Sentence Embedding Alignment for Lifelong Relation Extraction

Hong Wang[†], **Wenhan Xiong**[†], **Mo Yu**[‡*], **Xiaoxiao Guo**[‡*], **Shiyu Chang**[‡], **William Yang Wang**[†]
[†] University of California, Santa Barbara
[‡] IBM Research
{hongwang600, xwhan, william}@cs.ucsb.edu, yum@us.ibm.com, {xiaoxiao.guo, shiyu.chang}@ibm.com

Abstract

Conventional approaches to relation extraction usually require a fixed set of pre-defined relations. Such requirement is hard to meet in many real applications, especially when new data and relations are emerging incessantly and it is computationally expensive to store all data and re-train the whole model every time new data and relations come in. We formulate such a challenging problem as lifelong relation extraction and investigate memory-efficient incremental learning methods without catastrophically forgetting knowledge learned from previous tasks. We first investigate a modified version of the stochastic gradient methods with a replay memory, which surprisingly outperforms recent state-of-the-art lifelong learning methods. We further propose to improve this approach to alleviate the forgetting problem by anchoring the sentence embedding space. Specifically, we utilize an explicit alignment model to mitigate the sentence embedding distortion of the learned model when training on new data and new relations. Experiment results on multiple benchmarks show that our proposed method significantly outperforms the state-of-the-art lifelong learning approaches.

1 Introduction

The task of relation detection/extraction aims to recognize entity pairs' relationship from given contexts. As an essential component for structured information extraction, it has been widely used in downstream tasks such as automatic knowledge-based completion (Riedel et al., 2013) and question answering (Yih et al., 2015; Yu et al., 2017).

Existing relation detection methods always assume a closed set of relations and perform once-and-for-all training on a fixed dataset. While making the evaluation straightforward, this setting clearly limits the usage of these methods in realistic applications, where new relations keep emerging over time. To build an evolving system which automatically keeps up with the dynamic data, we consider a more practical lifelong learning setting (also called *continual learning*) (Ring, 1994; Thrun, 1998; Thrun and Pratt, 2012), where a learning agent learns from a sequence of tasks, where each of them includes a different set of relations. In such scenarios, it is often infeasible to combine the new data with all previous data and re-train the model using the combined dataset, especially when the training set for each task is huge.

To enable efficient learning in such scenarios, recent lifelong learning research (Kirkpatrick et al., 2016; Lopez-Paz and Ranzato, 2017) propose to learn the tasks incrementally, while at the same time preventing catastrophic forgetting (McCloskey and Cohen, 1989; Ratcliff, 1990; McClelland et al., 1995; French, 1999), i.e., the model abruptly forgets knowledge learned on previous tasks when learning on the new task. Current lifelong learning approaches address such challenge by either preserving the training loss on previously learned tasks (GEM) (Lopez-Paz and Ranzato, 2017), or selectively dimming the updates on important model parameters (EWC) (Kirkpatrick et al., 2016). These methods usually involve adding additional constraints on the model's parameters or the updates of parameters by utilizing stored samples. Despite the effectiveness of these methods on simple image classification tasks, there is little research validating the practical usage of these methods in realistic NLP tasks. In fact, when applying these methods to our relation extraction task, we observe that they underperform a simple baseline that updates the model parameters (i.e., learning by SGD) with a mix

[*] Co-mentoring
[-] Code and dataset can be found in this repository: `https://github.com/hongwang600/Lifelong_Relation_Detection`

Proceedings of NAACL-HLT 2019, pages 796–806
Minneapolis, Minnesota, June 2 - June 7, 2019. ©2019 Association for Computational Linguistics

of stored samples from previous tasks and new samples from the incoming task. We further test this simple baseline on commonly used continual learning benchmarks and get similar observations.

In this work, we thoroughly investigate two existing continual learning algorithms on the proposed lifelong relation extraction task. We observe that recent lifelong learning methods only operate on the models' parameter space or gradient space, and do not explicitly constraint the feature or embedding space of neural models. As we train the model on the new task, the embedding space might be distorted a lot, and become infeasible for previous tasks. We argue that the embedding space should not be distorted much in order to let the model work consistently on previous tasks. To achieve this, we propose an alignment model that explicitly anchors the sentence embeddings derived by the neural model. Specifically, the alignment model treats the saved data from previous tasks as anchor points and minimizes the distortion of the anchor points in the embedding space in the lifelong relation extraction. The aligned embedding space is then utilized for relation extraction. Experiment results show that our method outperforms the state-of-the-art significantly in accuracy while remaining efficient.

The main contributions of this work include:

- We introcduce the lifelong relation detection problem and construct lifelong relation detection benchmarks from two datasets with large relation vocabularies: SimpleQuestions (Bordes et al., 2015) and FewRel (Han et al., 2018).
- We propose a simple memory replay approach and find that current popular methods such as EWC and GEM underperform this method.
- We propose an alignment model which aims to alleviate the catastrophic forgetting problem by slowing down the fast changes in the embedding space for lifelong learning.

