Continual Relation Learning via Episodic Memory Activation and Reconsolidation

Xu Han[1*], **Yi Dai**[1*], **Tianyu Gao**[1], **Yankai Lin**[2],
Zhiyuan Liu[1†], **Peng Li**[2], **Maosong Sun**[1], **Jie Zhou**[2]
[1] State Key Lab on Intelligent Technology and Systems,
Institute for Artificial Intelligence,
Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]Pattern Recognition Center, WeChat AI, Tencent Inc., China
{hanxu17,daiy17,gty16}@mails.tsinghua.edu.cn
{liuzy,sms}@mail.tsinghua.edu.cn
{yankailin,patrickpli,withtomzhou}@tencent.com

Abstract

Continual relation learning aims to continually train a model on new data to learn incessantly emerging novel relations while avoiding catastrophically forgetting old relations. Some pioneering work has proved that storing a handful of historical relation examples in episodic memory and replaying them in subsequent training is an effective solution for such a challenging problem. However, these memory-based methods usually suffer from overfitting the few memorized examples of old relations, which may gradually cause inevitable confusion among existing relations. Inspired by the mechanism in human long-term memory formation, we introduce episodic memory activation and reconsolidation (EMAR) to continual relation learning. Every time neural models are activated to learn both new and memorized data, EMAR utilizes relation prototypes for memory reconsolidation exercise to keep a stable understanding of old relations. The experimental results show that EMAR could get rid of catastrophically forgetting old relations and outperform the state-of-the-art continual learning models. The code and datasets are released on https://github.com/thunlp/ContinualRE.

1 Introduction

Relation extraction aims at detecting relations between entities from text, e.g., extracting the relation "*the president of*" from the given sentence "*Newton served as the president of the Royal Society*", which could serve as external resource for various downstream applications (Dong et al., 2015; Xiong et al., 2017; Schlichtkrull et al., 2018). The conventional RE methods (Riedel et al., 2013; Zeng et al., 2014; Lin et al., 2016) mostly focus on recognizing relations for a fixed pre-defined relation set, and cannot handle rapidly emerging novel relations in the real world.

Some researchers therefore explore to detect and learn incessantly emerging relations in an open scenario. As shown in Figure 1, their efforts can be formulated into a two-step pipeline: (1) **Open Relation Learning** extracts phrases and arguments to construct patterns of specific relations, and then discovers unseen relation types by clustering patterns, and finally expands sufficient examples of new relation types from large-scale textual corpora; (2) **Continual Relation Learning** continually uses those expanded examples of new relations to train an effective classifier. The classifier is trained on a sequence of tasks for handling both existing and novel relations, where each task has its own relation set. Although continual relation learning is vital for learning emerging relations, there are rare explorations for this field.

A straightforward solution is to store all historical data and re-train models every time new relations and examples come in. Nevertheless, it is computationally expensive since relations are in sustainable growth. Moreover, the huge example number of each relation makes frequently mixing new and old examples become infeasible in the real world. Therefore, storing all data is not practical in continual relation learning. In view of this, the recent preliminary work (Wang et al., 2019) indicates that the main challenge of continual relation learning is the catastrophic forgetting problem, i.e., it is hard to learn new relations and meanwhile avoid forgetting old relations, considering memorizing all the data is almost impossible.

[*] indicates equal contribution
[†] Corresponding author

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6429–6440
July 5 - 10, 2020.

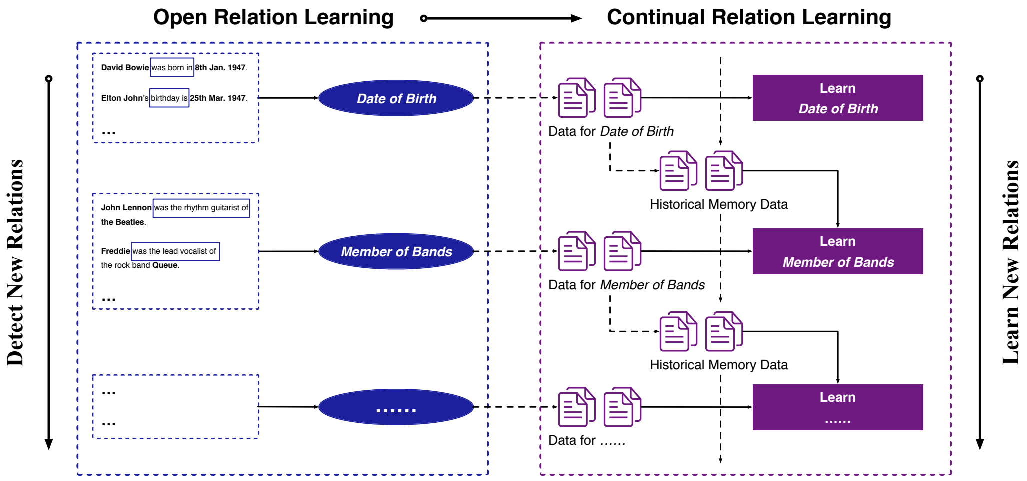


Figure 1: The whole pipeline to detect and learn new relations in an open scenario.

Recent work (Shin et al., 2017; Kemker and Kanan, 2018; Chaudhry et al., 2019) has shown that the memory-based approaches, maintaining episodic memory to save a few training examples in old tasks and re-training memorized examples during training new tasks, are one of the most effective solutions to the catastrophic forgetting problem, especially for continual learning in NLP scenarios (Wang et al., 2019; d'Autume et al., 2019). However, existing memory-based models still suffer from an overfitting problem: when adapting them for continual relation learning, they may frequently change feature distribution of old relations, gradually overfit a few examples in memory, and finally become confused among old relations after long-term training.

In fact, these memory-based methods are similar to long-term memory model of mammalian memory in neuroscience (McClelland et al., 1995; Bontempi et al., 1999). Although researchers in neuroscience are not clear about secrets inside the human brain, they reach a consensus that the formation of long-term memory relies on continually replaying and consolidating information (Tononi and Cirelli, 2006; Boyce et al., 2016; Yang et al., 2014), corresponding to the episodic memory and memory replay in continual learning models. Yet later work (Nader et al., 2000; Lee et al., 2004; Alberini, 2005) in neuroscience indicates that reactivation of consolidated memory triggers a reconsolidation stage to continually maintain memory, and memory is easy to be changed or erased in this stage. To apply some reconsolidation exercises can help memory go through this stage and keep long-term memory stable. Intuitively, the existing memory-based models seem like continual memory activation without reconsolidation exercises, and thus become sensitive and volatile.

