Unsupervised Domain Adaptation for Event Detection using Domain-specific Adapters

Nghia Ngo Trung[1], **Duy Phung**[1] **and Thien Huu Nguyen**[2]
[1] VinAI Research, Vietnam
[2] Department of Computer and Information Science, University of Oregon,
Eugene, OR 97403, USA
{v.nghiant66,v.duypv1}@vinai.io, thien@cs.uoregon.edu

Abstract

Due to the multi-dimensional variation of textual data, detection of event triggers from new domains can become a lot more challenging. This prompts a need to research on domain adaptation methods for event detection task, especially for the most practical unsupervised setting. Recently, large transformer-based language models, e.g. BERT, have become essential to achieve top performance for event detection. However, their unwieldy nature also prevents effective adaptation across domains. To this end, this work proposes a **D**omain-specific **A**dapter-based **A**daptation (**DAA**) framework to improve the adaptability of BERT-based models for event detection across domains. By explicitly representing data from different domains with separate adapter modules in each layer of BERT, **DAA** introduces a novel joint representation learning mechanism and a Wasserstein distance-based technique for data selection in adversarial learning to substantially boost the performance on target domains. Extensive experiments and analysis over different datasets (i.e., LitBank, TimeBank, and ACE-05) demonstrate the effectiveness of our approach.

1 Introduction

Event detection (ED) is an important component in the overall event extraction pipeline, which plays a crucial role in any natural language understanding system. The goal of ED is to identify event triggers in a given text and classify them into one of several pre-defined types. Formally, according to the ACE-05 annotation guideline , each event trigger is a phrase (usually a single verb or nominalization), which evokes that event in the context of the associating event mention. For example, the word "*fired*" is the trigger word for an event of type *Attack* in the following sentence: *"The police **fired** tear gas and water cannons in street battles with activists."* Tackling ED problem involves both locating the event triggers and categorizing them into specific event types, therefore can be a quite challenging task due to the intricate dependency among triggers, events, and contexts in linguistic data. The complication is further amplified by domain shift problem when text are collected from multiple different domains.

The majority of prior approaches on ED relied on the basic supervised learning assumption where training and testing data follow the same distribution. Several works further evaluated their methods on cross-domain setting where their models were trained using data from one domain and tested on another, without leveraging any adaptation mechanism to alleviate the domain shift problem (Nguyen and Grishman, 2015; Yubo et al., 2015; Hong et al., 2018b). To this end, our work explores the general problem of domain adaptation for ED where data comes from two different source and target domains. In particular, we focus on the unsupervised setting that requires no annotations for target data, and the model has to learn to make use of both labeled source and unlabeled target samples to improve its performance on target domain. To our knowledge, this is the first work on unsupervised domain adaptation (UDA) for ED in the literature.

The most prominent approach for UDA is a representation learning method based on the theory of learning from different domains developed by Ben-David et al. (2010). The main result provided a way to bound the loss of a model on target domain with its performance on source domain using a domain-divergence term and an optimal joint error term (which is assumably negligible). Ganin et al. (2016) adopted this idea for deep learning architecture in their domain-adversarial neural network (DANN). They employed a domain-adversarial training procedure in which a domain classifier is learned concurrently and adversarially with the network's fea-

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 4015–4025

ture extractor, resulting in a not only discriminative but also domain-invariant joint representation for data from both domains. While DANN and its variants are very well-studied in computer vision's domain adaption researches, their NLP counterparts are pale in comparison, especially for a newly established architecture like BERT. There have been only several works that adopted DANN to align the contextualized representations learned by BERT across domains (Lin et al., 2020; Naik and Rosé, 2020; Wright and Augenstein, 2020). Lin et al. (2020) even observed negative effect when applying adversarial training compare with simply fine-tuning BERT on in-domain data. One explanation is that the pre-training of BERT on massive corpora has already induces a somewhat general representation, thus DANN has little effect while the fine-tuning process using source dataset could cause over-fitting on the corresponding domain due to the immense capacity of the model. To this end, we propose fixing the parameters of the already universal language model while leveraging multiple adapter modules for domain-adversarial training process. More specifically, inspired by the works of Liu et al. (2017a) and Houlsby et al. (2019) on effective multi-task learning, we augment the pre-trained BERT model by adding three different adapters to create a shared-private architecture. Two source and target adapters which take as inputs data from their respective domains to capture private properties of each, and a joint adapter that encodes every sample in a subspace shared across domains through adversarial training. Orthogonality constraints together with a self-supervised auxiliary task are employed to ensure the representations of all adapters are informative while also attaining the above desired properties.

Recently, Ma et al. (2019) and Aharoni and Goldberg (2020) have shown that BERT's representations are extremely effective at clustering text to their respective domains, and a small subset of good in-domain data can already provide significant boosts in target performance while the rest only provide little to no improvement, in some cases even degrade model's out-of-domain generalization. Considering this, we explicitly find hard instances to leave out when learning to extract the domain-invariant features. Our data selection component estimates and minimizes the cost of transport between source and target marginal representation distributions based on the Wasserstein-1 distance (also refer to as Earth Mover distance). Arjovsky et al. (2017) pointed out that the relative strength of the topologies induced by this distance is much weaker than that of KL-divergence used by adversarial training. Therefore, it could serve as a good necessary condition for DANN component to achieve optimal alignment. The faraway source instances that induce the highest transportation costs are those out-of-distribution samples that may introduce noise and hurt adaptation performance. Accordingly, they are omitted from the domain-adversarial training process. The entire computation makes use of representations from source and target adapters, thus implicitly provides informative signals from domain-specific adapters to joint adapter without interrupting the joint representation learning procedure.