2 Problem Definition

Generic definition of lifelong learning problems In lifelong learning, there is a sequence of K tasks $\{\mathcal{T}^{(1)}, \mathcal{T}^{(2)}, \dots, \mathcal{T}^{(K)}\}$. Each task $\mathcal{T}^{(k)}$ is a conventional supervised task, with its own label set $L^{(k)}$ and training/validation/testing data $(T^{(k)}_{\text{train}}, T^{(k)}_{\text{valid}}, T^{(k)}_{\text{test}})$, each of which is a set of labeled instances $\{(x^{(k)}, y^{(k)})\}$. Note that $x^{(k)}$ is the input data of the context and candidate relations, and $y^{(k)}$ is the ground-truth label. The goal of lifelong learning is to learn a classification model f. At each step k, f observes the task $\mathcal{T}^{(k)}$, and optimizes the loss function on its training data with a loss function $\ell(f(x), y)$. At the same time, we require the model f learned after step k could still perform well on the previous $k-1$ tasks. That is, we evaluate the model by using the average accuracy of k tasks at each step as $\frac{1}{k}\sum_{j=1}^{k} acc_{f,j}$.

To make f perform well on the previous tasks, during the lifelong learning process, we usually allow the learner to maintain and observe a memory $\mathcal{M}$ of samples from the previous tasks. Practically, with the growth of the number of tasks, it is difficult to store all the task data[1]. Therefore, in lifelong learning research, the learner is usually constrained on the memory size, denoted as a constant B. Thus at each step k, the learner is allowed to keep training samples from $\{\mathcal{T}^{(j)}|j = 1, \dots, k-1\}$ with size less or equal to B.

Lifelong relation detection In this paper we introduce a new problem, *lifelong relation detection*. Relation detection is an important task that aims to detect whether a relation exists between a pair of entities in a paragraph. In many real-world scenarios, relation detection naturally forms a lifelong learning problem because new relation types emerge as new knowledge is constantly being discovered in various domains. For example, in the Wikidata (Vrandečić and Krötzsch, 2014) knowledge graph, the numbers of new items and properties are constantly increasing[2]. So we need to keep collecting data and updating the model over time in order to handle newly added relations.

The problem of lifelong relation detection has the same definition as above with only one difference: during prediction time, we hope to know whether an input paragraph contains any relation observed before. Therefore at time k, given an input x from task $j'<k$, instead of predicting an $y \in L^{(j')}$, we predict $y^{(k)} \in \bigcup_{j=1}^{k} L^{(j)}$. That says, the candidate label set is expanding as the learner observes more tasks, and the difficulty of each previous task is increasing over time as well.

[1] Even the data can be stored, it is unrealistic to make full usage of the stored data. For example, random sampling from all previous task data (e.g., for the methods in Section 4) will become statistically inefficient.

[2] https://www.wikidata.org/wiki/Wikidata:News

3 Evaluation Benchmarks for Lifelong Learning

3.1 Previous non-NLP Benchmarks

Lifelong MNIST MNIST is a dataset of handwriting ten digits (LeCun, 1998), where the input for each sample is an image, and the label is the digit the image represents. Two variants of the MNIST dataset were proposed for lifelong learning evaluation. One is MNIST *Permutations* (Kirkpatrick et al., 2016), where a task is created by rearranging pixels according to a fixed permutation. K different permutations are used to generate K tasks. Another variant is MNIST *Rotations* (Lopez-Paz and Ranzato, 2017), where each task is created by rotating digits by a fixed angle. K angles are chosen for creating K tasks. In our experiments, we follow (Lopez-Paz and Ranzato, 2017) to have $K = 20$ tasks for each benchmark.

Lifelong CIFAR CIFAR (Krizhevsky and Hinton, 2009) is a dataset used for object recognition, where the input is an image, and the label is the object the image contains. Lifelong CIFAR100 (Rebuffi et al., 2017a) is a variant of CIFAR-100 (CIFAR with 100 classes) by dividing 100 classes into K disjoint subsets. Each task contains samples from $\frac{100}{K}$ classes in one subset. Following (Lopez-Paz and Ranzato, 2017), we have $K = 20$ tasks, where each of them has 5 labels.

3.2 The Proposed Lifelong Relation Detection Benchmarks

Lifelong FewRel FewRel (Han et al., 2018) is a recently proposed dataset for few-shot relation detection. There are 80 relations in this dataset. We choose to create a lifelong benchmark based on FewRel because there are a sufficient number of relation labels. We extract the sentence-relation pairs from FewRel and build our lifelong FewRel benchmark as follows. Each sample contains a sentence with the ground-truth relation it refers, and a set of 10 randomly chosen false relations from all the whole relations set. The model is required to distinguish the right relation from the candidates. We apply K-Means over the averaged word embeddings of the relation names and divide 80 relations into 10 disjoint clusters. This results in 10 tasks in this benchmark, and each task contains relations from one cluster. Candidate relations will be masked if they do not appear in the history tasks.