Inspired by the reconsolidation mechanism in human long-term memory formation, we introduce episodic memory activation and reconsolidation (EMAR) to continual relation learning in this paper. More specifically, when training models on new relations and their examples, we first adopt memory replay to activate neural models on examples of both new relations and memory, and then utilize a special reconsolidation module to let models avoid excessively changing and erasing feature distribution of old relations. As the core of relation learning is to grasp relation prototypes rather than rote memorization of relation examples, our reconsolidation module requires models to be able to distinguish old relation prototypes after each time memory is replayed and activated. As compared with pioneering explorations to improve episodic memory replay (Chaudhry et al., 2019; Wang et al., 2019), with toughly keeping feature distribution of old relations invariant, EMAR is more flexible in feature spaces and powerful in remembering relation prototypes.

We conduct sufficient experiments on several RE datasets, and the results show that EMAR effectively alleviates the catastrophic forgetting problem and significantly outperforms the state-of-the-art continual learning models. Further experiments and analyses indicate the reasons for the effectiveness of EMAR, proving that it can utilize a few examples in old tasks to reconsolidate old relation prototypes and keep better distinction among old relations after long-term training.

2 Related Work

The conventional RE work, including both supervised RE models (Zelenko et al., 2003; Zhou et al., 2005; Gormley et al., 2015; Socher et al., 2012; Liu et al., 2013; Zeng et al., 2014; Nguyen and Grishman, 2015; dos Santos et al., 2015; Xu et al., 2015; Liu et al., 2015; Miwa and Bansal, 2016) and distantly supervised models (Bunescu and Mooney, 2007; Mintz et al., 2009; Riedel et al., 2010; Hoffmann et al., 2011; Zeng et al., 2015; Lin et al., 2016; Han et al., 2018a; Baldini Soares et al., 2019), focuses on extracting predefined relations from text. Yet in the real world, new relations are rapidly emerging, and it is impossible to train models with a fixed dataset once to cover all relations. Hence, some researchers pay their attention to relation learning in various open scenarios, in order to detect and learn relations without pre-defined relation sets. As we introduced before, learning incessantly emerging relations consists of two important steps: open relation learning and continual relation learning.

There have been many efforts for open relation learning, including pattern extraction (Banko et al., 2007; Fader et al., 2011; Mausam et al., 2012; Del Corro and Gemulla, 2013; Angeli et al., 2015; Petroni et al., 2015; Stanovsky and Dagan, 2016; Mausam, 2016; Cui et al., 2018), relation discovery (Yao et al., 2011; Marcheggiani and Titov, 2016), relation clustering (Shinyama and Sekine, 2006; Elsahar et al., 2017; Wu et al., 2019), and data collection (Riloff et al., 1999; Etzioni et al., 2005; Pantel and Pennacchiotti, 2006; Rozenfeld and Feldman, 2008; Nakashole et al., 2011; Zhu et al., 2009; Gao et al., 2020). However, for continual relation learning, there are still only some preliminary explorations for it. Following continual learning setting[1] (Ring, 1994; Thrun and Pratt, 2012) in machine learning, Wang et al. (2019) first explore continual relation learning.

Existing continual learning methods focus on three research directions: (1) consolidation-based methods (Kirkpatrick et al., 2017; Zenke et al., 2017; Li and Hoiem, 2017; Liu et al., 2018; Ritter et al., 2018) which consolidate the model parameters important to previous tasks and reduce their learning weights; (2) dynamic architecture methods (Chen et al., 2016; Rusu et al., 2016; Fernando et al., 2017) which dynamically expand model architectures to learn new tasks and effectively prevent forgetting old tasks. Yet model size growing dramatically with increasing tasks makes these methods unsuitable for NLP applications; (3) memory-based methods (Lopez-Paz and Ranzato, 2017; Rebuffi et al., 2017; Shin et al., 2017; Kemker and Kanan, 2018; Aljundi et al., 2018; Chaudhry et al., 2019) remember a few examples in old tasks and continually learn them with emerging new tasks to alleviate catastrophic forgetting. Among these methods, the memory-based methods have been proven to be the most promising for NLP tasks, including both relation learning (Wang et al., 2019) and other NLP tasks (d'Autume et al., 2019; Sun et al., 2019).

Inspired by reconsolidation in human memory formation, we introduce episodic memory activation and reconsolidation (EMAR) to alleviate the overfitting problem of the existing memory-based methods and better learn relations continually.

3 Methodology

3.1 Task Definition and Overall Framework

Continual relation learning trains models on a sequence of tasks, where the k-th task has its own training set $\mathcal{T}_k$, validation set $\mathcal{V}_k$, and query set $\mathcal{Q}_k$. Each set of the k-th task, e.g. $\mathcal{T}_k = \{(x_1^{\mathcal{T}_k}, y_1^{\mathcal{T}_k}), \ldots, (x_N^{\mathcal{T}_k}, y_N^{\mathcal{T}_k})\}$, consists of a series of examples and their corresponding relation labels, where N is the example number of $\mathcal{T}_k$. Each example $x_i^{\mathcal{T}_k}$ and its label $y_i^{\mathcal{T}_k}$ indicate that $x_i^{\mathcal{T}_k}$ can express the relation $y_i^{\mathcal{T}_k} \in \mathcal{R}_k$, where $\mathcal{R}_k$ is the relation set of the k-th task.

More specifically, models will be trained on $\mathcal{T}_k$ at the k-th step to learn the new relations in $\mathcal{R}_k$. As relations are emerging and accumulating, continual relation learning requires models to perform well on both the k-th task and previous $k-1$ tasks. Hence, after training on $\mathcal{T}_k$, models will be evaluated on $\tilde{\mathcal{Q}}_k = \bigcup_{i=1}^{k} \mathcal{Q}_i$, and required to classify each query example into the all known relation set $\tilde{\mathcal{R}}_k = \bigcup_{i=1}^{k} \mathcal{R}_i$. Therefore, the evaluation will be more and more difficult with the growth of tasks.