2 Related Work

Prior ED works have focused on the in-domain setting (Li et al., 2013; Chen et al., 2015; Nguyen et al., 2016; Yang and Mitchell, 2016; Nguyen and Grishman, 2018; Sha et al., 2018; Liu et al., 2017b; Tong et al., 2020; Nguyen et al., 2021), the cross-domain evaluation (Nguyen and Grishman, 2016; Hong et al., 2018a), the few/low-shot learning scenario (Lai et al., 2020a,b). Our work is different from those prior work as we explore a new formulation for ED with unsupervised domain adaptation where unlabeled data in the target domain is utilized to improve domain-invariant representation learning.

Recently, some efforts have been made to study the domain-related knowledge encoded in BERT's representations (Aharoni and Goldberg, 2020), and methods to leverage it to improve performances on domain-specific tasks, such as pre-training on additional data (Gururangan et al., 2020), fine-tuning using intermediate tasks (Phang et al., 2018; Garg et al., 2020), and data selection (Ma et al., 2019; Aharoni and Goldberg, 2020). Another line of research regarding multi-task learning shares a common goal of creating a universal representation space for all data with domain adaptation. Previous approaches made use of multiple encoders to set up a shared-private architecture (Bousmalis et al., 2016; Liu et al., 2017a), which usually is impractical for BERT-based models because of theirs sizes. By fixing a pre-trained BERT as the base for general representations, Houlsby et al. (2019) and Stickland and Murray (2019) proposed to adapt the

model to each task by adding small task-specific layers between BERT's layers and updating only them when fine-tuning on the corresponding task.

3 Model

Throughout this work, we formulate ED task as a token-level multi-class classification problem (Nguyen and Grishman, 2015; Ngo et al., 2020). For UDA setting in particular, we have a labeled source dataset $\mathcal{D}^{\mathbf{s}}_{\mathbf{XY}} = \{(\mathbf{X}^{\mathbf{s}}_i, y^{\mathbf{s}}_i)\}_{i=1}^{N^{\mathbf{s}}}$ of $N^{\mathbf{s}}$ samples from source domain $\mathbf{s}$ and an unlabeled set of $N^{\mathbf{t}}$ samples $\mathcal{D}^{\mathbf{t}}_{\mathbf{X}} = \{\mathbf{X}^{\mathbf{t}}_i\}_{i=1}^{N^{\mathbf{t}}}$ drawn from target domain $\mathbf{t}$. Each $\mathbf{X}^{\mathbf{s}}_i$ is a pair consists and an event mention $W^{\mathbf{s}}_i = (w^{\mathbf{s}}_{i1}, w^{\mathbf{s}}_{i2}, \cdots, w^{\mathbf{s}}_{im})$ (m is the fixed number of words), and a trigger position u ($1 \leq u \leq m$) corresponding to the word $w^{\mathbf{s}}_{iu}$. An encoder computes its latent representation $x^{\mathbf{s}}_i$, which is then used by the event classifier to predict an event of type $y^{\mathbf{s}}_i$. For target domain, parallel notations are used $x^{\mathbf{t}}_i$ and $y^{\mathbf{t}}_i$ (only accessible in target domain's test dataset)

3.1 Baseline Model

As this is the first work on UDA for ED, this section aims to establish a baseline of the task for further research. Recent works have shown a substantial boost in performance for the standard supervised setting of ED by leveraging contextual embedding of large self-attention based language models (Wang et al., 2019; Lai et al., 2020c). Accordingly, we utilize a pre-trained BERT's encoder, together with its domain-adversarial variant to create a strong baseline for the UDA setting.

Without any domain adaptation mechanism, our **BERT** baseline only follows cross-domain evaluation setting as previous works. The model is fully fine-tuned on source domain dataset while at test time, data from target domain is used to evaluate its performance.

On the other hand, the **BERT+DANN** baseline takes advantage of the availabel unlabeled target data through adversarial training. Specifically, a domain classification task is learned concurrently with the main downstream task, using unlabeled samples and their domain labels from both domains. By pushing the encoder to both minimize the event classfication loss and maximally misdirect domain predictor, the resulting representation can be indiscriminate with respect to the shift between the domains while also discriminative for the main learning task.

Finally, to demonstrate to ability of adapter-based tuning approach to retain the original's model performance, we also evaluate a **BERT+Adapter** baseline. Following recommendation from Pfeiffer et al. (2021), we augment a pre-trained BERT model by injecting a single bottleneck adapter module between the encoder's layers. Then, the fine-tuning process proceeds in the same manner as that of the **BERT** baseline, but only parameters of the adapter modules get updated in this case.

3.2 Adapter-based Domain Representation

Pre-trained BERT model was previously optimized for the task of masked language model in unsupervised manner on several extremely large corpora. The diversity of these unlabeled text also pushes the network to be a good starting point for learning domain-invariant features, which would be lost if we fully fine-tune it on source domain task. Accordingly, we make use of a fixed pre-trained BERT model as the base of our adapters.

