Effective Unsupervised Domain Adaptation with Adversarially Trained Language Models

Thuy-Trang Vu[◇] **Dinh Phung**[†] **Gholamreza Haffari**[†]
Faculty of Information Technology, Monash University, Australia
[◇]{trang.vuthithuy},[†]{first.last}@monash.edu

Abstract

Recent work has shown the importance of adaptation of broad-coverage contextualised embedding models on the domain of the target task of interest. Current self-supervised adaptation methods are simplistic, as the training signal comes from a small percentage of *randomly* masked-out tokens. In this paper, we show that careful masking strategies can bridge the knowledge gap of masked language models (MLMs) about the domains more effectively by allocating self-supervision where it is needed. Furthermore, we propose an effective training strategy by adversarially masking out those tokens which are harder to reconstruct by the underlying MLM. The adversarial objective leads to a challenging combinatorial optimisation problem over *subsets* of tokens, which we tackle efficiently through relaxation to a variational lower-bound and dynamic programming. On six unsupervised domain adaptation tasks involving named entity recognition, our method strongly outperforms the random masking strategy and achieves up to +1.64 F1 score improvements.

1 Introduction

Contextualised word embedding models are becoming the foundation of state-of-the-art NLP systems (Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019; Raffel et al., 2019; Brown et al., 2020; Clark et al., 2020). These models are pretrained on large amounts of raw text using self-supervision to reduce the labeled data requirement of target tasks of interest by providing useful feature representations (Wang et al., 2019a). Recent work has shown the importance of further training of pre-trained masked language models (MLMs) on the target domain text, as the benefits of their contextualised representations can deteriorate substantially in the presence of domain mismatch (Ma et al., 2019; Xu et al., 2019; Wang et al., 2019c; Gururangan et al., 2020). This is particularly crucial in unsupervised domain adaptation (UDA), where there is no labeled data in the target domain (Han and Eisenstein, 2019) and the knowledge from source domain labeled data is transferred to the target domain via a common representation space. However, current self-supervised adaptation methods are simplistic, as the training signal comes from a small percentage of *randomly* masked-out tokens. Thus, it remains to investigate whether there exist more effective self-supervision strategies to bridge the knowledge gap of MLMs about the domains to yield higher-quality adapted models.

A key principle of UDA is to learn a common embedding space of both domains which enables transferring a learned model on source task to target task. It is typically done by further pretraining the MLM on a combination of both source and target data. Selecting relevant training examples has been shown to be effective in preventing the negative transfer and boosting the performance of adapted models (Moore and Lewis, 2010; Ruder and Plank, 2017). Therefore, we hypothesise that the computational effort of the further pretraining should concentrate more on learning words which are specific to the target domain or undergo semantic/syntactic shifts between the domains.

In this paper, we show that the adapted model can benefit from careful masking strategy and propose an adversarial objective to select subsets for which the current underlying MLM is less confident. This objective raises a challenging combinatorial optimisation problem which we tackle by optimising its variational lower bound. We propose a training algorithm which alternates between tightening the variational lower bound and learning the parameters of the underlying MLM. This involves proposing an efficient dynamic programming (DP) algorithm to sample from the distribution over the

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 6163–6173,
November 16–20, 2020.

space of masking subsets, and an effective method based on Gumbel softmax to differentiate through the subset sampling algorithm.

We evaluate our adversarial strategy against the random masking and other heuristic strategies including POS-based and uncertainty-based selection on UDA problem of six NER span prediction tasks. These tasks involve adapting NER systems from the news domain to financial, twitter, and biomedical domains. Given the same computational budget for further self-supervising the MLM, the experimental results show that our adversarial approach is more effective than the other approaches, achieving improvements up to +1.64 points in Fscore and +2.23 in token accuracy compared to the random masking strategy.

2 Uunsupervised DA with Masked LMs

UDA-MLM. This paper focuses on the UDA problem where we leverage the labeled data of a related source task to learn a model for a target task without accessing to its labels. We follow the two-step UDA procedure proposed in AdaptaBERT consisting of a domain tuning step to learn a common embedding space for both domains and a task tuning step to learn to predict task labels on source labeled data (Han and Eisenstein, 2019). The learned model on the source task can be then zero-shot transferred to the target task thanks to the assumption that these tasks share the same label distribution.

This domain-then-task-tuning procedure resembles the pretrain-then-finetuning paradigm of MLM where the domain tuning shares the same training objective with the pretraining. In domain tuning step, off-the-shelf MLM is further pretrained on an equal mixture of randomly masked-out source and target domain data.

Self-Supervision. The training principle of MLM is based on self-supervised learning where the labels are automatically generated from unlabeled data. The labels are generated by covering some parts of the input, then asking the model to predict them given the rest of the input.

More specifically, a subset of tokens is sampled from the original sequence $\boldsymbol{x}$ and replaced with [MASK] or other random tokens (Devlin et al., 2019).[1] Without loss of generality, we assume that all sampled tokens are replaced with [MASK]. Let us denote the set of masked out indices by S, the ground truth tokens by $\boldsymbol{x}_S = \{x_i | i \in S\}$, and the resulting *puzzle* by $\boldsymbol{x}_{\bar{S}}$ which is generated by masking out the sentence tokens with indices in S. The training objective is to minimize the negative log likelihood of the ground truth,

$$\min_{\theta} - \sum_{\boldsymbol{x} \in \mathcal{D}} \log Pr(\boldsymbol{x}_S | \boldsymbol{x}_{\bar{S}}; \mathcal{B}_\theta) \quad (1)$$

where $\mathcal{B}_\theta$ is the MLM parameterised by θ, and $\mathcal{D}$ is the training corpus.

