Unleash GPT-2 Power for Event Detection

Amir Pouran Ben Veyseh[1], **Viet Dac Lai**[1],
Franck Dernoncourt[2], **and Thien Huu Nguyen**[1]
[1] Department of Computer and Information Science, University of Oregon,
Eugene, OR 97403, USA
[2] Adobe Research, San Jose, CA, USA
{apouranb,vietl,thien}@cs.uoregon.edu,
franck.dernoncourt@adobe.com

Abstract

Event Detection (ED) aims to recognize mentions of events (i.e., event triggers) and their types in text. Recently, several ED datasets in various domains have been proposed. However, the major limitation of these resources is the lack of enough training data for individual event types which hinders the efficient training of data-hungry deep learning models. To overcome this issue, we propose to exploit the powerful pre-trained language model GPT-2 to generate training samples for ED. To prevent the noises inevitable in automatically generated data from hampering training process, we propose to exploit a teacher-student architecture in which the teacher is supposed to learn anchor knowledge from the original data. The student is then trained on combination of the original and GPT-generated data while being led by the anchor knowledge from the teacher. Optimal transport is introduced to facilitate the anchor knowledge-based guidance between the two networks. We evaluate the proposed model on multiple ED benchmark datasets, gaining consistent improvement and establishing state-of-the-art results for ED.

1 Introduction

An important task of Information Extraction (IE) involves Event Detection (ED) whose goal is to recognize and classify words/phrases that evoke events in text (i.e., event triggers). For instance, in the sentence "*The organization* ***donated*** *2 million dollars to humanitarian helps.*", ED systems should recognize "*donated*" as an event trigger of type *Pay*. We differentiate two subtasks in ED, i.e., Event Identification (EI): a binary classification problem to predict if a word in text is an event trigger or not, and Event Classification (EC): a multi-class classification problem to classify event triggers according to predefined event types.

Several methods have been introduced for ED, extending from feature-based models (Ahn, 2006; Liao and Grishman, 2010a; Miwa et al., 2014) to advanced deep learning methods (Nguyen and Grishman, 2015; Chen et al., 2015; Nguyen et al., 2016c; Sha et al., 2018; Zhang et al., 2020b; Nguyen et al., 2021). Although deep learning models have achieved substantial improvement, their requirement of large training datasets together with the small sizes of existing ED datasets constitutes a major hurdle to build high-performing ED models. Recently, there have been some efforts to enlarge training data for ED models by exploiting unsupervised (Huang et al., 2016; Yuan et al., 2018) or distantly-supervised (Keith et al., 2017; Nguyen and Nguyen, 2018; Araki and Mitamura, 2018) techniques. The common strategy in these methods is to exploit unlabeled text data that are rich in event mentions to aid the expansion of training data for ED. In this work, we explore a novel approach for training data expansion in ED by leveraging the existing pre-trained language model GPT-2 (Radford et al., 2019) to automatically generate training data for models. Motivated by the promising performance of GPT models for text generation, we expect our approach to produce effective data for ED in different domains.

Specifically, we aim to fine-tune GPT-2 on existing training datasets so it can generate new sentences annotated with event triggers and/or event types, serving as additional training data for ED models. One direction to achieve this idea is to explicitly mark event triggers along with their event types in sentences of an existing ED dataset that can be used to fine-tune the GPT model for new data generation. However, one issue with this direction is that in existing ED datasets, numbers of examples for some rare event types might be small, potentially leading to the poor tuning performance of GPT and impairing the quality of generated examples for such rare events. In addition, large num-

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics
and the 11th International Joint Conference on Natural Language Processing, pages 6271–6282
August 1–6, 2021.

bers of event types in some ED datasets might make it more challenging for the fine-tuning of GPT to differentiate event types and produce high-quality data. To this end, instead of directly generating data for ED, we propose to use GPT-2 to only generate samples for the event identification task to simplify the generation and achieve data with better annotated labels (i.e., output sentences only are only marked with positions of event triggers). As such, to effectively leverage the generated EI data to improve ED performance, we propose a multi-task learning framework to train the ED models on the combination of the generated EI data and the original ED data. In particular, for every event trigger candidate in a sentence, our framework seeks to perform two tasks, i.e., EI to predict a binary label for being an event trigger or not, and ED to predict the event type (if any) evoked by the word via a multi-class classification problem. An input encoder is shared for both tasks that allow training signals from both generated EI data and original ED data to contribute to the representation learning in the encoder (i.e., transferring knowledge in generated EI data to ED models).

Despite the simplification to EI for better annotated labels of data, the generated sentences might still involve noises due to the inherent nature of the language generation, e.g., grammatically wrong sentences, inconsistent information, or incorrect event trigger annotations. As such, it is crucial to introduce mechanisms to filter the noises in generated data to enable effective transfer learning from generated EI data. To this end, prior works for GPT-based data generation for other tasks has attempted to directly remove noisy generated examples before actual usage for model training via some heuristic rules (Anaby-Tavor et al., 2020; Yang et al., 2020). However, heuristic rules are brittle and restricted in their coverage so they might overly filter the generated data or incorrectly retain some noisy generated samples. To address this issue, we propose to preserve all generated data for training and devise methods to explicitly limit impacts of noisy generated sentences in the models. In particular, we expect the inclusion of generated EI data into the training process for ED models might help to shift the representations of the models to better regions for ED. As such, we argue that this representation transition should only occur at a reasonable rate as drastic divergence of representations due to the generated data might be associated with noises in the data. Motivated by this intuition, we propose a novel teacher-student framework for our multi-task learning problem where the teacher is trained on the original clean ED datasets to induce anchor representation knowledge for data. The student, on the other hand, will be trained on both generated EI data and original ED data to accomplish transfer learning. Here, the anchor knowledge from the teacher will be leveraged to guide the student to prevent drastic divergence of representation vectors for noisy information penalization. Consequently, we propose a novel anchor information to implement this idea, seeking to maintain the same level of differences between the generated and original data (in terms of representation vectors) for both the teacher and the student (i.e., generated-vs-original data difference as the anchor). At the core of this techniques involves the computation of distance/difference between samples in generated and original data. In this work, we envision two types of information that models should consider when computing such distances for our problem: (1) representation vectors of the models for the examples, and (2) event trigger likelihood scores of examples based on the models (i.e., two examples in the generated and original data are more similar if they both correspond to event triggers). As such, we propose to cast this distance computation problem of generated and original data into an Optimal Transport (OT) problem. OT is an established method to compute the optimal transportation between two data distributions based on the probability masses of data points and their pair-wise distances, thus facilitating the integration of the two criteria of event trigger likelihoods and representation vectors into the distance computation between data point sets.