For handling the catastrophic forgetting in continual relation learning, an episodic memory module $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots\}$ is set to store a few examples of historical tasks, each memory module $\mathcal{M}_k = \{(x_1^{\mathcal{M}_k}, y_1^{\mathcal{M}_k}), \ldots, (x_B^{\mathcal{M}_k}, y_B^{\mathcal{M}_k})\}$ stores several examples and labels that come from $\mathcal{T}_k$, where $(x_i^{\mathcal{M}_k}, y_i^{\mathcal{M}_k}) \in \mathcal{T}_k$ and B is the constrained memory size for each task.

As shown in Figure 2, when models are trained

[1] Some work names it lifelong or incremental learning.

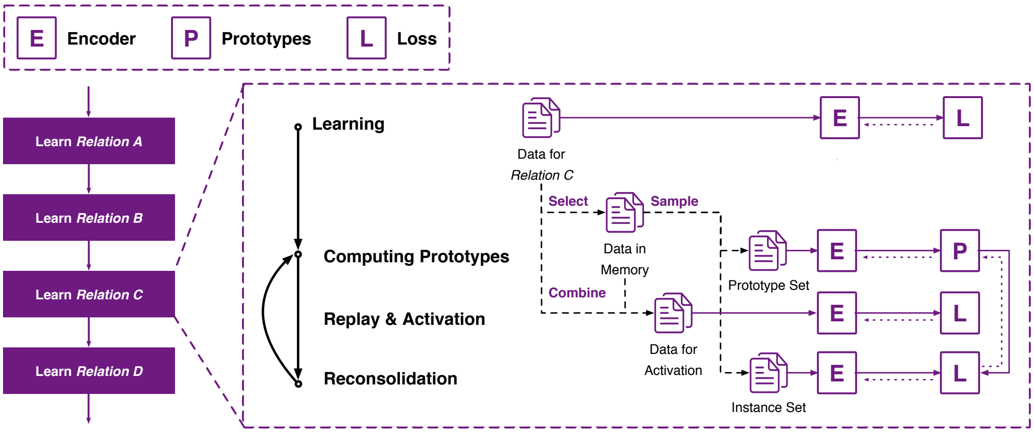


Figure 2: A simple example of continually learning four tasks (each task has only one relation: A, B, C, D respectively) to demonstrate the overall framework of episodic memory activation and reconsolidation during continual relation learning. The purple solid lines and dotted lines represent the forward and backward propagation respectively. The black dotted lines represent the data flow.

on the k-th task, our framework includes several steps to learn new relations and meanwhile avoid forgetting old relations: (1) First (Section 3.3), we fine-tune the example encoder on the training set $\mathcal{T}_k$ of the k-th task to let the model be aware of new relation patterns. (2) Second (Section 3.4), for each relation in the k-th relation set $\mathcal{R}_k$, we select its informative examples and store the examples into the episodic memory $\mathcal{M}_k$. (3) Finally (Section 3.5), we iteratively adopt memory replay and activation as well as memory reconsolidation to learn new relation prototypes while strengthening distinguishing old relation prototypes.

Besides, we will introduce how to train models as well as predict relations for query examples in Section 3.6. As the example encoder is used in all other steps, we first introduce it in Section 3.2 before other steps.

3.2 Example Encoder

Given an example x, we adopt an example encoder to encode its semantic features for detecting and learning relations. To be specific, we first tokenize the given example into several tokens, and then input the tokenized tokens into neural networks to compute its corresponding embedding. As extracting relations from sentences is related to those entities mentioned in sentences, we thus add special tokens into the tokenized tokens to indicate the beginning and ending positions of those entities. For simplicity, we denote such an example encoding operation as the following equation,

$$\boldsymbol{x} = f(x), \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^d$ is the semantic embedding of x, and d is the embedding dimension. Note that the encoder is not our focus in this paper, we select bidirectional long short-term memory (BiLSTM) (Bengio et al., 1994) as representative encoders to encode examples. In fact, other neural text encoders like convolutional neural networks (Zeng et al., 2014) and pre-trained language models (Devlin et al., 2019) can also be adopted as example encoders.

3.3 Learning for New Tasks

When the k-th task is arising, the example encoder has not touched any examples of new relations before, and cannot extract the semantic features of them. Hence, we first fine-tune the example encoder on $\mathcal{T}_k = \{(x_1^{\mathcal{T}_k}, y_1^{\mathcal{T}_k}), \ldots, (x_N^{\mathcal{T}_k}, y_N^{\mathcal{T}_k})\}$ to grasp new relation patterns in $\mathcal{R}_k$. The loss function of learning the k-th task is as follows,

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) = -\sum_{i=1}^{N}\sum_{j=1}^{|\tilde{\mathcal{R}}_k|} \delta_{y_i^{\mathcal{T}_k}=r_j} \times \\ \log \frac{\exp(g(f(x_i^{\mathcal{T}_k}), \boldsymbol{r}_j))}{\sum_{l=1}^{|\tilde{\mathcal{R}}_k|}\exp(g(f(x_i^{\mathcal{T}_k}), \boldsymbol{r}_l))},\end{aligned} \tag{2}$$

where $\boldsymbol{r}_j$ is the embedding of the j-th relation $r_j \in \tilde{\mathcal{R}}_k$ in the all known relation set $\tilde{\mathcal{R}}_k$, $g(\cdot, \cdot)$ is the function to compute similarities between embeddings (e.g. cosine similarity), and $\boldsymbol{\theta}$ is the parameters that can be optimized, including the example encoder parameters and relation embeddings. If $y_i^{\mathcal{T}_k}$ equals r_j, $\delta_{y_i^{\mathcal{T}_k}=r_j} = 1$, otherwise

$\delta_{y_i^{\mathcal{T}_k}=r_j} = 0$. For each new relation, we first randomly initialize its embedding and then optimize Eq. (2).

3.4 Selecting Examples for Memory

After several epochs of learning for new tasks with Eq. (2), we store a few examples from $\mathcal{T}_k$ into the memory $\mathcal{M}_k$. More specifically, we select informative and diverse examples from $\mathcal{T}_k$ to cover new relation patterns as much as possible, which can make the memory effectively approximate the feature distribution of relations.

