DAFS: A Domain Aware Few Shots Generative Model for Event Detection

Nan Xia[1], Hang Yu[1*], Yin Wang[1], Junyu Xuan[2] and Xiangfeng Luo[1]

[1]School of Computer Engineering and Science, Shanghai University, Shang Da Street No.99, Shang Hai, 200444, China.
[2]Australian artificial intelligence institute, University of Technology Sydney, 15 Broadway Ultimo, Sydney State, 2007, Australia.

*Corresponding author(s). E-mail(s): hang.yu@shu.edu.cn;
Contributing authors: shkklt@shu.edu.cn;
wangyin2018@shu.edu.cn; junyu.xuan@uts.edu.au;
luoxf@shu.edu.cn;

Abstract

More and more large-scale pre-trained models show apparent advantages in solving the event detection (ED), i.e., a task to solve the problem of event classification by identifying trigger words. However, this kind of model heavily depends on labeled training data. Unfortunately, there is not enough labeled training data for some particular areas, such as finance, due to the high cost of the data annotation process. Besides, the manually labeled training data has many problems like uneven sampling distribution, poor diversity, and massive long-tail data. Recently, some researchers have used the generative model to label data. However, training the generative models needs rich domain knowledge, which cannot be obtained from a few shots. Therefore, we propose a Domain Aware Few Shots (DAFS) generative model that can generate domain based training data through a relatively small amount of labeled data. First, DAFS utilizes self-supervised information from various categories of sentences to calculate words' transition probability under different domain and retain key triggers in each sentence. Then, we apply our joint algorithm to generate labeled training data that considers both diversity and effectiveness. Experimental results demonstrate that the training data generated

by DAFS significantly improves the performance of ED in actual financial data. Especially when there are no more than 20 training data, DAFS can still ensure the generative quality to a certain extent. It also obtains new state-of-the-art results on ACE2005 multilingual corpora.

Keywords: event detection, domain-aware, joint algorithm, self-supervised

1 Introduction

Automatic event extraction is a fundamental task of information extraction. Generally speaking, event detection (ED) aims at identifying event triggers which is a key step of event extraction. For example, from the sentence "*It's been ten minutes since I got home, and George called*", systems should detect the event of "*Movement* : *Transport*" triggered by "*got home*", and the event of "*Contact* : *Phone Write*" triggered by "*called*".

Most of the ED methods before the year of 2018 applied a word-wise classification paradigm, which has achieved significant progress [1]. Afterwards, with the rise of new pre-trained model BERT [2], the method of representation learning can obtain semantic information in sentence more precisely, as it is known that word-wise ED models suffer from the trigger word ambiguity and semantic loss problems [1]. For instance, we can't directly detect the event of "bankrupt" in sentence "*Will the bankruptcy caused by the financial crisis affect Ali*?". Although it has the trigger word "*bankruptcy*", it does not mean anything happened in a real financial situation . The pre-trained model can learn the language of this interrogative state through fine-tune mechanism, but it needs more data of this type.

Futhermore, we summarize the similarities and differences between training data and test data in real data in Table 1. The first line in Table 1 can easily recognize because of similar trigger words in both training and test corpus. Additionally, the second line in Table 1 represents the data without similar trigger words but with special semantic between training and test corpus. As shown in the example, although they all have the trigger word "*fire*", the data in TD is a negative sentence pattern, so it does not belong to *Label* 8. And the