Lifelong SimpleQuestions SimpleQuestions is a KB-QA dataset containing single-relation questions (Bordes et al., 2015). (Yu et al., 2017) created a relation detection dataset from SimpleQuestions that contains samples of question-relation pairs. For each sample, a candidate set of relations is also provided. Similar to lifelong FewRel, we divide relations into 20 disjoint clusters by using K-Means. This results in 20 tasks, and each task contains relations from one cluster.

4 Simple Episodic Memory Replay Algorithm for Lifelong Learning

Catastrophic forgetting is one of the biggest obstacles in lifelong learning. The problem is particularly severe in neural network models, because the learned knowledge of previous tasks is stored as network weights, while a slight change of weights when learning on the new task could have an unexpected effect on the behavior of the models on the previous tasks (French, 1999).

Currently, the memory-based lifelong learning approaches, which maintain a working memory of training examples from previous tasks, are proved to be one of the best solutions to the catastrophic forgetting problem. In this section, we first propose a memory-based lifelong learning approach, namely Episodic Memory Replay (EMR), which uses the working memory by sampling stored samples to replay in each iteration of the new task learning. Surprisingly, such a straightforward approach with a clear motivation was never used in previous research. We first compare EMR with the state-of-the-art memory-based algorithm Gradient Episodic Memory (GEM). We also show that the EMR outperforms GEM on many benchmarks, suggesting that it is likely to be among the top-performed lifelong learning algorithms, and it should never be ignored for comparison when developing new lifelong learning algorithms.

4.1 Episodic Memory Replay (EMR)

EMR is a modification over stochastic gradient descent algorithms. It replays randomly sampled data from memory while training on a new task, so the knowledge of previous tasks could be retained in the model. After training on each task k, EMR selects several training examples to store in the memory $\mathcal{M}$, denoted as $\mathcal{M} \bigcap T_{\text{train}}^{(k)}$.[3]

[3] (Rebuffi et al., 2017b) propose to dynamically change the size of memory set for each task during training. The

To handle the scalability, EMR stochastically replays the memory. Specifically, when training on task k with each mini-batch $D_{\text{train}}^{(k)} \subset T_{\text{train}}^{(k)}$, EMR samples from the memory $\mathcal{M}$ to form a second mini-batch $D_{\text{replay}}^{(k)} \subset \mathcal{M}$. Then two gradient steps are taken on the two mini-batches of $D_{\text{train}}^{(k)}$ and $D_{\text{replay}}^{(k)}$. Note that EMR could work with any stochastic gradient optimization algorithm, such as SGD, Adagrad, AdaDelta, and Adam, to optimize the model f with the mixed mini-batches.

We try two variations of $D_{\text{replay}}^{(k)}$ sampling: first, *task-level sampling*, which samples from one previous task j each time, i.e., $D_{\text{replay}}^{(k)} \subset \mathcal{M} \bigcap T_{\text{train}}^{(j)}$. Second, *sample-level sampling*, which samples all over the memory, i.e., $D_{\text{replay}}^{(k)} \subset \mathcal{M}$.

The two approaches differ in the task instance sampling probability. The task-level approach assumes a uniform distribution over tasks, while the sample-level approach has a marginal distribution on tasks that is proportional to the number of their training data in $\mathcal{M}$.[4] When tasks are balanced like MNIST and CIFAR, or when the stored data in the memory for different tasks are balanced, the two approaches become equivalent.

However, the sample-level strategy could sometimes make the code implementation more difficult: for some lifelong learning benchmarks such as MNIST Rotation, MNIST Permutation, and CIFAR-100 used in (Lopez-Paz and Ranzato, 2017), the tasks could differ from each other in the input or output distribution, leading to different computation graphs for different training examples. From our preliminary study, the task-level approach could always give results as good as those of the sample-level approach on our lifelong relation detection benchmarks (see Table 1) , so in our experiments in Section 6 we always use the task-level approach.

4.2 Comparing EMR with State-of-the-art Memory-based Lifelong Algorithm

In this part, we will first thoroughly introduce a state-of-the-art memory-based lifelong learning algorithm called Gradient Episodic Memory (GEM) (Lopez-Paz and Ranzato, 2017), and then compare EMR with it in both time complexity and experimental results on several benchmarks.