After encoding all examples of the k-th task $\mathcal{T}_k$ into $\{\boldsymbol{x}_1^{\mathcal{T}_k}, \dots, \boldsymbol{x}_N^{\mathcal{T}_k}\}$, we apply K-Means to cluster these example embeddings, where the number of clusters is the memory size B. Then, for each cluster, we select the example closest to the cluster centroid and record which relation these selected examples belong to. We denote this selected example set $\mathcal{C}_k$. By counting the example number in $\mathcal{C}_k$ for each relation, we can describe the relation importance in this task: more selected examples of a relation indicates more importance. As the limited memory size, for those more important relations, we select at least $\lfloor \frac{B}{|\mathcal{R}_k|} \rfloor$ examples, yet for those less important ones, we select at most $\lceil \frac{B}{|\mathcal{R}_k|} \rceil$ examples. If a relation does not have enough examples to fill its allocated memory, this memory will be re-allocated for other relations.

For each relation, we also use K-Means to cluster its own examples, and the number of current clusters is its allocated example number in the memory. For each cluster, we select the example closest to the cluster centroid, and store this example into the memory $\mathcal{M}_k$.

3.5 Replay, Activation and Reconsolidation

After fine-tuning the example encoder for $\mathcal{T}_k$ and selecting informative examples for $\mathcal{M}_k$, we iteratively adopt **computing prototypes**, **memory replay and activation**, and **memory reconsolidation** to strengthen identifying new relation patterns and keep distinguishing old relation patterns.

Computing Prototypes

By combining all examples in the episodic memory, we achieve the whole memory set $\tilde{\mathcal{M}}_k = \bigcup_{i=1}^{k} \mathcal{M}_i$. As we aim to grasp relation prototypes rather than rote memorization of relation examples, for each known relation $r_i \in \tilde{\mathcal{R}}_k$, we sample a prototype set $\mathcal{P}_i = \{x_1^{\mathcal{P}_i}, \dots, x_{|\mathcal{P}_i|}^{\mathcal{P}_i}\}$, where each example $x_i^{\mathcal{P}_i}$ comes from $\tilde{\mathcal{M}}_k$ and its label equals r_i, and compute its prototype embedding,

$$\boldsymbol{p}_i = \frac{\sum_{j=1}^{|\mathcal{P}_i|} f(x_j^{\mathcal{P}_i})}{|\mathcal{P}_i|}, \tag{3}$$

where $\boldsymbol{p}_i$ is the relation prototype embedding of $r_i \in \tilde{\mathcal{R}}_k$.

Memory Replay and Activation

In memory replay and activation, the whole memory set $\tilde{\mathcal{M}}_k$ and the k-th training set $\mathcal{T}_k$ will be combined into an activation set $\mathcal{A}_k = \tilde{\mathcal{M}}_k \cup \mathcal{T}_k = \{(x_1^{\mathcal{A}_k}, y_1^{\mathcal{A}_k}), \dots, (x_M^{\mathcal{A}_k}, y_M^{\mathcal{A}_k})\}$ to continually activate models to learn new relations and remember old relations, where M is the total example number of both $\tilde{\mathcal{M}}_k$ and $\mathcal{T}_k$. The loss function is

$$\begin{aligned}\mathcal{L}^{\mathcal{A}}(\boldsymbol{\theta}) = -\sum_{i=1}^{M}\sum_{j=1}^{|\tilde{\mathcal{R}}_k|} \delta_{y_i^{\mathcal{A}_k}=r_j} \times \\ \log \frac{\exp(g(f(x_i^{\mathcal{A}_k}), \boldsymbol{r}_j))}{\sum_{l=1}^{|\tilde{\mathcal{R}}_k|} \exp(g(f(x_i^{\mathcal{A}_k}), \boldsymbol{r}_l))}.\end{aligned} \tag{4}$$

Memory Reconsolidation

As we mentioned before, just conducting memory replay and activation will lead to the overfitting problem, and in the end, models only remember a handful of memorized examples after long-term training. Meanwhile, the core of learning relations is to grasp relation prototypes rather than rote memorization of relation examples. Hence, every time conducting memory replay and activation to grasp both new and old relations, we adopt a memory reconsolidation module to strengthen this process, which seems like conducting reconsolidation exercises to keep long-term memory stable in the human brain.

For each known relation $r_i \in \tilde{\mathcal{R}}_k$, we sample its instance set $\mathcal{I}_i = \{x_1^{\mathcal{I}_i}, \dots, x_{|\mathcal{I}_i|}^{\mathcal{I}_i}\}$ as is similar to sampling $\mathcal{P}_i$, where each example $x_i^{\mathcal{I}_i} \in \mathcal{I}_i$ also comes from $\tilde{\mathcal{M}}_k$ and its label equals r_i. The loss function of the memory reconsolidation is

$$\begin{aligned}\mathcal{L}^{\mathcal{R}}(\boldsymbol{\theta}) = -\sum_{i=1}^{|\tilde{\mathcal{R}}_k|}\sum_{j=1}^{|\mathcal{I}_i|} \\ \log \frac{\exp(g(f(x_j^{\mathcal{I}_i}), \boldsymbol{p}_i))}{\sum_{l=1}^{|\tilde{\mathcal{R}}_k|} \exp(g(f(x_j^{\mathcal{I}_i}), \boldsymbol{p}_l))},\end{aligned} \tag{5}$$

where $\boldsymbol{p}_l$ is the relation prototype embedding of $r_l \in \tilde{\mathcal{R}}_k$ computed by Eq. (3).

