Continual Few-Shot Named Entity Recognition via Data-Free Distillation

Abstract

Previous work in continual learning for Named Entity Recognition (NER) relies on the assumption that there exists abundance of labeled data in the new datasets arriving over time. This assumption is usually unrealistic since the token-level annotations required by NER training are laborious and scarce, especially for new (unseen) classes. We present the first work to study continual few-shot learning for NER, which is more general, but as a result, more challenging, compared to continual learning for NER. To alleviate the problem of catastrophic forgetting in continual few-shot learning, we reconstruct synthetic training data of the previously seen classes from the NER model and further develop a framework that distills from the existing model with both synthetic data, and real data from the current training set. Experimental results on several NER benchmarks show that our approach achieves significant improvements over existing baselines.

1 Introduction

Existing Named Entity Recognition (NER) models are usually trained on a large scale dataset with predefined entity classes, then deployed for entity extraction on the test data without further adaptation or refinement. However, real-world applications are usually dynamically evolving, *i.e.*, NER models may be expected to continually learn new entity classes as required by the users, but classes that were not initially available for training. In this case, one challenge is that the training data of old entity classes may not be available due to privacy concerns or memory limitations [7]. Then, the model can easily degrade in terms of the performance of existing classes when being fine-tuned with only annotations of new entity classes, *i.e.*, *catastrophic forgetting*. In addressing this problem, previous work in continual learning for NER [9] regularizes the current model by distilling from the previous model trained on old (existing) classes, using text from the training dataset of new classes. However, this requires abundance of data in the new dataset being used for distillation. Such an assumption is usually unrealistic since the token-level annotations required by NER training are labor-consuming and scarce, especially for the new unseen classes. In this paper, we study a more realistic setting, *i.e.*, continual few-shot learning for NER, where the model (i) is continually learning on new classes with few annotations, and (ii) does not require access to training data for old classes.

Compared with the setting of continual learning, continual few-shot learning for NER is more challenging scenario. First, few-shot datasets in continual few-shot learning may not contain enough information for the trained model to generalize during testing. Second, it is more challenging to solve the catastrophic forgetting problem in continual few-shot learning. In continual learning for NER [9], the same training sequence may contain entities of different classes. Therefore, when the training dataset for new classes is sufficiently large, its context, *i.e.*, words labeled as not from entities of new classes, will likely also contain abundant entities of the old classes. That is, the new training data can be regarded as an unlabeled *replay* dataset of the existing entity classes. With such a replay, catastrophic forgetting can be simply addressed by distilling from the previous model [9]. However, in continual few-shot learning, there may not exist sufficient (if any) entities of the old classes for distillation with the only few samples in the current dataset.

Efficient Natural Language and Speech Processing Workshop at the 35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

In this paper, we propose a framework to enable continual few-shot learning for NER. Provided there is not enough data samples from old classes for replay with few-shot datasets, compared with the setting of continual learning for NER, we consider generating synthetic replay of old classes by inverting the NER model. Specifically, given the current model that has been trained on the old classes, we optimize the token embeddings of the synthetic data being generated, so that predictions from the existing model can contain old entity classes. To ensure the synthetic (reconstructed) data to be realistic, we propose to adversarially match the hidden features of tokens from the synthetic data and those from the readily available training text for new classes. Consequently, the synthetic data will encourage knowledge preservation of old classes, considering that it is obtained from the previous model that was trained on the old classes. Further, since the synthetic data can be constructed in a large amount, *e.g.*, thousands of sentences, it will provide more diverse context for training of the current model, thus preventing over fitting to few samples of the new classes. With the generated synthetic data, we propose a framework that trains the NER model with annotations of the new classes, while distilling from the previous model with both the synthetic data and real text from the new training data. Our contributions are summarized as follows:

- To the best of our knowledge, this is the first work to study continual few-shot learning for NER, which is more practical and challenging, compared with continual learning for NER.
- We approach the problem by proposing a framework that distills from the existing model with both, real data of new entity classes and synthetic data reconstructed from the model as a replay of old entity classes. Experimental results demonstrate that our approach significantly improves over existing baselines.

