HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

GRADUATION THESIS

Continual Relation Extraction via Task-specific Prompt Pool and Representation Generation

NGUYỄN VĂN THANH TÙNG
tung.nvt190090@sis.hust.edu.vn

Major: Data Science

Supervisor:	MSc. Ngô Văn Linh	_____________
		Signature
Department:	Computer Science	
School:	School of Information and Communications Technology	

HANOI, 08/2023

ACKNOWLEDGMENT

I am profoundly grateful to all those who have contributed to the successful completion of this thesis.

First and foremost, I extend my heartfelt thanks to my family, friends, and loved ones for their unwavering support and encouragement throughout this arduous journey. Your unwavering belief in me has been my driving force, propelling me to persevere and achieve the best possible results.

I am immensely indebted to my dedicated teachers at Hanoi University of Science and Technology (HUST), especially those from the School of Information and Communication Technology (SOICT), for their invaluable guidance, mentorship, and profound insights that have shaped my academic growth and enriched my understanding.

A special thank you goes to my esteemed supervisor, MSc. Ngô Văn Linh, whose expertise, patience, and encouragement have been instrumental in refining my research and bringing it to fruition. Your invaluable feedback and constructive criticism have significantly improved the quality of this work.

Lastly, I humbly acknowledge the determination and relentless effort I have poured into this endeavor, overcoming challenges and embracing learning opportunities along the way.

Each of you has played a significant role in my academic journey, and I am sincerely thankful for the unwavering support you have provided. Without your encouragement, guidance, and belief in me, this accomplishment would not have been possible. Once again, thank you all for being part of this journey and making this achievement a reality.

ABSTRACT

Continual Relation Extraction (CRE) involves classifying the semantic relationships between entities in text while accommodating a continually expanding set of relation types. Here, entities refer to specific named entities, such as people, organizations, locations, dates, or any other relevant information in the text. The relationships between these entities are crucial for understanding the context and meaning of the text.

To mitigate Catastrophic Forgetting (CF) in CRE, most existing approaches use memory buffers to facilitate rehearsal of old knowledge when learning new tasks. Recently, prompt-based methods have emerged as strong alternatives to rehearsal-based approaches, showcasing impressive success in the Computer Vision (CV) domain. However, these methods do have certain drawbacks, including inaccurate prompt selection, insufficient strategies for reducing forgetting on shared parameters, and limited optimization of cross-task and within-task variances.

In my proposed method, I leverage a specific Prompt Pool for each task, acknowledging the differences among within-task data while maximizing cross-task variances. Moreover, I integrate a generative model to reinforce previously acquired knowledge on shared layers and parameters without explicitly storing data. Extensive experiments validate the effectiveness of my approach, surpassing state-of-the-art (SOTA) rehearsal-based and prompt-based methods in CRE on 2 datasets: FewRel and TACRED. By achieving superior results on these datasets, the proposed method demonstrates its capability to address the limitations of existing approaches and advance the state of Continual Relation Extraction.

Author of Thesis

Nguyễn Văn Thanh Tùng

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Definition
BERT	Bidirectional Encoder Representation from Transformer
CF	Catastrophic Forgetting
CL	Continual Learning
CRE	Continual Relation Extraction
CV	Computer Vision
L2P	Learning to Prompt
NLP	Natural Language Processing
RE	Relation Extraction

CHAPTER 1. INTRODUCTION

1.1 Continual Relation Extraction

Relation Extraction (RE) is a task in Natural Language Processing (NLP) that involves automatically identifying and extracting relationships between entities mentioned in a given text. Entities refer to specific elements such as persons, organizations, locations, or other objects, while the relationships represent the connections or interactions between these entities. The choice of RE as a research problem arises from the growing significance of NLP applications in various domains, including Information Retrieval [1], Question-Answering Systems [2], Knowledge Graph Construction [3], and more. With the exponential growth of digital data, there is an urgent need for automated techniques that can efficiently analyze and extract valuable knowledge from the vast textual sources available on the internet and other repositories.

The field of Continual Learning (CL), also known as lifelong learning or incremental learning, addresses the challenge of adapting models to new tasks without completely forgetting previously learned knowledge. In the context of Continual Relation Extraction (CRE), continual learning is crucial to ensure that RE models can effectively handle changing language patterns, evolving relationships, and new contexts over time. A critical problem that arises in CRE is Catastrophic Forgetting (CF). As the model is exposed to new data to learn additional relations or adapt to new domains, it tends to forget previously learned relations, leading to a degradation in performance on previously encountered tasks. Catastrophic forgetting hinders the model's ability to maintain proficiency in all previously learned relations, limiting its utility in real-world scenarios where data continuously evolves. To address the challenges of RE and CF, there is an urgent need to develop robust and efficient CRE methods. These methods should enable RE models to effectively acquire new knowledge while retaining their understanding of previously learned relations without experiencing significant performance drops.

Recently, Continual Relation Extraction (CRE) has emerged as a compelling subject, combining vital aspects from two distinct fields: Continual Learning (CL) and Relation Extraction (RE). This convergence addresses real-world issues, making it a significant and topical area of interest. The goal of this thesis is to investigate and propose novel CRE techniques that can facilitate the continual learning of RE models. By exploring various approaches, the thesis aims to enhance the adaptability and generalization capabilities of RE models, enabling them to excel in dynamic NLP environments with ever-changing data distributions and linguistic patterns.

1.2 Background and Problems of Research

In the field of Continual Relation Extraction (CRE), researchers have made significant strides in addressing the challenge of learning new relations while maintaining accuracy on previously learned ones without the need for retraining. This progress has been primarily achieved through rehearsal-based continual learning methods [4]–[6], which have proven successful in mitigating the issue of Catastrophic Forgetting (CF). The essence of rehearsal-based approaches lies in their inspiration drawn from memory-based techniques used in continual learning [7]–[9]. The rehearsal mechanism (see Figure 1.1, left) is facilitated by a memory buffer of past task data, allowing the model to reinforce its knowledge of past relations while learning new tasks, effectively mitigating catastrophic forgetting. During training, the model selectively stores and replays samples from past relations as it encounters new relations, maintaining a diverse and representative dataset that covers both old and new relationships. The rationale behind this approach is to refresh the model's memory of previously learned knowledge and prevent it from completely forgetting past experiences.

Despite the effectiveness of rehearsal-based methods, concerns have been raised regarding the long-term storage of data and privacy standards. In response to these concerns, rehearsal-free continual learning techniques [10], [11] have gained considerable attention as an alternative approach. These methods employ innovative strategies that allow the model to adapt and learn from new information without explicitly storing past data. This aspect is particularly crucial in sensitive domains like medicine, finance, or others, where data privacy is of utmost importance.

Recently, substantial progress has been made in the domain of rehearsal-free CL, particularly with advancements in prompting techniques [12]. These novel approaches, as proposed in L2P [13], DualPrompt [14], CODAPrompt [15] have demonstrated remarkable success in computer vision tasks, even surpassing state-of-the-art memory-based methods.

Figure 1.1 provides an overview of the L2P approach in comparison to conventional continual learning methods. While L2P also utilizes representative features from pre-trained models, it takes a different approach during the continual learning process. Instead of adjusting the parameters of the pre-trained model, L2P keeps the model unchanged and focuses on learning a set of instructions known as prompts. These prompts dynamically guide the models to solve specific tasks. The

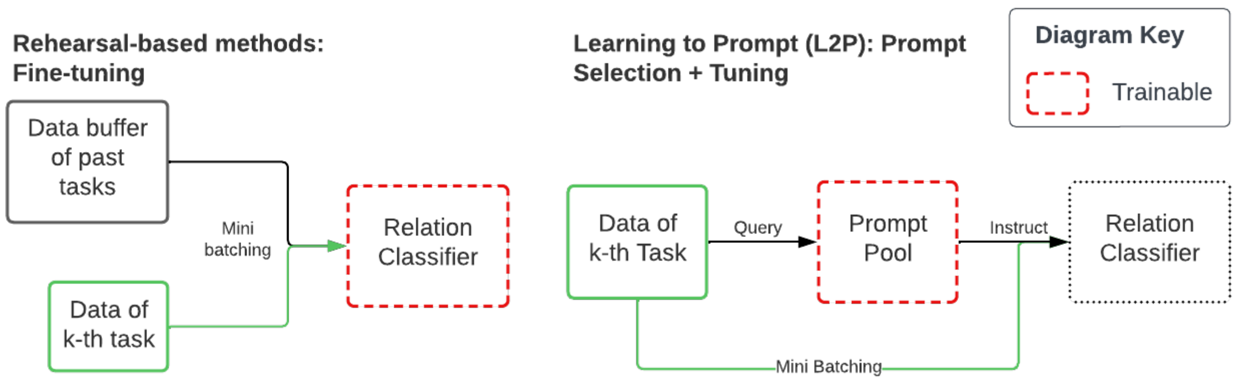

Figure 1.1. Previous frameworks: Compared to rehearsal-based methods, L2P simplifies learning by using a single backbone model and a Prompt Pool. It stores task-specific knowledge in the Prompt Pool, eliminating the need for a data buffer of past tasks to prevent forgetting. L2P selects and updates prompts from the pool on a per-instance basis.

prompts are organized within a shared memory space called the Prompt Pool, using a key-value structure. L2P incorporates a query mechanism to dynamically access a subset of task-relevant prompts based on the input features of each instance, allowing the model to adapt to different tasks effectively.

The distinguishing factor of these prompting-based techniques is their focus on tuning auxiliary embeddings known as prompts, instead of directly fine-tuning the pre-trained Transformer-based encoder. These prompts can be dynamically inserted into the training process, catering to each instance's features [15] or utilized as task-specific features [13], [14]. By fine-tuning the prompts, the model adapts to new relations while preserving its ability to retain previously learned knowledge. The advantages of these rehearsal-free methods extend beyond their ability to alleviate privacy concerns. They also offer practical benefits in terms of computational efficiency, as they do not rely on maintaining large memory buffers for rehearsal. Moreover, they can readily adapt to different tasks without requiring extensive retraining, making them a promising avenue for CL in various real-world applications.

1.3 Research Objectives and Conceptual Framework

The primary goal of this research project is to explore and propose effective solutions for Continual Relation Extraction (CRE) using rehearsal-free techniques. While rehearsal-based methods have shown promise in mitigating catastrophic forgetting, they raise concerns related to data privacy and long-term data storage. In response to these concerns, rehearsal-free continual learning techniques have emerged as an alternative approach that allows models to learn new information

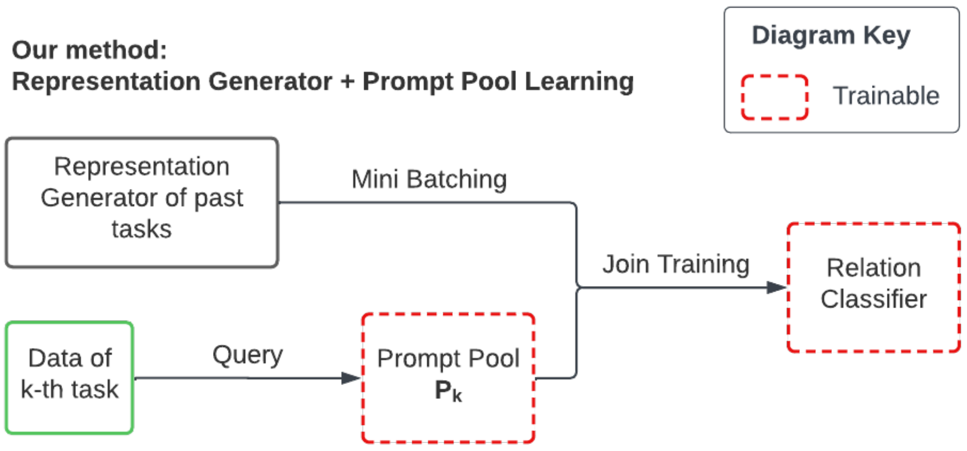

Figure 1.2. My framework: At the k-th task, I employ a task-specific Prompt Pool $\mathbf{P}_k$ instead of a shared Prompt Pool to prevent the update of the Prompt Pool from affecting the retention of information related to previous tasks. Additionally, I use a representation generator to synthesize past-task representations, reinforce the relation classifier with old knowledge.

without explicitly storing past data. In this section, I present the conceptual framework for my proposed solutions to address the existing challenges of rehearsal-free continual learning in CRE.