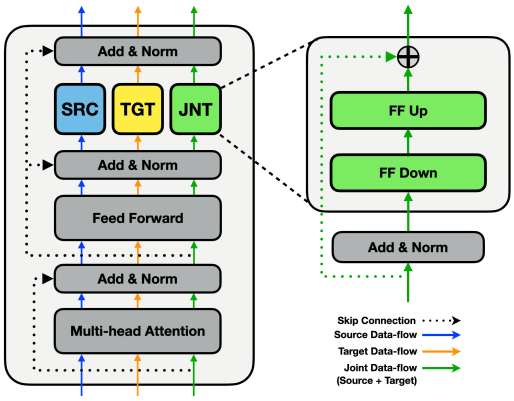

Figure 1: Domain-specific adapters inside the original BERT's layers.

An adapter for each domain To explicitly create a shared-private representation subspace of each domain, we inject three adapters into the same base encoder. Formally, adapter modules $a^{\mathbf{s}}_i$, $a^{\mathbf{t}}_i$, $a^{\mathbf{j}}_i$ are added on top of each BERT's layer. While these modules share the same architecture, they take in as inputs data only from their corresponding sources:

$$\mathbf{A}^{\mathbf{d}}(x^{\mathbf{d}}_i) = a^{\mathbf{d}}_L \circ b_{L-1} \circ \cdots \circ a^{\mathbf{d}}_1 \circ b_1(x^{\mathbf{d}}_i)$$

where b_i is the fixed layer of BERT ($1 \leq i \leq L$), $\mathbf{d} \in \{\mathbf{s}, \mathbf{t}, \mathbf{j}\}$, and $\mathcal{D}^{\mathbf{j}}_{\mathbf{X}} = \mathcal{D}^{\mathbf{s}}_{\mathbf{X}} \cup \mathcal{D}^{\mathbf{t}}_{\mathbf{X}}$. The joint adapter $\mathbf{A}^{\mathbf{j}}$ is our main representation which will be used by event detection head h_c for source

classification task:

$$\mathcal{L}_c = \frac{1}{n^{\mathbf{s}}} \sum_{i=1}^{n^{\mathbf{s}}} -y_i^{\mathbf{s}} log\left(h_c\left(\mathbf{A}^{\mathbf{j}}\left(x_i^{\mathbf{s}}\right)\right)\right)$$

On the other hand, the domain-specific adapters $\mathbf{A}^{\mathbf{s}}$ and $\mathbf{A}^{\mathbf{j}}$ will only be used to help $\mathbf{A}^{\mathbf{j}}$ find to the optimal joint-domain space while simultaneously retain good performance on downstream task.

Adapter architecture: There are a variety of ways that one can design the adapter modules' architecture. Following the observations from Pfeiffer et al. (2021), we choose ours to be the most efficient but also effective, which is a singular bottleneck neural network with skip-connection, taking features computed by BERT's feed-forward sub-layer as inputs. The adapter module in layer l can be decoupled into two parts $a_l^{\mathbf{d}} = a_l^{\mathbf{d},up} \circ a_l^{\mathbf{d},dw}$, where $a_l^{\mathbf{d},dw} : \mathbb{R}^{d_{model}} \rightarrow \mathbb{R}^c$ and $a_l^{\mathbf{d},up} : \mathbb{R}^c \rightarrow \mathbb{R}^{d_{model}}$ (as shown in figure 1). Despite tripling the added parameters from adapter modules, by setting $c \ll d_{model}$, the amount needed to be tuned is still only less than 10% that of the original network. Additionally, the factorized features enable effective adaptation by making use of the low-dimensional down-sampled representation, while also boosting classification performance by leveraging the free parameters of the up-sampling projection, as described in the next sections.

3.3 Joint Representation Learning

To learn a joint representation that is as general as possible while also maintaining its discriminative property, we propose to combined two mechanisms with complementary effects : a layer-wise domain-adversarial (LDA) component and an adapter-wise domain disentanglement (ADD) component.

3.3.1 Domain-adversarial Training

LDA apply domain-adversarial training to the representation of $\mathbf{A}^{\mathbf{j}}$. Multiple refinements to the original DANN are introduced to mitigate its flaws and learn better domain-invariant features.

Dimension Reduction It is known that discriminative features computed by high-level layers usually lie on low dimensional manifolds. As a result, naively applying adversarial training for BERT's representations, which require high dimension to capture contexts, can lead to gradient vanishing problem. We leverage the adapter's architecture to tackle this issue. Instead of the full dimension outputs of layers, we align domains based on the down-sampled version of the representations, computed by $a_i^{\mathbf{j},dw}$. In consequence, an adapter module can be viewed as a two-step adaption: a down-sampling projection step that extracts domain-invariant features and a following up-sampling projection step which transforms the extracted general features into task discriminative ones.