Algorithm 1 Train EMAR for the k-th task

```
Require: The training set $\mathcal{T}_k$ of the $k$-th task
Require: The emerging relation set $\mathcal{R}_k$ of the $k$-th task
Require: The memory module $\tilde{\mathcal{M}}_{k-1}$ before learning $\mathcal{T}_k$
Require: The known relation set $\tilde{\mathcal{R}}_{k-1}$ before learning $\mathcal{T}_k$
Initialize the relation embeddings for $\mathcal{R}_k$
$\tilde{\mathcal{R}}_k \leftarrow \tilde{\mathcal{R}}_{k-1} \cup \mathcal{R}_k$
for $i \leftarrow 1$ to $epoch_1$ do
    Update $\boldsymbol{\theta}$ with $\nabla\mathcal{L}$ on $\mathcal{T}_k$
end for
Select informative examples from $\mathcal{T}_k$ to store into $\mathcal{M}_k$
$\tilde{\mathcal{M}}_k \leftarrow \tilde{\mathcal{M}}_{k-1} \cup \mathcal{M}_k$
$\mathcal{A}_k \leftarrow \tilde{\mathcal{M}}_k \cup \mathcal{T}_k$
for $i \leftarrow 1$ to $epoch_2$ do
    for relation $r_j \in \tilde{\mathcal{R}}_k$ do
        Sample $\mathcal{P}_j$ from $\tilde{\mathcal{M}}_k$ and compute its relation prototype embedding $\boldsymbol{p}_j$
    end for
    for $j \leftarrow 1$ to $iter_1$ do
        Update $\boldsymbol{\theta}$ with $\nabla\mathcal{L}^{\mathcal{A}}$ on $\mathcal{A}_k$
    end for
    for $j \leftarrow 1$ to $iter_2$ do
        Sample $\mathcal{I}_i$ from $\tilde{\mathcal{M}}_k$ for each known relation $r_i$
        Update $\boldsymbol{\theta}$ with $\nabla\mathcal{L}^{\mathcal{R}}$ on $\{\mathcal{I}_1, \ldots, \mathcal{I}_{|\tilde{\mathcal{R}}_k|}\}$
    end for
end for
```

3.6 Training and Prediction

For training the k-th task, we first use $\mathcal{L}(\boldsymbol{\theta})$ to optimize parameters for several epochs. Then, we select examples for the memory, and iteratively optimize parameters with $\mathcal{L}^{\mathcal{A}}(\boldsymbol{\theta})$ and $\mathcal{L}^{\mathcal{R}}(\boldsymbol{\theta})$ until convergence. More details about the training process are shown in Algorithm 1.

After finishing the k-th task, for each known relation $r_i \in \tilde{\mathcal{R}}_k$, we collect all its memorized examples $\mathcal{E}_i = \{x_1^{\mathcal{E}_i}, \ldots, x_S^{\mathcal{E}_i}\}$ in the whole memory $\tilde{\mathcal{M}}_k$, where S is the example number of r_i in the memory, and compute final relation prototype for prediction,

$$\tilde{\boldsymbol{p}}_i = \frac{\boldsymbol{r}_i + \sum_{j=1}^{S} f(x_j^{\mathcal{E}_i})}{1+S}, \tag{6}$$

where $\boldsymbol{r}_i$ is the relation embedding of r_i used in Eq. (2) and Eq. (4). For each query example x in $\tilde{\mathcal{Q}}_k$, we define its score function for the relation r_i:

$$s(x, r_i) = g(f(x), \tilde{\boldsymbol{p}}_i), \tag{7}$$

where $\tilde{\boldsymbol{p}}_i$ is the final prototype of the relation r_i computed by Eq. (6). Finally, the prediction y for the query x is calculated by:

$$y = \arg\max_{r_i \in \tilde{\mathcal{R}}_k} s(x, r_i). \tag{8}$$

	FewRel		SimpleQ		TACRED	
	W	A	W	A	W	A
Lower Bound	18.9	20.8	63.2	56.9	12.3	9.5
EWC	27.1	30.2	67.2	59.0	14.5	14.5
GEM	49.2	59.8	84.1	79.6	-	-
AGEM	36.1	42.5	77.6	72.2	15.7	16.0
EMR	51.0	62.0	85.2	80.8	28.7	35.6
EA-EMR	56.6	67.3	**87.8**	82.4	30.5	40.5
EMAR	**66.0**	**77.9**	85.2	**83.7**	**44.5**	**54.4**
Upper Bound	81.9	85.8	88.9	84.1	74.3	77.0

Table 1: Accuracy (%) of models on three benchmarks. "W" stands for the **W**hole performance, and "A" stands for the **A**verage performance. The results of FewRel and SimpleQ come from Wang et al. (2019). The result of TACRED comes from our implemented models.

4 Experiments

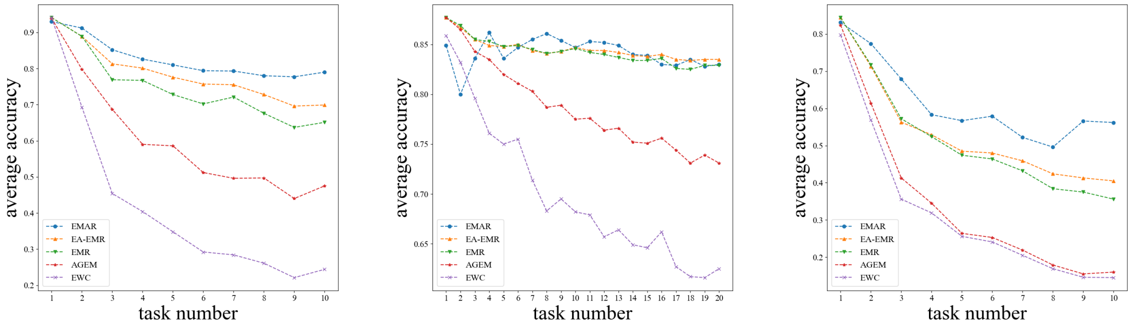
4.1 Datasets

We carry out our experiments on three benchmark datasets:

(1) **FewRel** (Han et al., 2018b). FewRel is a RE dataset that contains 80 relations and $56,000$ examples in total. We follow the settings from Wang et al. (2019) to make FewRel a continual learning benchmark: FewRel is split into 10 clusters of relations, leading to 10 tasks and each relation just belongs to only one task. Each example in these tasks is related to a relation and a candidate set of 10 randomly selected relations for evaluation.

(2) **SimpleQuestions** (Bordes et al., 2015). SimpleQuestions (SimpleQ) is a knowledge base question answering dataset that contains $108,442$ questions, and Yu et al. (2017) construct a relation detection dataset based on it, where questions are linked to relations. Like FewRel, we follow the settings from Wang et al. (2019): SimpleQ is split into 20 clusters of relations to construct 20 tasks. As each question in SimpleQ has been related to a candidate set for evaluation, we do not randomly sample candidate sets again for SimpleQ.

(3) **TACRED** (Zhang et al., 2017). TACRED is a RE dataset that contains 42 relations and $21,784$ examples. Similar to FewRel, we also split TACRED into 10 clusters of relations to construct 10 tasks, and randomly sample candidate relation sets consisting of 10 relations for each examples. Considering there is a special relation "n/a" (not available) in TACRED, we filter out these examples with the relation "n/a" and use the left examples for continual TACRED.