2 Problem Definition

Assume there is a stream of NER datasets $D^1, \ldots, D^t$, at time steps t and annotated with disjoint entity classes, where $D^t = \{(X_i^t, Y_i^t)\}_{i=1}^{|D^t|}$, for $t \geq 1$ contains c^t entity classes. Here $X_i^t = [x_{i,1}^t, \cdots, x_{i,N_i}^t]$ and $y_i^t = [y_{i,1}^t, \cdots, y_{i,N_i}^t]$ are the NER token and label sequences, respectively, with length N_i, and $|D^t|$ is the size of the dataset. Dataset D^1 is the base dataset, assumed of reasonably large size for the base classes. The datasets $\{D^t\}_{t\geq 1}$ are the few-shot datasets with about K samples for each class. In continual few-shot learning, the NER model will be incrementally trained with $D^1, D^2, \ldots$, over time, with data from D^t only available at the t^{th} time step. Figure 1(a) shows an example of annotations for different steps of continually few-shot learning on classes of *PER*, *LOC*, and *TIME*. After being trained with D^t, the model will be evaluated jointly on all entity classes encountered in $D^1, \cdots, D^t$, *i.e.*, we do not learn separate prediction modules for each time step.

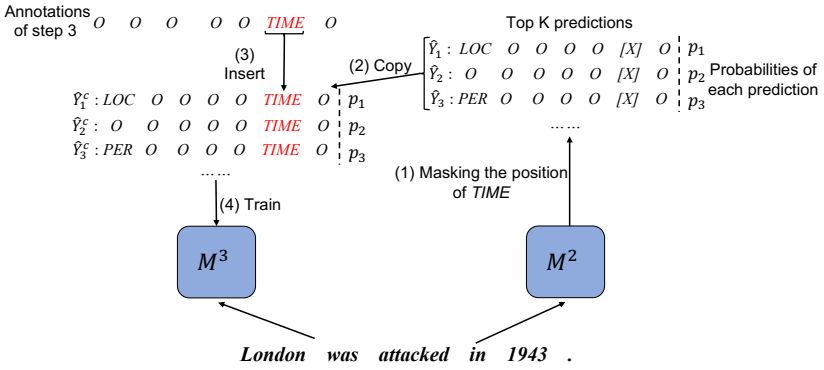
3 Framework for Training Without Forgetting

Following [1, 10], we employ the *BERT-CRF* model for NER training, which consists of a BERT$_{base}$ [3] encoder with a linear projection and a conditional random field (CRF) [6] layer for prediction. For time step $t > 1$, the model M^t is expected to learn about the new classes from D^t, while not forgetting the knowledge from $\{D^k\}_{k=1}^{t-1}$. Assume we have already obtained a synthetic dataset $D_r^t = \{E_i^{t,r}, Y_i^{t,r}\}_{i=1}^{|D_r^t|}$ of previous entity classes from $\{D^k\}_{k=1}^{t-1}$, where $E_i^{t,r} = [e_{i,1}^{t,r}, \cdots, e_{i,N}^{t,r}]$ and $Y_i^{t,r} = [y_{i,1}^{t,r}, \cdots, y_{i,N}^{t,r}]$ are the reconstructed token embeddings and reference label sequence that contains annotations of only old classes. We will discuss the construction of the synthetic D_r^t below. Given the current training data D^t and M^{t-1} that has been trained on D^{t-1}, we propose to train M^t by distilling from M^{t-1} with both the real data from D^t and synthetic data from D_r^t.

Distilling with Real Data D^t: The distillation from M^{t-1} to M^t involves matching the output distributions between M^t to M^{t-1}. However, let X be an input sequence from D^t at step t, the CRF layer outputs correspond to a sequence-level distribution $P_{M^t}(Y|X)$, *i.e.*, probabilities for all possible label sequences of X, the cardinality of which, grows exponentially large with the length of X. Following the current state-of-the-art approach of NER distillation [12], we approximate the sequence-level output distribution of CRF with only its top S predictions. Specifically, for model

Input Sequence:	*Emily*	*from*	*California*	*was*	*born*	*in*	*1990*	.
Step 1 :	*PER*	*O*	*O*	*O*	*O*	*O*	*O*	*O*
Step 2 :	*O*	*O*	*LOC*	*O*	*O*	*O*	*O*	*O*
Step 3 :	*O*	*O*	*O*	*O*	*O*	*O*	*TIME*	*O*
Prediction :	*PER*	*O*	*LOC*	*O*	*O*	*O*	*TIME*	*O*

(a)



(b)

Figure 1: (a)An example of annotations and expected model predictions in continual few-shot learning. In our experiments, we do not assume the same sentence is shared by datasets from different time steps. (b) Distilling with D^3 of (a). M^2 and M^3 are models of step 2 and 3, respectively. We replace predictions on the position of "1943" from M^2 with "TIME" from D^3 before training on M^3.