I acknowledge that existing rehearsal-free methods have inherent weaknesses. Firstly, they lack effective strategies to mitigate forgetting on shared parameters. Examples of such shared parameters include the shared Prompt Pool [13], the task-agnostic G-Prompt [14], and/or the shared classifier. Secondly, methods that rely on prompt selection to determine task identity, most notably, DualPrompt [14], suffer from severely inaccurate prompt selection (see Table 4.2) due to the unsupervised $key \times query$ mechanism. Finally, these methods have very limited optimization of both cross-task and within-task variances. For instance, L2P [13] use a common Prompt Pool where each sample is paired with multiple prompts selected through a $key \times query$ mechanism. As a result, instances from different tasks often share one or more prompts, leading to reduced cross-task variance. This issue becomes particularly prominent in CRE, where instances from different relation classes might frequently have very similar contexts, as shown in the following example:

"`[X] is a professor at [Z university]`" and
"`[X] is advised by a professor at [Z university]`"

To address these weaknesses, I will propose a novel prompting method for CRE that uses separate Prompt Pools for each task, enhancing cross-task disparity while recognizing within-task variances in the data. Moreover, to reduce catastrophic forgetting on shared parameters, I will employ a generative model to synthesize latent representations of data for replay during training, thereby enhancing the model's performance without the need for explicit data retention. Additionally, I will train an additional generative model to acquire unprompted pseudo-representations, which will be used to improve the quality of the sub-module responsible for selecting task-specific prompts, addressing the inaccuracies present in existing unsupervised prompt selection approaches.

The framework of my proposed solution is depicted in Figure 1.2, highlighting the different components that contribute to addressing the identified weaknesses in existing rehearsal-free methods.

1.4 Contributions

Building upon the background and challenges presented above, my research proposed innovative solution to enhance rehearsal-free CL techniques for CRE. My contributions are as follows:

a. Novel Prompting Method for CRE: My first solution involves introducing a novel prompting-based method tailored specifically for Continual Relation Extraction. During training, I associate each task with a separate Prompt Pool. This approach enhances the cross-task disparity, ensuring that each task relation is represented distinctively, while also recognizing within-task variances present in the data. This helps the model effectively adapt to new relations without forgetting previous ones, leading to improved performance in continual learning scenarios.

b. Reducing Catastrophic Forgetting on Shared Parameters: One of the weaknesses observed in existing rehearsal-free methods is their struggle with shared parameters, such as the shared Prompt Pool and classifier. To address this, I propose harnessing the power of a generative model to synthesize latent representations of data for replay. This avoids the need for explicit data retention while allowing us to control the quantity of rehearsal data effectively. The synthesized prompted representations are used to train the module responsible for relation classification, leading to significant improvements in mitigating catastrophic forgetting on shared parameters.

c. Enhancing Task-Specific Prompt Selection: To improve the quality of the submodule responsible for selecting task-specific prompts, I introduce an additional generative model. This model acquires unprompted pseudo-representations, which are then used to train the task predictor. The current approach used in rehearsal-free methods relies solely on the similarity of representations for prompt selection, which can lead to inaccuracies. By incorporating a supervised training process using unprompted pseudo-representations, I aim to enhance the accuracy of task-specific prompt selection and achieve better overall performance.

By implementing and evaluating this proposed framework, my research aims to contribute to the field of Continual Relation Extraction and provide a more privacy-friendly, efficient, and accurate approach to continual learning in real-world applications.

1.5 Organization of Thesis

My thesis is structured as follows:

Chapter 1 serves as an introduction to the Continual Relation Extraction (CRE) problem. It begins by offering a comprehensive overview of existing approaches used to tackle this challenge, highlighting their strengths and limitations. In doing so, it sheds light on the persisting issues faced by these methods. Within this context, a novel solution is presented to address the aforementioned problems. The proposed method not only effectively tackles these challenges but also outlines the specific contributions made to the field.

In Chapter 2, a detailed review of existing research is presented, where the focus of the study is carefully defined. The exploration begins by discussing important topics like Bidirectional Encoder Representations from Transformers (BERT), Continual Learning (CL), and Continual Relation Extraction (CRE). Each subject is explained with examples. Furthermore, the chapter explores notable efforts in CRE that have used methods without rehearsal. These successful works are analyzed to gain valuable insights and identify potential areas for improvement. This thorough examination sets the foundation for the future developments in the research.

In Chapter 3, I focus on explaining my proposed method to address the CRE problem. The chapter starts by presenting a detailed data flow diagram, which visually represents the architecture of the method. This diagram acts as a foundation, making it easier to understand the entire process. Besides, using the pseudo-code of algorithm as a guide, the following sections of Chapter 3 delve into the finer details of each component. I provide clear and detailed explanations to ensure that readers fully understand how the method works without any confusion. This structured approach in Chapter 3 ensures that readers grasp the proposed method thoroughly, making it easier for them to transition into the later stages of the research.

In Chapter 4, I present a series of thorough experiments conducted on the FewRel and TACRED datasets. These experiments demonstrate how well my proposed regularization method performs, especially in the difficult context of CRE. To make the experiments transparent and reproducible, I provide detailed information about the hyper-parameter settings at the Appendix section. This allows readers to replicate the experiments easily and better understand the effectiveness of my method. By conducting these experiments, Chapter 4 strengthens the credibility of my regularization approach and confirms its reliability as a strong solution for the CRE problem.

Chapter 5 acts as the final chapter, summarizing the main findings from the research and discussing the limitations of the project. It provides a comprehensive overview, highlighting the key takeaways and insights gained throughout the study. In addition to presenting the limitations, this chapter explores potential future paths of work. By addressing these identified limitations directly, the overall approach of the proposed method can be improved and refined. This forward-looking analysis lays the foundation for future researchers to build upon the current research and make significant progress in the field of CRE. Chapter 5 serves as the culmination of the research, emphasizing its importance and contributing to the broader academic discussion with its comprehensive evaluation of conclusions, limitations, and future directions.

At the end of the thesis, there is the Appendix section, which provides a valuable resource containing a diverse range of supplementary information to support and enhance the main content of the article. In this section, readers can access various materials and details that offer additional context and insights into the research.

CHAPTER 2. LITERATURE REVIEW

2.1 Scope of Research

In Continual Learning (CL), this study explores three major scenarios, which will be discussed in detail in Subsection 2.2.1. However, for the Continual Relation Extraction (CRE) task, the focus will be solely on the Class-Incremental Learning scenario. This choice allows for a fair comparison with previous works on the same problem. Moreover, Class-Incremental Learning represents the most practical and challenging scenario among the three. Within the Class-Incremental Learning scenario, the thesis centers on rehearsal-free based methods, specifically the prompt-based continual learning approach. This approach is currently dominant and attractive in the field of continual learning.

The main objective of this research is to investigate the effectiveness of prompting methods in CRE. The focus will be on designing the architecture of the prompts in the model and developing methods that can utilize these prompts effectively, addressing the limitations of current prompt-based continual learning methods. To ensure a fair comparison with state-of-the-art CRE works, Bidirectional Encoder Representations from Transformers (BERT) language model will be used as the base encoder. This choice is motivated by BERT's extensive use in both rehearsal-based and rehearsal-free methods, making it a suitable benchmark for evaluation.

2.2 Related Works

2.2.1 Continual Learning

Recent strides in Deep Learning [16] have revolutionized the development of intelligence systems, enabling them to tackle complex problems across various domains like Natural Language Processing [17] and Computer Vision [18]. However, a significant limitation in these studies is their focus on the traditional setting, where tasks are predetermined and fixed during training and testing. This approach falls short when faced with real-life scenarios, where new tasks may arise unexpectedly during deployment.

To overcome this challenge, the paradigm of Continual Learning [19] has been proposed. Continual Learning, also known as Lifelong Learning [20], Sequential Learning [21], or Incremental Learning [22], aims to enable models to learn from an ongoing stream of data associated with different tasks and domains. In practical terms, this means that continual learners must handle numerous consecutive tasks, with the number of novel classes rapidly increasing over time. The goal is for the continual learner to perform well on all tasks it has been trained on, which includes both previously encountered datasets and newly emerging datasets.

A key issue in Continual Learning is the stability-plasticity trade-off. Here, stability refers to maintaining accuracy on previously learned tasks, while plasticity relates to effectively learning new tasks. Modern deep learning models often adapt quickly to new knowledge but suffer from Catastrophic Forgetting [23], where they forget previously learned information. This phenomenon is a fundamental problem in the continual learning paradigm, prompting extensive research to find effective solutions.

In Continual Learning, researchers have put forth three primary scenarios based on the tasks that models need to tackle. These scenarios are known as Task-Incremental Learning [24], Domain-Incremental Learning [25], and Class-Incremental Learning [26]. Task-Incremental Learning refers to situations where test-time performance depends on the task or context identity. On the other hand, Class-Incremental Learning or Domain-Incremental Learning comes into play when test-time performance is not influenced by task identity. The key distinction lies in the way task identity is handled. In Domain-Incremental Learning, models don't need to infer task identity, while in Class-Incremental Learning, they do. This makes Class-Incremental Learning the most challenging problem among the three scenarios. For a clearer perspective, the differences between the three scenarios are summarized in Table 2.1, as shown below.

Scenarios	*Necessary during the test phrase*
Task-Incremental Learning	Solved tasks so far, with provided task-ID
Domain-Incremental	Solved tasks so far, without a provided task-ID
Class-Incremental	Solved tasks so far and inferring task-ID

Table 2.1: Three scenarios for continual learning (Source: [27])

2.2.2 Bidirectional Encoder Representations from Transformers

a, Overview

Bidirectional Encoder Representations from Transformers (BERT) is a groundbreaking natural language processing (NLP) model introduced by Google AI language in 2018 [28]. It represents a significant advancement in pre-trained language representation techniques, enabling the model to understand the context of words in a sentence by capturing both left and right context simultaneously. Traditional language models like Word2Vec [29] and GloVe [30] were unidirectional, which means they could only consider the words to the left or right of a given word during training. This limitation often resulted in shallow representations, as the models lacked a complete understanding of the sentence's context. BERT, however, introduced bidirectional learning, which revolutionized NLP tasks and established new state-of-the-art (SOTA) results across various benchmarks.

b, Bert Tokenization

A crucial step in BERT's pre-training process is tokenization, which involves breaking down raw text into smaller units called tokens, making it suitable for input into the model. Tokenization is a fundamental step in NLP that converts continuous text into discrete tokens, usually words or subwords. However, traditional tokenization techniques fall short when dealing with languages with complex morphological structures and out-of-vocabulary (OOV) words. BERT, on the other hand, introduced a novel tokenization approach called WordPiece tokenization [31] [32], which effectively addresses these challenges.