Layer-wise Alignment To enhance the alignment capability of our model, domain-adversarial training is applied on every layer's output. In particular, we incorporate the asymmetric relaxation of DANN (Wu et al., 2019):

$$\mathcal{L}_d^l = -\frac{1}{N}\sum_{i=1}^{N}\left[d_i\, log\left(\frac{h_d(a_l^{\mathbf{j},dw}(x_i))}{1+\beta_j}\right) + (1-d_i) log\left(1-\frac{h_d(a_l^{\mathbf{j},dw}(x_i))}{1+\beta}\right)\right]$$

where d_i is domain label of samples x_i, $N = n^{\mathbf{s}} + n^{\mathbf{t}}$ is minibatch size, and $\beta_l \geq 0$ is a hyper-parameter controlling the maximal difference of the two marginal distributions ($\beta_l = 0$ is the original formulation). This modification addresses the target shift scenario where domain-adversarial training is unable to achieve optimal solution. As outlined by Rogers et al. (2021), lower-level layers of BERT contain quite broad knowledge, thus encode more random distribution when projected into label space. In contrast, high-level ones are gradually more task-specific, effectively reducing the possible amount of label shift between the two domains. Therefore, we adopt the following relaxation annealing strategy:

$$\mathcal{L}_d = \sum_{l=1}^{L} \mathcal{L}_d^l(a_l^{\mathbf{j},dw}, \beta_l, h_{d,l}^{\mathbf{j}}), with\ \beta_l = 2^{3-l}$$

where each term on the right-hand side is a different relaxed domain classification loss computed by a separate domain classifier $h_{d,l}^{\mathbf{j}}$.

3.3.2 Adapter-wise Domain Disentanglement

The role of ADD component is to ensure the shared-private relationship among adapters. We want the joint adapter $\mathbf{A}^{\mathbf{j}}$ to encode a shared representation space containing common information between domains and no domain-specific information, while the private adapters $\mathbf{A}^{\mathbf{s}}$ and $\mathbf{A}^{\mathbf{t}}$ should only accommodate distinct knowledge that belong exclusively to their corresponding domains. Following

the work of Liu et al. (2017a) and Bousmalis et al. (2016), an orthogonality constraint is imposed using the following similarity loss function:

$$\mathcal{L}_s = \|\mathbf{A_s^j}^\top \mathbf{A_s^s}\|_F^2 + \|\mathbf{A_t^j}^\top \mathbf{A_t^t}\|_F^2$$

where $\|.\|_F$ is the Frobenius norm and $\mathbf{A_{d_2}^{d_1}}$ is a matrix whose rows are the outputs of adapter $\mathbf{A^{d_1}}$ taking inputs from domain $\mathbf{d_2}$. Minimizing $\mathcal{L}_s$ will force $\mathbf{A^j}$ to be in a complementary subspace with $\mathbf{A^s}$ and $\mathbf{A^t}$, encouraging independency among adapters and removing domain-specific noises that may contaminate the joint representation. However, whereas $\mathbf{A^j}$ is trained to be informative for the downstream classification, $\mathbf{A^s}$ and $\mathbf{A^t}$ are not constrained by any task, which potentially could lead to a trivial solution where the network learns to map each representation into the same orthogonal space with $\mathbf{A^j}$ while not having any expressive capability of their corresponding domains. To address this issue, we incorporate a self-supervised component, using the popular Masked Language Modeling (MLM) as our unsupervised task. The token predictor $h_m : \mathbb{R}^{d_{model}} \rightarrow \mathbb{R}^V$ (V is the vocabulary size), is shared between source and target domains:

$$\mathcal{L}_m = \sum_{i=1}^{N_{mask}} \mathcal{L}_m^{\mathbf{s}}(x_i^{\mathbf{s}}) + \mathcal{L}_m^{\mathbf{t}}(x_i^{\mathbf{t}})$$

$$\mathcal{L}_m^{\mathbf{d}}(x_i^{\mathbf{d}}) = -w_i^{\mathbf{d}} \log h_m \left(\mathbf{A^j}(x_i^{\mathbf{d}}) + \mathbf{A^d}(x_i^{\mathbf{d}})\right)$$

where N_{mask} is the number of randomly masked input tokens, following the original procedure in Devlin et al. (2019). The benefit of adding the MLM component is twofold. On one hand, it serves as a constraint to learn informative representations for domain-specific adapters. On the other hand, it also help conditioning joint adapter $\mathbf{A^j}$ on unsupervised knowledge of unlabeled target data, which can have a positive impact on target domain's performance.

3.4 Data Selection

Considering the Wasserstein-1 distance between the distributions generating source and target marginal representations $\mathbb{P}_{\mathbf{X}}^{\mathbf{s}}$ and $\mathbb{P}_{\mathbf{X}}^{\mathbf{t}}$, which can be written as:

$$W(\mathbb{P}_{\mathbf{X}}^{\mathbf{s}}, \mathbb{P}_{\mathbf{X}}^{\mathbf{t}}) = \sup_{\|f\|_L \leq 1} \mathbb{E}\left[f(x^{\mathbf{s}})\right] - \mathbb{E}\left[f(x^{\mathbf{t}})\right]$$

There are several advantages of using this distance as the proxy for data selection mechanism. First, Wasserstein distance takes into account the geometry of the actual data distributions. Thus, it is intuitive to use it to evaluate the discrepancy between marginal distributions and pick source samples that are geometrically close to samples from target distribution. Furthermore, it has been proven by Arjovsky et al. (2017) that the minimization of KL-divergence, on which LDA component based to update $\mathbf{A^j}$, also implies the minimal Wasserstein distance between the corresponding distributions. Therefore, leaving out the most far-a-way samples based on this distance should provide a good necessary condition for LDA to achieve optimal alignment from source to target domain.