M^{t-1}, we have,

$$\hat{P}_{M^{t-1}}(Y|X) = [P_{M^{t-1}}(\hat{Y}_1|X), \cdots, P_{M^{t-1}}(\hat{Y}_S|X), 1 - \sum_{s=1}^{S} P_{M^{t-1}}(\hat{Y}_s|X)], \tag{1}$$

where $\{\hat{Y}_s\}_{s=1}^S$ are the top S most probable predictions of label sequence from M^{t-1}. In this way, the output from the CRF of M^{t-1} becomes tractable. However, M^t still cannot be trained with such an output from M^{t-1}. This is because M^{t-1} was not trained with the new classes in D^t. Therefore, when X is from D^t, M^{t-1} will have wrong predictions on the tokens labeled as being from entities of new classes. In order to distill with M^{t-1}, we propose a correction for $\{\hat{Y}_s\}_{s=1}^S$. Figure 1(b) shows an example of such a process. Specifically, on the positions of the sequence where D^t has labeled as new classes, we replace the predictions in $\{\hat{Y}_s\}_{s=1}^S$ with the annotations from D^t. We denote the corrected set of predictions as $\{\hat{Y}_s^c\}_{s=1}^S$. For training of M^t, we first calculate the predicted distribution of M^t with respect to $\{\hat{Y}_s^c\}_{s=1}^S$, as

$$\hat{P}_{M^t}(Y|X) = [P_{M^t}(\hat{Y}_1^c|X), \cdots, P_{M^t}(\hat{Y}_S^c|X), 1 - \sum_{s=1}^{S} P_{M^t}(\hat{Y}_s^c|X)], \tag{2}$$

where we compute the predicted probabilities from M^t with regard to $\{\hat{Y}_s^c\}_{s=1}^S$ from M^{t-1}. Then, M^t can be trained by minimizing the cross entropy between $\hat{P}_{M^{t-1}}(Y|X)$ and $\hat{P}_{M^t}(Y|X)$ via

$$L^{real}(D^t) = -\frac{1}{|D^t|} \sum_{X \in D^t} CE(\hat{P}_{M^{t-1}}(Y|X), \hat{P}_{M^t}(Y|X)), \tag{3}$$

where $CE(\cdot, \cdot)$ is the cross entropy function. Note that the definition of *O* is different in M^{t-1} and M^t. Take Figure 1(b) as an example, the prediction of *O* in step 2 corresponds to both *O* and *TIME* for step 3, since *TIME* is not in the target entity classes of step 2. However, we know that tokens

annotated as *O* in step 3 are not *TIME*. Therefore, we can safely copy the prediction of *O* in $\{\hat{Y}_s^c\}_{s=1}^S$ from M^2 for training of M^3.

Distilling with Synthetic Data D_r^t: Different from data from D^r, in which we know tokens annotated as *O* are not from the new classes, data from D_r^t is reconstructed from M^{t-1} and only contains labels for the previous classes. In this case, any token predicted with "O" from M^{t-1} can be potentially labeled as *O* or the new classes by M_t. Therefore, with D_r^t, it is unclear how to correct the output of CRF from M^{t-1}, *i.e.* $\{\hat{Y}_s\}_{s=1}^S$, for training of M^t. Here, we resort to another approach that decomposes the output from CRF, *i.e.*, sequence level label distribution, into marginal label prediction for each token, using the forward-backward method in [6]. For each token with embedding e, let $p_e^t = [p_{e,O}^t; p_{e,C^{t-1}}^t; p_{e,c^t}^t]$ and $p_e^{t-1} = [p_{e,O}^{t-1}; p_{e,C^{t-1}}^{t-1}]$ be the predicted marginal distribution on the token from M^t and M^{t-1} with $C^{t-1} = \sum_{k<t} c^k$. $p_{e,O}^t, p_{e,O}^{t-1} \in \mathbb{R}$ are the probabilities for class O, whereas $p_{e,C^{t-1}}^t, p_{e,C^{t-1}}^{t-1} \in \mathbb{R}^{C^{t-1}}$ are the probabilities for entity classes encountered up to step $t-1$. Further, $p_{e,c^t}^t \in \mathbb{R}^{c^t}$ are probabilities for the new classes in step t. We first collapse p_e^t by computing $\hat{p}_e^t = [\mathrm{sum}([p_{e,O}^t; p_{e,c^t}^t]); p_{e,C^{t-1}-1}^t]$, where we merge the predictions of *O* and c^t new classes. In this way, $\hat{p}_e^t$ will have the same dimension as p_e^{t-1}. Let E_r^t be the set of token embeddings for all tokens contained in D_r^t and $KL(\cdot||\cdot)$ is the KL divergence. The distillation loss for D_r^t is