Extensive experiments and analysis reveal the effectiveness of the proposed approach for ED in different domains, establishing new state-of-the-art performance on the ACE 2005, CySecED and RAMS datasets.

2 Model

We formulate the task of Event Detection as a word-level classification problem as in prior work (Nguyen and Grishman, 2015; Ngo et al., 2020). Formally, given the sentence $S = [w_1, w_2, \ldots, w_n]$ and the candidate trigger word w_t, the goal is to predict the event type l from a pre-defined set of event types L. Note that if the word w_t is not a trigger word, the gold event type is $None$. Our proposed approach for this task consist of two stages: (1) Data Augmentation: to employ natural language generation to augment existing training datasets for ED, (2) Task Modeling: to propose a deep learning model for ED, exploiting available training data.

2.1 Data Augmentation

As presented in the introduction, our motivation in this work is to explore a novel approach for training data augmentation for ED based on the powerful pre-trained language model for text generation GPT2. Our overall strategy involves using some existing training dataset $\mathcal{O}$ for ED (i.e., original data) to fine-tune GPT-2. The fine-tuned model is then employed to generate a new labeled training set $\mathcal{G}$ (i.e., synthetic data) that will be combined with the original data $\mathcal{O}$ to train models for ED.

To simplify the training data generation task and enhance the quality of the synthetic data, we seek to generate data only for the subtask EI of ED where synthesized sentences are annotated with positions of their event triggers (i.e., event types for triggers are not required for the generation to avoid the complication with rare event types for fine-tuning). To this end, we first enrich each sentence $S \in \mathcal{O}$ with positions of event triggers that it contains to facilitate the GPT fine-tuning process. Formally, assume that $S = w_1, w_2, \ldots, w_n$ is a sentence of n words with only one event trigger word located at w_t, the enriched sentence S' for S would have the form: $S' = [BOS, w_1, \ldots, TRG_s, w_t, TRG_e, \ldots, w_n, EOS]$ where TRG_s and TRG_e are special tokens to mark the position of the event trigger, and BOS and EOS are special tokens to identify the beginning and the end of the sentence. Next, the GPT-2 model will be fine-tuned on the enriched sentences S' of $\mathcal{O}$ in an auto-regressive fashion (i.e., predicting the next token in S' given prior ones). Finally, using the fine-tuned GPT-2, we generate a new dataset $\mathcal{G}$ of $|\mathcal{O}|$ sentences ($|\mathcal{G}| = |\mathcal{O}|$) to achieve a balanced size. Here, we ensure that only generated sentences that contain the special tokens TRG_s and TRG_e (i.e., involving event trigger words) are added into $\mathcal{G}$, allowing us to identify the candidate trigger word in our word-level classification formulation for ED. As such, the combination $\mathcal{A}$ of the synthetic data $\mathcal{G}$ and the original data $\mathcal{O}$ ($\mathcal{A} = \mathcal{O} \cup \mathcal{G}$) will be leveraged to train our ED model in the next step.

To assess the quality of the synthetic data, we randomly select 200 sentences from $\mathcal{G}$ (generated by the fine-tuned GPT-2 model over the popular ACE 2005 training set for ED) and evaluate them regarding grammatical soundness, meaningfulness, and inclusion and correctness of annotated event triggers (i.e., whether the words between the tokens TRG_s and TRG_e evoke events or not). Among the sampled set, we find that 17% of the sentences contains at least one type of such errors.

2.2 Task Modeling

This section describes our model for ED to overcome the noises in the generated data $\mathcal{G}$ for model training. As discussed in the introduction, we employ the Teacher-Student framework with multi-task learning to achieve this goal. In the proposed framework, the teacher and student employs a base deep learning model with the same architecture and different parameters.

Base Model: Following the prior work (Wang et al., 2019), our base model consists of the BERT$_{base}$ model to represent each word w_i in the input sentence S with a vector e_i. Formally, the input sentence $[[CLS], w_1, w_2, \ldots, w_n, [SEP]]$ is fed into the BERT$_{base}$ model and the hidden states of the last layer of BERT are taken as the contextualized embeddings of the input words, i.e., $E = [e_1, e_2, \ldots, e_n]$. Note that if w_i contains more than one word-piece, the average of its word-piece embeddings is used for e_i. In our experiments, we find that fixing the BERT$_{base}$ parameters achieve higher performance. As such, to fine-tune the contextualized embeddings E for ED, we employ a Bi-directional Long Short-Term Memory (BiLSTM) network to consumes E; its hidden states, i.e., $H = [h_1, h_2, \ldots, h_n]$, are then employed as the final representations for the words in S. Finally, to create the final vector V for ED prediction, the max-pooled representation of the sentence, i.e., $\bar{h} = MAX\_POOL(h_1, h_2, \ldots, h_n)$, is concatenated with the representation of the trigger candidate, i.e., h_t. V is consumed by a feed-forward network, whose last layer has $|L|$ neurons, followed by a softmax layer to predict the distribution $P(\cdot|S, t)$ over possible event types in L. To train the model, we use negative log-likelihood as the loss function: $\mathcal{L}_{pred} = -\log P(l|S, t)$ where l is the gold label.