3 Adversarially Trained Masked LMs

Given a finite computational budget, we argue that it should be spent wisely on new tokens or those having semantic/syntactic shifts between the two domains. Our observation is that such tokens would pose more challenging puzzles to the MLM, i.e. the model is less confident when predicting them. Therefore, we propose to strategically select subsets for which the current underlying MLM $\mathcal{B}_\theta$ is less confident about its predictions:

$$\min_{\theta} \max_{S \in \mathcal{S}_K} - \log Pr(\boldsymbol{x}_S | \boldsymbol{x}_{\bar{S}}; \mathcal{B}_\theta) \quad (2)$$

Henceforth, we assume that the size of the masked set K for a given sentence $\boldsymbol{x}$ is fixed. For example in BERT (Devlin et al., 2019), K is taken to be $15\% \times |\boldsymbol{x}|$ where $|\boldsymbol{x}|$ denotes the length of the sentence. We denote all possible subsets of indices in a sentence with a fixed size by $\mathcal{S}_K$.

3.1 Our Variational Formulation

The masking strategy learning problem described in eqn (2) is a minimax game of two players: the puzzle generator to select the subset resulting in the most challenging puzzle, and the MLM $\mathcal{B}_\theta$ to best solve the puzzle by reconstructing the masked tokens correctly. As optimising over the subsets is a hard combinatorial problem over the discrete space of $\mathcal{S}_K$, we are going to convert it to a continuous optimisation problem.

We establish a variational lower bound of the objective function over S using the following inequality,

$$\max_{S \in \mathcal{S}_K} - \log Pr(\boldsymbol{x}_S | \boldsymbol{x}_{\bar{S}}; \mathcal{B}_\theta) \geq \quad (3)$$

$$\max_{\phi} \sum_{S \in \mathcal{S}_K} -q(S|\boldsymbol{x}; \pi_\phi) \log Pr(\boldsymbol{x}_S | \boldsymbol{x}_{\bar{S}}; \mathcal{B}_\theta) \quad (4)$$

[1] In BERT implementation, 15% tokens in $\boldsymbol{x}$ are selected; among them 80% are replaced with [MASK], 10% are replaced with random tokens, and 10% are kept unchanged.

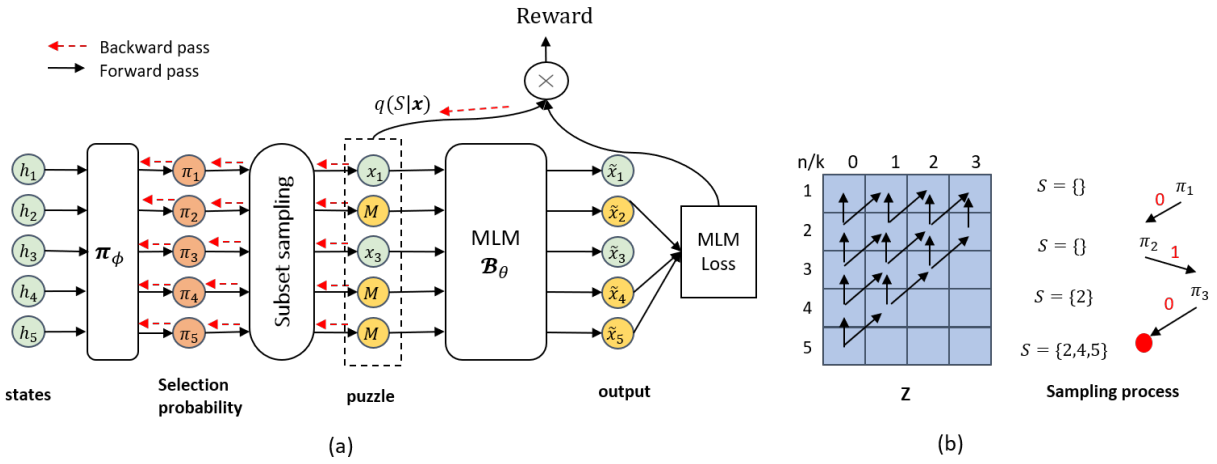

Figure 1: (a) Our adversarial learned masking strategy for MLM includes a puzzle generator to estimate selection probability, a subset sampling procedure and the MLM model. The red dash arrow shows the gradient flow when updating the puzzle generator. (b) Masked subset sampling procedure with dynamic programming.

where $q(.)$ is the variational distribution provided by a neural network π_ϕ. This variational distribution $q(S|\boldsymbol{x};\pi_\phi)$ estimates the distribution over all subset of size K. It is straightforward to see that the weighted sum of negative log likelihood of all possible subsets is always less than the max value of them. Our minimax training objective is thus,

$$\min_\theta \max_\phi \sum_{S\in\mathcal{S}_K} -q(S|\boldsymbol{x};\pi_\phi)\log Pr(\boldsymbol{x}_S|\boldsymbol{x}_{\bar{S}};\mathcal{B}_\theta) \quad (5)$$

$$q(S|\boldsymbol{x},\pi_\phi) = \prod_{i\in S}\pi_\phi(i|\boldsymbol{x})\prod_{i'\notin S}(1-\pi_\phi(i'|\boldsymbol{x}))/\mathcal{Z} \quad (6)$$

where $\mathcal{Z}$ is the partition function making sure the probability distribution sums to one,

$$\mathcal{Z} = \sum_{S'\in\mathcal{S}_K}\prod_{i\in S'}\pi_\phi(i|\boldsymbol{x})\prod_{i'\notin S'}(1-\pi(i'|\boldsymbol{x})). \quad (7)$$

