Meta Representation Learning for Continual Few-shot Event Detection

Junting Wang, Kejie Zhao, Zhenwei He
University of Illinois Urbana-Champaign
junting3@illinois.edu, kejiez2@illinois.edu, zhenwei4@illinois.edu

Abstract

Event types are usually predefined while the ontology of event types keep expanding and changing in the real word scenarios. Continual event detection always suffers from high computation cost or catastrophic forgetting. We propose to adapt a meta-learning based framework to address the issues in continual learning. The proposed framework is composed of two modules, Representation Module and Adaption Module and training with the objective of Online-aware Meta-learning. Experiments results demonstrate the effectiveness of the proposed method that significantly outperforms the classical pretrain + finetune method.

1 Introduction

Event detection (ED) aims to identify and classify different event triggers in a sentence into specific event types. For example, in the sentence *"They killed by hostile fire in Iraqi"*, the *"fire"* can be identified as an event trigger which can be classified as the *"Attack"* event type.

Existing supervised event detection methods usually assumes that the ontology of the event types (also called event classes) is pre-defined and the number of event types is fixed. However, in the real word scenario, the event detection is usually continual, which means the ontology of event types can keep changing and expanding: new types of events and more fine-grained sub-types are constantly added to the system. Therefore. although recent years the supervised event detection methods have been put forward and showed promising performance (Chen et al., 2015; Du and Cardie, 2020; Liu et al., 2020; Lu et al., 2021), they are not suitable for continual settings/learning (called called continual learning or lifelong learning or incremental learning)(Ring et al., 1994; Thrun, 1998; Cao et al., 2020).

There are other problems need to be considered in the continual event detection.

The costs on the memory and computing can be dramatically high. A natural approach to address the continual event detection is to save the old data and re-train the whole model on the combination of old data and new data. However, the data stream for the continual event detection can be endless. Preserving the data for all the old event types demands extremely high cost on the storage. Also, re-training the model when the data come can be highly expensive on computation.

Another problem for the continual event detection is the *catastrophic forgetting*(McCloskey and Cohen, 1989; French, 1999). Another intuitive method to handle the continual event detection problem is to finetune pre-trained models on new data. However, this method can cause catastrophic forgetting problem: after finetuning the models on new event types, the accuracy of identifying old types suffer from significant dropping.

Much efforts have been tired to address the catastrophic forgetting problem. The major methods can be categorized into three classes:

1. Significant parameters based methods: try to preserve important parameters of the models learned on the old classes when learning the new classes (Kirkpatrick et al., 2017; Aljundi et al., 2018a).
2. Rehearsal methods: combine older samples into the new data during training, by replaying older data such as Rebuffi et al. (2017); Hou et al. (2019), by utilizing generative models trained on older data such as Shin et al. (2017), by leveraging knowledge distillation that produces representations or targets from older predictors (Li and Hoiem, 2017; Cao et al., 2020).
3. Sparse representation methods, which leave room for future learning and mitigate the catastrophic forgetting issues (Liu et al., 2019; Aljundi et al., 2018b).

Moreover, the distribution of event types is naturally imbalanced in the natural language. This is also called Long-tail distribution (Yu et al., 2021), which means many event types only have a small number of data points while some event types have a large of number of mentions. Previous methods (Nguyen et al., 2016; Cao et al., 2020) tend to focus on the frequent event types which can gain low performance on rare event types. Predicting the events types with few data points is also called *few-shot event detection*, and a method using *Causal Intervention* was introduced to address the few-shot problem (Chen et al., 2021).

Therefore, this paper proposes a method trying to address all the problems above:

- Learn new types and dynamic ontology when new data come;
- Reduce high cost on the computation and storage for systems;
- Avoid the catastrophic forgetting problem when learning new knowledge;
- Handle the long-tail distribution problem for event types.

Meta-learning based methods have achieved great success in various learning paradigms, i.e. few-shot learning, online learning and continual learning (Harrison et al., 2019; Javed and White, 2019). They achieve state-of-the-art performance on tasks such as few-shot image classification. Because the meta-learning methods proved to be useful in:

- Online learning – learn new types and dynamic ontology when new data come;
- No need to save much old data – reduce high cost on the computation and storage for systems;
- Continual learning – avoid the catastrophic forgetting problem when learning new knowledge;
- Few-shot learning – handle the long-tail distribution problem for event types.

this paper is motivated to explore a meta-learning framework that can be effectively applied to continual event detection setup and also address the long-tail problem. The main contribution of this paper:

- We adapt the meta-learning framework based on Javed and White (2019) to get meta-representation to address the four major problems above in the continual event detection;
- Our base model results show significant improvements over both the fine-tune and pre-trained fine-tune baselines.;
- Our model-agnostic representation module can be used as pre-trained language model initialization for better performance
- Our framework can be easily adapted to other continual Information Extraction (IE) tasks.