(a) FewRel (b) SimpleQuestions (c) TACRED

Figure 3: Changes in accuracy (%) with increasing tasks through the continual learning process.

	FewRel						SimpleQ						TACRED					
	10		25		50		10		25		50		10		25		50	
	W	A	W	A	W	A	W	A	W	A	W	A	W	A	W	A	W	A
EWC	-	-	-	-	21.3	24.4	-	-	-	-	63.9	62.5	-	-	-	-	14.5	14.5
AGEM	29.0	34.0	33.8	39.0	41.2	47.5	69.1	66.1	72.2	69.2	76.2	73.1	14.7	14.5	15.0	15.5	15.7	16.0
EMR	42.0	54.1	49.0	60.5	53.6	65.1	81.5	77.4	84.9	81.0	86.9	82.9	21.8	26.5	25.7	31.6	28.7	35.6
EA-EMR	49.0	61.2	54.9	66.4	59.1	69.9	**83.3**	78.7	**86.4**	**82.0**	**87.9**	83.5	23.0	30.0	27.7	37.0	30.5	40.5
EMAR	**53.8**	**69.1**	**62.5**	**74.9**	**66.0**	**77.9**	80.9	**78.7**	84.6	81.4	85.2	**83.7**	**31.0**	**36.3**	**37.8**	**48.5**	**44.5**	**54.4**

Table 2: Accuracy (%) of models with different memory sizes. All the results come from our implemented models.

4.2 Experimental Settings

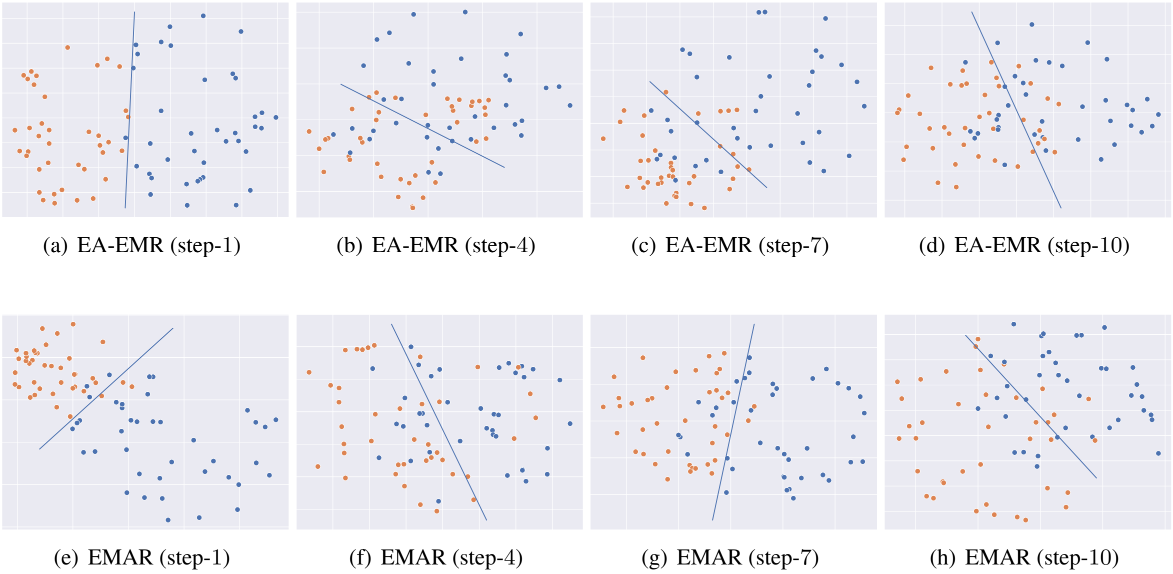
We use two evaluation settings including **whole performance**, which calculates the accuracy on the whole test set of all tasks, and **average performance**, which averages the accuracy on all seen tasks. After having seen all tasks, we use the final whole performance and average performance to evaluate the overall performance of continual relation learning. As average performance highlights the performance of handling catastrophic problem, and thus it is the main metric to evaluate models.

As the task sequence has influence on final model performance, we implement the baseline models by ourselves based on the toolkit[2] released by Wang et al. (2019). For fair comparison, we unify the random seeds in our experiments completely consistent with the seeds in Wang et al. (2019), so that the task sequence can be completely consistent with Wang et al. (2019). For other settings, such as hidden embedding dimension and pre-trained input embeddings, we also follow the settings in Wang et al. (2019).

4.3 Baselines

We evaluate our model and several baselines on the benchmarks, and select two theoretical models to measure the lower and upper bounds: (1) **Lower Bound**, which continually fine-tunes models for each new task without memorizing any historical examples; (2) **Upper Bound**, which remembers all examples in history and continually re-train models with all data. In fact, this model serves as the ideal upper bound for the performance of continual relation learning; (3) **EWC** (Kirkpatrick et al., 2017), which adopts elastic weight consolidation to add special L_2 regularization on parameter changes. Then, EWC uses Fisher information to measure the parameter importance to old tasks, and slow down the update of those parameters important to old tasks; (4) **EMR** (Parisi et al., 2019), a basic memory-based method, which memorizes a few historical examples and simply conduct memory replay. Every time a new task comes in, EMR mixes memorized examples and new examples together to fine-tune models; (5) **GEM** (Lopez-Paz and Ranzato, 2017), an extension of EMR, which adds a constraint on directions of new gradients to make sure that optimization directions do not conflict with gradients on old tasks; (6) **AGEM**

[2] https://github.com/hongwang600/Lifelong_Relation_Detection


(a) EA-EMR (step-1) (b) EA-EMR (step-4) (c) EA-EMR (step-7) (d) EA-EMR (step-10)

(e) EMAR (step-1) (f) EMAR (step-4) (g) EMAR (step-7) (h) EMAR (step-10)

Figure 4: A visualization of features learnt by EA-EMR and EMAR at different training steps on FewRel. For each image, we use the support vector machine to acquire its best linear boundary and draw it as the blue line.

	Step-1	Step-4	Step-7	Step-10
EA-EMR	**98.8**	65.0	78.8	73.8
EMAR	92.5	**75.0**	**87.5**	**80.0**

Table 3: Classification accuracy (%) based on the features learnt by EA-EMR and EMAR in Figure 4.