The number of possible subsets is $|\mathcal{S}_K| = \binom{|\boldsymbol{x}|}{K}$, which grows exponentially with respect to K. In §4, we provide efficient dynamic programming algorithm for computing the partition function and sampling from this exponentially large combinatorial space. In the following, we present our model architecture and training algorithm for the puzzle generator ϕ and MLM θ parameters based on the variational training objective in eqn (5).

3.2 Model Architecture

We learn the masking strategy through the puzzle generator network as shown in Figure 1. It is a feed-forward neural network assigning a selection probability $\pi_\phi(i|\boldsymbol{x})$ for each index i given the original sentence $\boldsymbol{x}$, where ϕ denote the parameters. Inputs to the puzzle generator are the feature representations $\{\boldsymbol{h}_i\}_{i=1}^n$ of the original sequence $\{\boldsymbol{x}_i\}_{i=1}^n$. More specifically, they are output of the last hidden states of the MLM. The probability of perform masking at position i is computed by applying sigmoid function over the feed-forward net output $\pi_\phi(i|\boldsymbol{x}) = \sigma(\text{FFNN}(\boldsymbol{h}_i))$. From these probabilities, we can sample the masked positions in order to further train the underlying MLM $\mathcal{B}_\theta$.

3.3 Optimising the Variational Bound

We use an alternating optimisation algorithm to train the MLM $\mathcal{B}_\theta$ and the puzzle generator π_ϕ (Algorithm 1). The update frequency for π_ϕ is determined via a mixing hyperparameter β.

Training the MLM. Fixing the puzzle generator, we can train the underlying MLM model using gradient descent on MLM objective in eqn (1),

$$\min_\theta \mathbb{E}_{q(S|\boldsymbol{x};\pi_\phi)}[-\log Pr(\boldsymbol{x}_S|\boldsymbol{x}_{\bar{S}};\mathcal{B}_\theta)] \quad (8)$$

where we approximate the expectation by sampling. That is, $\mathbb{E}_{q(S|\boldsymbol{x};\pi_\phi)}[-\log Pr(\boldsymbol{x}_S|\boldsymbol{x}_{\bar{S}};\mathcal{B}_\theta)]$ is approximated by

$$\frac{1}{M}\sum_{m=1}^{M} -\log Pr(\boldsymbol{x}_{S_m}|\boldsymbol{x}_{\bar{S}_m};\mathcal{B}_\theta) \quad (9)$$

where $S_m \sim q(S|\boldsymbol{x};\pi_\phi)$. In §4.2, we present an efficient sampling algorithm based on a sequential decision making process involving discrete choices, i.e. whether to include an index i or not.

Algorithm 1 Adversarial Training Procedure

Input: data $\mathcal{D}$, update freq. β, masking size K
Output: generator π_ϕ, MLM $\mathcal{B}_\theta$

Let $\phi \leftarrow \phi_0; \theta \leftarrow \theta_0$
while stopping condition is not met **do**
  for $\boldsymbol{x} \in \mathcal{D}$ **do**
    $S, q(S) \leftarrow \text{subsetSampling}(\boldsymbol{x}, \pi_\phi, K)$
    Update the MLM using Eq. (8)
    if coinToss(β)==Head **then**
      Compute reward
      $r \leftarrow -\log Pr(\boldsymbol{x}_S|\boldsymbol{x}_{\bar{S}}; \mathcal{B}_\theta)$
      Update the generator using Eq. (10)
    end if
  end for
end while
return θ, ϕ

Training the Puzzle Generator. Fixing the MLM, we can train the puzzle generator by considering $-\log Pr(\boldsymbol{x}_S|\boldsymbol{x}_{\bar{S}}; \mathcal{B}_\theta)$ as the reward, and aim to optimise the expected reward,

$$\max_{\phi} \mathbb{E}_{q(S|\boldsymbol{x};\pi_\phi)}[-\log Pr(\boldsymbol{x}_S|\boldsymbol{x}_{\bar{S}}; \mathcal{B}_\theta)]. \quad (10)$$

We may aim to sample multiple index sets $\{S_1, .., S_M\}$ from $q(S|\boldsymbol{x}; \pi_\phi)$, and then optimise the parameters of the puzzle generator by maximizing the Monte Carlo estimate of the expected reward. However, as sampling each index set S_m corresponds to a sequential decision making process involving *discrete* choices, we cannot backpropagate through the sampling process to learn the parameters of the puzzle generator network. Therefore, we rely on the Gumbel-Softmax trick (Jang et al., 2017) to deal with this issue and backpropagate through the parameters of π_ϕ, which we will cover in §4.3.

4 Sampling and Differentiating Subsets

4.1 A DP for the Partition Function

In order to sample from the variational distribution in eqn (6), we need to compute its partition function in eqn (7). Interestingly, the partition function can be computed using dynamic programming (DP).