Gradient Episodic Memory (GEM) The key idea of GEM (Lopez-Paz and Ranzato, 2017) is to constrain the new task learning with previous task data stored in memory. Specifically, it constrains the gradients during training with the following operation. When training on task k, for each mini-batch $D_{\text{train}}^{(k)} \subset T_{\text{train}}^{(k)}$, it first computes the gradient $g_{\text{train}}^{(k)}$ on $D_{\text{train}}^{(k)}$, and the average gradients on the stored data of each previous task j, denoted as $g_{\text{task}}^{(j)}$. More concretely, we define

$$g_{\text{task}}^{(j)} = \frac{\sum_{i'} \nabla \ell(f(x_{i'}^{(j)}), y_{i'}^{(j)})}{|\mathcal{M} \bigcap T_{\text{train}}^{(j)}|},$$

where $j<k$, $\ell(\cdot)$ is the loss function, and $(x_{i'}^{(j)}, y_{i'}^{(j)}) \in \mathcal{M} \bigcap T_{\text{train}}^{(j)}$, i.e. $(x_{i'}^{(j)}, y_{i'}^{(j)})$ is a training instance in $\mathcal{T}^{(j)}$ that was stored in memory $\mathcal{M}$. Then the model f is updated along the gradient $\tilde{g}$ that solves the following problem:

$$\begin{aligned} \min_{\tilde{g}} \quad & ||\tilde{g} - g_{\text{train}}^{(k)}||^2 \\ \text{s.t.} \quad & \langle \tilde{g}, g_{\text{task}}^{(j)} \rangle \geq 0, \ j = 1, \ldots, k-1. \end{aligned}$$

$\tilde{g}$ is the closest gradient to the gradient on the current training mini-batch, $g_{\text{train}}^{(k)}$, without decreasing performance on previous tasks much since the angle between $\tilde{g}$ and $g_{\text{task}}^{(j)}$ is smaller than 90°.

Time Complexity One difference between EMR and GEM is that EMR deals with unconstrained optimization and does not require the gradient projection, i.e., solving $\tilde{g}$. But since the model f is deep networks, empirically the time complexity is mainly dominated by the computation of forward and backward passes. We analyze the time complexity as below:

In task k, suppose the mini-batch size is $|D|$ and the memory replay size is m, our EMR takes $|D| + m$ forward/backward passes in each training batch. Note that m is a fixed number and set to be equal to the number of instances stored for each previous task in our experiments. While for GEM, it needs to compute the gradient of all the data stored in the memory $\mathcal{M}$, thus $|D| + |\mathcal{M}|$ forward/backward passes are taken. Its complexity is largely dominated by the size $|\mathcal{M}|$ (upper bounded by the budget B). When the budget B is large, with the number of previous tasks increases, $\mathcal{M}$ grows linearly, and GEM will become infeasible.

followup work and this paper all use fixed sets, and we will investigate the usage of dynamic sets in future work.

[4] The two approaches hence favor different evaluation metrics – the former fits macro averaging better and the latter fits micro averaging better.

Task	EMR		GEM
	sample	task	
MNIST Rotation	–	0.828	**0.860**
MNIST Permutation	–	0.824	**0.826**
CIFAR-100	–	**0.675**	**0.675**
FewRel	0.606	**0.620**	0.598
SimpleQuestions	0.804	**0.808**	0.796

Table 1: The average accuracy across all the tasks at last time step for EMR and GEM on both non-NLP and our lifelong relation detection benchmarks. For the experiments on MNIST and CIFAR, we follow the setting in (Lopez-Paz and Ranzato, 2017) (see Appendix A.2 for details). For the experiments on FewRel and SimpleQuestions, we use the same setting in Section 6. We only implement task-level EMR for MNIST and CIFAR because of the relatively easy implementation.

Superior Empirical Results of EMR The EMR algorithm is much simpler compared to the GEM. However, one interesting finding of this paper is that the state-of-the-art GEM is unnecessarily more complex and more inefficient, because EMR, a simple stochastic gradient method with memory replay, outperforms it on several benchmarks.

The results are shown in Table 1. The numbers are the average accuracy, i.e. $\frac{1}{k}\sum_{j=1}^{k} acc_{f,j}$, at last time step. For both algorithms, the training data is randomly sampled to store in the memory, following (Lopez-Paz and Ranzato, 2017). On lifelong relation detection, the EMR outperforms GEM on both of our created benchmarks. To further show its generalizability, we apply the EMR to previous lifelong MNIST and CIFAR benchmarks and compare to the results in (Lopez-Paz and Ranzato, 2017) with all the hyperparameters set as the same. Still, EMR performs similarly to GEM except for the MNIST Rotation benchmark.[5]

From the above results, we learned the lesson that previous lifelong learning approaches actually fail to show improvement compared to doing memory replay in a stochastic manner. We hypothesise that GEM performs worse when there is positive transfer among tasks, making the gradient projection an inefficient way to use gradients computed from memory data. Therefore, in the next section, we start with the basic EMR and focus on more efficient usage of the historical data.

5 Embedding Aligned EMR (EA-EMR)

Based on our basic EMR, this section proposes our solution to lifelong relation detection. We improve the basic EMR with two motivations: (1) previous lifelong learning approaches work on the parameter space. However, the number of parameters in a deep network is usually huge. Also, deep networks are highly non-linear models, and the parameter dimensions have complex interactions, making the Euclidean space of parameters not a proper delegate of model behavior (French, 1999). That is, a slight change in parameter space could affect the model prediction unexpectedly. The above two reasons make it hard to maintain deep network behaviors on previous tasks with constraints or Fisher information. Therefore, we propose to alleviate catastrophic forgetting in the hidden space (i.e., the sentence embedding space). (2) for each task, we want to select the most informative samples to store in the memory, instead of random sampling like in (Lopez-Paz and Ranzato, 2017). Therefore the budget of memory can be better utilized.