2 Problem Formulation

At first, introduce some Automatic Content Extraction (ACE) terminologies to help understand the Event Detection tasks (Cao et al., 2020): "**Event trigger** refers to a word that most clearly expresses the occurrence of an event. **Event arguments** are participants of the event. **Event mention** refers to a phrase or sentence within which an event is described."

Because Continual Event Detection (CED) in Natural Language problem can be viewed as a kind of Continual Learning Prediction (CLP), this paper, which majorly exploits meta-learning based methods, formulate the CED problem based on the CLP problem formulation from Javed and White (2019). A Continual Event Detection (CED) in Natural Language problem consists of an endless data stream:

$$\mathcal{T} = (X_1, Y_1), (X_2, Y_2), \ldots, (X_t, Y_t), \ldots$$

for inputs X_t (event mentions) and targets Y_t (event types), from sets $\mathcal{X}$ and $\mathcal{Y}$. The random vector Y_t is sampled according to an unknown distribution $p(Y|X_t)$. We define $S_k = (X_j + 1, Y_j + 1), (X_j + 2, Y_j + 2), \ldots, (X_j + k, Y_j + k)$ as a trajectory of length k which is smapled randomly from the CET problem $\mathcal{T}$, and $p(S_k|\mathcal{T})$ represents a distribution over all the trajectories S_k.

Our goal is to learn a function f_W, θ which predict Y_t for X_t, and minimize the loss function below for a CED problem:

$$\begin{aligned} &\mathcal{L}_{CED}(W, \theta) \\ &\overset{\text{def}}{=} \mathbb{E}\left[\ell\left(f_{W,\theta}(X), Y\right)\right] \\ &= \int \left[\int \ell\left(f_{W,\theta}(x), y\right) p(y \mid x) dy\right] \mu(x) dx \end{aligned} \tag{1}$$

where W and θ are the parameters to be updated to minimize the objective $\mathcal{L}_{CED}$.

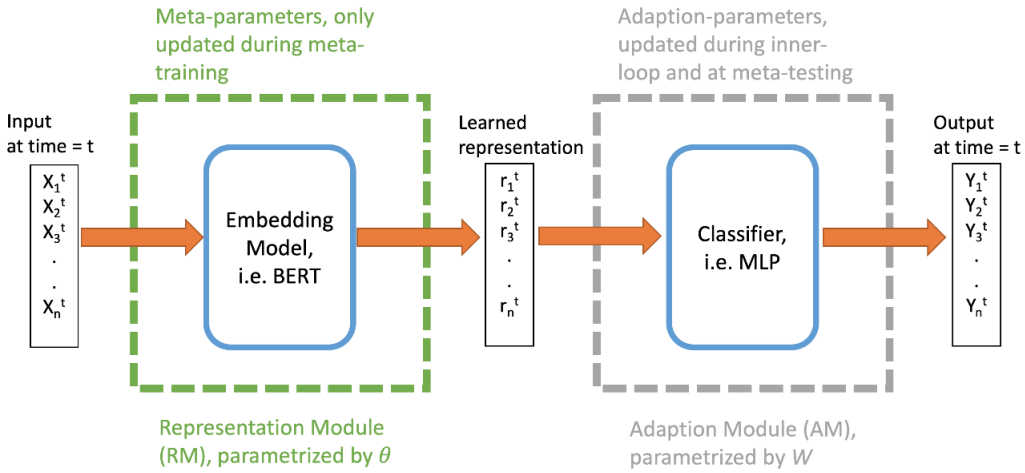

Figure 1: Our proposed framework composed of two modules: Representation Module and Adaption Module

3 Method

3.1 Meta-learning Representation Learning

Here we propose a model-agnostic meta-learning framework composed of two modules: $\phi_\theta(X)$, a deep Representation Module (RM) parametrized by θ - from X to $\mathbb{R}^h$; g_W, an Adaption Module (AM) parameterized by W - from $\mathbb{R}^h$ to y. The architecture of this composed module, $f_{\theta,w} = g_W(\phi_\theta(X))$ is shown in figure 1 . We follow a similar setup as **: θ is a meta-parameter that is learned by minimizing the meta-objective and is only updated during meta-training. Then, W is learned from trajectory using fully online stream data in a single pass.