$$L^{syn}(D_r^t) = \mathbb{E}_{e \in E_r^t} KL(\hat{p}_e^t || p_e^{t-1}), \tag{4}$$

General Objective: The general objective of M^t for training at step t is given by

$$L^t = L^{real}(D^t) + \alpha L^{syn}(D_r^t), \tag{5}$$

where α is a parameter balancing training between real data from D^t and synthetic data from D_r^t.

4 Synthetic Data Reconstruction

Now we describe how to reconstruct D_r^t from M^{t-1}. Given a randomly sampled label sequence Y containing the old entity classes from $\{D^k\}_{k<t}$, we seek to reconstruct the embedding sequence E corresponding to its training data. In doing so, we randomly initialize embeddings E, then optimize the parameters of E with gradient descent so that its output with M^{t-1} matches the expected label sequence Y. Formally, we optimize E by minimizing the training loss of the CRF as

$$L^{crf} = -\log P_{M^{t-1}}(Y|E). \tag{6}$$

One problem of such reconstruction is that *the resulting synthetic E may not be realistic.* Here, we propose to encourage synthetic data to be more realistic by leveraging the real data from D^t.

Let $h^{l,syn}(E_r^t)$ be the hidden state from the l^{th} layer of the BERT encoder in M^{t-1}, regarding the set of synthetic token embeddings, E_r^t, from D_r^t. Similarly, let $h^{l,real}(emb(X^t))$ be the output hidden states from the l^{th} layer of M^{t-1}, regarding the set of real tokens, X^t, from D^t. Moreover, $emb(\cdot)$ is the embedding layer. We propose to adversarially match $h^{l,syn}(E_r^t)$ and $h^{l,real}(emb(X^t))$ so that hidden states from the real and synthetic are not far away from each other. In this way, the reconstructed embeddings from D_r^t are likely to be more realistic. Specifically, let F^l be a binary discriminator module, *i.e.*, one layer linear projection with sigmoid output, whose inputs are the real and synthetic the hidden states,

$$\begin{aligned} F^{l,*} = \ & \mathrm{argmin}_{F^l} -\mathbb{E}_{h \in h^{l,syn}(E_r^t)} \log F^l(h) - \mathbb{E}_{h \in h^{l,real}(emb(X^t))} \log(1 - F^l(h)), \\ & L_{adv}^l = \mathbb{E}_{h \in h^{l,syn}(E_r^t)} \log(1 - F^{l,*}(h)). \end{aligned} \tag{7}$$

Consequently, the final loss for reconstructing D_r^t is

$$L^r = L^{crf} + \beta L^{adv}, \tag{8}$$

where $L^{adv} = \sum_{l \in l^s} L_{adv}^l$. $l^s = 2, 4, \cdots, 12$, *i.e.*, we match every two layers of the BERT encoder in M^{t-1}. β is a balancing parameter.

Another problem we should consider is that *the real data D^t and synthetic data D_r^t may contain different sets of entity classes*, *i.e.*, the few-shot dataset D^t may not contain entities of old classes

Table 1: 5-shot learning with CoNLL2003.

Method	Step 1	Step 2	Step 3	Step 4	Avg ≥ 2
Continual NER	87.89	59.54	51.09	42.98	51.20
EWC++	88.35	68.23	60.34	50.97	59.85
FSLL	88.35	68.49	61.66	52.71	60.95
AS-DFD	88.35	68.87	60.32	52.99	60.73
Ours ($\alpha = 0$)	88.35	60.09	52.16	44.31	52.19
Ours ($\beta = 0$)	88.35	69.11	60.54	53.86	61.13
Ours (all tokens)	88.35	69.78	62.33	58.74	63.62
Ours	88.35	**71.31**	**63.76**	**59.37**	**64.18**

Table 2: 10-shot learning with CoNLL2003.