The WordPiece tokenization algorithm is a subword-based tokenization technique. It breaks down words into smaller units, called subwords, or pieces, allowing the model to handle OOV words by composing them from known subwords. This approach significantly reduces the size of the vocabulary and enables BERT to efficiently process a vast number of words and subwords.

Here's an overview of the BERT tokenization process:

1. Word Splitting. The input text is first split into words. For example, the sentence `"I love natural language processing"` would be broken down into `['I', 'love', 'natural', 'language', 'processing'].`

2. Subword Tokenization. Each word is further tokenized into subwords using the WordPiece algorithm. For instance, `"processing"` might be split into `['process', '##ing']`. The double hash `(##)` indicates that it is a continuation of the previous subword and helps retain the context.

3. Special Tokens. BERT requires special tokens to distinguish different parts of the input and perform various tasks like sentence-pair classification or language modeling. Two special tokens are added at the beginning and end of the sentence: [CLS] and [SEP]. [CLS] stands for `"classification"` and is used to represent the entire sentence, while [SEP] separates sentences in sentence-pair tasks.

4. Padding and Truncation. As BERT processes inputs in batches, all sequences need to have the same length. Sequences longer than the maximum allowed length are truncated, and shorter sequences are padded with a special padding token [PAD] to match the desired length.

5 .IDs converter. After tokenization, the text is converted into token IDs using

BERT's vocabulary. Each token is mapped to a unique ID, allowing BERT to process the input as a sequence of integers rather than raw text.

BERT's ability to capture context and handle complex linguistic structures is attributed to its tokenization scheme. By leveraging subwords, BERT can handle OOV words, understand contextually similar words, and improve the generalization of the model across various NLP tasks. The BERT tokenization process lays the groundwork for the model's success and continues to be a key component in the development of advanced NLP models.

c, Bert Word Embeddings

Unlike traditional word embeddings like Word2Vec or GloVe, which create static representations of words, Bert generates dynamic contextual embeddings. These embeddings capture the meaning of a word based on its context in a given sentence, allowing for a deeper understanding of word semantics and improving the performance of various NLP tasks.

Bert utilizes a deep transformer architecture, enabling bidirectional processing of text. It learns by predicting masked words within a sentence and understanding the relationships between words in both forward and backward directions, which will be described in the next sections. As a result, it captures the intricacies of language and is better equipped to handle polysemy (multiple meanings of a word) and synonymy (different words with the same meaning).

Example:
Let's consider the sentence: `"The bank was situated by the river."`. With traditional word embeddings, the word `"bank"` would have a static representation regardless of its context. However, using BERT Word Embeddings, the context of the word is considered. In this case, `"bank"` could refer to a financial institution or the side of a river. With BERT, the word `"bank"` will have different embeddings based on the context:
`"bank"` (financial institution) $\longrightarrow$ `[0.457, -0.824, 0.135, ...]`
`"bank"` (side of a river) $\longrightarrow$ `[-0.673, 0.245, -0.732, ...]`
As shown in the example, BERT Word Embeddings provide distinct representations for the same word based on its context, leading to a more nuanced understanding of language and improved performance in various NLP tasks.

d, Bert's Pre-training Process

BERT's remarkable success can be attributed to its sophisticated pre-training process, which equips the model with robust language understanding capabilities. During pre-training, BERT undergoes two crucial tasks that contribute to its effective language context comprehension. It is pretrained on a vast corpus of unlabelled text, including the entire Wikipedia (2.5 billion words) and book corpus (0.8 billion words), using two strategies known as Masked Language Model (MLM) and Next Sentence Prediction (NSP). These strategies are visually represented in the left of the figure below:

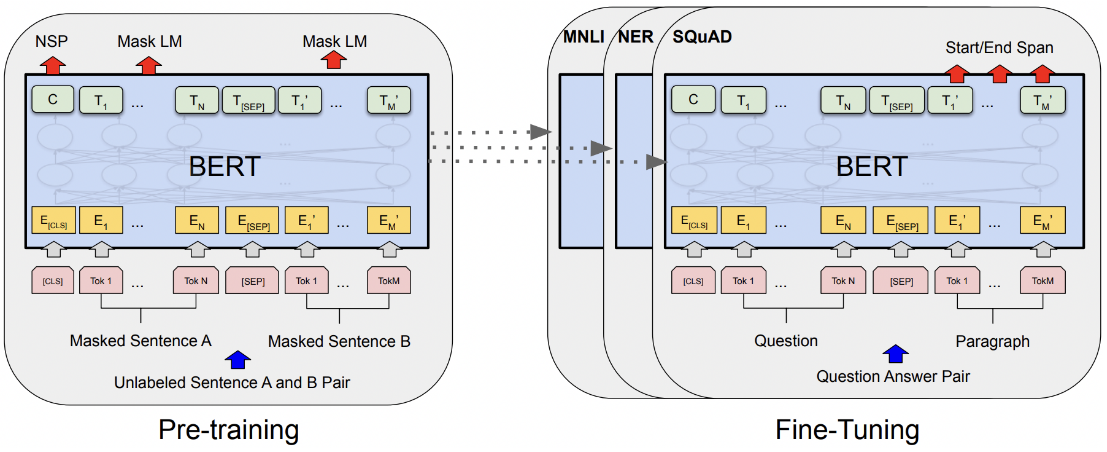

Figure 2.1. Overall pre-training and fine-tuning procedures for BERT (Source: [28]): Except for the output layers, the architectures remain consistent in both pre-training and fine-tuning stages. The pre-trained model parameters are utilized to initialize models for various downstream tasks. During fine-tuning, all parameters are adjusted. Each input example is prefixed with a special symbol, [CLS], while [SEP] serves as a special separator token, effectively separating questions and answers.

Masked Language Model (MLM). The first task in BERT's pre-training process is the Masked Language Model (MLM). In this task, BERT is exposed to a vast corpus of text and randomly masks certain words within sentences. The model's objective is then to predict the missing words based on the context provided by the surrounding words. This unique approach allows BERT to learn bidirectional relationships between words and their contextual nuances. By understanding both the left and right contexts of a word, BERT can capture the intricate dependencies between words in a sentence. As a result, BERT gains a deeper understanding of sentence context and can effectively grasp the meaning of complex sentences.

Example of MLM:
Original Sentence: `"The capital of France is [MASK]."`
Predicted Word: `"Paris"`
By considering the context of the surrounding words, BERT predicts that the missing word is `"Paris"`, which completes the sentence with the proper context.

Next Sentence Prediction (NSP). In the Next Sentence Prediction task, BERT is presented with pairs of sentences from the training corpus. The model learns to predict whether the second sentence follows the first one in the original text or not. Mastering this task enables BERT to understand the relationships between sentences and the broader context in which words are used. This skill becomes invaluable in various natural language processing (NLP) tasks, such as question-answering and sentiment analysis, where understanding the connections between sentences is crucial.

Example of NSP: For a pair of sentences:
(A) `"The sun is shining brightly in the sky."`
(B) `"It's a beautiful day."`
In this example, BERT would predict that sentence (B) does not follow sentence (A) consecutively in the original text.

In conclusion, BERT's pre-training process, with the Masked Language Model and Next Sentence Prediction tasks, equips the model with a strong foundation for language understanding. The bidirectional learning and context comprehension achieved during pre-training allow BERT to excel in a wide range of natural language processing tasks, making it a pivotal breakthrough in the field of NLP.

e, Fine-Tuning BERT for Natural Language Processing Tasks

BERT has demonstrated exceptional performance in a wide range of NLP tasks, including text classification, named entity recognition, question answering, and more. Fine-tuning BERT (see Figure 2.1, right) involves taking the pre-trained BERT model and adapting it to specific downstream tasks by training it on task-specific datasets. This process helps leverage BERT's contextual language understanding and transfer its knowledge to new tasks.

Below is an overview of the fine-tuning process for BERT:

1. Pre-trained BERT Model. BERT is pre-trained on a large corpus of text from diverse sources, utilizing a masked language modeling (MLM) objective and a next sentence prediction (NSP) objective. This pre-training allows BERT to learn rich contextual representations of words, capturing both left and right context within a sentence as descibed above.

2. Task-Specific Data Collection. To fine-tune BERT for a particular NLP task, a task-specific dataset is required. This dataset should be labeled and aligned with the task's objective. For instance, for sentiment analysis, the dataset should contain sentences or documents labeled with sentiment scores (e.g., positive, negative, or neutral).

3. Tokenization. The text in the task-specific dataset is tokenized into subword units using BERT's tokenization scheme, WordPiece. Tokenization converts words into subword tokens and ensures that the input text adheres to BERT's maximum token limit.

4. Model Architecture. BERT comes in different sizes, such as BERT-base (uncased) and BERT-large (uncased). The choice of architecture depends on the computational resources available and the complexity of the task. The BERT architecture consists of a multi-layer bidirectional Transformer encoder, which is responsible for generating contextual embeddings.

5. Fine-Tuning Objective. During fine-tuning, the pre-trained BERT model is adapted to the task-specific data. The model is then fine-tuned using a task-specific objective function (e.g., cross-entropy loss for classification tasks). The weights of BERT are updated during the fine-tuning process, while the task-specific layers are randomly initialized.

6. Training Procedure. The fine-tuning process involves iteratively training the adapted BERT model on the task-specific dataset. The learning rate, batch size, and number of training epochs are hyperparameters that need to be carefully tuned to achieve the best results.

7. Task-Specific Layers. In some cases, additional task-specific layers are added on top of BERT to further tailor the model to the specific task. These layers can include fully connected layers, pooling operations, or attention mechanisms, depending on the nature of the task.

8. Evaluation and Tuning. Once the fine-tuning process is complete, the model is evaluated on a separate validation set or through cross-validation to assess its performance. Based on the evaluation results, hyper-parameters can be adjusted and further fine-tuning iterations may be performed to achieve better performance.

9. Inference. After fine-tuning, the BERT model is ready for deployment and can be used for inference on new, unseen data. The fine-tuned model can be utilized to make predictions, classify text, answer questions, or perform other specific NLP tasks.

In summary, fine-tuning BERT allows for the adaptation of the pre-trained language model to various NLP tasks, enabling powerful and efficient natural language understanding across a wide range of applications. It leverages the general language knowledge captured during pre-training to provide strong performance on diverse downstream tasks, making it a go-to choice for many NLP practitioners.

2.2.3 Problem Formulation of CRE

In this research project, I will be concentrating solely on the continual learning setting concerning the classification task, specifically related to the Continual Relation Extraction problem. To clearly define the classification problem in the context of Continual Relation Extraction, I present a concrete formalization below. CRE [33], [34] involves training a model on a sequential series of tasks $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_K\}$ where each task $\mathcal{T}_k$ has a distinct training set denoted as $\mathcal{D}_k$, and a corresponding relation set denoted as $\mathcal{R}_k$. Each task $\mathcal{T}_k$ follows the conventional framework of supervised classification [35], [36], comprising a series of $\mathcal{N}_k$ samples and their corresponding labels $\mathcal{D}_k = \{(\boldsymbol{x}_i^k, y_i^k)\}_{i=1}^{\mathcal{N}_k}$, where $\boldsymbol{x}_i^k$ represents the input data consisting of natural language text and an entity pair, and y_i^k belongs to the relation label set $\mathcal{R}_k$. The main goal of CRE is to train a model that can learn from new tasks without affecting its performance on the tasks it has already observed. Specifically, after learning the k-th task, the model must be capable of determining the relation of an entity pair to one of the relations in the set $\hat{\mathcal{R}}_k$. Here $\hat{\mathcal{R}}_k$ represents the set of relations observed up to the k-th task, defined as $\hat{\mathcal{R}}_k = \bigcup_{i=1}^{k} \mathcal{R}_i$.