Let us denote by $Z(j, k)$ the partition function of all subsets of size k from the index set $\{j, .., |\boldsymbol{x}|\}$. Hence, the partition function of the q distribution is $Z(1, K)$. The DP relationship can be written as,

$$\begin{aligned} Z(j-1, k) = (1 - \pi(j-1|\boldsymbol{x}))Z(j, k) \\ + \pi_\phi(j-1|\boldsymbol{x})Z(j, k-1). \end{aligned} \quad (11)$$

The initial conditions are $Z(j, 0) = 1$ and

$$Z(|\boldsymbol{x}| - k + 1, k) = \prod_{i=|\boldsymbol{x}|-k+1}^{|\boldsymbol{x}|} \pi_\phi(j|\boldsymbol{x}) \quad (12)$$

corresponding to two special terminal cases in selection process in which we have picked all K indices, and we need to select all indices left to fulfil K.

This amounts to a DP algorithm with the time complexity $\mathcal{O}(K|\boldsymbol{x}|)$.

Algorithm 2 Sampling Procedure

Function: subsetSampling
Input: datapoint $\boldsymbol{x}$, prob. π_ϕ, masking size K
Output: subset S, sample log probability l

Let $S \leftarrow \emptyset; l \leftarrow 0; j \leftarrow 0$
Calculate DP table Z using Eq. (11)
while $|S| < K$ **do**
  $j \leftarrow j + 1$
  $q_{j,Y} \leftarrow q_j(\text{Y}|S_{j-1}, \pi_\phi)$ // using eqn (13)
  $q_{j,N} \leftarrow 1 - q_{j,Y}$
  $\epsilon_{j,Y}, \epsilon_{j,N} \sim \text{Gumbel}(0, 1)$
  $o_j \leftarrow \text{argmax}_{o \in \{Y,N\}} \log q_{j,o} + \epsilon_{j,o}$
  $l \mathrel{+}= \log \text{softmax}(\log q_{j,o} + \epsilon_{j,o})\big|_{o=o_j}$
  if $o_j == Y$ **then**
    $S \leftarrow S \cup \{j\}$
  end if
end while
return S, l

4.2 Subset Sampling for MLMs

The DP in the previous section also gives rise to the sampling procedure. Given a partial random subset S_{j-1} with elements chosen from the indices $\{1, .., j-1\}$, the probability of including the next index j, denoted by $q_j(\text{yes}|S_{j-1}, \pi_\phi)$, is

$$\frac{\pi_\phi(j|\boldsymbol{x})Z(j+1, K-1-|S_{j-1}|)}{Z(j, K-|S_{j-1}|)} \quad (13)$$

where $Z(j, k)$ values come from the DP table. Hence, the probability of *not* including the index j is

$$q_j(\text{no}|S_{j-1}, \pi_\phi) = 1 - q_j(\text{yes}|S_{j-1}, \pi_\phi). \quad (14)$$

Domain	NER Dataset	Num. tokens/Num. sent. Train	Dev.	Test	Unlab. Corpus	Num. tokens /Num. sent.
NEWS	CoNLL2003	203.6k/14.0k	51.36k/3.3k	46.4k/3.5k	-	-
TWEET	WNUT2016	46.5k/2.4k	16.3K/1k	61.9k/3.8k	Sentiment140	20M/1.4M
FIN	FIN	41.0k/1.2k	-	13.3k/303	SEC Filing 2019	155M/5.5M
BIOMED	JNLPBA	445.0k/16.8k	47.5k/1.7k	101.0k/3.9k	PubMed	4.3B/181M
BIOMED	BC2GM	355.4k/12.6k	71.0k/2.5k	143.5k/5.0k	PubMed	4.3B/181M
BIOMED	BioNLP09	227.7k/7.5k	44.2k/1.4k	74.6k/2.5k	PubMed	4.3B/181M
BIOMED	BioNLP11EPI	161.6k/5.7k	54.8k/1.9k	116.1k/4.1k	PubMed	4.3B/181M

Table 1: Data statistics of named entity span prediction tasks and unlabled additional pretraining corpus.

In case the next index is chosen to be in the sample, then $S_{j+1} = S_j \cup \{j+1\}$; otherwise $S_{j+1} = S_j$.

The sampling process entails a sequence of binary decisions (Figure 1.b) in an underlying Markov Decision Process (MDP). It is an iterative process, which starts by considering the index one. At each decision point j, the sampler's action space is to whether include (or not include) the index j into the partial sample S_j based on eqn (13). We terminate this process when the partially selected subset has K elements.

The sampling procedure is described in Algorithm 2. In our MDP, we actually sample an index by generating Gumbel noise in each stage, and then select the choice (yes/no) with the maximum probability. This enables differentiation through the sampled subset, covered in the next section.

4.3 Differentiating via Gumbel-Softmax

Once the sampling process is terminated, we then need to backpropagate through the parameters of π_ϕ, when updating the parameters of the puzzle generator according to eqn (10).