Approximate Wasserstein Distance Following the approximation from Shen et al. (2018), we employ a data selection head h_w to estimate the Wasserstein distance between two representation distributions of $\mathbf{A^s}$ and $\mathbf{A^t}$ by maximizing the following empirical loss with respect to θ_w:

$$\mathcal{L}_{\hat{w}} = \frac{1}{n^{\mathbf{s}}} \sum_{i=1}^{n^{\mathbf{s}}} h_w\left(\mathbf{A^s}\left(x_i^{\mathbf{s}}\right)\right) - \frac{1}{n^{\mathbf{t}}} \sum_{i=1}^{n^{\mathbf{t}}} h_w\left(\mathbf{A^t}\left(x_i^{\mathbf{t}}\right)\right)$$

For the above approximation to work, we need to enforce the Lipschitz constraint, which will force the hypothesis class of h_w to be 1-Lipschitz. Following Gulrajani et al. (2017), a gradient penalty $\mathcal{L}_{gr}$ is added to the loss, resulting in the overall estimation problem for the Wasserstein distance as

$$\max_{\theta_w} \mathcal{L}_w = \mathcal{L}_{\hat{w}} - \lambda_{gr}\mathcal{L}_{gr}$$

$$\mathcal{L}_{gr}(\mathbf{A^d}) = (||\nabla_{\mathbf{A^d}} h_w(\mathbf{A^d})||_2 - 1)^2$$

where $\mathbf{d} \in \{\mathbf{s}, \mathbf{t}\}$ and λ_{gr} is a hyper-parameter.

Data Selection based on Wasserstein Distance To avoid negative transfer problem in case of highly dissimilar domains, we propose to use a data selection mechanism based on the estimated Wasserstein distance. By minimizing the empirical distance using $\mathbf{A^s}$ and $\mathbf{A^t}$, we find the representations that achieve the shortest transport distance between source and target samples. Then, a subset of $\hat{n}^{\mathbf{s}}$ source samples is selected with the lowest $h_w(.)$ scores, which corresponds to the $\hat{n}^{\mathbf{s}}$ shortest distances to target domain. These source instances will be used by the joint adapter $\mathbf{A^j}$, together with target unlabeled data, to learn domain-invariant features in LDA.

System	In-domain(bn+nw)			Out-of-domain (bc)			Out-of-domain (cts)			Out-of-domain (wl)			Out-of-domain (un)		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
BERT	77.5	77.5	77.5	75.2	71.8	73.5	75.1	69.9	72.4	70.2	57.9	60.6	68.9	67.5	68.2
BERT+DANN	77.4	75.8	76.6	72.8	69.4	70.9	73.4	39.9	51.2	69.2	50.5	58.4	68.8	59.2	63.6
BERT+Adapter	76.8	76.2	76.7	78.5	72.9	75.6	77.3	69.5	73.2	64.3	56.9	60.3	72.4	69.0	70.6
DAA	79.7	75.7	**77.7**	78.5.1	75.6	**76.9**	78.4	73.2	**75.6**	66.2	60.3	**63.1**	73.5	71.3	**72.3**

Table 1: Unsupervised domain adaptation for event detection. Performance on the **ACE-05** test datasets for different domains.

System	In-domain (TimeBank)			Out-of-domain (LitBank)		
	P	R	F	P	R	F
LSTM+DANN	69.3	87.5	77.3	25.6	72.9	37.9
BiLSTM+DANN	74.2	79.4	76.7	26.3	72.0	38.6
BERT	79.6	84.3	81.9	28.1	84.8	42.2
BERT+DANN	79.8	85.6	82.6	30.3	80.8	44.1
DAA	90.9	88.4	**89.6**	40.0	81.3	**53.6**

Table 2: Performance on TimeBank-to-LitBank.

System	In-domain (LitBank)			Out-of-domain (TimeBank)		
	P	R	F	P	R	F
LSTM+DANN	61.1	61.6	61.3	89.0	18.9	31.2
BiLSTM+DANN	66.1	62.8	64.4	92.9	18.5	30.9
BERT	73.5	72.7	73.1	88.1	28.2	42.7
BERT+DANN	71.9	71.3	71.6	85.0	35.0	49.6
DAA	77.7	75.6	**76.7**	83.2	48.5	**61.1**

Table 3: Performance on LitBank-to-TimeBank.

3.5 Alternating Minimization

Taking it all together, our final training objective is given as:

$$\mathcal{L}_{total} = \mathcal{L}_c + \lambda_d \mathcal{L}_d + \lambda_w \mathcal{L}_w + \lambda_s \mathcal{L}_s + \lambda_m \mathcal{L}_m$$

where $\lambda_d, \lambda_w, \lambda_s, \lambda_m$ are hyper-parameters which help to balance the importance of the corresponding loss with the main event detection loss. Of the five terms on the right-hand side, the domain discrepancy losses ($\mathcal{L}_d$ and $\mathcal{L}_w$) require optimization of different directions with respect to the added heads and the adapters, resulting in a min-max optimization problem. Previous works that made used of domain-adversarial training usually applied gradient reversal layer to train the feature extractors. We find this approach to be unstable and cause performance degradation. Following suggestions from Goodfellow et al. (2014) and Shu et al. (2018), we design an alternating minimization process that is compatible with our learning algorithm whilst also stabilizing the domain-adversarial training. In the first stage, all parameters are fixed except for those of domain-adversarial heads and data selection head. This step corresponds to the estimation of corresponding distance functions between domains given the current representations. After repeatedly updating for k times (k is a hyper-parameter that controls the trade-off between computation and accuracy of the divergence estimations), a subset of source minibatch can be selected based on the approximated Wasserstein distance, which will be used for domain-adversarial training of joint adapter in next step. The following stage, while keeping the previously updated heads fixed, updates the rest of the model's parameters, using the standard gradient descent algorithm. All maximization problems of discrepancy losses are converted into minimization using reversed domain labels.