Example:
Let's say I have a research project aimed at training a Natural Language Processing model for Continual Relation Extraction in text. I have a series of tasks $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3$ that represent different relation extraction tasks. The Example Dataset is shown in Table 2.2.

Task $\mathcal{T}_1$: Training set $\mathcal{D}_1$ consists of 2 samples, each containing a sentence and an entity pair along with their corresponding relation label.
Relation Set $\mathcal{R}_1$: ["PlaceOfBirth", "Discovered"]
Observed Relation Set $\hat{\mathcal{R}}_1$: ["PlaceOfBirth", "Discovered"]

Task $\mathcal{T}_2$: Training set $\mathcal{D}_2$ consists of 2 samples, each following the same format as Task $\mathcal{T}_1$, but with different relation labels.
Relation Set $\mathcal{R}_2$: ["Created", "Invented"]
Observed Relation Set $\hat{\mathcal{R}}_2$: ["PlaceOfBirth", "Discovered", "Created", "Invented"]

Task $\mathcal{T}_3$: Training set $\mathcal{D}_3$ consists of 2 samples, each following the same format as Task $\mathcal{T}_1$ and $\mathcal{T}_2$, but with additional new relation labels.
Relation Set $\mathcal{R}_3$: ["BoilingPoint", "Awarded"]
Observed Relation Set $\hat{\mathcal{R}}_3$: ["PlaceOfBirth", "Discovered", "Created", "Invented", "BoilingPoint", "Awarded"]

In this example, I have three tasks, each with its own training set and relation set. The goal of the Continual Relation Extraction problem is to train a model on

Sample	Sentence	Entity Pair (E1, E2)	Relation Label	Task
1	Barack Obama was born in Honolulu.	("Barack Obama", "Honolulu")	"PlaceOfBirth"	$\mathcal{T}_1$
2	Albert Einstein discovered the theory of relativity.	("Albert Einstein", "Theory of Relativity")	"Discovered"	$\mathcal{T}_1$
3	The Mona Lisa was painted by Leonardo da Vinci.	("Mona Lisa", "Leonardo da Vinci")	"Created"	$\mathcal{T}_2$
4	Thomas Edison invented the electric light bulb.	("Thomas Edison", "Electric Light Bulb")	"Invented"	$\mathcal{T}_2$
5	Water boils at 100 degrees Celsius.	("Water", "100 degrees Celsius")	"BoilingPoint"	$\mathcal{T}_3$
6	Marie Curie won the Nobel Prize in Physics.	("Marie Curie", "Nobel Prize in Physics")	"Awarded"	$\mathcal{T}_3$

Table 2.2: Samples of Example Dataset for CRE

these tasks sequentially, such that after learning each task, the model is capable of predicting relations for entity pairs using the union of the relations observed up to that point. For instance, after learning Task $\mathcal{T}_2$, the model should be able to determine relations like ["Created", "Invented"] which are in the set $\hat{\mathcal{R}}_2$ (the union of relation sets $\mathcal{R}_1$ and $\mathcal{R}_2$). Similarly, after learning Task $\mathcal{T}_3$, the model should be able to predict relations like "BoilingPoint" and "Awarded," which are in the set $\hat{\mathcal{R}}_3$ (the union of relation sets $\mathcal{R}_1$, $\mathcal{R}_2$, and $\mathcal{R}_3$).

2.2.4 Prompt-based Continual Learning

In the rapidly evolving field of Natural Language Processing (NLP), the concept of $prompts$ has gained significant prominence. A prompt can be defined as a piece of text or an instruction provided to a language model, serving the crucial purpose of guiding the model in generating a desired response. By presenting a prompt, I give the model a starting point or a context, helping it understand the specific task at hand [12], [37]–[40].

Initially, prompts were primarily seen as explicit words or phrases added to the input to influence the model's output. However, as NLP has advanced, the use of prompts has evolved significantly. Nowadays, they go beyond simple additions to the input and have become an essential technique for fine-tuning large pre-trained transformer-based encoders on specific downstream tasks without altering their parameters [41]–[43].

The idea of prompts has proven to be remarkably influential and versatile, inspiring various methods that effectively utilize prompting techniques in combination with pre-trained transformer-based encoders. These approaches leverage prompts as high-level instructions for lifelong class-incremental models, enabling models to continually learn and adapt to new tasks. Among such approaches, three particularly noteworthy ones stand out: L2P [13], DualPrompt [14], and CODA-Prompt [15].

Learning to Prompt (L2P):

L2P [13] is one of the leading approaches that leverage prompting to facilitate lifelong class-incremental learning. However, due to space limitations in the main thesis, I can only provide a detailed description of L2P. For a comprehensive understanding of the DualPrompt [14] and CODA-Prompt [15] methods, I will present them in Appendix A.1. In this section, we will delve into the L2P method in the context of Contextual Relation Extraction (CRE), utilizing a frozen BERT [28] as the pre-trained transformer-based encoder. To facilitate this approach, a $prompt$ is defined as $P \in \mathbb{R}^{L_p \times D}$, where L_p denotes the prompt length (a user-defined hyperparameter), and D represents the dimensionality of the embeddings.

The L2P method, depicted in Figure 1.1 (right), leverages a Prompt Pool comprising M prompts, denoted as $\mathbf{P} = \{P_i\}_{i=1}^{M}$. Each prompt in the pool is an embedding with the same dimensionality as the output representation of the pre-trained transformer model. To apply L2P, I start by encoding each data instance $\boldsymbol{x}$ using the pre-trained BERT, yielding an output representation called the $query$, denoted as $q(\boldsymbol{x})$. Besides, each prompt in the pool is associated with a prompt key. To construct an enriched representation for the input instance $\boldsymbol{x}$, L2P selects the top N prompts whose prompt keys have the smallest distance to $q(\boldsymbol{x})$. These selected prompts are then concatenated with the embedded input instance $\boldsymbol{x}_e$, leading to the formation of $\boldsymbol{x}_p = [P_e; \boldsymbol{x}_e]$, where P_e represents the set of the chosen top N prompts. Subsequently, the enriched input $\boldsymbol{x}_p$ is forwarded through the pre-trained transformer encoder. This process allows the model to integrate both the original input representation and the relevant information from the selected prompts during the encoding process.

Research in the field has demonstrated that prompt-based learning, as exemplified by the L2P method, surpasses the performance of regular fine-tuning approaches. This approach has proven successful in transfer learning scenarios, particularly when training task-specific prompts, as it facilitates effective knowledge transfer from pre-trained models to downstream tasks.

3.1 Overview

My approach offers a compelling solution to the Continual Relation Extraction (CRE) problem. It revolves around three key steps, which are meticulously outlined in Section 3.2. In the following sections, each of these steps will be thoroughly explored to provide a comprehensive understanding of this groundbreaking method.

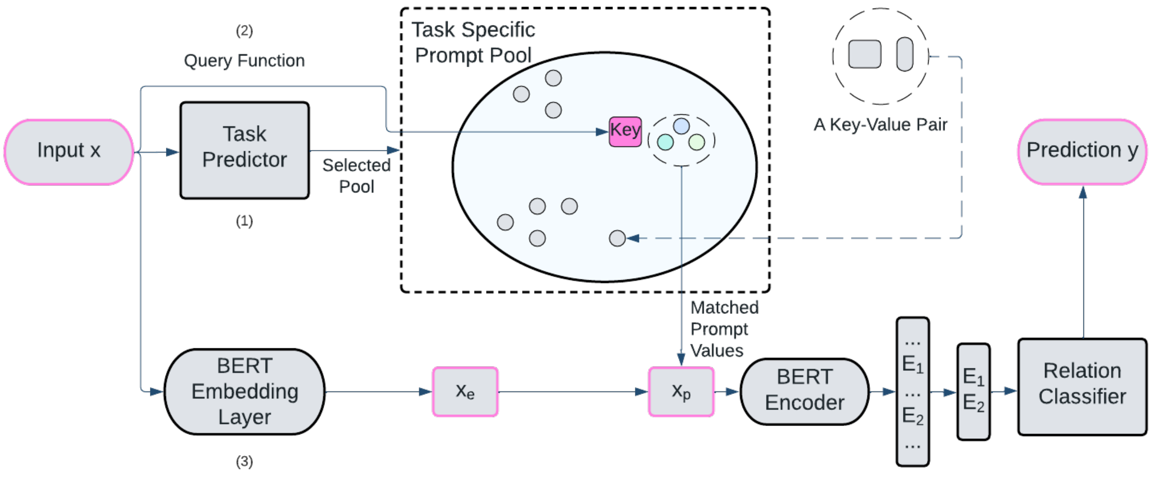

Figure 3.1. Data Flow Diagram: Initially, the Task Predictor predicts the task identity of the input $\boldsymbol{x}$, enabling the selection of the corresponding Prompt Pool. Subsequently, the input $\boldsymbol{x}$ queries this Prompt Pool to identify prompts whose corresponding keys are closest to the $query$ of $\boldsymbol{x}$. The chosen prompts are then prepended to the start of the embedded input $\boldsymbol{x}_e$, creating the prompted input $\boldsymbol{x}_p$. The combined $\boldsymbol{x}_p$ is fed into the Bert Encoder, where the two embeddings corresponding to the positions of the entities E_1 and E_2 are concatenated. Finally, the resulting concatenated embeddings is passed to the Class Classifier, which predicts the relation label y of the input $\boldsymbol{x}$.

First of all, we need to understand the method that I propose to use the Prompt Pool corresponding to each task. The Prompt Pool $\mathbf{P}_k$ corresponds to the task $\mathcal{T}_k$, and it is intended to store information related to task $\mathcal{T}_k$ only. Thus, the samples $\boldsymbol{x}$ of task $\mathcal{T}_k$ instructed by Prompt Pool $\mathbf{P}_k$ are better classified by the classifier.

Figure 3.1 shows how an input sentence $\boldsymbol{x}$ is forwarded through the modules, and the final result is a predicted relation label y corresponding to $\boldsymbol{x}$. This process consists of three steps: (1) First, $\boldsymbol{x}$ is forwarded via the Task Predictor. The Task Predictor is a classifier that helps classify $\boldsymbol{x}$ into the correct corresponding task, thereby selecting the corresponding Prompt Pool for $\boldsymbol{x}$. (2) Second, $\boldsymbol{x}$ queries the selected Prompt Pool from step (1). We can see that this Prompt Pool is created using prompts, which are simply key-value pairs. The query of $\boldsymbol{x}$ to the pool will match and select the prompts whose key is closest to the key of $\boldsymbol{x}$. (3) Finally, $\boldsymbol{x}$ is forwarded through the BERT Embedding Layer, creating embeddings $\boldsymbol{x}_e$. This $\boldsymbol{x}_e$ will be appended to the values of the prompts selected in step (2), forming $\boldsymbol{x}_p$. $\boldsymbol{x}_p$ is then forwarded through the BERT Encoder, extracting two corresponding embeddings of two entities, E1 and E2. These two embeddings are forwarded to the relation classifier, which serves to classify the relation bet ween two entities E1, E2 in the input sentence $\boldsymbol{x}$, and the result is the prediction y.