We adapted Online-aware Meta-learning (OML) as our meta-objective for updating θ in Representation Module. OML objective could help maximize fast adaption for training the Representation Module as well as minimizing interference and is defined as:

$$\begin{aligned} &\min_{W,\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \text{OML}(W, \theta) \\ &= \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \sum_{\mathcal{S}_k^j \sim p(\mathcal{S}_k|\mathcal{T}_i)} [\mathcal{L}_i(U(W, \theta, \mathcal{S}_k^j)] \end{aligned} \tag{2}$$

where $p(\mathcal{T})$ is the assumed distribution for this continual learning problem; $\mathcal{S}_k^j = (X_{j+1}^i, Y_{j+1}^i), ..., (X_{j+k}^i, Y_{j+k}^i)$, a random trajectory of length k sampled from $p(\mathcal{T})$. And $U(W, \theta, \mathcal{S}_k^j) = (W_{t+k}, \theta)$ is an updated function where W_{t+k} is the weights of Adaption Module after k steps of parameter updates, where the jth update step in U is on parameters (W_{t+j-1}, θ) with samples (X_{j+t}^i, Y_{j+t}^i). We can see that OML only uses one data point from the trajectory, $\mathcal{S}_k^j$, which would help the model overcome the common problems of catastrophic forgetting in continual learning.

Here, we use BERT * as our Representation Module due to its strong linguistic representation learning ability and a one-layer fully-connected network as our Adaption Module.

3.2 Meta-Training and Meta-Testing Workflow

As stated in section 3.1, we update both Representation Module and Adaption Module only in meta-training phase, which is composed of four steps as shown in figure 2 below:
1. Suppose there is a data stream $\mathcal{T} = (X_0, Y_0), (X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n)$, we randomly sample a trajectory $\mathcal{S}_{support}$ of length k as our support set for inner updates and another trajectory $\mathcal{S}_{query}$ for evaluation.
2. We use $\mathcal{S}_{support}$ to do k continual gradient updates on Adaption Module following OML training objective, from W_1 to W_2, ..., to W_k.
3. Next, we run evaluation of $\mathcal{S}_{query}$ using the updated network to obtain a loss; this loss would be back-propagated with respect to the initial parameters, θ of RM and W_1 of AM.
4. Finally we update both of RM and AM, from θ, W_1 to θ', W_1'.

For meta-testing, we would freeze Representation Module and simply update the Adaption Module following the step 2 in meta-training using the support set and then make predictions on query