Method	Step 1	Step 2	Step 3	Step 4	Avg ≥ 2
Continual NER	87.89	59.77	54.03	46.94	53.58
EWC++	88.35	66.32	62.69	55.14	61.38
FSLL	88.35	68.34	63.59	56.00	62.71
AS-DFD	88.35	68.95	59.54	53.22	60.57
Ours ($\alpha = 0$)	88.35	60.26	55.46	47.69	54.47
Ours ($\beta = 0$)	88.35	69.60	60.56	54.59	61.68
Ours (all tokens)	88.35	70.26	61.25	58.69	63.40
Ours	88.35	**70.75**	**64.60**	**60.02**	**65.12**

in D_r^t. In this case, for the token embeddings of old classes in D_r^t, *s.t.*, $\{e_{i,j}|y_{i,j}^{t,r} \neq O\}$, matching the hidden states of these embeddings with those from D^t will distract these embedding from being optimized into the entities of old classes, which we will show in the experiments. Therefore, we overload the definition of E_r^t in (4) by excluding embeddings of the old entity classes in D_r^t from matching, *i.e.*, $E_r^t = \{e_{i,j}|y_{i,j}^{t,r} = O\}$, while X^t contains all the real tokens from D^t. Algorithm 1 in the Supplementary material shows the complete procedure for constructing D_r^t.

Since D_r^t contains entities of old classes from previous steps, distilling with $L^{syn}(D_r^t)$ will help preserving knowledge of old entity classes, *i.e.*, avoiding catastrophic forgetting, without accessing the real data from previous steps. Additionally, with D_r^t, M^t is no longer trained with only few samples from D^t, thus is less likely to overfit. This is because D_r^t can be constructed in a relative larger scale, *e.g.*, several hundred sentences, within the computation limit. Compared with training only with D^t, D_r^t provides more diverse text information for M^t during training. Moreover, the entity of old classes from D_r^t can be regarded as negative samples for training of the new classes in D^t, reducing the confusion between old and new classes for M^t during training.

5 Related Work

Continual Learning: [9] studies continual learning with different classes, building a unified NER classifier for all the classes encountered over time. There are two problems regarding this method. Firstly, [9] only works with a non-CRF-based model. Secondly, it assumes that a large amount of data for the new classes is available, which is unrealistic since annotations for unseen classes are usually scarce. In this work, we assume only few-shot datasets are available for the new classes, *i.e.*, continual few-shot learning, which was proposed in [11], yet not studied in the context of NER.

Few-Shot Learning: The current state-of-the-art works on few-shot learning for NER can be categorized as metric-learning-based method and data-augmentation-based method. The former involves predicting by learning to compare token features with class prototypes [4] or stored tokens from the query set [13]. These works build a separate prediction module for the target classes and ignore the performance of base classes during evaluation, thus incompatible with continual learning. On the other hand, [5] avoids overfitting of few-shot learning by augmenting with noisy or unlabeled data. Our approach is similar to [5], in that we augment few-shot training of the current step with the reconstructed data from the model of the previous step.

6 Experiments

6.1 Datasets

Following the previous work of continual learning for NER [9], we experiment with two datasets: CoNLL2003 and Ontonote 5.0. Table 1 and 2 list the entity classes used for each step for continual few-shot learning. CoNLL2003 contains 4 entity classes. We experiment with one entity class for each step and consider the average performance of eight permutations of ordering as in [9].

Baselines and Ablation Study: We compare with the state-of-the art work of continual learning for NER (*Continual NER*) [9].Additionally, we implement EWC++ [2] with $\alpha = 0$, *i.e.*, using weights regularization to avoid forgetting instead of synthetic data from D_r^t. We also implement FSLL [8], a state-of-the-art method of continual few-shot learning for **image classification** with metric learning. We further include AS-DFD [7], the state-of-the-art method of data-free distillation in **text classification**. Specifically, we construct D_r^t with the adversarial regularization described in AS-DFD instead of (8). The ablation study also includes *Ours (all tokens)*, which matches all the