At test time, a new sample x_{test} will go through the trained joint adapter $\mathbf{A^j}$ to produce domain-invariant representation $\mathbf{A^j}(x_{test})$, which is then used by prediction head h_c to produce the corresponding event label.

4 Experiments

We evaluate our model on two related tasks: event identification and event detection. Given a trigger word in the context of an event mention, the former is formulated as a binary classification problem in which the goal is to determine if the trigger word expresses an event, while the latter is a multi-class classification task that requires model to assign the predicted label into one of the pre-defined 34 event types (include 1 negative type).

4.1 Datasets

TimeBank dataset (Pustejovsky et al., 2003) a fine-grained temporally annotated corpus of events and their positions and ordering in time. The text of the dataset were chosen from a wide range of sources from the news media domain. Events are annotated in a binary manner.

LitBank dataset (Sims et al., 2019) a recently introduced corpus of literary events. The dataset contains excerpts of 100 literary works from the Project Gutenberg corpus. Labels for events are binary.

System	In-domain(bn+nw)			Out-of-domain (bc)			Out-of-domain (cts)			Out-of-domain (wl)			Out-of-domain (un)		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
DAA-D	75.8	79.7	77.7	74.4	76.8	75.5	78.4	70.0	73.9	67.2	57.1	61.7	71.3	70.3	70.8
DAA-W	79.2	76.4	77.7	77.8	73.5	75.6	80.8	70.1	75.1	70.2	53.7	60.9	74.0	66.7	70.1
DAA-M	78.1	76.5	77.3	80.4	71.0	75.4	77.0	69.4	73.3	68.9	55.1	61.2	73.5	68.6	71.0
DAA-S	77.5	77.1	77.3	78.7	72.4	75.4	79.2	70.7	74.7	65.5	57.9	61.5	72.7	67.9	70.2
DAA	79.7	75.7	**77.7**	78.5	75.6	**76.9**	78.4	73.2	**75.6**	66.2	60.3	**63.1**	73.5	71.3	**72.3**

Table 4: Ablation study. Performances on the ACE-05 test datasets for different domains.

Automatic Content Extraction 2005 (ACE-05) dataset (Walker et al., 2005) a densely annotated corpus collected from 6 different domains: Newswire (nw) - 20%, Broadcast news (bn) - 20%, Broadcast conversation (bc) - 15%, Weblog (wl) - 15%, Usenet Newsgroups (un) - 15%, Conversational Telephone Speech (cts) - 15%. Events of the dataset are categorized into 33 types.

4.2 Experimental Setup

4.2.1 Unsupervised Domain Adaptation Setting

To formulate the unsupervised domain adaptation setting from the origin dataset of each task, we split the target domain's documents into two parts at the ratio of 1 to 4, a training dataset without labels which models have access to when learning, and a test dataset that models are evaluated on. For event identification, transfer experiments are performed in two ways: `LitBank-to-TimeBank`, and the reserve direction, `TimeBank-to-LitBank`. In event detection experiments, we combine samples from two closely related domains, `nw` and `bn`, to create a sizeable labeled training source dataset. Then, each of the other domains is considered the target domain of a single adaptation setting.

4.2.2 Implementation and Hyper-parameters

Our model leverage the pre-trained BERT-base model as the fixed foundation for all adapters, each of which has a down-sampled dimension of 96. All of the downstream heads are implemented as feed-forward networks with activation functions between layers. We train all models using batch size of 150, which composes of 90 source samples (60 of which will be used for domain-adversarial training) and 60 target samples. Weights of the losses are chosen from a grid-search of range $[0.01, 0.05, 0.1, 0.2, 0.5, 1, 5]$ using `bc` domain as development dataset. Every experiment is run 5 times epochs with different random seeds and the performance is reported using the average result of the 5 runs.

4.2.3 Baseline

We compare the proposed model **DAA** with several other baselines. In particular, for the task of event identification, the performance of domain-adversarial models implemented in Naik and Rosé (2020) are considered. Regarding the event detection task, our baselines include adaption results of **BERT** and **BERT+Adapter** models fine-tuned using only source dataset, and finally **BERT+DANN** which making use of unlabeled target data through adversarial training.

4.3 Experimental Result

Event Identification The results of our event identification experiments are presented in tables 2 and 3. In both settings, our proposed model **DAA** outperforms naive implementation of domain-adversarial on BERT by about 10 points in F1. We also note that high precision is observed from models transferring from `LitBank-to-TimeBank`, while the other direction has high recall. This imbalance is caused by the extreme disparity between the two adaptation settings, which our model manages to address and thus significantly improves out-of-domain performance in both cases.