More concretely, let us assume that we would like to sample a subset S. As mentioned in previous section, we need to decide about the inclusion of the next index j given the partial sample so far S_{j-1} based on the eqn (13). Instead of uniform sampling, we can equivalently choose one of these two outcomes as follows

$$o_j^* = \underset{o_j \in \{\text{yes,no}\}}{\operatorname{argmax}} \log q_j(o_j|S_{j-1}, \pi_\phi) + \epsilon_{o_j} \quad (15)$$

where the random noise ϵ_{o_j} is distributed according to standard Gumbel distribution. Sampling a subset then amounts to a sequence of argmax operations. To backpropagate through the sampling process, we replace the argmax operators with softmax, as argmax is not differentiable. That is,

$$Pr(o_j) = \frac{\exp(\log q_j(o_j|S_{j-1},\pi_\phi)+\epsilon_{o_j})}{\sum_{o'_j} \exp(\log q_j(o'_j|S_{j-1},\pi_\phi)+\epsilon_{o'_j})}. \quad (16)$$

The log product of the above probabilities for the decisions in a sampling *path* is returned as l in Algorithm 2, which is then used for backpropagation.

5 Experiments

We evaluate our proposed masking strategy in UDA for named entity span prediction tasks coming from three different domains.

5.1 Unsupervised Domain Adaptation Tasks

Source and Target Domain Tasks. Our evaluation is focused on the problem of identifying named entity spans in domain-specific text without access to labeled data. The evaluation tasks comes from several named entity recognition (NER) dataset including WNUT2016 (Strauss et al., 2016), FIN (Salinas Alvarado et al., 2015), JNLPBA (Collier and Kim, 2004), BC2GM (Smith et al., 2008), BioNLP09 (Kim et al., 2009), and BioNLP11EPI (Kim et al., 2011). Table 1 reports data statistics.

These datasets cover three domains social media (TWEETS), financial (FIN) and biomedical (BIOMED). We utilize the CoNLL-2003 English NER dataset in newstext domain (NEWS) as the source task and others as the target. We perform domain-tuning and source task-tuning, followed by zero-shot transfer to the target tasks, as described in §2. Crucially, we do not use the labels of the training sets of the target tasks, and only use their sentences for domain adaptation. Since the number of entity types are different in each task, we convert all the labels to entity span in IBO scheme. This ensures that all tasks share the same set of labels consisting of three tags: I, B, and O.

Extra Target Domain Unlabeled Corpora. As the domain tuning step can further benefit from additional unlabeled data, we create target domain unlabeled datasets from the available corpora of relevant domains. More specifically, we use publicly available corpora, Sentiment140 (Go et al., 2009), SEC Filing 2019[2] (DeSola et al., 2019) PubMed (Lee et al., 2020) for the TWEET, FIN and BIOMED domains respectively (Table 1). From the unlabeled corpora, the top 500K and 1M similar sentences to the training set of each target task are extracted based on the average n-gram similarity where $1 \leq n \leq 4$, resulting in extra target domain unlabeled corpora.

5.2 Masking Strategies for MLM Training

We compare our adversarial learned masking strategy approach against random and various heuristic masking strategies which we propose:

- **Random.** Masked tokens are sampled uniformly at random, which is the common strategy in the literature (Devlin et al., 2019; Liu et al., 2019).
- **POS-based strategy.** Masked tokens are sampled according to a non-uniform distribution, where a token's probability depends on its POS tag. The POS tags are obtained using spaCy.[3] Content tokens such as verb (VERB), noun (N), adjective (ADJ), pronoun (PRON) and adverb (ADV) tags are assigned higher probability (80%) than other content-free tokens such as PREP, DET, PUNC (20%).
- **Uncertainty-based strategy.** We select those tokens for which the current MLM is most uncertain for the reconstruction, where the uncertainty is measured by the entropy. That is, we aim to select those tokens with high Entropy$[Pr_i(.|\boldsymbol{x}_{\bar{S}_i}; \mathcal{B}_\theta)]$, where $\boldsymbol{x}_{\bar{S}_i}$ is the sentence $\boldsymbol{x}$ with the ith token masked out, and $Pr_i(.|\boldsymbol{x}_{\bar{S}_i}; \mathcal{B}_\theta)$ is the predictive distribution for the ith position in the sentence.

  Calculating the predictive distribution for each position requires one pass through the network. Hence, it is expensive to use the exact entropy, as it requires $|\boldsymbol{x}|$ passes. We mitigate this cost by using $Pr_i(.|\boldsymbol{x}; \mathcal{B}_\theta)$ instead, which conditions on the original unmasked sentence. This estimation only costs one pass through the MLM.
- **Adversarial learned strategy.** The masking strategy is learned adversarially as in §3. The puzzle-generator update frequency β (Algorithm 1) is set to 0.3 for all experiments.

These strategies only differ in how we choose the candidate tokens. The number of to-be-masked tokens is the same in all strategies (15%). Among them, 80% are replaced with [MASK], 10% are replaced with random words, the rest are kept unchanged as in (Devlin et al., 2019). In our experiments, the masked sentences are generated dynamically on-the-fly.

To evaluate the models, we compute precision, recall and F1 scores on a per token basis. We report average performance of five runs.

5.3 Implementation Details

Our implementation is based on Tensorflow library (Abadi et al., 2016)[4]. We use BERT-Base model architecture which consists of 12 Transformer layers with 12 attention heads and hidden size 768 (Devlin et al., 2019) in all our experiments. We use the cased wordpiece vocabulary provided in the pretrained English model. We set learning rate to 5e-5 for both further pretraining and task tuning. Puzzle generator is a two layer feed-forward network with hidden size 256 and dropout rate 0.1.