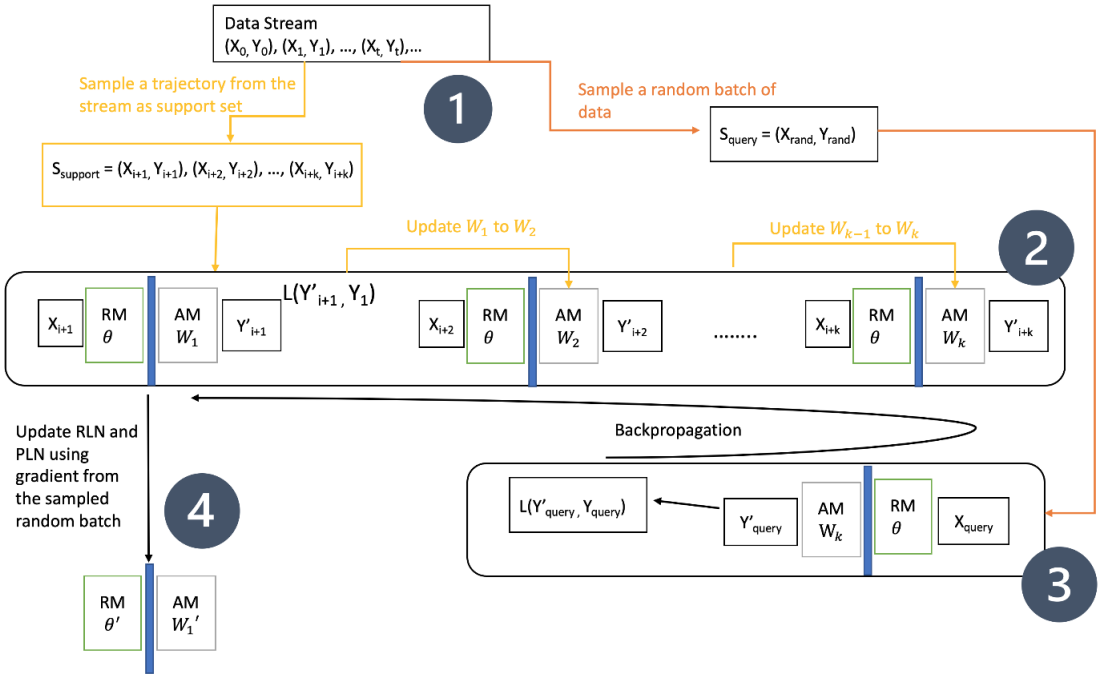

Figure 2: Meta-training Workflow

set with the updated Adaption Module and freeze Representation Module.

Dataset	MAVEN	ACE
# Event types	30	30
# Instances	23806	2216

Table 1: Dataset Statistics

4 Experiment

4.1 Dataset and Continual Few-shot Event Detection Task

We adopted two widely used datasets for event detection and created the continual few-shot event detection tasks for evaluation. The dataset statistic are shown in Table 1.

ACE 2005 English [1]: ACE 2005 English contains 33 event types. To conduct our few-shot experiment, we filtered out events that have less than 10 interactions.

MAVEN (Wang et al., 2020): MAVEN dataset contains 168 event types and is much more diverse compared to the ACE05 dataset. Each event types also has a significantly more instances. Due to limited computational resources, we choose keep the top 50 event types by number of instances.

[1] https://catalog.ldc.upenn.edu/LDC2006T06

Continual Few-shot Event Detection: There is no available benchmark for continual few-shot event detection. Therefor, we propose the following construction method: for a given dataset, we first construct a few-shot event detection dataset following a similar setup as (Chen et al., 2021). In particular, for every event type, we subsample its instances to simulate a few-shot condition. For each event type, we randomly sample few instances as the support set, and the all other instances are used as the query set for evaluating. For the continual event detection, we followed the setup in (Cao et al., 2020). We exploit the data of the top 10 most frequent event classes for continual learning. One new event class will be available for the model at each time, and we test the model on all previously seen event types.

4.2 Meta-Training and Meta-Testing Detail

Meta-Training:

Meta-Testing:

4.3 Baselines

We compare out approach with the following baselines:

Finetune: The model is simply fine-tuned whenever a stream of data arrives.

Pretrain + finetune: We fine-tune the language model that are pretrained on the meta-training set.

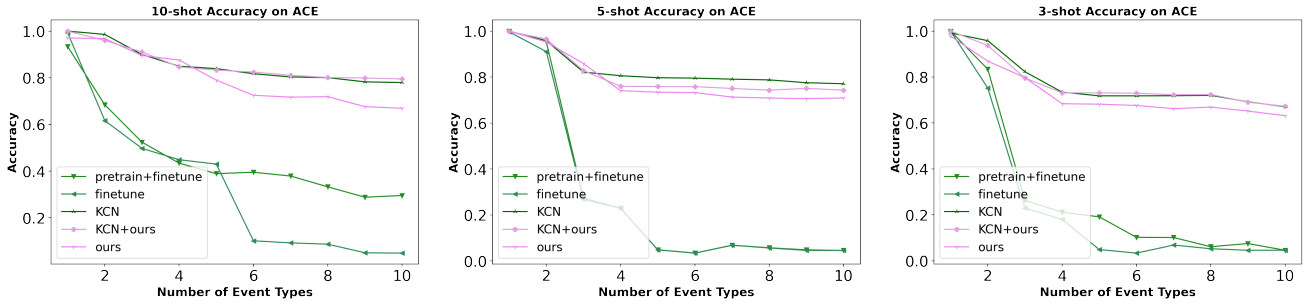

Figure 3: 10-shot, 5-shot, and 3-shot results using ACE05 as meta-testing set

KCN(Cao et al., 2020): KCN focuses on incremental event detection and uses prototype enhanced retrospection and hierarchical distillation to mitigate the adverse effects of semantic ambiguity and class imbalance. We initialized the language model used in KCN with a pretrained language model that is trained on the meta-training set to ensure a fair comparison.