As the synthetic sentences in $\mathcal{G}$ only involve information about positions of event triggers (i.e., no event types included), we cannot directly combine $\mathcal{G}$ with $\mathcal{O}$ to train ED models with the loss $\mathcal{L}_{pred}$. To facilitate the integration of $\mathcal{G}$ into the training process, we introduce an auxiliary task of EI for the multi-task learning in the training process, seeking to predict the binary label l_{aux} for the trigger candidate w_t in S, i.e., $l_{aux} = 1$ if w_t is an event trigger. To perform this auxiliary task, we employ another feed-forward network, i.e., $\texttt{FF}_{aux}$, which also consumes the overall vector V as input. This feed-forward network has one neuron with the sigmoid activation function in the last layer to estimate the event trigger likelihood score: $P(l_{aux} = 1|S, t) = \texttt{FF}_{aux}(V)$. Finally, to train the base model with the auxiliary task, we exploit the binary cross-entropy loss: $\mathcal{L}_{aux} = -(l_{aux} \log(\texttt{FF}_{aux}(V)) + (1 - l_{aux}) \log(1 - \texttt{FF}_{aux}(V)))$. Note that the main ED task and the auxiliary EI task are done jointly in a single training process where the loss $\mathcal{L}_{pred}$ for ED is computed only for the original data $\mathcal{O}$. The loss $\mathcal{L}_{aux}$, in contrast, will be obtained for both original and synthetic data in $\mathcal{A}$.

Knowledge Consistency: The generated data $\mathcal{G}$ is not noise-free. As such, training the ED model on $\mathcal{A}$ could lead to inferior performance. To address this issue, as discussed in the introduction, we propose to first learn the anchor knowledge from the original data $\mathcal{O}$, then use that to lead the model training on $\mathcal{A}$ to prevent drastic divergence from the anchor knowledge (i.e., knowledge consistency promotion), thus constraining the noises. Hence, we propose a teacher-student network, in which the teacher is first trained on $\mathcal{O}$ to learn the anchor knowledge. The student network will be trained on $\mathcal{A}$ afterward leveraging the consistency guidance with the induced anchor knowledge from the teacher. We will also use the student network as the final model for our ED problem in this work.

In our framework, both teacher and student networks will be trained in the multi-task setting with ED and EI tasks. In particular, the training losses for both ED and EI will be computed based on $\mathcal{O}$ for the teacher (the loss to train the teacher is: $\mathcal{L}_{pred} + \tau\mathcal{L}_{aux}$ where τ is a trade-off parameter). In contrast, the combined data $\mathcal{A}$ will be used to compute the EI loss for the student while the ED loss for the student can only be computed on the original data $\mathcal{O}$. As such, we propose to enforce the knowledge consistency between the two networks for both the main task ED and the auxiliary task EI during the training of the student model. First, to achieve the knowledge consistency for ED, we seek to minimize the KL divergence between the teacher-predicted label-probability distribution and the student-predicted label-probability distributions. Formally, for a sentence $S \in \mathcal{O}$, the label-probability distributions of the teacher and the student, i.e., $P_t(\cdot|S, t)$ and $P_s(\cdot|S, t)$ respectively, are employed to compute the KL-divergence loss $\mathcal{L}_{KL} = -\Sigma_{l \in L} P_t(l|S, t) \log(\frac{P_t(l|S,t)}{P_s(l|S,t)})$. By decreasing the KL-divergence during the student's training, the model is encouraged to make similar predictions as the teacher for the same original sentence, thereby preventing noises to mislead the student. Note that different from traditional teacher-student networks that employ KL to achieve knowledge distillation on unlabelled data (Hinton et al., 2015), the KL divergence in our model is leveraged to enforce knowledge consistency to prevent noises in labeled data automatically generated by GPT-2.

Second, for the auxiliary task EI, instead of enforcing the student-teacher knowledge consistency via similarity predictions, we argue that it will be more beneficial to leverage the difference between the original data $\mathcal{O}$ and the generated data $\mathcal{G}$ as an anchor knowledge to promote consistency. In particular, we expect that the student which is trained on $\mathcal{A}$, should discern the same difference between $\mathcal{G}$ and $\mathcal{O}$ as the teacher which is trained only on the original data $\mathcal{O}$. Formally, during student training, for each mini-batch, the distances between the original data and the generated data detected by the teacher and the student are denoted by $d^T_{\mathcal{O},\mathcal{G}}$ and $d^S_{\mathcal{O},\mathcal{G}}$, respectively. To enforce the $\mathcal{O}$-$\mathcal{G}$ distance consistency between the two networks, the following loss is added into the overall loss function: $\mathcal{L}_{dist} = \frac{|d^T_{\mathcal{O},\mathcal{G}} - d^S_{\mathcal{O},\mathcal{G}}|}{|B|}$, where $|B|$ is the mini-batch size. The advantage of this novel knowledge consistency enforcement compared to the KL-divergence is that it explicitly exploits the different nature of the original and generated data to facilitate the mitigation of noises in the generated data.