5.4 Empirical Results

Under the same computation budget to update the MLM, we evaluate the effect of masking strategy in the domain tuning step under various size of additional target-domain data: none, 500K and 1M. We continue pretraining BERT on a combination of unlabeled source (CoNLL2003), unlabeled target task training data and additional unlabeled target domain data (if any). If target task data is smaller, we oversample it to have equal size to the source data. The model is trained with batch size 32 and max sequence length 128 for 50K steps in 1M target-domain data and 25K steps in other cases. It equals to 3-5 epochs over the training set. After domain tuning, we finetune the adapted MLM on the source task labeled training data (CoNLL2003) for three epochs with batch size 32. Finally, we evaluate the resulting model on target task. On the largest dataset, random and POS strategy took around 4 hours on one NVIDIA V100 GPU while entropy

[2] http://people.ischool.berkeley.edu/~khanna/fin10-K/

[3] https://spacy.io/

[4] Source code is available at https://github.com/trangvu/mlm4uda

Task	UDA rand	UDA pos	UDA ent	UDA adv	UDA + 500K target-domain rand	UDA + 500K target-domain pos	UDA + 500K target-domain ent	UDA + 500K target-domain adv	UDA + 1M target-domain rand	UDA + 1M target-domain pos	UDA + 1M target-domain ent	UDA + 1M target-domain adv
WNUT2016	**47.11**	46.79†	46.95†	47.03†	46.93	47.69†	47.84†	**48.01**†	52.36	52.01†	**52.74**†	52.53†
FIN	21.55	22.53†	22.73†	**23.38**†	24.70	26.70†	26.63†	**26.85**†	25.96	26.95†	26.96†	**28.94**†
JNLPBA	27.44	28.06†	28.22†	**30.06**†	29.92	**30.56**†	30.47†	30.31†	31.01	30.91†	**31.59**†	31.54†
BC2GM	28.31	28.50	**30.81**†	29.01†	31.13	31.85†	31.83†	**32.38**†	31.35	31.70†	32.01†	**32.49**†
BioNLP09	26.37	27.53†	29.21†	**29.24**†	31.38	31.03†	34.33†	**35.05**†	32.16	33.51†	34.99†	**35.41**†
BioNLP11EPI	32.69	33.51†	**34.81**†	34.59†	42.41	42.81†	**42.83**†	42.64	43.11	43.47†	43.31	**43.61**†
$\bar{\Delta}$	-	+0.58	+1.54	**+1.64**	-	+0.70	+1.26	**+1.46**		+0.43	+0.94	**+1.43**

Table 2: F1 score of name entity span prediction tasks in three UDA scenarios which differ in the amount of additional target-domain data. rand, pos, ent and adv denote the random, POS-based, uncertainty-based, and adversarial masking strategy respectively. $\bar{\Delta}$ row reports the average improvement over random masking across all tasks. **Bold** shows the highest score of task on each UDA setting. † indicates statistically significant difference to the random baseline with p-value ≤ 0.05 using bootstrap test.

	Task	rand	mix-pos	mix-ent	mix-adv
UDA + 500K	WNUT2016	46.93	51.17	52.40	**52.56**
	FIN.	24.70	26.95	27.36	**28.30**
	JNLPBA	29.92	29.22	31.65	**32.99**
	BC2GM	31.13	32.11	**32.68**	32.60
	BioNLP09	31.38	33.17	34.27	**34.91**
	BioNLP11EPI	42.41	42.73	**43.43**	43.08
	$\bar{\Delta}$	-	+3.10	+4.17	**+4.61**
UDA + 1M	WNUT2016	52.36	52.40	52.64	**52.95**
	FIN.	25.96	27.86	28.51	**29.08**
	JNLPBA	31.01	31.77	32.07	**32.26**
	BC2GM	31.35	31.76	32.43	**32.52**
	BioNLP09	32.61	34.49	35.67	**35.78**
	BioNLP11EPI	43.11	43.96	**44.81**	44.27
	$\bar{\Delta}$	-	+1.05	+1.70	**+1.82**

Table 3: F1 score in UDA with additional data under several mixed masking strategies. **Bold** shows the highest score of task on each UDA setting.

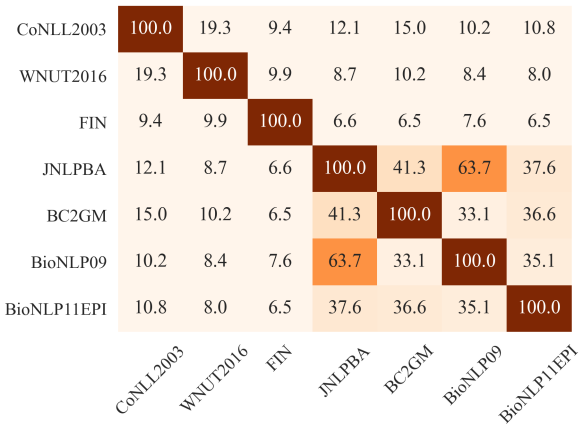

Figure 2: Vocabulary overlap (%) between NER tasks.

and adversarial approach took 5 and 7 hours respectively. The task tuning took about 30 minutes.

Results are shown in Table 2. Overall, strategically masking consistently outperforms random masking in most of the adaptation scenarios and target tasks. As expected, expanding training data with additional target domain data further improves performance of all models. Comparing to random masking, prioritising content tokens over content-free ones can improve up to 0.7 F1 score in average. By taking the current MLM into account, uncertainty-based selection and adversarial learned strategy boost the score up to 1.64. Our proposed adversarial approach yields highest score in 11 out of 18 cases, and results in the largest improvement over random masking across all tasks in both UDA with and without additional target domain data.