3.2 Framework

Algorithm 1 $\mathcal{T}_k$ training process

```
Require: Training k-th dataset $\mathcal{D}_k$, current relation set $\mathcal{R}_k$, history relation set
    $\hat{\mathcal{R}}_{k-1}$, task-specific Prompt Pool set $\hat{\mathbf{P}}_{k-1}$, generative model set $\hat{\mathbf{G}}_{k-1}$
Ensure: Task-specific Prompt Pool set $\hat{\mathbf{P}}_k$, generative model set $\hat{\mathbf{G}}_k$, relation clas-
    sifier $ClassHead$, task predictor $TaskHead$
 1: $\mathbf{P}_k$ establish                                   ▷ $k$-th Prompt Pool
 2: for $e_{id} \leftarrow 1$ to $training\_epoch$ do
 3:     for $each$ mini batch $\boldsymbol{x}_B \in \mathcal{D}_k$ do
 4:         Update $\mathbf{P}_k$ and $g_\phi$ by $\nabla L_{CE}$ on $\boldsymbol{x}_B$
 5:     end for
 6: end for
 7: $\hat{\mathcal{R}}_k \leftarrow \hat{\mathcal{R}}_{k-1} \cup \mathcal{R}_k$
 8: $\hat{\mathbf{P}}_k \leftarrow \hat{\mathbf{P}}_{k-1} \cup \mathbf{P}_k$
 9: $\mathbf{G}_{z_k} \leftarrow \emptyset$, $\mathbf{G}_{query_k} \leftarrow \emptyset$                 ▷ $k$-th GMMs set
10: for $each\ r \in \mathcal{R}_k$ do
11:     $\boldsymbol{x}^r \leftarrow \mathcal{D}_k^r$
12:     $\mathbf{G}_{z_k} \cup = \mathbf{GMM}(z(\boldsymbol{x}^r, \mathbf{P}_k))$
13:     $\mathbf{G}_{query_k} \cup = \mathbf{GMM}(query(\boldsymbol{x}^r))$
14: end for
15: $\hat{\mathbf{G}}_{z_k} \leftarrow \hat{\mathbf{G}}_{z_{k-1}} \cup \mathbf{G}_{z_k}$
16: $\hat{\mathbf{G}}_{query_k} \leftarrow \hat{\mathbf{G}}_{query_{k-1}} \cup \mathbf{G}_{query_k}$
17: $\mathcal{D}'_z \leftarrow \emptyset$, $\mathcal{D}'_{query} \leftarrow \emptyset$                  ▷ Pseudo Datasets
18: for $each\ r \in \hat{\mathcal{R}}_k$ do
19:     $\mathcal{D}'_z \leftarrow \mathcal{D}'_z \cup Sample(\mathbf{G}_z^r)$
20:     $\mathcal{D}'_{query} \leftarrow \mathcal{D}'_{query} \cup Sample(\mathbf{G}_{query}^r)$
21: end for
22: for $each$ mini batch $z_B \in \mathcal{D}'_z$ do
23:     Update $ClassHead$ by $\nabla L_{CE}$ on $z_B$
24: end for
25: for $each$ mini batch $query_B \in \mathcal{D}'_{query}$ do
26:     Update $TaskHead$ by $\nabla L_{CE}$ on $query_B$
27: end for
28: Return $\hat{\mathbf{P}}_k, \hat{\mathbf{G}}_k, ClassHead, TaskHead$
```

If Figure 3.1 depicted the fundamental architecture of our approach, the Algorithm 1 here outlines the specific steps of our methodology for generative replay and prompting to enhance CRE. Our approach involves three primary stages:

(1) Prompt Pool learning for a new task (line $1 \sim 8$): When the data for the k-th task arrives, consisting of dataset $\mathcal{D}_k$ and its corresponding relation set $\mathcal{R}_k$, a task-specific Prompt Pool $\mathbf{P}_k$ is initialized and employed to be associated with the k-th task. This Prompt Pool is trained alongside a simple relation classifier g_ϕ to classify the relations in $\mathcal{R}_k$ based on the dataset $\mathcal{D}_k$. Throughout this learning process, the pre-trained BERT model remains frozen, while knowledge pertaining to relations in $\mathcal{R}_k$ is acquired and stored in $\mathbf{P}_k$.

(2) Generative Models (line $9 \sim 16$): For each new relation $r \in \mathcal{R}_k$ trained in Step 1, we utilize two Gaussian Mixture Models (GMMs): G^r_{query} and G^r_z. These models serve to preserve the distribution of high-dimensional representations of all the samples labeled by r. Specifically, G^r_{query} captures the distribution of the unprompted relation representations, while G^r_z learns the distribution of prompted relation representations. We refer to the collections of GMMs representing the k-th task as $\mathrm{G}_k = \{\mathrm{G}_{query_k}, \mathrm{G}_{z_k}\}$. Furthermore, we denote the collections of GMMs learned up to the k-th task as $\hat{\mathrm{G}}_k = \{\hat{\mathrm{G}}_{query_k}, \hat{\mathrm{G}}_{z_k}\}$. These GMMs will be utilized to generate inputs for training classifiers. The purpose of generating the $query$ is to determine the task identity, while the generated z is employed for relation classification, as outlined in Step 3 below.

(3) Training the Task Predictor and Relation Classifier (line $17 \sim 27$): To classify an unlabeled sample $\boldsymbol{x}$, we employ two distinct neural networks: the task predictor, denoted as $TaskHead$, and the relation classifier, referred to as $ClassHead$. The $TaskHead$ utilizes the $query$ representations derived from $\hat{\mathrm{G}}_{query_k}$ as its input and focuses on predicting the task identity $t \in [1, 2, ..., k]$ of $\boldsymbol{x}$. On the other hand, the $ClassHead$ operates on the z representations sampled from $\hat{\mathrm{G}}_{z_k}$ to accurately classify the relation class $r \in \hat{\mathcal{R}}_k$ of the samples.

During the training process, our main focus lies on Steps 1 and 2. Step 3 is specifically utilized for evaluating the model and does not directly impact the training procedure.

3.3 Task-specific Prompt Pool Learning

In Section 2.2.4, we discussed the concept of a Prompt Pool. However, employing a shared Prompt Pool for all tasks can lead to limitations such as forgetfulness or knowledge noise. To overcome these challenges, we propose the implementation of a task-specific Prompt Pool, where each task has its own dedicated Prompt Pool for learning purposes. The Prompt Pool is defined as follows:

$$\mathbf{P} = \{(\boldsymbol{k}_1, P_1), (\boldsymbol{k}_2, P_2), ..., (\boldsymbol{k}_M, P_M)\} \tag{3.1}$$

In this equation, M represents the number of prompts in the pool. Each prompt P_i is assigned a learnable key k_i, where $\boldsymbol{k}_i \in \mathbb{R}^{D_k}$ and D_k denotes the prompt key dimension. The set of all keys is denoted as $\boldsymbol{K} = \{\boldsymbol{k}_i\}_{i=1}^{M}$.

Given a pre-trained language model BERT and a tokenized input sentence $\boldsymbol{x} \in \mathbb{R}^L$, where L represents the length of the input sentence, we follow [3], [44] to insert special tokens, specifically $[E_{11}]/[E_{12}]$ and $[E_{21}]/[E_{22}]$ to indicate the starting and ending positions of the head and tail entities, respectively. It is important to note that the pre-trained language model BERT remains frozen throughout the learning process. Additionally, the token embeddings associated with the special tokens $[E_{11}, E_{12}, E_{21}, E_{22}]$ are also kept frozen after completing the first task in order to ensure consistency.

Afterward, the pre-trained embedding layer f_e projects the tokenized sentence into the embedding feature $\boldsymbol{x}_e = f_e(\boldsymbol{x}) \in \mathbb{R}^{L \times D}$, where D represents the embedding dimension. We can modify the input embedding in the following manner:

$$\boldsymbol{x}_p = [P_{s_1}; P_{s_2}; ...; P_{s_N}; \boldsymbol{x}_e] \tag{3.2}$$

Here, the symbol ; represents concatenation along the token length dimension. We aim to empower the input instance to determine the prompts $\{s_i\}_{i=1}^{N}$ to select through $key \times query$ matching. To achieve this, we employ the entire pre-trained BERT model as a frozen feature extractor. It serves as a query function, denoted as $q : \mathbb{R}^L \to \mathbb{R}^{D_k}$. This function encodes the input $\boldsymbol{x}$ into the same dimension as the prompt keys. Specifically, $q(\boldsymbol{x}) = f_r(\boldsymbol{x}_e)[[E_{11}, E_{21}], :]$, where we extract the feature vectors corresponding to the representation of the start token of the two entities.

Denote $\gamma : \mathbb{R}^{D_k} \times \mathbb{R}^{D_k} \to \mathbb{R}$ as a function that scores the match between the $query$ and prompt key (cosine similarity can be used as it has been proven to work well in experiments). Given an input $\boldsymbol{x}$, we use $q(\boldsymbol{x})$ to lookup the top-N keys by simply solving the objective:

$$\boldsymbol{K}_{\boldsymbol{x}} = \underset{s \in [1,M]}{\operatorname{argmin}} \sum_{i=1}^{N} \gamma(q(\boldsymbol{x}), \boldsymbol{k}_{si}) \tag{3.3}$$

where $\boldsymbol{K}_{\boldsymbol{x}}$ denote a subset of the top-N keys that are specifically chosen for $\boldsymbol{x}$ from the set $\boldsymbol{K}$. A task-specific Prompt Pool allows the customization of prompts based on the requirements and characteristics of each individual task. By tailoring the prompts to the specific task at hand, the model can focus on the relevant information and patterns needed to excel in that particular task. This adaptability enhances the model's performance and improves its learning efficiency for each task.

Optimization Objective for task-specific Prompt Pool Learning: In the learning process, for each new task $\mathcal{T}_k$, a new Prompt Pool $\mathbf{P}_k$ is created. During each training step, following the aforementioned strategy, N prompts are selected, and the corresponding adapted embedding feature, denoted as $\boldsymbol{x}_p$, is inputted to the pre-trained encoder f_r and the final classifier g_ϕ, which is parameterized by ϕ. The objective is to optimize the end-to-end training loss function, which can be summarized as follows:

$$\min_{\mathbf{P}_k,\phi} \mathcal{L}(g_\phi(f_r^C(\boldsymbol{x}_p)), y) + \lambda \sum_{s_i \in \boldsymbol{K}_{\boldsymbol{x}}} \gamma(q(\boldsymbol{x}), \boldsymbol{k}_{s_i}) \tag{3.4}$$

where $f_r^C(\boldsymbol{x}_p) = Concat(f_r(\boldsymbol{x}_p)[[E_{11}, E_{21}], :])$ and $\boldsymbol{K}_{\boldsymbol{x}}$ is obtained with Equation 3.4. In order to provide further clarification, it should be noted that the representation of two special tokens, namely $[E_{11}]$ and $[E_{21}]$, refers to the combined representation of the head and tail entities before being inputted into the classification head g_ϕ. The first term in the Equation 3.4 employs the softmax cross-entropy loss, while the second term represents a loss function that encourages the prompt keys associated with the query features to be closer in proximity. The scaling factor λ is a scalar used to adjust the weight of the loss term.