We also implemented a model of (Yu et al., 2021) following our experimental setup. However, this method could only reach an accuracy of 0.4 while our method or KCN could reach 0.7. There are some possible explanations for this result:

1. There might be some implementation errors of (Yu et al., 2021) or a different configuration of hyper-parameters is required due to the new few-shots learning. For example, changing the default learning rate (1e-4) in (Yu et al., 2021) to (6e-4) could help improve the final accuracy from 0.1 to 0.4. And some other hyper-parameters, i.e. the size of experience replay are also sensitive to the few-shots learning. We believe that a more concrete hyper-paramter search could further improve the performance.
2. (Yu et al., 2021) is not designed for a few-shot learning setting; there are thousands of training data in the original experiments while there are only 10 training data in our setting. However, we believe that we could combine our model with his proposed method of knowledge distillation + knowledge transfer strategy.
3. (Yu et al., 2021) adopts a different formulation of incremental event detection.

We test our meta-learned representation for continual few-shot event detection by using it as the language model initialization for KCN. (KCN+ours), and adding a simple fully connected layer for fine-tuning(ours).

4.4 Experimental Setup

While meta-testing on the ACE05 dataset, we used the MAVEN dataset as the meta-training set and vice versa. We conducted our experiments in 10-shot, 5-shot and 3-shot settings. We use accuracy as the evaluation metric. We choose a simple fully connected layer for the adaptation module. All experiments were conducted on the HAL cluster (Kindratenko et al., 2020). Our implementation has been submitted on Canvas.

4.5 Experimental Results

As expected, in Figure 3 both the fine-tune and pretrain + finetune suffer from catastrophic interference. Their performance dropped significantly after 1 increment in event types. On the other hand, our meta-learned representation module with a simple fine-tune fully connected layer demonstrates strong performance across different few-shot setups. However, it is not able to achieve the same level of performance as KCN and KCN + ours. Our speculation is that simply fine-tuning the model on the meta-learned representation module suffers from some degree of forgetting problem as it lacks the exmapler/memory set to store previous knowledge. Moreover, it cannot learn from previous examples as effectively as KCN since there is no distillation strategy. On the other hand KCN + ours outperforms KCN under the 10 shot setting, and is on par with KCN for 5-shot and 3-shot setup.

4.6 Case Study

We also tested our representation module to see if it can be generalized to larger dataset. We used ACE05 as the meta-training set and MAVEN as the meta testing set.

As we can see, both ours and KCN + ours have worse performance than KCN. We suspect the is-

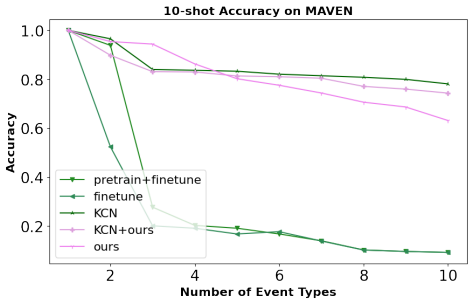

Figure 4: 10-shot results using MAVEN as meta-testing set

sue is that MAVEN is a significantly larger and more diverse dataset compared to ACE05. Our meta-learned representation module might not generalize well enough to give robust performance on the continual few-shot event detection task.

5 Related Works

5.1 Few-shot Event Detection

Many methods have been proposed to address the few-shot event detection problem. Bronstein et al. (2015) identify new event with feature-based method by collecting seed triggers. Deng et al. (2020) divide the few-shot event detection into trigger extraction and few-shot classification. Feng et al. (2020) conduct few-show classification without triggers on sentence-level. Chen et al. (2021) using the causal intervention to address the few-shot problem.

