.5
temperature_τ	0.1
contrastive_sample_number	500
soft_prompt_initialization	random
soft_prompt_length	3
soft_prompt_number	4
ChatGPT_temperature	0
generated_number_FewRel	2
generated_number_TACRED	5

Table 5: Hyperparameters setting.

B. More Experimental Results

B.1. Influence of Memory Size

Memory size is an important factor for memory-based continual learning models. Since CRE assumes that the model continues to learn new tasks, the number of tasks(samples) may be very large in the real-world scenario, and the memory size is limited, so it can not save more samples, but only save the key sample. Generally speaking, memory sampling and augmentation are employed to ensure that CFRE works practically in real-world scenarios.

FewRel (*10-way 10-shot*)								
Method	$\mathcal{T}^1$	$\mathcal{T}^2$	$\mathcal{T}^3$	$\mathcal{T}^4$	$\mathcal{T}^5$	$\mathcal{T}^6$	$\mathcal{T}^7$	$\mathcal{T}^8$
Finetune	95.43	45.67	28.26	21.69	17.94	13.83	12.24	10.70
Joint-train	**95.55**	**88.75**	**83.98**	**80.70**	**79.39**	**78.24**	**76.48**	**74.93**
EMAR+ACA‡ (Wang et al., 2022)	95.43	84.23	74.67	71.00	68.61	65.39	63.48	61.75
InfoCL‡ (Song et al., 2023)	95.08	83.34	76.03	72.05	70.09	66.97	65.09	63.02
RP-CRE† (Cui et al., 2021)	95.19	79.21	74.72	71.39	67.62	64.43	63.08	61.46
CRL† (Zhao et al., 2022)	95.01	82.08	79.52	75.48	69.41	66.49	64.86	62.95
CRECL† (Hu et al., 2022)	95.63	83.81	78.06	71.28	68.32	66.76	64.95	63.01
ERDA† (Qin and Joty, 2022)	92.68	66.59	56.33	48.62	40.51	37.21	36.39	33.51
SCKD‡ (Wang et al., 2023)	95.43	86.51	79.72	76.01	73.69	70.45	68.22	66.58
CPL (ours)	<u>95.54</u>	<u>88.51</u>	<u>81.92</u>	<u>77.42</u>	<u>74.71</u>	<u>72.25</u>	<u>69.63</u>	<u>67.49</u>

TACRED (*5-way 10-shot*)								
Method	$\mathcal{T}^1$	$\mathcal{T}^2$	$\mathcal{T}^3$	$\mathcal{T}^4$	$\mathcal{T}^5$	$\mathcal{T}^6$	$\mathcal{T}^7$	$\mathcal{T}^8$
Finetune	86.13	25.17	19.82	17.71	15.25	10.71	10.12	8.45
Joint-train	**88.37**	**83.79**	**76.75**	**72.15**	**70.21**	**66.60**	**64.29**	**61.42**
EMAR+ACA‡ (Wang et al., 2022)	86.10	75.68	51.03	42.39	47.26	45.69	45.02	41.51
InfoCL‡ (Song et al., 2023)	85.42	74.64	62.49	57.80	54.65	49.81	46.24	44.25
RP-CRE† (Cui et al., 2021)	86.68	78.43	69.43	60.71	55.84	51.17	47.27	47.16
CRL† (Zhao et al., 2022)	87.81	77.68	63.31	56.51	53.21	52.42	48.54	46.46
CRECL† (Hu et al., 2022)	83.88	73.45	59.24	53.51	49.27	47.41	45.15	44.33
ERDA† (Qin and Joty, 2022)	79.37	51.28	36.97	29.39	27.80	25.18	24.47	22.37
SCKD‡ (Wang et al., 2023)	<u>88.28</u>	80.77	72.41	65.57	64.90	58.60	55.65	53.42
CPL (ours)	86.52	<u>81.78</u>	<u>75.55</u>	<u>68.04</u>	<u>66.65</u>	<u>61.72</u>	<u>60.61</u>	<u>58.57</u>

Table 6: Main results on FewRel and TACRED in 10-shot setting. † are reported in (Wang et al., 2023), ‡ are we re-running the origin code. The **best results** are in bold, and the <u>second-highest</u> is underlined.