3.4 Gaussian Mixture Replay for Relation Representation

To effectively retain the acquired knowledge from prior tasks, we employ a technique in which we capture the distributions of observed relations to replay relation samples. For each relation, we store a distribution called $\mathbf{G}_z^r$, which is derived from applying a Gaussian Mixture Model (GMM) to the set $\mathcal{D}_z^r = \{z^r\}$. Here:

$$z^r = f_r(\boldsymbol{x}_p^r)[[E_{11}, E_{21}], :] \tag{3.5}$$

represents the high-dimensional prompted representation of the input $\boldsymbol{x}^r$ belonging to the relation r. Besides z, we have introduced the concept of a $query$ corresponding to each sample $\boldsymbol{x}$ in Section 3.3. Specifically, we utilize:

$$query^r = q(\boldsymbol{x}^r) = f_r(\boldsymbol{x}_e^r)[[E_{11}, E_{21}], :] \tag{3.6}$$

as the representation of the $query$ result for each sample $\boldsymbol{x}^r$ belonging to relation r. Finally, the distribution of the $query$ results corresponding to each of these relations will be saved as G^r_{query}.

This implies that we possess a generative model for every relation, capable of reconstructing the corresponding high-dimensional prompted representation and $query$ representation for each sample belonging to that relation. The reason for utilizing GMM in this context is its minimal memory requirement, which scales linearly with the representation of a sample. It only necessitates the storage of means and variances that represent a cluster component in the mixture model.

3.5 Task Predictor and Relation Classifier

Task Predictor: Since each task we have learned has its own Prompt Pool, as suggested in Section 3.3, we know that the newly established Prompt Pool is learning to serve the sample of current task during training. However, during testing, we need to figure out the which Prompt Pool corresponds to a given sample. For that reason, we propose to learn a task predictor called $TaskHead$. Once the $TaskHead$ is trained, it can be used to predict the task identity of new, unseen samples. Obtaining the task identity also means that the appropriate Prompt Pool has been detected at test time.

We make use of the learned $\hat{\mathrm{G}}_{query_k}$ (Section 3.4) to generate the representation set $query^r$ corresponding to each relation $r \in \hat{\mathcal{R}}_k$, getting the dataset $\mathcal{D}'_{query} = \{\mathcal{D}'^r_{query}\}_{r\in\hat{\mathcal{R}}_k}$, which contains the generated $query$ representations of all samples belonging to the observed relations. This $\mathcal{D}'_{query}$ dataset is used to train the task predictor $TaskHead$, which is a simple **MLP(.)** neural network consisting only of linear layers.

Relation Classifier: In Section 3.3, we noted the existence of the task-specific Prompt Pool set $\hat{\mathbf{P}}_k$ until the k-th task. Additionally, in Section 3.4, we discovered that by using G^r_z, we can generate z^r, which represents the representation of the sample $\boldsymbol{x}^r$ associated with its corresponding Prompt Pool $\mathbf{P}^r$. Expanding upon the concept presented in the $TaskHead$ learning, we can apply the same approach to generate a dataset $\mathcal{D}'_z = \{\mathcal{D}'^r_z\}_{r\in\hat{\mathcal{R}}_k}$ for training a relation classifier called $ClassHead$. $ClassHead$ is a simple model, specifically a **MLP(.)** neural network.

SWAG training: Stochastic Weight Averaging Gaussian (SWAG) [45] is a technique that combines stochastic gradient descent and Bayesian inference to estimate model uncertainty in deep learning. It improves generalization by averaging multiple snapshots of the model's weights during training. Additionally, SWAG incorporates Gaussian sampling, which effectively explores the weight space by estimating the covariance matrix of the weights. By applying SWAG to both the $TaskHead$ and $ClassHead$, we can find enhanced solutions and potentially achieve better performance than a standard **MLP(.)** model.

CHAPTER 4. NUMERICAL RESULTS

4.1 Datasets

We evaluate our methods and all baselines on two datasets: TACRED [46] and FewRel [47].

FewRel. FewRel contains 80 relation types with a total of 56,000 samples. To make it suitable for CRE, we split the dataset into 10 non-overlapping groups, simulating the training of 10 tasks with sequentially arriving data. We follow the configurations outlined in [48] and use the original training set and validation set as the basis for our experiments.

TACRED. The TACRED dataset is an imbalanced dataset for relation extraction (RE), consisting of 42 relations (including the $no\_relation$ class) and a total of 106,264 samples. We adopt the experimental settings proposed by [49], similar to the CRL approach presented in [4].

4.2 Baselines

First, we compare our model with the SOTA rehearsal-free continual learning methods described in Section 2.2.4, specifically L2P [13], DualPrompt [14], and CODA-Prompt [15]. We evaluate our method against these baselines in two scenarios: when they operate completely rehearsal-free and when they utilize a small memory buffer for replay. Since these baselines were originally designed for computer vision tasks, we re-implement them for CRE by using BERT [28] as the backbone pre-trained encoder. Besides, building upon the model architecture as outlined in **CRL** [4], our **MLP(.)** includes three linear layers in both the $TaskHead$ and $ClassHead$. The first two layers serve as feature extractors, while the final layer possesses an output dimension corresponding to the number of relations to be classified. All implementations are carried out using PyTorch.

Additionally, we compare our proposed methods with several established baselines for CRE. **EA-EMR** [48] introduced a technique that combines memory replay and embedding alignment to address catastrophic forgetting. **CML** [51] presented a curriculum-meta learning approach to tackle order-sensitivity and catastrophic forgetting in CRE. **EMAR + BERT** [52] proposed a method based on memory activation and reconsolidation to preserve prior knowledge. **RP-CRE** [49] utilized a memory network to refine sample embeddings with relation prototypes, aiming to prevent catastrophic forgetting. **CRECL** [44] combined a classification network and a prototypical contrastive network to alleviate the problem of

FewRel										
Model	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
L2P	98.6	47.9	36.5	25.7	21.7	19.2	14.1	11.9	14.4	11.6
L2P w/ buffer	98.6	90.3	81.8	78.3	73.5	71.2	68.3	66.9	65.6	62.3
DualPrompt	98.8	49.2	41.4	27.7	20.9	20.2	14.1	12.2	14.0	11.5
DualPrompt w/ buffer	98.8	94.4	91.5	89.8	88.3	85.9	83.5	81.2	79.0	75.3
CODA-Prompt	98.8	52.4	42.4	28.6	24.4	21.6	14.9	12.1	15.1	11.8
CODA-Prompt w/ buffer	**98.8**	94.5	92.0	90.5	89.4	87.8	86.5	85.1	84.0	82.6
Ours	97.9	**95.5**	**93.5**	**92.4**	**91.1**	**89.5**	**88.4**	**87.5**	**86.3**	**84.8**
Ours w/o SWAG	97.7	95.1	93.0	91.7	90.6	89.0	87.7	86.7	85.4	83.8
TACRED										
Model	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
L2P	96.6	40.8	32.4	24.1	19.7	15.4	13.6	9.1	10.8	9.7
L2P w/ buffer	96.6	91.4	86.9	82.0	79.3	74.7	72.7	69.7	68.8	66.6
DualPrompt	96.6	40.0	33.4	23.8	19.9	15.2	13.8	9.6	11.1	11.4
DualPrompt w/ buffer	96.6	92.1	86.1	81.9	79.5	76.0	74.4	72.1	71.6	70.0
CODA-Prompt	95.9	40.8	34.2	24.7	19.9	15.1	14.1	12.1	12.3	12.2
CODA-Prompt w/ buffer	95.9	92.5	87.6	83.7	81.7	79.5	77.4	76.4	75.5	73.9
Ours	**98.4**	**94.3**	**90.2**	**86.2**	**84.7**	**82.6**	**81.3**	**79.7**	**79.5**	**78.3**
Ours w/o SWAG	98.3	94.2	90.0	86.0	84.5	82.1	80.5	78.6	78.4	77.3

Table 4.1: Performance of our methods (%) on all observed relations at each stage of learning, in comparison with SOTA rehearsal-free Continual Learning baselines. All results are generated from our implementations. Regarding the baselines which use a buffer, the buffer size is 10 samples from each relation type.

catastrophic forgetting. **CRL** [4] employed a contrastive replay mechanism and knowledge distillation to maintain learned knowledge. **EMAR+ACA** [5] improves **CRE** by using a data augmentation mechanism to enhance the robustness of the learned model. **CRE-DAS** [6] employed memory-insensitive relation prototypes and memory augmentation to overcome overfitting; they also introduced integrated training and focal knowledge distillation to enhance performance on analogous relations. **CDec+ACA** [50] proposed a classifier decomposition framework to address representation biases by promoting robust representation learning while preserving previous knowledge.

4.3 Experimental Results

Rehearsal-free CRE. As mentioned earlier, Table 4.1 demonstrates that the current SOTA prompt-based methods for rehearsal-free continual learning produce significantly inferior results in CRE. This provides strong evidence that, at present, these methods are unable to effectively mitigate catastrophic forgetting across all domains. Without a strategy to mitigate forgetting for the shared parameters, these models are susceptible to excessive plasticity.

When we introduce a memory buffer to assist them with the reinforcement of old

FewRel										
Model	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
DualPrompt	0.0	64.1	0.0	0.0	0.0	0.0	57.3	65.0	0.0	0.0
DualPrompt w/ buffer	0.0	71.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**88.8**
Ours	**88.5**	**86.2**	**86.9**	86.4	**85.2**	**86.9**	**87.7**	**86.0**	**85.4**	82.5
Ours w/o SWAG	86.9	85.4	84.1	**86.5**	84.7	86.0	86.6	84.2	84.6	82.5
TACRED										
Model	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
DualPrompt	7.4	0.0	63.0	69.0	0.0	51.1	0.0	0.0	0.0	2.5
DualPrompt w/ buffer	0.0	0.0	0.0	0.0	17.7	0.0	0.0	**81.3**	0.0	61.3
Ours	**83.8**	**79.5**	**84.2**	**76.2**	**80.4**	**74.8**	**79.7**	69.3	**83.7**	81.5
Ours w/o SWAG	83.7	77.5	83.6	73.6	79.9	74.4	77.8	68.4	83.0	**83.4**

Table 4.2: Task prediction precision for each task ($\%$) at testing time after training the 10-th task, in comparison with DualPrompt. All results are generated from our implementations. Regarding the baselines which use a buffer, the buffer size is 10 samples from each relation type.

knowledge, their performances improve considerably. However, since these methods lack more nuanced strategies specifically to CRE, their performances still lag behind the SOTA CRE methods shown in Table 4.3. In comparison, our method outperforms all rehearsal-free baselines for continual learning in the realm of CRE by substantial margins. Without SWAG, our method achieves final accuracies of 83.8% and 77.3% on FewRel and TACRED, respectively. These results are 3.4% and 2.9% higher than the best outcomes achieved among the baselines. Incorporating SWAG into our method further enhances its performance, yielding an additional 1% increase in final accuracy for both FewRel and TACRED.

Furthermore, as we have also mentioned earlier, the $key \times query$ mechanism for task prediction (i.e., prompt selection) can be inaccurate. To prove this, we will compare the task prediction accuracy of our method against that of DualPrompt. We chose DualPrompt because they dedicate a separate prompt for each task during training, which makes it simple to evaluate their prompt selection during testing time. The results are shown in Table 4.2. DualPrompt's strategy for task-specific prompt selection is to find the prompt whose key is closest to the query of the input instance and to pull that key closer to the query by integrating the $key \times query$ similarity into the loss function. This means the key of a task will eventually converge to the task-mean unsupervised representation. Since these representations are outputs of a generic pre-trained model, it is unrealistic to expect the representations of the tasks to be well-separated. As a result, this prompt selection method has resulted in atrocious accuracy. Our method, which uses a dedicated task predictor trained on synthesized query representations, achieves much better task prediction accuracy.