A remaining question for our proposed knowledge consistency concerns how to assess the difference between the original and the generated data from the perspective of the teacher, i.e., $d^T_{\mathcal{O},\mathcal{G}}$, and the student networks, i.e., $d^S_{\mathcal{O},\mathcal{G}}$. In this section, we will describe our method from the perspective of the student (the same method is employed for the teacher network). In particular, we define the difference between the original and the generated data as the cost of transforming $\mathcal{O}$ to $\mathcal{G}$ such that for the transformed data the model will make the same predictions as $\mathcal{G}$. How can we compute the cost of such transformation? To answer this question, we propose to employ Optimal Transport (OT) which is an established method to find the efficient transportation (i.e., transformation with the lowest cost) of one probability distribution to another one. Formally, given the probability distributions $p(x)$ and $q(y)$ over the domains $\mathcal{X}$ and $\mathcal{Y}$, and the cost function $C(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$ for mapping $\mathcal{X}$ to $\mathcal{Y}$, OT finds the optimal joint distribution $\pi^*(x, y)$ (over $\mathcal{X} \times \mathcal{Y}$) with marginals $p(x)$ and $q(y)$, i.e., the cheapest transportation from $p(x)$ to $q(y)$, by solving the following problem:

$$\pi^*(x,y) = \min_{\pi \in \Pi(x,y)} \int_{\mathcal{Y}} \int_{\mathcal{X}} \pi(x,y) C(x,y) dx dy \\ \textbf{s.t. } x \sim p(x) \text{ and } y \sim q(y), \tag{1}$$

where $\Pi(x, y)$ is the set of all joint distributions with marginals $p(x)$ and $q(y)$. Note that if the distributions $p(x)$ and $q(y)$ are discrete, the integrals in Equation 1 are replaced with a sum and the joint distribution $\pi^*(x, y)$ is represented by a matrix whose entry (x, y) represents the probability of transforming the data point $x \in \mathcal{X}$ to $y \in \mathcal{Y}$ to convert the distribution $p(x)$ to $q(y)$. By solving the problem in Equation 1[1], the cost of transforming the discrete distribution $p(x)$ to $q(y)$ (i.e., Wasserstein distance $Dist_W$) is defined as: $Dist_W = \Sigma_{x \in \mathcal{X}} \Sigma_{y \in \mathcal{Y}} \pi^*(x, y) C(x, y)$.

In order to utilize OT to compute the transformation cost between $\mathcal{O}$ and $\mathcal{G}$, i.e., $d^S_{\mathcal{O},\mathcal{G}}$, we propose to define the domain $\mathcal{X}$ and $\mathcal{Y}$ as the representation spaces of the sentences in $\mathcal{O}$ and $\mathcal{G}$, respectively, obtained from the student network. In particular, a data point $x \in \mathcal{X}$ represents a sentence $X_o \in \mathcal{O}$. Similarly, a data point $y \in \mathcal{Y}$ stands for a sentence $Y_g \in \mathcal{G}$. To define the cost function $C(x, y)$ for OT, we compute the Euclidean distance between the representation vectors of the sentences X_o and Y_g (obtained by max-pooling over representations of their words): $C(x, y) = \left\|\bar{h}^X_o - \bar{h}^Y_g\right\|$ where $\bar{h}^X_o = MAX\_POOL(h^X_{o,1}, \ldots, h^X_{o,|X_o|})$, $\bar{h}^Y_g = MAX\_POOL(h^Y_{g,1}, \ldots, h^Y_{g,|Y_g|})$, and $h^X_{o,i}$ and $h^Y_{g,i}$ are the representation vectors of the i-th words of X_o and Y_g, respectively, obtained from the student's BiLSTM. Also, to define the discrete distribution $p(x)$ for OT over $\mathcal{X}$, we employ the event trigger likelihood $Score^X_o$ for the trigger candidate of each sentence X_o in $\mathcal{X}$ that is returned by the feed-forward network $\texttt{FF}^S_{aux}$ for the auxiliary task EI in the student model, i.e, $Score^X_o = \texttt{FF}^S_{aux}(X_o)$. Afterward, we apply the softmax function over the scores of the original sentences in the current mini-batch to obtain $p(x)$, i.e., $p(x) = Softmax(Score^X_o)$. Similarly, the discrete distribution $q(y)$ is defined as $q(y) = Softmax(Score^Y_g)$. To this end, by solving the OT problem in Equation 1 and obtaining the efficient transport plan $\pi^*(x, y)$ using this setup, we can obtain the distance $d^S_{\mathcal{O},\mathcal{G}}$. In the same way, the distance $d^T_{\mathcal{O},\mathcal{G}}$ can be computed using the representations and event trigger likelihoods from the teacher network. Note that in this way, we can integrate both representation vectors of sentences and event trigger likelihoods into the distance computation between data as motivated in the introduction.

Finally, to train the student model, the following combined loss function is used in our framework: $\mathcal{L} = \mathcal{L}_{pred} + \alpha\mathcal{L}_{aux} + \beta\mathcal{L}_{KL} + \gamma\mathcal{L}_{dist}$, where α, β, and γ are the trade-off parameters.

3 Experiments

3.1 Datasets, Baselines & Hyper-Parameters

To evaluate the effectiveness of the proposed model, called the GPT-based data augmentation model for ED with OT (GPTEDOT), we conduct experiments on the following ED datasets:

ACE 2005 (Walker et al., 2006): This dataset annotates 599 documents for 33 event types that cover different text domains(e.g., news, weblog or conversation documents). We use the same preprocessing script and data split as prior works (Lai et al., 2020c; Tong et al., 2020b) to achieve fair comparisons. In particular, the data split involves 529/30/40 articles for train/dev/test sets respectively. For this dataset, we compare our model with prior state-of-the-art models reported in the recent works (Lai et al., 2020c; Tong et al., 2020b), including BERT-based models such as DMBERT, AD-DMBERT (Wang et al., 2019), DRMM, EKD (Tong et al., 2020b), and GatedGCN (Lai et al., 2020c).