We further explore the mix of random masking and other masking strategies. We hypothesise that the combination strategies can balance the learning of challenging tokens and effortless tokens when forming the common semantic space, hence improve the task performance. In a minibatch, 50% of sentences are masked according to the corresponding strategy while the rest are masked randomly. Results are shown in Table 3. We observe an additional performance to the corresponding single-strategy model across all tasks.

5.5 Analysis

Domain Similarity. We quantify the similarity between source (CoNLL2003) and target domains by vocabulary overlap between the domains (excluding stopwords). Figure 2 shows the vocabulary overlap across tasks. As seen, all the target domains are dissimilar to the source domain, with FIN having the lowest overlap. FIN has gained the

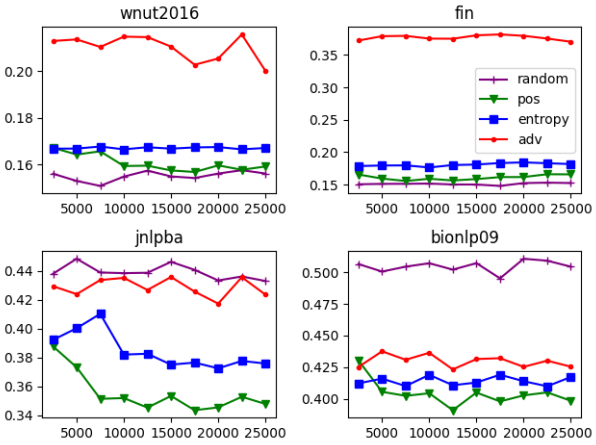

Figure 3: Average density ratio of masked-out tokens of every 2500 training steps in UDA setting.

largest improvement from the adversarial strategy in the UDA results in Tables 2 and 3. As expected, the biomedical datasets have relatively higher vocabulary overlap with each other.

Density Ratio of Masked Subsets. We analyze the density ratio of masked-out tokens in the target and source domains

$$r(w) = \max(1 - \frac{Pr_s(w)}{Pr_t(w)}, 0)$$

where $Pr_s(w)$ and $Pr_t(w)$ is the probability of token w in source and target domains, respectively. These probabilities are according to unigram language models trained on the training sets of the source and target tasks. The higher value of $r(w)$ means the token w is new or appears more often in the target text than in the source. Figure 3 plots the density ratio of masked-out tokens during domain tuning time for four UDA tasks. Comparing to other strategies, we observed that adversarial approach tends to select tokens which have higher density ratio, i.e. more significant in the target.

Syntactic Diversity in Masked Subset. Table 4 describes the percentage of POS tags in masked subset selected by different masking strategies. We observed that our method selects more tokens from the major POS tags (71%) compared to random (45%) and entropy-based (55%) strategies. It has chosen less nouns compared to the POS strategy, and more pronouns compared to all other strategies.

Tagging Accuracy of OOV and non-OOV. We compare the tagging accuracy of out-of-vocabulary (OOV) words which are in target domain but not presenting in source, and non-OOV tokens in Table 5. As seen, our adversarial masking strategy achieves higher accuracy on both OOV and non-OOV tokens in most cases.

POS Tag	rand	pos	ent	adv
ADJ	9%	17%	11%	13%
VERB	8%	16%	10%	17%
NOUN	25%	51%	31%	34%
PRON	1%	2%	1%	3%
ADV	2%	4%	2%	4%
Others	55%	10%	45%	29%

Table 4: The tag ratio of the POS tags of tokens in masked subset on BIONLP11 under different masking strategies.

6 Related Work

Unsupervised Domain Adaptation. The main approaches in neural UDA include discrepancy-based and adversarial-based methods. The discrepancy-based methods are based on the usage of the maximum mean discrepancy or Wasserstein distance as a regularizer to enforce the learning of domain non-discriminative representations (Shen et al., 2018). Inspired by the Generative Adversarial Network (GAN) (Goodfellow et al., 2014), the adversarial-based methods learn a representation that is discriminative for the target task and indiscriminative to the shift between the domains (Ganin and Lempitsky, 2015).

Domain Adaptation with MLM. Performance of fine-tuned MLM can deteriorate substantially on the presence of domain mismatch. The most straightforward domain adaptation approach in MLM is to adapt general contextual embedding to a specific domain (Lee et al., 2020; Alsentzer et al., 2019; Chakrabarty et al., 2019), that is to further improve pretrained MLM by continuing to pretrain language models on related domain or similar tasks (Gururangan et al., 2020), or via intermediate task which is also referred to as STILTs (Phang et al., 2018). Recent works have proposed two-step adaptive domain adaptation framework which consists of domain tuning and task finetuning (Ma et al., 2019; Xu et al., 2019; Wang et al., 2019c; Logeswaran et al., 2019). They have demonstrated that domain tuning is necessary to adapt MLM with both domain knowledge and task knowledge before finetuning, especially when the labelled data