5.1 Embedding Alignment for Lifelong Learning

This section introduces our approach which performs lifelong learning in the embedding space, i.e., the Embedding Aligned EMR (EA-EMR).

In EA-EMR, for each task k, besides storing the original training data $(x^{(k)}, y^{(k)})$ in the memory $\mathcal{M}$, we also store the embeddings of $x^{(k)}$. In the future after a new task is trained, the model parameters are changed thus the embeddings for the same $(x^{(k)}, y^{(k)})$ would be different. Intuitively, a lifelong learning algorithm should allow such parameter changes but ensure the changes do not distort the previous embedding spaces too much.

Our EA-EMR alleviates the distortion of embedding space with the following idea: if the embedding spaces at different steps are not distorted much, there should exist a simple enough transformation a (e.g., a linear transformation in our case) that could transform the newly learned embeddings to the original embedding space, without much performance degeneration on the stored instances. So we propose to add a transformation a on the top of the original embedding and learn the basic model f and the transformation a automatically. Specifically, at the k-th task, we start with the model $f^{(k-1)}$, and the transformation $a^{(k-1)}$,

[5] Even on MNIST Rotation, it has achieved a competitive result, since the conventional training on shuffled data from all the tasks in this benchmark gives $\sim$ 0.83 according to (Lopez-Paz and Ranzato, 2017).

that trained on the previous $k-1$ tasks. We want to learn the basic model f and the transformation a such that the performance on the new task and stored instances are optimized without distorting the previous embedding spaces much.

$$\min_{f(\cdot),a(\cdot)} \sum_{(x,y)\in D^{(k)}_{\text{train}}} \ell(a(f(x)),y)+ \\ \sum_{(x,y)\in D^{(k)}_{\text{replay}}} \left(\ell(a(f(x)),y)+\|a(f(x))-a^{(k-1)}(f^{(k-1)}(x))\|^2\right)$$

We propose to minimize the above objective through two steps. In the first step, we optimize the basic model f by:

$$\min_{f(\cdot)} \sum_{(x,y)\in D^{(k)}_{\text{train}} \cup D^{(k)}_{\text{replay}}} \ell\left(a^{(k-1)}(f(x)),y\right)$$

This step mainly focuses on learning the new task without performance drop on the stored samples.

In second step, we optimize a to keep the embedding space of the current task and restore the previous embedding space of all stored samples:

$$\min_{a(\cdot)} \sum_{(x,y)\in D^{(k)}_{\text{train}}} \|a(f(x))-a^{(k-1)}(f(x))\|^2 \\ + \sum_{(x,y)\in D^{(k)}_{\text{replay}}} \|a(f(x))-a^{(k-1)}(f^{(k-1)}(x))\|^2$$

Embedding Alignment on Relation Detection Model We introduce how to add embedding alignment to relation detection models. The basic model we use is a ranking model that is similar to HR-BiLSTM (Yu et al., 2017). Two BiLSTMs (Hochreiter and Schmidhuber, 1997) are used to encode the sentence and relation respectively given their GloVe word embedding (Pennington et al., 2014). Cosine similarity between the sentence and relation embedding is computed as the score. Relation with maximum score is predicted by the model for the sentence. Ranking loss is used to train the model[6]. This base model is our model f, which is trained on a new task k at each step and results in an updated model $f^{(k)}$. Our proposed approach (Figure 1) inserts an alignment model a to explicitly align to embedding space for stored instances and maintain the embedding space of the current task. Note that the label y (the relation here) also has embedding, so it needs to pass through the alignment model a as well.

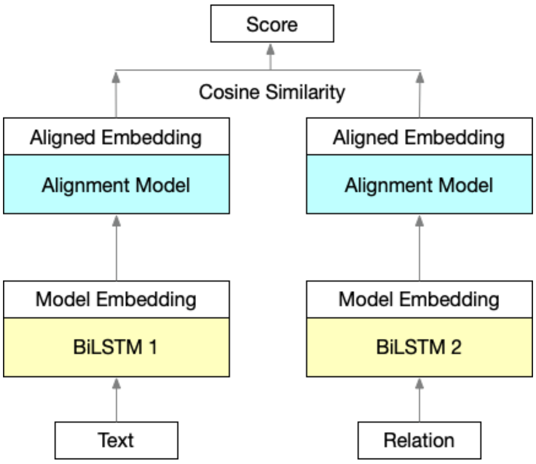


Figure 1: This figure shows how we add the alignment model (a linear model in our case) on the basic relation detection model, where two BiLSTMs are used to encode the text and relation, and cosine similarity between their embeddings are computed as the score.