(Chaudhry et al., 2019), the extension of GEM, which takes the gradient on sampled memorized examples from memory as the only constraint on the optimization directions of the current task; (7) **EA-EMR** (Wang et al., 2019), which introduces memory replay and embedding aligned mechanism to enhance previous tasks and mitigate the embedding distortion when trained on new tasks. EA-EMR is also an extension of EMR, and the state-of-the-art on continual relation learning.

4.4 Overall Results

Table 1 shows the overall performance on three benchmarks under two different settings. From the table, we can see that (1) our proposed EMAR significantly outperforms other baselines and achieves state-of-the-arts almost in all settings. On the SimpleQ dataset, the performance of EMAR is close to EA-EMR and EMR. The reason is perhaps that the SimpleQ benchmark is over simple (even the weakest Lower Bound achieves relatively high results close to Upper Bound). On other benchmarks, EMAR outperforms all the baseline models with a large margin, showing the superiority of our proposed episodic memory activation and reconsolidation mechanism. (2) There is still a huge gap between our model and the upper bound. It indicates there remains lots of things to be explored in continual relation learning.

To further investigate how accuracy changes while learning new tasks, we show the average performance of models at each step in Figure 3. As shown in the figure, we can observe that: (1) With increasing numbers of tasks, the performance of all the models decreases in some degree. This indicates that catastrophically forgetting old relations is inevitable, and it is indeed one of the major difficulty for continual relation learning. (2) The memory-based methods significantly outperform the consolidation-based method, which demonstrates the memory-based methods could alleviate the problem of catastrophic forgetting to some extent. (3) Our proposed EMAR achieves a much better results compared to state-of-the-art model EA-EMR. It shows the effectiveness of our memory reconsolidation, and further indicates understanding relation prototypes is more important and reasonable than rote memorization of examples.

4.5 Effect of Memory Size

Memory size indicates the number of remembered examples for each task. In this section, we investigate the effect of memory size for the performance of baselines and our proposed model. We compare three memory sizes: 10, 25 and 50. As existing work does not report the results with different memory size, we re-implement baseline models by ourselves in this experiment. The results are shown in Table 2. We can find that: (1) With the increasing memory size, the performance of all models improves respectively, which shows that the memory size is one of the key factor determining the performance of continual relation learning models. (2) On both FewRel and TACRED, our EMAR keeps performing the best under different memory sizes, and even achieves comparable results with other models of larger memory sizes. It indicates adopting relation prototypes in EMAR is a more effective way to utilize memory compared with existing memory-based methods.

4.6 Effect of Prototypes and Reconsolidation

To show the effectiveness of prototypes and reconsolidation, we give a case study demonstrating the changing of feature spaces learnt by EA-EMR and EMAR (ours). We sample two relations from the training set and 40 examples per relation from the test set. Then we train EA-EMR and EMAR with the sampled training data respectively and visualize the changes of the sampled 40 instances in the feature spaces at different steps.

From Figure 4, we can see that EMAR learns better features of instances after multi-step training: the embedding space of EMAR is more sparse and features from two relations are more distinguishable. On the other hand, the features learnt by EA-EMR become more dense with increasing steps, thus harder to classify.

This phenomenon is mainly due to the different approaches of constraining features used by EA-EMR and EMAR. The L_2 regularization used in EA-EMR for keeping the instance distribution of old relations leads to higher density in the feature space and smaller distances between different relations after several training steps. On the contrary, EMAR avoids models from forgetting previous relations by relation prototypes. Compared with EA-EMR, using prototypes for reconsolidation is a more flexible constraint, allowing EMAR to utilize larger feature spaces for representing examples and prototypes.

To quantitatively analyze the case, we use the support vector machine to acquire linear boundaries for each image in Figure 4 and list the classification results in Table 3. The quantitative results in the table show that embeddings learnt by EMAR achieve better classification performance, which further supports our above observations.

5 Conclusion and Future Work

To alleviate catastrophically forgetting old relations in continual relation learning, we introduce episodic memory activation and reconsolidation (EMAR), inspired by the mechanism in human long-term memory formation. Compared with existing memory-based methods, EMAR requires models to understand the prototypes of old relations rather than to overfit a few specific memorized examples, which can keep better distinction among relations after long-term training. We conduct experiments on three benchmarks in relation extraction and carry out extensive experimental results as well as empirical analyses, showing the effectiveness of EMAR on utilizing memorized examples. For future work, how to combine open relation learning and continual relation learning together to complete the pipeline for emerging relations still remains a problem, and we will continue to work on it.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (No. 2018YFB1004503) and the National Natural Science Foundation of China (NSFC No. 61732008, 61772302). Tianyu Gao is supported by 2019 Tencent Rhino-Bird Elite Training Program and Tsinghua University Initiative Scientific Research Program.

References

Cristina M Alberini. 2005. Mechanisms of memory stabilization: are consolidation and reconsolidation similar or distinct processes? *Trends in neurosciences*, 28(1):51–56.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of ECCV*, pages 139–154.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of ACL-IJCNLP*, pages 344–354.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of ACL*, pages 2895–2905.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of IJCAI*, pages 2670–2676.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Bruno Bontempi, Catherine Laurent-Demir, Claude Destrade, and Robert Jaffard. 1999. Time-dependent reorganization of brain circuitry underlying long-term memory storage. *Nature*, 400(6745):671.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*.

Richard Boyce, Stephen D Glasgow, Sylvain Williams, and Antoine Adamantidis. 2016. Causal evidence for the role of rem sleep theta rhythm in contextual memory consolidation. *Science*, 352(6287):812–816.

Razvan Bunescu and Raymond Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of ACL*, pages 576–583.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019. Efficient lifelong learning with a-gem. In *Proceedings of ICLR*.

Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. 2016. Net2Net: Accelerating learning via knowledge transfer. In *Proceedings of ICLR*.

Lei Cui, Furu Wei, and Ming Zhou. 2018. Neural open information extraction. In *Proceedings of ACL*, pages 407–413.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. In *Proceedings of NIPS*.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: Clause-based open information extraction. In *Proceedings of WWW*, pages 355–366.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. Question answering over Freebase with multi-column convolutional neural networks. In *Proceedings of ACL-IJCNLP*, pages 260–269.