Task	Model	acc.	non-OOV	OOV
WNUT2016	rand	23.04	21.88	24.99
	pos	23.78	22.77	25.48
	ent	23.95	**22.95**	25.62
	adv	**24.20**	22.79	**26.57**
FIN	rand	27.66	25.01	30.88
	pos	28.51	27.23	29.36
	ent	28.09	**27.67**	31.21
	adv	**29.36**	27.56	**33.90**
JNLPBA	rand	7.77	7.86	7.50
	pos	**9.74**	**9.83**	**9.50**
	ent	8.79	8.81	8.74
	adv	7.92	7.89	8.01
BC2GM	rand	11.38	11.35	11.48
	pos	13.09	12.88	13.89
	ent	13.19	12.89	14.28
	adv	**14.53**	**14.44**	**14.84**
BioNLP09	rand	9.49	8.88	10.2
	pos	9.45	10.51	8.22
	ent	13.11	15.67	10.14
	adv	**14.82**	**18.45**	**10.61**
BioNLP11EPI	rand	13.16	27.40	6.57
	pos	14.02	28.28	7.43
	ent	**14.28**	**28.70**	**7.59**
	adv	13.76	28.56	6.89

Table 5: Tagging accuracy of in-vocabulary (non-OOV) and out-of-vocabulary (OOV) words in UDA + 500K in-domain data.

in target task is extremely small. Our experiment setting is similar to Han and Eisenstein (2019)'s work. However, we focus on learning masking strategy to boost the domain-tuning step.

Adversarial Learning. Recent research in adversarial machine learning has either focused on attacking models with adversarial examples (Alzantot et al., 2018; Iyyer et al., 2018; Ebrahimi et al., 2018), or training models to be robust against these attacks (Zhou et al., 2019). Wang et al. (2019b); Liu et al. (2020) propose the use of adversarial learning for language models. They consider autoregressive LMs and train them to be robust against adversarial perturbations of the word embeddings of the target vocabulary.

7 Conclusion

We present an adversarial objective for further pretraining MLM in UDA problem. The intuition behind the objective is that the adaptation effort should focus on a subset of tokens which are challenging to the MLM. We establish a variational lower bound of the objective function and propose an effective sampling algorithm using dynamic programming and Gumbel softmax trick. Comparing to other masking strategies, our proposed adversarial masking approach has achieve substantially better performance on UDA problem of named entity span prediction for several domains.

Acknowledgments

This material is based on research sponsored by Air Force Research Laboratory and DARPA under agreement number FA8750-19-2-0501. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The authors are grateful to the anonymous reviewers for their helpful comments. The computational resources of this work are supported by the Google Cloud Platform (GCP), and by the Multi-modal Australian ScienceS Imaging and Visualisation Environment (MASSIVE) (www.massive.org.au).

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin

Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Tuhin Chakrabarty, Christopher Hidey, and Kathy McKeown. 2019. IMHO fine-tuning improves claim detection. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 558–563, Minneapolis, Minnesota. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. {ELECTRA}: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Nigel Collier and Jin-Dong Kim. 2004. Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pages 73–78, Geneva, Switzerland. COLING.

Vinicio DeSola, Kevin Hanna, and Pri Nonis. 2019. Finbert: pre-trained model on sec filings for financial natural language tasks.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1180–1189. JMLR.org.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of ACL*.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4238–4248, Hong Kong, China. Association for Computational Linguistics.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of BioNLP'09 shared task on event extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 1–9, Boulder, Colorado. Association for Computational Linguistics.

Jin-Dong Kim, Sampo Pyysalo, Tomoko Ohta, Robert Bossy, Ngan Nguyen, and Jun'ichi Tsujii. 2011. Overview of BioNLP shared task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 1–6, Portland, Oregon, USA. Association for Computational Linguistics.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. 2020. Adversarial training for large neural language models. *arXiv preprint arXiv:2004.08994*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*,

pages 3449–3460, Florence, Italy. Association for Computational Linguistics.

Xiaofei Ma, Peng Xu, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2019. Domain adaptation with BERT-based domain classification and data selection. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 76–83, Hong Kong, China. Association for Computational Linguistics.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with Bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 372–382, Copenhagen, Denmark. Association for Computational Linguistics.

Julio Cesar Salinas Alvarado, Karin Verspoor, and Timothy Baldwin. 2015. Domain adaption of named entity recognition to support credit risk assessment. In *Proceedings of the Australasian Language Technology Association Workshop 2015*, pages 84–90, Parramatta, Australia.

Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. 2018. Wasserstein distance guided representation learning for domain adaptation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Larry Smith, Lorraine K Tanabe, Rie Johnson nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M Friedrich, Kuzman Ganchev, et al. 2008. Overview of biocreative ii gene mention recognition. *Genome biology*, 9(2):S2.

Benjamin Strauss, Bethany Toma, Alan Ritter, Marie-Catherine de Marneffe, and Wei Xu. 2016. Results of the WNUT16 named entity recognition shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 138–144, Osaka, Japan. The COLING 2016 Organizing Committee.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

Dilin Wang, Chengyue Gong, and Qiang Liu. 2019b. Improving neural language modeling via adversarial training. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6555–6565, Long Beach, California, USA.

Huazheng Wang, Zhe Gan, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, and Hongning Wang. 2019c. Adversarial domain adaptation for machine reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2510–2520, Hong Kong, China. Association for Computational Linguistics.

Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2324–2335, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Yilun Zhou, Julie Shah, and Steven Schockaert. 2019. Learning household task knowledge from WikiHow descriptions. In *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*, pages 50–56, Macau, China. Association for Computational Linguistics.