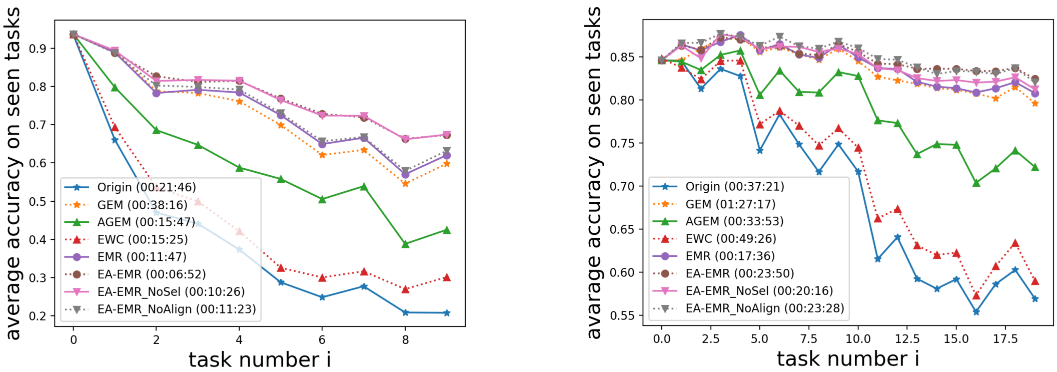
5.2 Selective Storing Samples in Memory

When the budget of memory is relatively smaller, how to select previous samples will greatly affect the performance. Ideally, in order to make the memory best represents a previous task, we hope to choose diverse samples that best approximate the distribution of task data. However, distribution approximation itself is a hard problem and will be inefficient due to its combinatorial optimization nature. Therefore, many recent works such as GEM ignore this step and randomly select samples from each task to store in the memory.

Rebuffi et al. (2017b) proposed to select exemplars that best approximate the mean of the distribution. This simplest distribution approximation does not give an improvement in our experiments because of the huge information loss. Therefore, we propose a better approach of sample selection by clustering over the embedding space from the model, and choose one representative from each cluster to store in the memory. More specifically, The embedding after alignment model is used to represent the input because the model makes prediction based on that. Then we apply K-Means (the number of clusters equals the budget given to the specific task) to cluster all the samples of the task. For each cluster, we select the sample closest to the centroid to store in the memory.

We leave more advanced approaches of representative sample selection and their empirical comparison to future work.

[6] Though the basic model is simple, it achieves reasonable results on the two datasets when training with all the data, i.e., 0.837 on FewRel and 0.927 on SimpleQuestions.



(a) FewRel

(b) SimpleQuestions

Figure 2: This figure shows the average accuracy of all the observed tasks on the benchmarks of lifelong FewRel and lifelong SimpleQuestions during the lifelong learning process. The average performance of 5 runs is reported, and the **average running time** is shown in the brackets.

6 Experiments

6.1 Experimental Setting

We conduct experiments on our lifelong benchmarks: lifelong SimpleQuestions (Bordes et al., 2015) and lifelong FewRel (Han et al., 2018) to compare our proposed methods EA-EMR, EA-EMR without Selection (EA-EMR_NoSel), EA-EMR without Alignment (EA-EMR_noAlign), and EMR with the following baselines.

• **Origin**, which simply trains on new tasks based on the previous model.

• **EWC** (Kirkpatrick et al., 2016), which slows down updates on important parameters by adding L_2 regularization of parameter changes to the loss.

• **GEM** (Lopez-Paz and Ranzato, 2017), which projects the gradient to benefit all the tasks so far by keeping a constraint for each previous task.

• **AGEM** (Anonymous, 2019), which only uses one constraint that the projected gradient should decrease the average loss on previous tasks.

On both FewRel and SimpleQuestions, the epoch to train on each task is set to be 3. Learning rate for the basic model is set to be 0.001. The hidden size of LSTM is set to be 200. The batch size is set to be 50. For each sample in the memory, 10 candidate relations is randomly chosen from all observed relations to alleviate the problem that new relations are emerging incessantly.

Parameters for our model and baselines are set as follows. For EA-EMR and EA-EMR_NoSel, when training the alignment model, the learning rate is set to be 0.0001, and the training epoch is set to be 20 and 10 for FewRel and SimpleQuestions respectively. For AGEM, 100 samples are randomly chosen from all the previous tasks to form a constraint. For EWC, we set the balancing parameter $\alpha = 100$. For GEM and EMR related methods, memory size of each task is set to be 50.

Method	FewRel		SimpleQuestions	
	Whole	Avg	Whole	Avg
Origin	0.189	0.208	0.632	0.569
Baselines				
GEM	0.492	0.598	0.841	0.796
AGEM	0.361	0.425	0.776	0.722
EWC	0.271	0.302	0.672	0.590
Ours				
Full EA-EMR	**0.566**	0.673	**0.878**	**0.824**
w/o Selection	0.564	**0.674**	0.857	0.812
w/o Alignment	0.526	0.632	0.869	0.820
w/o Alignment but keep the architecture	0.545	0.655	0.871	0.813
EMR Only	0.510	0.620	0.852	0.808

Table 2: This table shows the accuracy on the whole testing data ("Whole" column), and average accuracy on all observed tasks ("Avg" column) after the last time step. The average performance of 5 runs are listed here and the best result on each dataset is marked in bold.