We set the memory size to 1 for a fair comparison with other baselines. We also did experiments to analyze the memory size, and all results are reported in the 10-shot setting without data generation. From figure 6, with the increase in memory size, the performance of the models improves significantly, which proves the importance of memory size for continual learning. Notably, when the memory size increases to 10, that is, the memory contains all previous samples, and the performance is almost close to the Joint-train result.

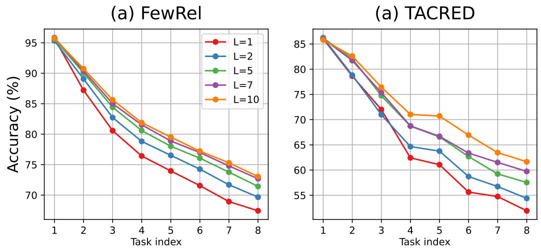


Figure 6: Results on memory size.

B.2. Results with 10-shot

Table 6 shows the 10-shot results on FewRel and TACRED datasets. From the results, our method still achieved the best performance in the 10-shot setting, which indicates that our method can alleviate catastrophic forgetting and overfitting even with more training samples.

C. Generated Samples from ChatGPT

To generate more diverse examples, you can set the temperature parameter in OpenAI API to 1, to ensure that the samples generated are as diverse as possible. But for the stability of the results, we set the temperature to 0 in our experiments. The memory size is set to 1, so we only have one example as the demonstration in the input. The prompt input is shown in Table 4, and we give some cases on FewRel dataset in Table 7.

From the results, we have some findings: (1) Our method is able to generate some high-quality samples, but the generated samples are still a little monotonous even though we've tried to make sure that the generated samples are as diverse as possible. With relation *"child"*, the words "mother", "father", "son" and "daughter" appear in almost all of the samples. (2) Without example in the input, ChatGPT sometimes doesn't understand the relation well and generates some lexically similar samples, but actually false relations.

Example	Cases
with	Relation: child Context: Michelle Obama is the mother of Sasha and Malia Obama, both of whom are well-known for their activism and public speaking engagements. Head Entity: Michelle Obama Tail Entity: Sasha and Malia Obama
with	Relation: child Context: Queen Elizabeth II is the mother of Prince Charles, Prince Andrew, Princess Anne, and Prince Edward, all of whom hold royal titles and duties. Head Entity: Queen Elizabeth II Tail Entity: Prince Charles, Prince Andrew, Princess Anne, and Prince Edward
with	Relation: child Context: Renowned actor Will Smith is the father of Jaden and Willow Smith, both of whom have followed in their father's footsteps and pursued careers in the entertainment industry. Head Entity: Will Smith Tail Entity: Jaden and Willow Smith
with	Relation: child Context: The famous singer Beyonce is the mother of Blue Ivy, Sir, and Rumi Carter, all of whom have captured the hearts of fans with their appearances in music videos and public events. Head Entity: Beyonce Tail Entity: Blue Ivy, Sir, and Rumi Carter
with	Relation: child Context: Former President Barack Obama and First Lady Michelle Obama are the parents of Malia and Sasha Obama, both of whom have grown up in the public eye and captivated audiences with their poise and intelligence. Head Entity: Barack Obama and Michelle Obama Tail Entity: Malia and Sasha Obama
without	Relation: child Context: Sarah watched proudly as her daughter, Emily, rode her bike without training wheels for the first time. Head Entity: Sarah Tail Entity: Emily
without	**Relation: child** Context: John and Mary are siblings. John is the eldest child in the family. Head Entity: John Tail Entity: Mary
without	Relation: child Context: The park was filled with families enjoying the sunny day, including Jane pushing her son, Noah, on the swings. Head Entity: Jane Tail Entity: Noah
without	Relation: child Context: The school auditorium was buzzing with excitement as Maria prepared to watch her daughter, Isabella, perform in the school play. Head Entity: Maria Tail Entity: Isabella
without	Relation: child Context: Jake beamed with pride as he watched his son, Liam, take his first steps towards him. Head Entity: Jake Tail Entity: Liam

Table 7: Generated cases on FewRel. **Errors** are highlighted in bold and keywords for judging relations are underlined.