Hady Elsahar, Elena Demidova, Simon Gottschalk, Christophe Gravier, and Frederique Laforest. 2017. Unsupervised open relation extraction. In *Proceedings of ESWC*, pages 12–16.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *AI*, 165:91–134.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of EMNLP*, pages 1535–1545.

Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. 2017. PathNet: Evolution channels gradient descent in super neural networks. *ai.google*.

Tianyu Gao, Xu Han, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2020. Neural snowball for few-shot relation learning. In *Proceedings of AAAI*.

Matthew R Gormley, Mo Yu, and Mark Dredze. 2015. Improved relation extraction with feature-rich compositional embedding models. In *Proceedings of EMNLP*, pages 1774–1784.

Xu Han, Pengfei Yu, Zhiyuan Liu, Maosong Sun, and Peng Li. 2018a. Hierarchical relation extraction with coarse-to-fine grained attention. In *Proceedings of EMNLP*, pages 2236–2245.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018b. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of EMNLP*, pages 4803–4809.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of ACL*, pages 541–550.

Ronald Kemker and Christopher Kanan. 2018. Fearnet: Brain-inspired model for incremental learning. In *Proceedings of ICLR*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of NAS*, pages 3521–3526.

Jonathan LC Lee, Barry J Everitt, and Kerrie L Thomas. 2004. Independent cellular processes for hippocampal memory consolidation and reconsolidation. *Science*, 304(5672):839–843.

Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *TPAMI*, 40(12):2935–2947.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of ACL*, pages 2124–2133.

ChunYang Liu, WenBo Sun, WenHan Chao, and Wanxiang Che. 2013. Convolution neural network for relation extraction. In *Proceedings of ICDM*, pages 231–242.

Xialei Liu, Marc Masana, Luis Herranz, Joost Van de Weijer, Antonio M Lopez, and Andrew D Bagdanov. 2018. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In *Proceedings of ICPR*, pages 2262–2268.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and WANG Houfeng. 2015. A dependency-based neural network for relation classification. In *Proceedings of ACL-IJCNLP*, pages 285–290.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Proceedings of NIPS*, pages 6467–6476.

Diego Marcheggiani and Ivan Titov. 2016. Discrete-state variational autoencoders for joint discovery and factorization of relations. *TACL*, 4:231–244.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of EMNLP-CoNLL*, pages 523–534.

Mausam Mausam. 2016. Open information extraction systems and downstream applications. In *Proceedings of IJCAI*, pages 4074–4077.

James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. 1995. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL-IJCNLP*, pages 1003–1011.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of ACL*, pages 1105–1116.

Karim Nader, Glenn E Schafe, and Joseph E Le Doux. 2000. Fear memories require protein synthesis in the amygdala for reconsolidation after retrieval. *Nature*, 406(6797):722.

Ndapandula Nakashole, Martin Theobald, and Gerhard Weikum. 2011. Scalable knowledge harvesting with high precision and high recall. In *Proceedings of WSDM*, pages 227–236.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the NAACL Workshop on Vector Space Modeling for NLP*, pages 39–48.

Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of COLING*, pages 113–120.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*.

Fabio Petroni, Luciano Del Corro, and Rainer Gemulla. 2015. CORE: Context-aware open relation extraction with factorization machines. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1763–1773.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. iCaRL: Incremental classifier and representation learning. In *Proceedings of CVPR*, pages 2001–2010.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of ECML-PKDD*, pages 148–163.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of NAACL*, pages 74–84.

Ellen Riloff, Rosie Jones, et al. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of AAAI*, pages 474–479.

Mark Bishop Ring. 1994. *Continual learning in reinforcement environments*. Ph.D. thesis, University of Texas at Austin Austin, Texas 78712.

Hippolyt Ritter, Aleksandar Botev, and David Barber. 2018. Online structured laplace approximations for overcoming catastrophic forgetting. In *Proceedings of NIPS*, pages 3738–3748.

Benjamin Rozenfeld and Ronen Feldman. 2008. Self-supervised relation extraction from the web. *Proceedings of KAIS*, 17(1):17–33.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *ai.google*.

Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of ACL-IJCNLP*, pages 626–634.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *Proceedings of ESWC*, pages 593–607.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. In *Proceedings of NIPS*, pages 2990–2999.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of NAACL-HLT*, pages 304–311.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of EMNLP*, pages 1201–1211.

Gabriel Stanovsky and Ido Dagan. 2016. Creating a large benchmark for open information extraction. In *Proceedings of EMNLP*, pages 2300–2305.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2019. LAMAL: Language modeling is all you need for lifelong language learning. *arXiv preprint arXiv:1909.03329*.

Sebastian Thrun and Lorien Pratt. 2012. *Learning to learn*. Springer Science & Business Media.

Giulio Tononi and Chiara Cirelli. 2006. Sleep function and synaptic homeostasis. *Sleep medicine reviews*, 10(1):49–62.

Hong Wang, Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. 2019. Sentence embedding alignment for lifelong relation extraction. In *Proceedings of NAACL-HLT*, pages 796–806.

Ruidong Wu, Yuan Yao, Xu Han, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2019. Open relation extraction: Relational knowledge transfer from supervised data to unsupervised data. In *Proceedings of EMNLP-IJCNLP*, pages 219–228.

Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of WWW*, pages 1271–1279.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of EMNLP*, pages 1785–1794.

Guang Yang, Cora Sau Wan Lai, Joseph Cichon, Lei Ma, Wei Li, and Wen-Biao Gan. 2014. Sleep promotes branch-specific formation of dendritic spines after learning. *Science*, 344(6188):1173–1178.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of EMNLP*, pages 1456–1466.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved neural relation detection for knowledge base question answering. In *Proceedings of ACL*, pages 571–581.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Proceedings of JMLR*, pages 1083–1106.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of EMNLP*, pages 1753–1762.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING*, pages 2335–2344.

Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *Proceedings of ICML*, pages 3987–3995.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of EMNLP*, pages 35–45.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of ACL*, pages 427–434.

Jun Zhu, Zaiqing Nie, Xiaojiang Liu, Bo Zhang, and Ji-Rong Wen. 2009. StatSnowball: A statistical approach to extracting entity relationships. In *Proceedings of WWW*, pages 101–110.