CySecED (Man Duc Trong et al., 2020): This dataset provides 8,014 event triggers for 30 event types from 300 articles of the cybersecurity domain (i.e., cybersecurity events). We follow the the same pre-processing and data split as the original work (Man Duc Trong et al., 2020) with 240/30/30 documents for the train/dev/test sets. To be consistent with other experiments and facilitate the data generation based on GPT-2, the experiments on Cy-

[1] It is worth mentioning that this problem is intractable so we solve its entropy-based approximation using the Sinkhorn algorithm (Peyre and Cuturi, 2019).

SecED are conducted at the sentence level where inputs for models involve sentences. As such, we employ the state-of-the-art sentence-level models reported in (Man Duc Trong et al., 2020), i.e., DMBERT (Wang et al., 2019), BERT-ED (Yang et al., 2019), as the baselines for CySecED.

RAMS (Ebner et al., 2020): This dataset annotates 9,124 event triggers for 38 event types. We use the official data split with 3,194, 399, and 400 documents for training, development, and testing respectively for RAMS. We also perform ED at the sentence level in this dataset. For the baselines, we utilize recent state-of-the-art BERT-based models for ED, i.e., DMBERT (Wang et al., 2019) and GatedGCN (Lai et al., 2020c). For a fair comparison, the performance of such baseline models is obtained via their official implementations from the original papers that are fine-tuned for RAMS.

For each dataset, we use its training and development data to fine-tune the GPT-2 model. We tune the hyperparameters for the proposed teacher-student architecture using a random search. All the hyperparameters are selected based on the F1 scores on the development set of the ACE 2005 dataset. The same hyper-parameters from this fine-tuning are then applied for other datasets for consistency. In our model we use the small version of GPT-2 to generate data. In the base model, we use BERT$_{base}$, 300 dimensions in the hidden states of BiLSTM and 2 layers of feed-forward neural networks with 200 hidden dimensions to predict events. The trade-off parameters τ, α, β and γ are set to 0.1, 0.1, 0.05, and 0.08, respectively. The learning rate is set to 0.3 for the Adam optimizer and the batch size of 50 are employed during training. Finally, note that we do not update the BERT model for word embeddings in this work due to its better performance on the development data of ACE 2005.

3.2 Results

Results of experiments on the ACE 2005 test set are shown in Table 1. The most important observation is that the proposed model GPTEDOT significantly outperforms all the baseline models ($p < 0.01$), thus showing the benefits of GPT-generated data and the teacher-student framework with knowledge consistency for ED in this work. In particular, compared to the BERT-based models that leverage data augmentation, i.e., AD-DMBERT (Wang et al., 2019) with semi-supervised and adversarial learning, DRMM (Tong et al., 2020a) with image-enhanced models, and EKD (Tong et al., 2020b) with external open-domain event triggers, the better performance of GPTEDOT highlights the advantages of GPT-2 to generate data for ED models.

Model	P	R	F1
CNN (Nguyen and Grishman, 2015)	71.8	66.4	69.0
DMCNN (Chen et al., 2015)	75.6	63.6	69.1
DLRNN (Duan et al., 2017)	77.2	64.9	70.5
ANN-S2 (Liu et al., 2017)	78.0	66.3	71.7
GMLATT (Liu et al., 2018)	78.9	66.9	72.4
GCN-ED (Nguyen and Grishman, 2018)	77.9	68.8	73.1
Lu's DISTILL (Lu et al., 2019)	76.3	71.9	74.0
TS-DISTILL (Liu et al., 2019)	76.8	72.9	74.8
DMBERT* (Wang et al., 2019)	77.6	71.8	74.6
AD-DMBERT* (Wang et al., 2019)	77.9	72.5	75.1
DRMM* (Tong et al., 2020a)	77.9	74.8	76.3
GatedGCN* (Lai et al., 2020c)	78.8	76.3	77.6
EKD* (Tong et al., 2020b)	79.1	78.0	78.6
GPTEDOT*	82.3	76.3	**79.2**

Table 1: Performance on the on ACE 2005 test set. * indicates models that use BERT for the encoding.

Model	P	R	F1
CNN (Nguyen and Grishman, 2015)	51.8	36.7	43.0
DMCNN (Chen et al., 2015)	47.5	38.7	43.2
GCN-ED (Nguyen and Grishman, 2018)	46.3	51.8	48.9
MOGANED (Yan et al., 2019)	53.7	59.6	56.5
CyberLSTM (Satyapanich et al., 2020)	42.5	29.0	34.5
DMBERT (Wang et al., 2019)	59.4	51.3	55.1
BERT-ED (Man Duc Trong et al., 2020)	60.2	56.1	58.1
GPTEDOT	65.9	64.1	**65.0**

Table 2: Comparison with state-of-the-art models on CySecED. All the models in this table use BERT.

Model	P	R	F1
DMBERT (Wang et al., 2019)	62.6	44.0	51.7
GatedGCN (Lai et al., 2020c)	66.5	59.0	62.5
GPTEDOT	55.5	78.6	**65.1**

Table 3: Model's performance on RAMS. All the models use BERT in this table.

Results of experiments on the CySecED test set are presented in Table 2. This table reveals that the teacher-student architecture GPTEDOT significantly improves the performance over previous state-of-the-art models for ED in cybersecurity domain. This is important as it shows that the proposed model is effective in different domains. In addition, our results also suggest that GPT-2 can be employed to generate effective data for ED in domains where data annotation for ED requires extensive domain expertise and expensive cost to obtain such as the cybersecurity events. Moreover, the higher margin of improvement for GPTEDOT on CySecED compared to the those on the ACE

2005 dataset suggests the necessity of using more training data for ED in technical domains.