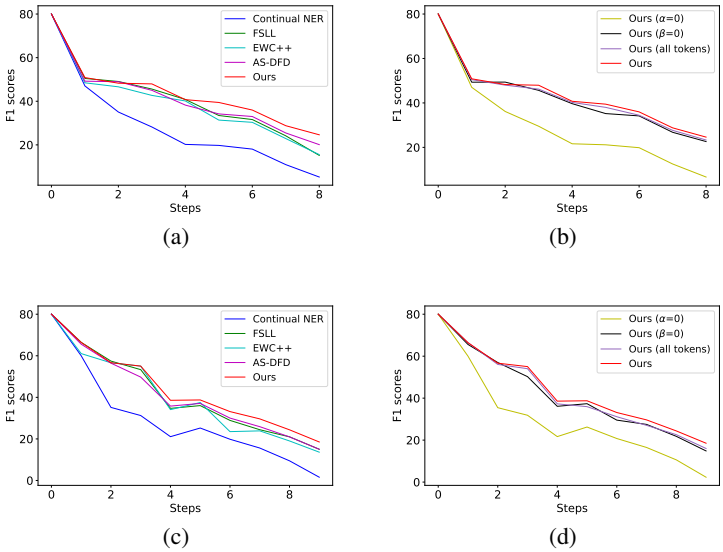


Figure 2: 5-shot continual few-shot learning for OntoNote 5.0 data. (a) and (b) are comparison with baselines and ablation studies, respectively, for combination P_1 (see Table 2 in the supplementary material). Similarly, (c) and (d) are results for P_2.

synthetic tokens in D_r^t with real tokens in D^t, instead of matching with only those labeled as *O* in D_r^t, as described after eq (8).

6.2 Results of Continual Few-Shot Learning

Table 1 and 2 shows the F1 scores of 5-shot and 10-shot from different steps of continual few-shot learning. Figure 2 shows the 5-shot results for OntoNote 5.0. Our methods outperforms all the considered baselines. Especially, *Continual NER* [9] has the worst result among all the methods. This is because the performance of *Continual NER* relies on a large amount of data from D^t for replay of previous entities. Therefore, it does not work well in the few-shot scenario, where D^t with only few samples may not contain entities of old classes for replay. Additionally, we find that the performance of *AS-DFD* [7] is slightly lower than *Ours* ($\beta = 0$), *i.e.*, distilling using data reconstructed with only L^{crf}. *AS-DFD* is designed for text classification, where they use the feature of the special token *[CLS]* from BERT for classification, while features of the non-special tokens (within text) are trained with an augmented task of language modeling. However, in NER, features of the non-special tokens are directly used for prediction. Thus, simultaneously training such features with language modeling may distract the model from learning the task specific information of NER. In the ablation study, we find that our adversarial matching indeed improves the quality of the synthetic data (*Ours* v.s. *Ours* ($\beta = 0$)), especially when excluding tokens of the reconstructed old entities from matching (*Ours* v.s. *Ours (all tokens)*).

7 Conclusion

In this work, we study continual few-shot learning for NER, which is a more challenging but practical scenario compared to continual learning of NER. To address the problem of catastrophic forgetting, we proposed to reconstruct synthetic training of the old entity classes from the model trained at the previous time step. Then, we proposed a framework that distills with both real data from the current training dataset and the synthetic data. Distilling with the synthetic data helps preserving knowledge of the old entity classes. Additionally, the synthetic data allows the model to be trained with a more diverse context, thus less likely to overfit to the few training samples of current step. Experimental results showed that our method outperforms existing baselines.

References

[1] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

[2] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547, 2018.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. *arXiv preprint arXiv:2006.05702*, 2020.

[5] Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. Few-shot named entity recognition: A comprehensive study. *arXiv preprint arXiv:2012.14978*, 2020.

[6] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[7] Xinyin Ma, Yongliang Shen, Gongfan Fang, Chen Chen, Chenghao Jia, and Weiming Lu. Adversarial self-supervised data-free distillation for text classification. *arXiv preprint arXiv:2010.04883*, 2020.

[8] Pratik Mazumder, Pravendra Singh, and Piyush Rai. Few-shot lifelong learning. *arXiv preprint arXiv:2103.00991*, 2021.

[9] Natawut Monaikul, Giuseppe Castellucci, Simone Filice, and Oleg Rokhlenko. Continual learning for named entity recognition. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*, 2021.

[10] Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. Portuguese named entity recognition using bert-crf. *arXiv preprint arXiv:1909.10649*, 2019.

[11] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12183–12192, 2020.

[12] Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Fei Huang, and Kewei Tu. Structure-level knowledge distillation for multilingual sequence labeling. *arXiv preprint arXiv:2004.03846*, 2020.

[13] Yi Yang and Arzoo Katiyar. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. *arXiv preprint arXiv:2010.02405*, 2020.