Event Detection Table 1 showcases the results of our event detection experiment. The main conclusions from the table include: (1) The **BERT** baseline performs decently without using any mechanism to address the discrepancy between domains. This is due to the generalization potential of large unsupervised pre-trained language model. However, naively adopting DANN for BERT has an adverse effect, notably reducing the performance of **BERT+DANN** on all target domains. This outcome is consistent with results from Lin et al. (2020), further emphasizing the need for a compatible implementation method for domain-adversarial training on BERT's representations. (2) The results of **BERT+Adapter** proves that adapter-based tuning procedure is not only able to retain performance but also prevent over-fitting through capacity reduction, therefore performing better than the

System	bc			cts			wl			un		
	P	R	F	P	R	F	P	R	F	P	R	F
Lower	77.5	73.9	75.6	74.1	74.7	74.4	67.5	56.7	61.6	73.0	69.4	71.0
Middle	78.1	73.4	75.7	78.5	70.4	74.2	66.4	56.7	61.2	71.6	70.8	71.2
Higher	79.2	73.3	76.1	77.5	71.2	74.2	67.4	55.5	60.9	72.9	68.1	70.4
Last	79.6	72.6	75.9	79.7	69.5	74.2	66.4	56.6	61.0	73.0	69.7	71.3
Up-Dim	77.6	73.2	75.3	74.5	71.7	73.0	66.6	53.1	59.1	69.0	67.7	68.4
No-Rel	77.8	75.1	76.5	76.1	74.2	75.2	67.2	58.0	62.3	72.2	70.6	71.4
Full	78.5	75.6	**76.9**	78.4	73.2	**75.6**	66.1	59.6	**62.6**	73.5	71.3	**72.3**

Table 5: Domain-adversarial analysis. Performance on the ACE-05 test datasets for different domains.

fully fine-tuned **BERT** in case where it follows too closely to source domain. (3) Finally, our proposed model **DAA** manages to achieves the best adaptation performance across all target domains. In settings where domains are closely related such as bc and cts, **DAA** is more robust and thus performs better on target domain. On the other hand, **DAA** significantly outperforms baselines (3 to 5 points increase in F1 score) when transferring to target domains that are highly dissimilar to source domains (wl and un).

4.4 Ablation Study

To examine the effect of each of the proposed component individually, We perform an extensive ablation analysis for our **DAA** model by measuring domain adaptation ability of each trained model, with a single main component discarded (by setting the weight of its associated loss to 0), on ACE-05. In table 4, **DAA-D**, **DAA-W**, **DAA-M**, and **DAA-S** correspond to performances of partial models with domain-adversarial training, data selection component, self-supervised task, and orthogonality constraint removed, respectively. Results from the study show that every incomplete model performs consistently worse compare to the full model. In particular, while in-domain performances are retained across settings, different domains experience varying degree of reduction in target performance depending on its relation with the source domain. Especially, data drawn from the domains of wl and un are substantially diverged from the source domain. Therefore, components that address domains' dissimilarity play important roles in improving adaptation capability, which is confirmed by the fact that models such as **DAA-W** and **DAA-D** have the lowest results.

4.5 Domain-adversarial Analysis

The central component of our architecture is undoubtedly LDA whose responsibility is to ensure joint adapter extracts domain-invariant features for classifying event triggers. From the negative results of **BERT+DANN**, finding an appropriate way to implement domain-adversarial training for BERT is an important question. This section aims to demonstrate the effectiveness of our layer-wise implementation of DANN.

We apply domain alignment to different portions of BERT. Specifically, we partitioned 12 layers of the BERT-base encoder into 3 levels - **Lower**, **Middle**, **Upper** - each corresponds to the only 4 layers whose representations are used by domain-adversarial training. In addition, we present results of **Last** and **Up-Dim**. The former is original implementation where last layer's output is aligned , while the latter is similar to our model **Full** except the representation with full dimension (768) is used instead of the down-sampled ones. Finally, **No-Rel** is the same as **Full** but no relaxation is used.

Table 5 showcases the results of our experiment. Overall, we observe performance degrades on all three partial adaptation settings However, the changes vary across domains in each situation, probably stemming from the fact that adversarial training addresses different degrees of domain shifts in each layer. Moreover, taking only the last layer's representation as input for DANN component performs worse compare to all other multi-layer counterparts. Notably, using representations with full dimension significantly reduces out-of-domain performances of model. This result confirms the benefit of the bottleneck architecture. Not only the alignment of down-sampled representations is more effective, but the free parameters of up-sampling layers also increase model's capacity for the main downstream task.

4.6 Domain Discrepancy Analysis

To verify the effect of our method on alleviating the negative impact of the domain shift problem on the learning process, we compare each model's performance on different settings with varying shift magnitudes. Specifically, for each target domain,

based on the learned Wasserstein distance between the two domains, we quantify the distance of each target domain sample (in evaluation dataset) to the source dataset and group them into 2 disjoint sets: **FAR** - 25% of target samples that are farthest from the source dataset, and **CLOSE** - 25% of those closest to source dataset. The domain adaptation performances on these sets for 2 target domain `bc` and `wl`, together with the set of in-domain examples **IN-DOM** from `bnnw` domain, are provided in Table 6. When adapting to `bc` domain which has a low discrepancy to source domain, the results for each setting show little variance, but we still observe the over-fitting of **BERT** as performance of out-of-domain settings is lower compared to its in-domain score. Moreover, **BERT+DANN** is able to improve on **FAR** set, however at the cost of degradation in the other two settings. In contrast, the negative effect of high discrepancy between domains is apparent in the case of `wl` domain, as the gaps between each setting are all above 10 points. Notably, the results of **BERT+DANN** are lower than that of **BERT**, indicating that naive implementation of **DANN** is not only unable to align between source and target domains, but also causes negative transfer when trying to learn domain-invariant representation. On the other hand, in both case, **DAA** is able to address the weakness of the baseline and improves the performance on **FAR** and **CLOSE** simultaneously.

	bc		wl		bnnw
	FAR	CLOSE	FAR	CLOSE	IN-DOM
BERT	72.4	75.6	43.1	59.2	77.5
BERT+DANN	73.2	76.6	35.3	52.4	76.6
DAA	74.8	76.4	50.9	64.4	77.7

Table 6: Domain adaptation performances in F1 score with different domain shift settings.