FewRel										
Model	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
EA-EMR	89.0	69.0	59.1	54.2	47.8	46.1	43.1	40.7	38.6	35.2
CML	91.2	74.8	68.2	58.2	53.7	50.4	47.8	44.4	43.1	39.7
EMAR+BERT	**98.8**	89.1	89.5	85.7	83.6	84.8	79.3	80.0	77.1	73.8
RP-CRE	97.9	92.7	91.6	89.2	88.4	86.8	85.1	84.1	82.2	81.5
CRECL	98.0	94.7	92.4	90.7	89.4	87.1	85.9	85.0	84.0	82.1
CRL	98.2	94.6	92.5	90.5	89.4	87.9	86.9	85.6	84.5	83.1
EMAR+ACA	98.3	95.0	92.6	91.3	90.4	89.2	87.6	87.0	86.3	84.7
CRE-DAS	98.1	**95.8**	**93.6**	91.9	91.1	89.4	88.1	86.9	85.6	84.2
CDec+ACA	98.4	95.4	93.2	92.1	91.0	**89.7**	88.3	87.4	**86.4**	84.8
Ours	97.9	95.5	93.5	**92.4**	**91.1**	89.5	**88.4**	**87.5**	86.3	**84.8**

TACRED										
Model	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
EA-EMR	47.5	40.1	38.3	29.9	24	27.3	26.9	25.8	22.9	19.8
CML	57.2	51.4	41.3	39.3	35.9	28.9	27.3	26.9	24.8	23.4
EMAR+BERT	96.6	85.7	81	78.6	73.9	72.3	71.7	72.2	72.6	71.0
RP-CRE	97.6	90.6	86.1	82.4	79.8	77.2	75.1	73.7	72.4	72.4
CRECL	97.3	93.6	90.5	86.1	84.6	82.1	79.4	77.6	77.9	77.4
CRL	97.7	93.2	89.8	84.7	84.1	81.3	80.2	79.1	79.0	78.0
EMAR+ACA	98.0	92.1	90.6	85.5	84.4	82.2	80.0	78.6	78.8	78.1
CRE-DAS	97.7	94.3	**92.3**	**88.4**	**86.6**	**84.5**	**82.2**	**81.1**	**80.1**	**79.1**
CDec+ACA	97.7	92.8	91.0	86.7	85.2	82.9	80.8	80.2	78.8	78.6
Ours	**98.4**	**94.3**	90.2	86.2	84.7	82.6	81.3	79.7	79.5	78.3

Table 4.3: Performance of our methods (%) on all observed relations at each stage of learning, in comparison with SOTA CRE baselines. The results of the baselines are directly taken from [50] and [6]

General CRE. Regarding the current most successful CRE baselines, the results in Table 4.3 demonstrate how our method compares to them. Even without retaining training data or directly fine-tuning BERT, our method still produces nearly equivalent results to the current SOTA baselines on both datasets. On FewRel, our proposed method achieves SOTA accuracy at most stages of training. Meanwhile, on TACRED, the highest result is only 0.8% higher than ours. This observation highlights the importance and effectiveness of our contributions, enabling a rehearsal-free CRE model like ours to achieve comparable, and in some cases, better performance compared to SOTA rehearsal-based methods.

CHAPTER 5. CONCLUSIONS

5.1 Summary

In this thesis, we have addressed several key problems in the context of rehearsal-free Continual Relation Extraction (CRE) and proposed an innovative framework to enhance continual learning techniques. Our contributions centered on representation generation for replay, accurate task prediction, and optimizing within-task and cross-task variability for prompting. Through empirical benchmarks, we have demonstrated the effectiveness of our model, surpassing the current state-of-the-art prompt-based baselines for continual learning and achieving competitive results compared to the most advanced CRE methods in general.

Our first contribution involved the introduction of a novel prompting-based method tailored specifically for Continual Relation Extraction. By associating each task with a separate Prompt Pool during training, we ensured that individual task relations were represented distinctly, while also accounting for within-task variances. This approach facilitated effective adaptation to new relations without forgetting previous ones, leading to improved performance in continual learning scenarios.

We also addressed the issue of catastrophic forgetting on shared parameters, such as the shared Prompt Pool and classifier, which has been a challenge in existing rehearsal-free methods. To overcome this, we harnessed the power of a generative model to synthesize latent representations of data for replay. This approach avoided the need for explicit data retention while allowing us to control the quantity of rehearsal data effectively. By using the synthesized prompted representations for training the relation classification module, we achieved significant improvements in mitigating catastrophic forgetting on shared parameters.

Furthermore, we enhanced the quality of the sub-module responsible for selecting task-specific prompts by introducing an additional generative model. This model acquired unprompted pseudo-representations, which were then used to train the task predictor. The traditional approach in rehearsal-free methods relied solely on the similarity of representations for prompt selection, leading to potential inaccuracies. Incorporating a supervised training process using unprompted pseudo-representations improved the accuracy of task-specific prompt selection and overall performance.

In conclusion, our research has made substantial progress in the field of Continual Relation Extraction, offering a more privacy-friendly, efficient, and accurate approach to continual learning in real-world applications. By addressing the limitations of existing rehearsal-free methods and introducing novel strategies for representation generation, accurate task prediction, and prompt optimization, we have advanced the state-of-the-art in continual learning for relation extraction. We believe that our contributions will inspire further research in this domain and lead to more robust and adaptable solutions for lifelong learning systems in various practical settings.

5.2 Limitations

Despite the promising results and effectiveness demonstrated by our proposed methods in mitigating the catastrophic problem, this research has certain limitations that need to be acknowledged.

Firstly, Knowledge Retention Challenge: Similar to previous works in the field, our approach struggles with retaining knowledge of past tasks. Despite incorporating improvements like utilizing Prompt Pools to store task-specific knowledge, the issue of forgetfulness still persists, especially when these pools are not appropriately determined at test time. This limitation hinders the full realization of long-term knowledge retention, potentially impacting the model's ability to generalize across diverse tasks effectively.

Secondly, Dependency on Explicit Knowledge: As mentioned in [48], our model's current reliance on the provision of explicit knowledge regarding the placement of entities within the sentence poses a significant limitation. This dependency restricts the applicability of our approach in scenarios where entity positions are not readily available or require extensive manual annotation, hindering its usability in real-world applications where such information may not be easily accessible or practical to obtain.

Lastly, Limited Generative Model Exploration: Another limitation of our research lies in the use of a specific generative model, namely Gaussian Mixture Models (GMMs). While GMMs have proven effective in certain cases, our study lacks exploration into alternative generative methods. This limitation might overlook potential improvements and efficiencies that could be achieved through the adoption of more advanced or tailored generative models for the given tasks.

It is important to recognize and address these limitations to provide a comprehensive understanding of the potential applicability and generalizability of our proposed methods. Future research should focus on tackling these challenges to further enhance the capabilities of the proposed approach in real-world applications.

5.3 Suggestions for Future Work

Future work should focus on addressing the identified limitations and enhancing the overall capabilities of the approach. Improving the determination of task identity to mitigate forgetfulness and better retain knowledge of previously encountered entities is a priority. Exploring deterministic models as an alternative to the current probabilistic approach could lead to more robust solutions for entity handling, reducing uncertainty and improving overall performance. Additionally, experimenting with various generative models beyond GMMs, such as Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), and Transformer-based models like GPT-3, may yield more accurate entity handling solutions. These efforts aim to push the boundaries of the approach and contribute to the advancement of entity handling in natural language processing, achieving more efficient and effective solutions.

REFERENCE

[1] C. Xiong, R. Power and J. Callan, "Explicit semantic ranking for academic search via knowledge graph embedding," in *Proceedings of the 26th International Conference on World Wide Web*, ser. WWW '17, Perth, Australia: International World Wide Web Conferences Steering Committee, 2017, pp. 1271–1279, ISBN: 9781450349130. DOI: `10.1145/3038912.3052558`. [Online]. Available: `https://doi.org/10.1145/3038912.3052558`.

[2] D. Sorokin and I. Gurevych, "Context-aware representations for knowledge base relation extraction," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 1784–1789. DOI: `10.18653/v1/D17-1188`. [Online]. Available: `https://aclanthology.org/D17-1188`.

[3] L. Baldini Soares, N. FitzGerald, J. Ling and T. Kwiatkowski, "Matching the blanks: Distributional similarity for relation learning," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 2895–2905. DOI: `10.18653/v1/P19-1279`. [Online]. Available: `https://aclanthology.org/P19-1279`.

[4] K. Zhao, H. Xu, J. Yang and K. Gao, "Consistent representation learning for continual relation extraction," in *Findings of the Association for Computational Linguistics: ACL 2022*, Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 3402–3411. DOI: `10.18653/v1/2022.findings-acl.268`. [Online]. Available: `https://aclanthology.org/2022.findings-acl.268`.

[5] P. Wang, Y. Song, T. Liu *et al.*, "Learning robust representations for continual relation extraction via adversarial class augmentation," 2022.

[6] W. Zhao, Y. Cui and W. Hu, "Improving continual relation extraction by distinguishing analogous semantics," in *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics)*, 2023.

[7] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Advances in neural information processing systems*, vol. 30, 2017.

[8] H. Shin, J. K. Lee, J. Kim and J. Kim, "Continual learning with deep generative replay," *Advances in neural information processing systems*, vol. 30, 2017.

[9] A. Chaudhry, M. Ranzato, M. Rohrbach and M. Elhoseiny, “Efficient lifelong learning with a-gem,” in *International Conference on Learning Representations (ICLR)*, 2019.

[10] J. S. Smith, J. Tian, S. Halbe, Y.-C. Hsu and Z. Kira, *A closer look at rehearsal-free continual learning*, 2023. arXiv: `2203.17269 [cs.LG]`.

[11] Z. Wang, Y. Liu, T. Ji *et al.*, “Rehearsal-free continual language learning via efficient parameter isolation,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 10 933–10 946. [Online]. Available: `https://aclanthology.org/2023.acl-long.612`.

[12] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi and G. Neubig, *Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing*, 2021. arXiv: `2107.13586 [cs.CL]`.

[13] Z. Wang, Z. Zhang, C.-Y. Lee *et al.*, “Learning to prompt for continual learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 139–149.

[14] Z. Wang, Z. Zhang, S. Ebrahimi *et al.*, “Dualprompt: Complementary prompting for rehearsal-free continual learning,” *European Conference on Computer Vision*, 2022.

[15] J. S. Smith, L. Karlinsky, V. Gutta *et al.*, “Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 11 909–11 919.

[16] D. A. Roberts, S. Yaida and B. Hanin, *The Principles of Deep Learning Theory*. Cambridge University Press, May 2022. DOI: `10.1017/9781009023405`. [Online]. Available: `https://doi.org/10.1017%2F9781009023405`.

[17] A. Torfi, R. A. Shirvani, Y. Keneshloo, N. Tavaf and E. A. Fox, *Natural language processing advancements by deep learning: A survey*, 2021. arXiv: `2003.01200 [cs.CL]`.

[18] S. Jamil, M. J. Piran and O.-J. Kwon, *A comprehensive survey of transformers for computer vision*, 2022. arXiv: `2211.06004 [cs.CV]`.

[19] L. Wang, X. Zhang, H. Su and J. Zhu, *A comprehensive survey of continual learning: Theory, method and application*, 2023. arXiv: `2302.00487 [cs.LG]`.

[20] S. Sodhani, M. Faramarzi, S. V. Mehta *et al.*, *An introduction to lifelong supervised learning*, 2022. arXiv: `2207.04354 [cs.LG]`.

[21] R. Aljundi, M. Rohrbach and T. Tuytelaars, *Selfless sequential learning*, 2019. arXiv: 1806.05421 [stat.ML].

[22] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov and J. van de Weijer, *Class-incremental learning: Survey and performance evaluation on image classification*, 2022. arXiv: 2010.15277 [cs.LG].