Finally, results of experiments on the RAMS test set are reported in Table 3. Consistent with our experiments on ACE 2005 and CySecED, our proposed model achieve significantly higher performance than existing state-of-the-art models ($p < 0.01$), thus further confirming the advantages of GPTEDOT for ED.

Model	P	R	F1
GPTEDOT (full)	82.4	75.0	**78.5**
Base$^{\mathcal{O}}$	78.2	73.7	75.9
Base$^{\mathcal{A}}$	75.8	73.9	74.9
Teacher^{-A}	76.9	78.1	77.5
Teacher^{-M}	75.8	77.9	76.9
Student^{-}OT	75.4	79.3	77.3
Student^{-}KL	76.8	77.3	77.0
Student^{+OT}	76.1	76.6	76.4
Student^{+KL}	77.1	76.7	76.9
OT^{-Rep}	76.8	77.3	77.0
OT^{-Score}	78.0	77.1	77.6

Table 4: Ablation study on the ACE 2005 dev set.

3.3 Ablation Study

This ablation study evaluates the effectiveness of different components in GPTEDOT for ED. First, for the importance of the generated data $\mathcal{G}$ from GPT-2 and the teacher-student architecture to mitigate noises, we examine the following baselines: (1) **Base**$^{\mathcal{O}}$: The baseline is the base model trained only on the original data $\mathcal{O}$, thus being equivalent to the teacher model and not using the student model; and (2) **Base**$^{\mathcal{A}}$: This baseline trains the base model on the combination of the original and generated data, i.e., $\mathcal{A}$, using the multi-learning setting (i.e., the teacher model is excluded).

Second, for the multi-task learning design in the teacher network, we explore the following ablated models: (3) **Teacher**$^{-A}$: This baseline removes the auxiliary task EI in the teacher from GPTEDOT. As such, the OT-based knowledge consistency for EI is also eliminated; (4) **Teacher**$^{-M}$: In this model, the main task ED is utilize to train the teacher, so the corresponding KL-based knowledge consistency for ED is also removed.

Third, for the design of the knowledge consistency losses in the student network, we evaluate the following baselines: (5) **Student**$^{-OT}$: This ablated model eliminates the OT-based knowledge consistency loss for the auxiliary task EI in the student's training of GPTEDOT (the auxiliary task is still employed for the teacher and the student); (6) **Student**$^{-KL}$: For this model, the KL-based knowledge consistency for the main task ED is ignored in the student's training; (7) **Student**$^{+OT}$: In this baseline, we use OT for the knowledge consistency on both the main and the auxiliary tasks. Here, for the main task ED, the cost function $C(x, y)$ for OT is still obtained via the Euclidean distances between representation vectors while the distributions $p(x)$ and $p(y)$ are based on the maximum probabilities of the label-probability distributions $P_s(.|X_o, t_o)$ and $P_s(Y_g, t_g)$ for the ED task; and (8) **Student**$^{+KL}$: This baseline employs the KL divergence between models' predicted distributions to enforce the teacher-student consistency for both the main task and the auxiliary task. To this end, for the auxiliary task EI, we convert the final activation of $\texttt{FF}_{aux}$ into a distribution with two data points (i.e., $[\texttt{FF}_{aux}(X), 1 - \texttt{FF}_{aux}(X)]$) to compute the KL divergence between the teacher and the student.

Finally, for the importance of Euclidean distances and event trigger likelihoods in the OT-based distance between $\mathcal{O}$ and $\mathcal{G}$ for knowledge consistency in EI, we investigate two baselines: (9) **OT**$^{-Rep}$: Here, to compute OT, we use constant cost between every pair of sentences, i.e., $C(x, y) = 1$ (i.e., ignoring representation-based distances); and (10) **OT**$^{-Score}$: This model uses uniform distributions for $p(x)$ and $q(y)$ to compute the OT (i.e., ignoring event trigger likelihoods).

We report the performance of the models (on the ACE 2005 development set) for the ablation study in Table 4. There are several observations from this table. First, the generated data $\mathcal{G}$ and the teacher-student architecture are necessary for GPTEDOT to achieve the highest performance. In particular, comparing with Base$^{\mathcal{O}}$, the better performance of GPTEDOT indicates the benefits of the GPT-generated data. Moreover, the better performance of Base$^{\mathcal{O}}$ over Base$^{\mathcal{A}}$ reveals that the simple combination of the synthetic and original data without any effective method to mitigate noises might be harmful. Second, the lower performance of Teacher^{-A} and Teacher^{-M} shows that both the auxiliary and the main task (i.e., multi-task learning) in the teacher are integral to produce the best performance. Third, the choice of methods to promote knowledge consistency is important and the proposed combination of KL and OT for the ED and EI tasks (respectively) are necessary. In particular, removing or replacing each of them with the other one (i.e., Student^{+OT} and Student^{+KL}) would de-

Dataset	Sentence
ACE 2005	I was totally shocked by the court's decision to agree with Sam Sloan after he TRG$_s$ **sued** TRG$_e$ his children.
CySecED	According to the last update by the company, the following techniques are used to protect against such TRG$_s$ **malware** TRG$_e$.
RAMS	The Russian officials TRG$_s$ **vowed** TRG$_e$ to bomb the ISIS bases after the last week's TRG$_s$ **attack** TRG$_e$.

Table 5: Generated sentences by GPT-2 for different datasets. Event triggers are shown in boldface that are surrounded by the special tokens TRG$_s$ and TRG$_e$ generated by GPT-2.