5.2 Continual Learning

Much efforts have been put to study the continual learning problems. Major methods can be divided into three categories or the combination of these categories: significant parameter based methods (Kirkpatrick et al., 2017; Aljundi et al., 2018a); rehearsal based methods (Rebuffi et al., 2017; Hou et al., 2019; Shin et al., 2017; Li and Hoiem, 2017; Cao et al., 2020); and sparse representation methods (Liu et al., 2019; Aljundi et al., 2018b). The significant parameter based methods try to recognize and reserve the significant parameters trained on old data. But it is hard to devise suitable metrics to evaluate the parameters. The rehearsal-based methods try to preserve previous knowledge via storing a few old dataRebuffi et al. (2017); Hou et al. (2019), via generative models learned on old dataShin et al. (2017), via knowledge distillation(Li and Hoiem, 2017). (Cao et al., 2020) indicates that these methods cannot handle semantic ambiguity problem and class imbalance problem in incremental event detection. So they proposed *Knowledge Consolidation Networks* based on both replay-based and knowledge distillation methods to handle these issues. However, they ignore the long-tail distribution problem (Yu et al., 2021).

6 Conclusion

In this paper, we propose a meta-learning based framework that could help address the common issues encountered in continual event detection, i.e. high cost for retrain and catastrophic forgetting, and our base model significantly outperforms the classical method of finetune+pretrain. Also, our proposed mode-agnostic representation modules could be easily integrated to other methods and adapted to other continual information extraction tasks. We found that our model is quite sensitive to hyper-parameters, i.e. learning rate and hidden size in Adaption Module; however, we did not run a concrete search of hyper-parameters due to the limitation of computation resources. Some future works could focus on the tuning of our proposed model. In addition, we could implement the combination of our method with (Yu et al., 2021) to see how much knowledge distillation and knowledge transfer would help improve our method. We could also run experiments on more data to get a more concrete conclusion.

7 Acknowledgements

We really appreciate the help from Professor Heng Ji for allowing us to use the ACE 2005 English dataset, and Pengfei Yu for us formulating the research problem.

References

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

Rahaf Aljundi, Marcus Rohrbach, and Tinne Tuytelaars. 2018b. Selfless sequential learning. *arXiv preprint arXiv:1806.05421*.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Ofer Bronstein, Ido Dagan, Qi Li, Heng Ji, and Anette Frank. 2015. Seed-based event trigger labeling: How far can event descriptions get us? 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 372–376.

Pengfei Cao, Yubo Chen, Jun Zhao, and Taifeng Wang. 2020. Incremental event detection via knowledge consolidation networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 707–717.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2021. Honey or poison? solving the trigger curse in few-shot event detection via causal intervention. *arXiv preprint arXiv:2109.05747*.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. 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 167–176.

Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 2020. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 151–159.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. *arXiv preprint arXiv:2004.13625*.

Rui Feng, Jie Yuan, and Chao Zhang. 2020. Probing and fine-tuning reading comprehension models for few-shot event extraction. *arXiv preprint arXiv:2010.11325*.

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

James Harrison, Apoorva Sharma, Chelsea Finn, and Marco Pavone. 2019. Continuous meta-learning without tasks. *arXiv preprint arXiv:1912.08866*.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. 2019. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 831–839.

Khurram Javed and Martha White. 2019. Meta-learning representations for continual learning. *arXiv preprint arXiv:1905.12588*.

Volodymyr Kindratenko, Dawei Mu, Yan Zhan, John Maloney, Sayed Hadi Hashemi, Benjamin Rabe, Ke Xu, Roy Campbell, Jian Peng, and William Gropp. 2020. Hal: Computer system for scalable deep learning. In *Practice and Experience in Advanced Research Computing*, PEARC '20, page 41–48, New York, NY, USA. Association for Computing Machinery.

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 the national academy of sciences*, 114(13):3521–3526.

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

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651.

Vincent Liu, Raksha Kumaraswamy, Lei Le, and Martha White. 2019. The utility of sparse representations for control in reinforcement learning. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 4384–4391.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2event: Controllable sequence-to-structure generation for end-to-end event extraction. *arXiv preprint arXiv:2106.09232*.

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.

Thien Huu Nguyen, Lisheng Fu, Kyunghyun Cho, and Ralph Grishman. 2016. A two-stage approach for extending event detection to new types via neural networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 158–165.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

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

Mark Bishop Ring et al. 1994. Continual learning in reinforcement environments.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. *arXiv preprint arXiv:1705.08690*.

Sebastian Thrun. 1998. Lifelong learning algorithms. In *Learning to learn*, pages 181–209. Springer.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. MAVEN: A massive general domain event detection dataset. In *Proceedings of EMNLP 2020*.

Pengfei Yu, Heng Ji, and Prem Natarajan. 2021. Lifelong event detection with knowledge transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5278–5290.