[23] J. Kirkpatrick, R. Pascanu, N. Rabinowitz *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017. DOI: 10.1073/pnas.1611835114. [Online]. Available: https://doi.org/10.1073%2Fpnas.1611835114.

[24] M. S. Hossain, P. Saha, T. F. Chowdhury, S. Rahman, F. Rahman and N. Mohammed, *Rethinking task-incremental learning baselines*, 2022. arXiv: 2205.11367 [cs.AI].

[25] Y. Dai, H. Lang, Y. Zheng, B. Yu, F. Huang and Y. Li, *Domain incremental lifelong learning in an open world*, 2023. arXiv: 2305.06555 [cs.CL].

[26] D.-W. Zhou, Q.-W. Wang, Z.-H. Qi, H.-J. Ye, D.-C. Zhan and Z. Liu, *Deep class-incremental learning: A survey*, 2023. arXiv: 2302.03648 [cs.CV].

[27] G. M. Van de Ven and A. S. Tolias, "Three scenarios for continual learning," *arXiv preprint arXiv:1904.07734*, 2019.

[28] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. [Online]. Available: https://aclanthology.org/N19-1423.

[29] T. Mikolov, K. Chen, G. Corrado and J. Dean, *Efficient estimation of word representations in vector space*, 2013. arXiv: 1301.3781 [cs.CL].

[30] J. Pennington, R. Socher and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. [Online]. Available: https://aclanthology.org/D14-1162.

[31] X. Song, A. Salcianu, Y. Song, D. Dopson and D. Zhou, "Linear-time wordpiece tokenization," *ArXiv*, vol. abs/2012.15524, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:229924282.

[32] X. Song, A. Salcianu, Y. Song, D. Dopson and D. Zhou, *Fast wordpiece tokenization*, 2021. arXiv: 2012.15524 [cs.CL].

[33] C. Hu, D. Yang, H. Jin, Z. Chen and Y. Xiao, "Improving continual relation extraction through prototypical contrastive learning," in *Proceedings of the 29th International Conference on Computational Linguistics*, Gyeongju, Republic of Korea: International Committee on Computational Linguistics, Oct. 2022, pp. 1885–1895. [Online]. Available: https://aclanthology.org/2022.coling-1.163.

[34] H. Zhang, B. Liang, M. Yang, H. Wang and R. Xu, "Prompt-based prototypical framework for continual relation extraction," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 2801–2813, 2022. DOI: 10.1109/TASLP.2022.3199655.

[35] B. Ji, J. Yu, S. Li *et al.*, "Span-based joint entity and relation extraction with attention-based span-specific and contextual semantic representations," in *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 88–99. DOI: 10.18653/v1/2020.coling-main.8. [Online]. Available: https://aclanthology.org/2020.coling-main.8.

[36] J. Wang and W. Lu, "Two are better than one: Joint entity and relation extraction with table-sequence encoders," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online: Association for Computational Linguistics, Nov. 2020, pp. 1706–1721. DOI: 10.18653/v1/2020.emnlp-main.133. [Online]. Available: https://aclanthology.org/2020.emnlp-main.133.

[37] T. Shin, Y. Razeghi, R. L. L. I. au2, E. Wallace and S. Singh, *Autoprompt: Eliciting knowledge from language models with automatically generated prompts*, 2020. arXiv: 2010.15980 [cs.CL].

[38] Y. Wei, T. Mo, Y. Jiang, W. Li and W. Zhao, *Eliciting knowledge from pretrained language models for prototypical prompt verbalizer*, 2022. arXiv: 2201.05411 [cs.CL].

[39] K. Zhou, J. Yang, C. C. Loy and Z. Liu, "Conditional prompt learning for vision-language models," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[40] A. Webson and E. Pavlick, *Do prompt-based models really understand the meaning of their prompts?* 2022. arXiv: 2109.01247 [cs.CL].

[41] B. Lester, R. Al-Rfou and N. Constant, “The power of scale for parameter-efficient prompt tuning,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3045–3059. DOI: 10.18653/v1/2021.emnlp-main.243. [Online]. Available: https://aclanthology.org/2021.emnlp-main.243.

[42] X. Liu, K. Ji, Y. Fu *et al.*, “P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 61–68. DOI: 10.18653/v1/2022.acl-short.8. [Online]. Available: https://aclanthology.org/2022.acl-short.8.

[43] Z. Wu, S. Wang, J. Gu *et al.*, “IDPG: An instance-dependent prompt generation method,” in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 5507–5521. DOI: 10.18653/v1/2022.naacl-main.403. [Online]. Available: https://aclanthology.org/2022.naacl-main.403.

[44] C. Hu, D. Yang, H. Jin, Z. Chen and Y. Xiao, “Improving continual relation extraction through prototypical contrastive learning,” in *Proceedings of the 29th International Conference on Computational Linguistics*, Gyeongju, Republic of Korea: International Committee on Computational Linguistics, Oct. 2022, pp. 1885–1895. [Online]. Available: https://aclanthology.org/2022.coling-1.163.

[45] W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov and A. G. Wilson, “A simple baseline for bayesian uncertainty in deep learning,” *Advances in neural information processing systems*, vol. 32, 2019.

[46] Y. Zhang, V. Zhong, D. Chen, G. Angeli and C. D. Manning, “Position-aware attention and supervised data improve slot filling,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, 2017, pp. 35–45. [Online]. Available: https://nlp.stanford.edu/pubs/zhang2017tacred.pdf.

[47] X. Han, H. Zhu, P. Yu *et al.*, “FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language*

Processing, Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 4803–4809. DOI: 10.18653/v1/D18-1514. [Online]. Available: https://aclanthology.org/D18-1514.

[48] H. Wang, W. Xiong, M. Yu, X. Guo, S. Chang and W. Y. Wang, "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)*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 796–806. DOI: 10.18653/v1/N19-1086. [Online]. Available: https://aclanthology.org/N19-1086.

[49] L. Cui, D. Yang, J. Yu *et al.*, "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)*, Online: Association for Computational Linguistics, Aug. 2021, pp. 232–243. DOI: 10.18653/v1/2021.acl-long.20. [Online]. Available: https://aclanthology.org/2021.acl-long.20.

[50] H. Xia, P. Wang, T. Liu, B. Lin, Y. Cao and Z. Sui, "Enhancing continual relation extraction via classifier decomposition," in *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics)*, 2023.

[51] T. Wu, X. Li, Y.-F. Li *et al.*, "Curriculum-meta learning for order-robust continual relation extraction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, pp. 10 363–10 369.

[52] X. Han, Y. Dai, T. Gao *et al.*, "Continual relation learning via episodic memory activation and reconsolidation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, Jul. 2020, pp. 6429–6440. DOI: 10.18653/v1/2020.acl-main.573. [Online]. Available: https://aclanthology.org/2020.acl-main.573.

APPENDIX

A. PROMPT-BASED CONTINUAL LEARNING

A.1 Dualprompt and Coda-Prompt

DualPrompt [14] improves upon L2P by having a dedicated prompt for each task during training, besides a task-agnostic prompt for every instance. The prompts is then either passed through the transformer encoder either by directly prepending them to the input embedding as in L2P; or via a technique named prefix-tuning. In prefix-tuning, prompts can be pass to several multi-head self-attention layers in the pre-trained transformer encoder; the choice of which layers to be prompted is a hyper-parameter. A prompt $P \in \mathbb{R}^{L_p \times D}$ is split into $P_k, P_v \in \mathbb{R}^{\frac{L_p}{2} \times D}$. We denote $\mathbf{H} \in \mathbb{R}^{L \times D}$ as the input matrix, where each line represents the embedding of a corresponding token. Normally, the calculation process in an arbitrary multi-head self-attention (MSA) layer is formulated as:

$$\begin{aligned}&\mathrm{MSA}(\mathbf{H}) = \mathrm{Concat}(h_1, ..., h_m)W_0\\ &\text{where } h_i = \mathrm{Attention}(\mathbf{H}W_i^Q, \mathbf{H}W_i^K, \mathbf{H}W_i^V),\end{aligned}$$

where W_0 , W_i^Q, W_i^K, and W_i^V are projection matrices and m is the number of heads [15]. In prefix-tuning, each head h_i calculation is modified by prepending the prompts into $\mathbf{H}$ before projecting them with W_i^K and W_i^V, as follows:

$$h_i = \mathrm{Attention}(\mathbf{H}W_i^Q, [P_k; \mathbf{H}]W_i^K, [P_v; \mathbf{H}]W_i^V).$$

[14] claimed in their paper that prompt tuning resulted in better outcomes.

Meanwhile, CODA-Prompt [15] also leverages prompts via prefix-tuning. Instead of using a pool of prompts, they use a pool of *prompt components* $\mathbf{P}$. Each prompt component $P_i \in \mathbb{R}^{L_p \times D}$ has its corresponding attention vector $\mathbf{A}_i \in \mathbb{R}^{L_p \times D}$, and prompt key $\mathbf{K}_i \in \mathbb{R}^{L_p \times D}$. $\mathbf{P}$, $\mathbf{A}$, and $\mathbf{K}$ are all learn-able parameters. The final prompt to be used with each instance $\boldsymbol{x}$ is a linear combination of the *prompt components*, $P_{\boldsymbol{x}} = \sum_{i=1}^{M} \alpha_{i\boldsymbol{x}} P_i$, where:

$$\begin{aligned}\alpha_{\boldsymbol{x}} &= \gamma(q(\boldsymbol{x}) \odot \mathbf{A}, \mathbf{K})\\ &= \{\gamma[q(\boldsymbol{x}) \odot \mathbf{A}_1, \mathbf{K}_1], ..., \gamma[q(\boldsymbol{x}) \odot \mathbf{A}_M, \mathbf{K}_M]\}.\end{aligned}$$

B. REPRODUCIBILITY CHECKLIST

B.1 Checklist

- **Source code with specifications of all dependencies, including external libraries**: My source code with necessary documentation for reproducibility will be released upon acceptance of the paper.
- **Description of computing infrastructure used**: In this work, we utilized a single NVIDIA A100 Tensor Core GPU with 80GB memory operated by the Google Colab Server for all experiments. PyTorch 2.0.1 and Huggingface Transformer 4.30.1 are used to implement the models.
- **Average runtime for each approach**: The training of my proposed model on the FewRel dataset took approximately 7 hours, while for the TACRED dataset, it took approximately 3 hours. Moving on to the testing phase, we observed that the model's evaluation time was approximately 20 minutes for the FewRel dataset and 3 minutes for the TACRED dataset in total.
- **Number of parameters in the model**: my proposed model has in total 114M parameters. As we freeze the BERT model, the number of learnable parameters is thus only 3.8M.
- **Explanation of evaluation metrics used, with links to code**: We use the same performance measures (average F1-scores on 5 permutations of task orders) as in prior work [4] for fair comparison.
- **Bounds for each hyper-parameter**: To tune the proposed model, we choose $gmm\_num\_components$ from [1, 3, 5], $encoder\_epochs$ from [10, 20, 50], $prompt\_pool\_lr$ from [2×10^{-5}, 5×10^{-5}, 1×10^{-4}], $classifier\_epochs$ from [100, 300, 500]. All other hyper-parameters will be set as default.
- **The method of choosing hyper-parameter values and the criterion used to select among them**: We tune the hyper-parameters for the proposed model using a random search. All the hyper-parameters are selected based on the F1 scores on both the FewRel and TACRED development sets, as each dataset has its own set of optimal hyper-parameters.