6.2 Lifelong Relation Detection Results

Evaluation Metrics We use two metrics to evaluate the performance of the model:

• Average performance on all seen tasks after time step k, which highlights the catastrophic problem:

$$\text{ACC}_{\text{avg}} = \frac{1}{k}\sum_{i=1}^{k} acc_{f,i}$$

• Accuracy on the whole testing data of all tasks:

$$\text{ACC}_{\text{whole}} = acc_{f,D_{\text{test}}}$$

Results on FewRel and SimpleQuestions We run each experiment 5 times independently by shuffling sequence of tasks, and the average performance is reported. The average accuracy over all observed tasks during the whole lifelong learning process is presented in Figure 2, and the accuracy on the whole testing data during the process is shown in Appendix A.1. We also list the result at last step in Table 2. From the results, we can see that EWC and GEM are better than the Origin baseline on both two datasets, which indicates that they are able to reduce the catastrophic forgetting problem. However, our EA-EMR perform significantly better than these previous state-of-the-arts. The proposed EMR method itself achieves better results than all baselines on both datasets. The ablation study shows that both the *selection* and the *alignment* modules help on both tasks.

The Effect of Embedding Alignment To investigate the effect of our embedding alignment approach, we conduct two ablation studies as below: First, we remove both the alignment loss in equation 5.1, as well as the alignment module a, which results in significant drop on most of the cases (the line "w/o Alignment" in Table 2). Second, to make sure that our good results do not come from introducing a deeper model with the module a, we propose to only remove the embedding alignment loss, but keep everything else unchanged. That means, we still keep the module a and the training steps, with the only change on replacing the loss in step 2 with the one in step 1 (the line "w/o Alignment but keep the architecture" in Table 2). We can see that this decreases the performance a lot. The above results indicate that by explicitly doing embedding alignment, the performance of the model can be improved by alleviating the distortion of previous embedding space.

Comparison of Different Sample Selection Strategies Here we compare different selection methods on lifelong FewRel and SimpleQuestions. EMR Only randomly choose samples. (Rebuffi et al., 2017b) propose to choose samples that can best approximate the mean of the distribution. We compare their sampling strategy (denoted as iCaRL) with our proposed method (K-Means) which encourages to choose diverse samples by choosing the central sample of the cluster in the embedding space. From the results in Table 3, we can see that our method outperforms iCaRL and the random baseline. While iCaRL is not significantly different from the random baseline.

Method	FewRel		SimpleQuestions	
	Whole	Avg	Whole	Avg
EMR Only	0.510	0.620	0.852	0.808
+ K-Means	**0.526**	**0.632**	**0.869**	**0.820**
+ iCaRL	0.501	0.615	0.854	0.806

Table 3: Comparison of different methods to select data for EMR. The accuracy on the whole testing data (”Whole” column), and average accuracy on all observed tasks (”Avg” column) is reported. We run each method 5 times, and give their average results.

7 Related Work

Lifelong Learning without Catastrophic Forgetting Recent lifelong learning research mainly focuses on overcoming the *catastrophic forgetting* phenomenon (French, 1999; McCloskey and Cohen, 1989; McClelland et al., 1995; Ratcliff, 1990), i.e., knowledge of previous tasks is abruptly forgotten when learning on a new task.

Existing research mainly follow two directions: the first one is *memory-based approach* (Lopez-Paz and Ranzato, 2017; Anonymous, 2019), which saves some previous samples and optimizes a new task with a forgetting cost defined on the saved samples. These methods have shown strength in alleviating catastrophic forgetting, but the computational cost grows rapidly with the number of previous tasks. The second direction is to *consolidate parameters that are important to previous tasks* (Kirkpatrick et al., 2016; Liu et al., 2018; Ritter et al., 2018; Zenke et al., 2017). For example, Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2016) slows down learning on weights that are important to previous tasks. These methods usually do not need to save any previous data and only train on each task once. But their abilities to overcome catastrophic forgetting are limited.

Lifelong Learning with Dynamic Model Architecture There is another related direction on dynamically changing the model structure (i.e., adding new modules) in order to learn the new task without interfering learned knowledge for previous tasks, such as (Xiao et al., 2014; Rusu et al., 2016; Fernando et al., 2017). These approaches could successfully prevent forgetting. However, they do not suit many lifelong settings in NLP. First, it cannot benefit from the positive transfer between tasks. Second, the size of the model grows dramatically with the number of observed tasks, which makes it infeasible for real-world problems where there are a lot of tasks.

Remark It is worth noting that the term lifelong learning is also widely used in (Chen et al., 2015; Chen, 2015; Shu et al., 2016, 2017), which mainly focus on how to represent, reserve and extract knowledge of previous tasks. These works belong to a research direction different from lifelong learning without catastrophic forgetting.