Error Type	Sentence Example	Proportion
Incompleteness	A federal judge on Monday settled TRG$_s$ **charges** TRG$_e$ against seven members of	18%
Repetition	Do you think the TRG$_s$ **attack** TRG$_e$ will happen to you or do you think the TRG$_s$ **attack** TRG$_e$ will happen to you?	15%
Inconsistency	this morning we were watching the news and heard the news about the tragic TRG$_s$ **death** TRG$_e$ of a young boy and her mother in Iraq.	12%
Missing Labels	Aaron Tramailer's story is the story of a woman who was forced into **suicide**.	29%
Incorrect Labels	The SEC is a very good place to TRG$_s$ hide TRG$_e$ money.	26%

Table 6: Samples of noisy generated sentences for the ACE 2005 dataset from GPT-2. Event triggers are shown in boldface and the special tokens TRG$_s$ and TRG$_e$ are generated by GPT-2.

$\|\mathcal{G}\|$	P	R	F1
0.5 * $\|\mathcal{O}\|$	80.3	72.4	76.2
1.0 * $\|\mathcal{O}\|$	82.4	75.0	78.5
2.0 * $\|\mathcal{O}\|$	81.3	73.3	77.1
3.0 * $\|\mathcal{O}\|$	78.4	71.8	75.0

Table 7: The performance of GPTEDOT on the ACE 2005 dev set with different sizes of the generated data $\mathcal{G}$.

crease the performance significantly. Finally, in the proposed consistency method based on OT for EI, it is beneficial to employ both representation-level distances (i.e., OT^{-Rep}) and models' predictions for event trigger likelihoods (i.e., OT^{-Score}) as removing any of them hurts the performance.

3.4 Analysis

To provide more insights into the quality of the synthetic data $\mathcal{G}$, we provide samples of sentences that are generated by the fine-tuned GPT-2 model on each dataset in Table 5. This table illustrates that the generated sentences also belong to the domains of the original data (i.e., the cybersecurity domain). As such, combining synthetic data with original data is promising for improving ED performance as demonstrated in our experiments.

As discussed earlier, the generated data $\mathcal{G}$ is not free of noise. In order to better understand the types of errors existing in generated sentences, we manually assess 200 sentences randomly selected from the set $\mathcal{G}$ generated by the fine-tuned GPT-2 model on the ACE 2005 dataset. We categorize the errors into five types and provide their proportions along with example for each error type in Table 6. This table shows that the majority of errors are due to missing labels (i.e., no special tokens TRG$_s$ and TRG$_e$ are generated) or incorrect labels (i.e., marked words are not event triggers of interested types) generated by the language model.

Finally, to study the importance of the size of the generated data to augment training set for ED, we conduct an experiment in which different numbers of generated samples in $\mathcal{G}$ (for the ACE 2005 dataset) are combined with the original data $\mathcal{O}$. The results are shown in Table 7. According to this table, the highest performance of the proposed model is achieved when the numbers of the generated and original data are equal. More specifically, decreasing the number of generated samples potentially limits the benefits of data augmentation. On the other hand, increasing the size of generated data might introduces extensive noises and become harmful to the ED models.

4 Related Work

Early methods for ED have employed feature-based techniques (Ahn, 2006; Ji and Grishman, 2008; Patwardhan and Riloff, 2009; Liao and Grishman, 2010a,b; Hong et al., 2011; McClosky et al., 2011; Li et al., 2013; Miwa et al., 2014; Yang and Mitchell, 2016). Later, advanced deep learning methods (Nguyen and Grishman, 2015; Chen et al., 2015; Nguyen et al., 2016a,b; Sha et al., 2018; Zhang et al., 2019; Yang et al., 2019; Nguyen and Nguyen, 2019; Zhang et al., 2020b) have been applied for ED. One challenge for ED research is the limited size of existing datasets that hinder the training of effective models. Prior works have attempted to address this issue via unsupervised (Huang et al., 2016; Yuan et al., 2018), semi-supervised (Liao and Grishman, 2010a; Huang and Riloff, 2012; Ferguson et al., 2018), distantly supervised (Keith et al., 2017; Nguyen and Nguyen, 2018; Zeng et al., 2017; Araki and Mitamura, 2018), and few/zero-shot (Huang et al., 2018; Lai et al., 2020a,b) learning. In this work, we propose a novel method to augment training data for ED by exploiting the powerful language model GPT-2 to automatically generate new samples.

Leveraging GPT-2 for augmenting training data has also been studied for other NLP tasks recently (e.g., relation extraction, commonsense reasoning) (Papanikolaou and Pierleoni, 2020; Zhang et al., 2020a; Yang et al., 2020; Madaan et al., 2020; Bosselut et al., 2019; Kumar et al., 2020; Anaby-Tavor et al., 2020; Peng et al., 2020). However, none of those works has explored GPT-2 for ED. In addition, existing methods only resort to heuristics to filter out noisy samples generated by GPT-2. In contrast, we propose a novel differentiable method capable of preventing noises from diverging representation vectors of the models for ED.

5 Conclusion

We propose a novel method for augmenting training data for ED using the samples generated by the language model GPT-2. To avoid noises in the generated data, we propose a novel teacher-student architecture in a multi-task learning framework. We introduce a mechanism for knowledge consistency enforcement to mitigate noises from generated data based on optimal transport. Experiments on various ED benchmark datasets demonstrate the effectiveness of the proposed method.

Acknowledgments

This research has been supported by the Army Research Office (ARO) grant W911NF-21-1-0112 and the NSF grant CNS-1747798 to the IUCRC Center for Big Learning. 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

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*.

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Jun Araki and Teruko Mitamura. 2018. Open-domain event detection using distant supervision. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

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)*.

Shaoyang Duan, Ruifang He, and Wenli Zhao. 2017. Exploiting document level information to improve event detection via recurrent neural networks. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

James Ferguson, Colin Lockard, Daniel S Weld, and Hannaneh Hajishirzi. 2018. Semi-supervised event extraction with paraphrase clusters. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. In *Proceedings of the Deep Learning Workshop at NeurIPS*.

Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare Voss, Jiawei Han, and Avirup Sil. 2016. Liberal event extraction and event schema induction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).*

Lifu Huang, Heng Ji, Kyunghyun Cho, and Clare R. Voss. 2018. Zero-shot transfer learning for event extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).*

Ruihong Huang and Ellen Riloff. 2012. Bootstrapped training of event extraction classifiers. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL).*

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).*

Katherine Keith, Abram Handler, Michael Pinkham, Cara Magliozzi, Joshua McDuffie, and Brendan O'Connor. 2017. Identifying civilians killed by police with distantly supervised entity-event extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245.*

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, Franck Dernoncourt, and Thien Huu Nguyen. 2020b. Extensively matching for few-shot learning event detection. In *Proceedings of the 1st Joint Workshop on Narrative Understanding, Storylines, and Events (NUSE) at ACL 2020.*

Viet Dac Lai, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020c. Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In *Proceedings of the 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).*

Shasha Liao and Ralph Grishman. 2010a. Filtered ranking for bootstrapping in event extraction. In *Proceedings of the International Conference on Computational Linguistics (COLING).*

Shasha Liao and Ralph Grishman. 2010b. Using document level cross-event inference to improve event extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).*

Jian Liu, Yubo Chen, and Kang Liu. 2019. Exploiting the ground-truth: An adversarial imitation based knowledge distillation approach for event detection. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI).*

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2018. Event detection via gated multilingual attention mechanism. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI).*

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

Yaojie Lu, Hongyu Lin, Xianpei Han, and Le Sun. 2019. Distilling discrimination and generalization knowledge for event detection via delta-representation learning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).*

Aman Madaan, Dheeraj Rajagopal, Yiming Yang, Abhilasha Ravichander, Eduard Hovy, and Shrimai Prabhumoye. 2020. Eigen: Event influence generation using pre-trained language models. *arXiv preprint arXiv:2010.11764.*

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 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

David McClosky, Mihai Surdeanu, and Christopher Manning. 2011. Event extraction as dependency parsing. In *BioNLP Shared Task Workshop.*

Makoto Miwa, Paul Thompson, Ioannis Korkontzelos, and Sophia Ananiadou. 2014. Comparable study of event extraction in newswire and biomedical domains. In *Proceedings of the International Conference on Computational Linguistics (COLING).*

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).*

Minh Van Nguyen and Thien Huu Nguyen. 2018. Who is killed by police: Introducing supervised attention for hierarchical lstms. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016a. Joint event extraction via recurrent neural 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, Lisheng Fu, Kyunghyun Cho, and Ralph Grishman. 2016b. A two-stage approach for extending event detection to new types via neural networks. In *Proceedings of the 1st ACL Workshop on Representation Learning for NLP (RepL4NLP)*.

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 (ACL)*.

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)*.

Thien Huu Nguyen, Adam Meyers, and Ralph Grishman. 2016c. New york university 2016 system for kbp event nugget: A deep learning approach. In *Proceedings of Text Analysis Conference (TAC)*.

Trung Minh Nguyen and Thien Huu Nguyen. 2019. One for all: Neural joint modeling of entities and events. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Yannis Papanikolaou and Andrea Pierleoni. 2020. Dare: Data augmented relation extraction with gpt-2. In *SciNLP workshop at the Conference on Automated Knowledge Base Construction (AKBC)*.

Siddharth Patwardhan and Ellen Riloff. 2009. A unified model of phrasal and sentential evidence for information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Baolin Peng, Chenguang Zhu, Michael Zeng, and Jianfeng Gao. 2020. Data augmentation for spoken language understanding via pretrained models. *arXiv preprint arXiv:2004.13952*.

Gabriel Peyre and Marco Cuturi. 2019. Computational optimal transport: With applications to data science. In *Foundations and Trends in Machine Learning*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Taneeya Satyapanich, Francis Ferraro, and Tim Finin. 2020. Casie: Extracting cybersecurity event information from text. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

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)*.

Meihan Tong, Shuai Wang, Yixin Cao, Bin Xu, Juanzi Li, Lei Hou, and Tat-Seng Chua. 2020a. Image enhanced event detection in news articles. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

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

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. 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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Haoran Yan, Xiaolong Jin, Xiangbin Meng, Jiafeng Guo, and Xueqi Cheng. 2019. Event detection with multi-order graph convolution and aggregated attention. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. Generative data augmentation for commonsense reasoning. In *Proceedings of the Findings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Quan Yuan, Xiang Ren, Wenqi He, Chao Zhang, Xinhe Geng, Lifu Huang, Heng Ji, Chin-Yew Lin, and Jiawei Han. 2018. Open-schema event profiling for massive news corpora. In *Proceedings of the Conference on Information and Knowledge Management (CIKM)*.

Ying Zeng, Yansong Feng, Rong Ma, Zheng Wang, Rui Yan, Chongde Shi, and Dongyan Zhao. 2017. Scale up event extraction learning via automatic training data generation. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Danqing Zhang, Tao Li, Haiyang Zhang, and Bing Yin. 2020a. On data augmentation for extreme multi-label classification. *arXiv preprint arXiv:2009.10778*.

Junchi Zhang, Yanxia Qin, Yue Zhang, Mengchi Liu, and Donghong Ji. 2019. Extracting entities and events as a single task using a transition-based neural model. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Yunyan Zhang, Guangluan Xu, Yang Wang, Daoyu Lin, Feng Li, Chenglong Wu, Jingyuan Zhang, and Tinglei Huang. 2020b. A question answering-based framework for one-step event argument extraction. In *IEEE Access, vol 8, 65420-65431*.