5 Conclusion

We present a novel framework for ED in UDA setting that effectively leverages the generalization capability of large pre-trained language models through a shared-private adapter-based architecture. A layer-wise domain-adversarial training process combined with a Wasserstein-based data selection addresses the discrepancy between domains and produces domain-invariant representations. The proposed model achieves state-of-the-art results on several adaptation settings across multiple datasets. In the future, we plan to extend our approach in the several directions: (1) We will devise a method to incorporate target domain's private adapter to further improve model's out-of-domain performance.; (2) We will adapt our framework to more general settings such as multi-source domain adaptation and domain generalization.; and (3) We will extend our work to novel domains for ED (Trong et al., 2020).

Acknowledgments

This research has been supported by the Army Research Office (ARO) grant W911NF-21-1-0112. This research is also based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the Better Extraction from Text Towards Enhanced Retrieval (BETTER) Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ARO, ODNI, IARPA, the Department of Defense, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875*.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

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 Annual Meeting of the Association for Computational Linguistics (ACL)*.

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 (NAACL-HLT)*.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

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. *arXiv preprint arXiv:2004.10964*.

Yu Hong, Wenxuan Zhou, jingli zhang jingli, Guodong Zhou, and Qiaoming Zhu. 2018a. Self-regulation: Employing a generative adversarial network to improve event detection. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yu Hong, Wenxuan Zhou, Jingli Zhang, Guodong Zhou, and Qiaoming Zhu. 2018b. Self-regulation: Employing a generative adversarial network to improve event detection. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. *Proceedings of the 36th International Conference on Machine Learning (ICML)*.

Viet Dac Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2020a. Exploiting the matching information in the support set for few shot event classification. In *Proceedings of the 24th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*.

Viet Dac Lai, Thien Huu Nguyen, and Frank Dernoncourt. 2020b. Extensively matching for few-shot learning event detection. In *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*.

Viet Dac Lai, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020c. Event detection: Gate diversity and syntactic importance scoresfor graph convolution neural networks. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Chen Lin, Steven Bethard, Dmitriy Dligach, Farig Sadeque, Guergana K Savova, and T. A. Miller. 2020. Does bert need domain adaptation for clinical negation detection? *Journal of the American Medical Informatics Association (JAMIA)*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017a. Adversarial multi-task learning for text classification. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017b. Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

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)*.

Aakanksha Naik and Carolyn Rosé. 2020. Towards open domain event trigger identification using adversarial domain adaptation. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Nghia Ngo, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Learning to select important context words for event detection. In *Proceedings of the 24th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*.

Minh Van Nguyen, Viet Dac Lai, and Thien Huu Nguyen. 2021. Cross-task instance representation interactions and label dependencies for joint information extraction with graph convolutional networks. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent

neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*.

Thien Huu Nguyen and Ralph Grishman. 2016. Modeling skip-grams for event detection with convolutional neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Thien Huu Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. Adapterfusion: Non-destructive task composition for transfer learning. *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

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

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003. The timebank corpus. In *Corpus linguistics*, volume 2003, page 40. Lancaster, UK.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2021. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics (TACL)*.

Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. 2018. Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. 2018. Wasserstein distance guided representation learning for domain adaptation. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. 2018. A dirt-t approach to unsupervised domain adaptation. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Matthew Sims, Jong Ho Park, and David Bamman. 2019. Literary event detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Asa Cooper Stickland and Iain Murray. 2019. BERT and pals: Projected attention layers for efficient adaptation in multi-task learning. *Proceedings of the 36th International Conference on Machine Learning (ICML)*.

Meihan Tong, Bin Xu, Shuai Wang, Yixin Cao, Lei Hou, Juanzi Li, and Jun Xie. 2020. Improving event detection via open-domain trigger knowledge. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Hieu Man Duc Trong, Duc Trong Le, Amir Pouran Ben Veyseh, Thuat Nguyen, and Thien Huu Nguyen. 2020. Introducing a new dataset for event detection in cybersecurity texts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2005. Ace 2005 multilingual training corpus. In *Technical report, Linguistic Data Consortium*.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019. Adversarial training for weakly supervised event detection. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Dustin Wright and Isabelle Augenstein. 2020. Transformer based multi-source domain adaptation. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yifan Wu, Ezra Winston, Divyansh Kaushik, and Zachary Lipton. 2019. Domain adaptation with asymmetrically-relaxed distribution alignment. *Proceedings of the 36th International Conference on Machine Learning (ICML)*.

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Chen Yubo, Xu Liheng, Liu Kang, Zeng Daojian, Zhao Jun, et al. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.