8 Conclusion

In this paper, we introduce lifelong learning into relation detection, and find that two state-of-the-art lifelong learning algorithms, GEM and EWC, are outperformed by a simple memory replay method EMR on many benchmarks. Based on EMR, we further propose to use embedding alignment to alleviate the problem of embedding space distortion, which we think is one reason that causes catastrophic forgetting. Also, we propose to choose diverse samples to store in the memory by conducting K-Means in the model embedding space. Experiments verify that our proposed methods significantly outperform other baselines.

Acknowledgement

This research was supported in part by a UCSB Chancellor's Fellowship, an IBM Faculty Award, and a DARPA Grant D18AP00044 funded under the DARPA YFA program. The authors are solely responsible for the contents of the paper, and the opinions expressed in this publication do not reflect those of the funding agencies.

References

Anonymous. 2019. Efficient lifelong learning with a-gem. In *Submitted to International Conference on Learning Representations*. Under review.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *CoRR*, abs/1506.02075.

Zhiyuan Chen. 2015. Lifelong machine learning for topic modeling and beyond. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 133–139. Association for Computational Linguistics.

Zhiyuan Chen, Nianzu Ma, and Bing Liu. 2015. Lifelong learning for sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 750–756. Association for Computational Linguistics.

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. *CoRR*, abs/1701.08734.

Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. 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, October 31 - November 4, 2018*, pages 4803–4809.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2016. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796.

Alex Krizhevsky and Geoffrey Hinton. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

Yann LeCun. 1998. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

Xialei Liu, Marc Masana, Luis Herranz, Joost van de Weijer, Antonio M. López, and Andrew D. Bagdanov. 2018. Rotate your networks: Better weight consolidation and less catastrophic forgetting. *CoRR*, abs/1802.02950.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6470–6479.

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):419–457.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Roger Ratcliff. 1990. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285–308.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. 2017a. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5533–5542.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017b. icarl: Incremental classifier and representation learning. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5533–5542. IEEE.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84.

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. *CoRR*, abs/1805.07810.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *CoRR*, abs/1606.04671.

Lei Shu, Bing Liu, Hu Xu, and Annice Kim. 2016. Lifelong-rl: Lifelong relaxation labeling for separating entities and aspects in opinion targets. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 225–235. Association for Computational Linguistics.

Lei Shu, Hu Xu, and Bing Liu. 2017. Lifelong learning crf for supervised aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 148–154. Association for Computational Linguistics.

Sebastian Thrun. 1998. *Lifelong Learning Algorithms*, pages 181–209. Springer US, Boston, MA.

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

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Tianjun Xiao, Jiaxing Zhang, Kuiyuan Yang, Yuxin Peng, and Zheng Zhang. 2014. Error-driven incremental learning in deep convolutional neural network for large-scale image classification. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 177–186, New York, NY, USA. ACM.

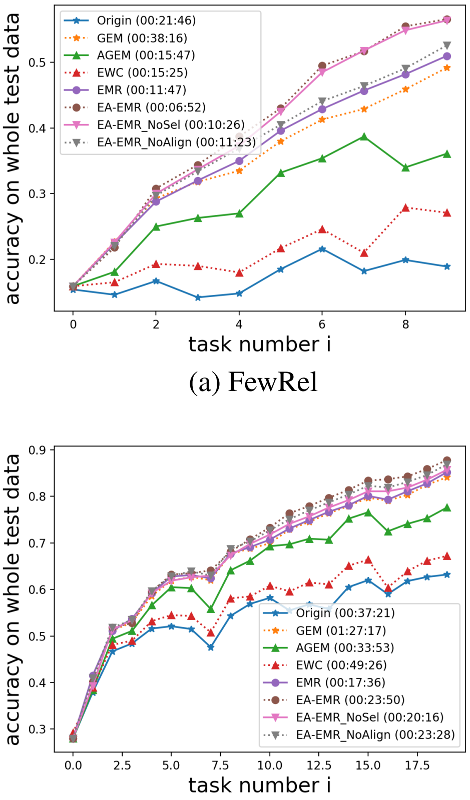
Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331. Association for Computational Linguistics.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved neural relation detection for knowledge base question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 571–581.

Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 3987–3995.

A Appendix

A.1 Performance on the whole testing data over time



(a) FewRel

(b) SimpleQuestions

Figure 3: This figure shows the accuracy on the whole testing data on the benchmark of lifelong FewRel and lifelong SimpleQuestions during the lifelong learning process. The average performance of 5 runs is reported, and the **average running time** is shown in the brackets.

The performance on the whole testing data over time is shown in Figure 3.

A.2 Experiment setting for MNIST and CIFAR

Following the setting in (Lopez-Paz and Ranzato, 2017), the size of memory for each task is set to be 256. The learning rate is set to be 0.1. The epoch for training the model on each task is set to be 1. Plain SGD and minibatch of 10 samples are used. For the MNIST dataset, each task has 1000 samples of 10 classes. For the CIFAR dataset, each task has 2500 samples of 5 classes.