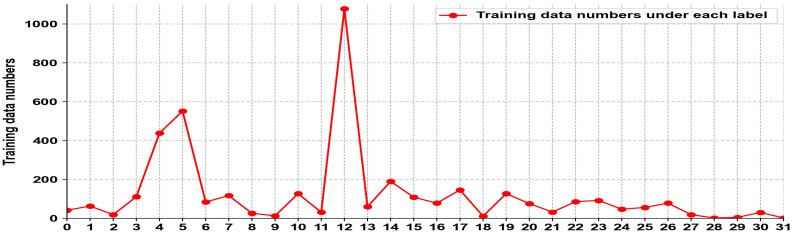
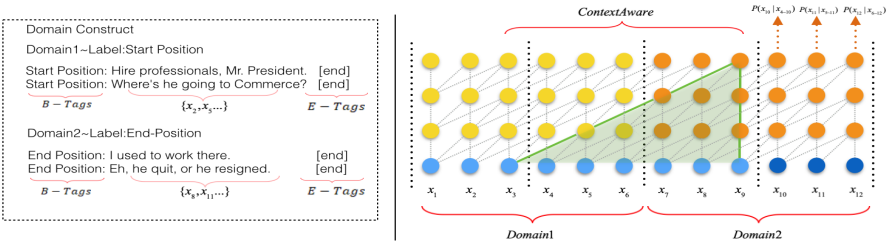


Fig. 1 Training data distribution of ACE2005 corpus under 33 categories. The number ranged from 2 to 1078 training data.

Table 1 Data difference between training and test corpus. 0 stands for Negative class. 8 stands for the label "Stop production due to accident"

Similarities and Differences	Example in Training Data (TRD)	Label	Example in Test Data (TD)	Label
1) Repetition triggers and similar semantic	Affected by the electric vehicle ***fire accident***,BYD's share price plunged yesterday.	8	A ***fire*** broke out in haipujia Technology Co., Ltd., resulting in shutdown	8
2) Repetition triggers but special semantic	The ***fire*** of Hai Pujia Technology Co., Ltd. led to the shutdown	8	Shangxi Co., Ltd. denied the ***fire*** incident	0
3) No repetition triggers but have similar semantic	***Fire*** at CICC gold subsidiary caused shutdown.	8	***Typhoon*** caused Dragon Group to stop production	8
4) No repetition triggers and no similar semantic	Suzhou Solid Technetium: serious ***fire***.	8	The ***explosion*** in Shi Zuishan coal mine has killed 19 people	8

third line in Table 1 represents the test data with no repetition triggers but with similar semantic to the training corpus. "*Typhon*" in TD is the triggered word that have never appeared in TRD. To improve parts 2 and 3, most pre-trained based methods for ED follow the supervised-learning paradigm, which requires lots of labeled data for training. However, annotating large amount of data accurately will cost a lot of labor. At this time, generation model becomes a way for people to do the research. VAE[3] and GAN [4] are committed to generating highly simulated data, but their training itself requires thousands of data to make loss converge. However, in the field of ED, the number of data for one class ranges from 2 to 1000 (As show in Fig.1). According to the statistics, there are 78.2% of trigger words in the benchmark ACE2005 that have a frequency of less than 5. Another generation methods focus on generating data by argument replacement and adjunct token rewriting [5]. But this method does little help to improve recall, because repeated semantics training data weakens the generalization ability of the classification model. Therefore, generating semantic diversity is also a factor we need to consider. In addition, the fourth part in Table 1 represent the data which most of models can hardly predict because neither similar trigger words nor similar semantic sentences appeared in training data. Embedding prior knowledge is a good method [6], but it still requires additional work of manual mapping and collection of new data sources. Specifically, to resolve these problems, this paper proposes a Domain Aware Few shots (DAFS) generative model which can generate diversity and effectiveness of labeled training data relying on few shots resource. Firstly, we do the domain construct to prepare for training data, and then we apply long-distance attention component (Transformer-XL) to fully train the context dependence of words among different domains. Secondly, we use a joint algorithm to generate data that can ensure diversity and effectiveness to the classification model, meanwhile we develop a simple data filter process



(a) Domain Construct (b) Generation Model

Fig. 2 Sample of Domain construction on the left and training process of DAFS model on the right. DAFS not only guarantees the learning of the potential relationship of key features in a domain, but also generates more abundant annotated corpus by combining the transfer probability of words outside the domain. $\{x_1, x_2, x_3\}$ stands for a sentence in a domain. x_1, x_3 stands for $B-Tags$, $E-Tags$ and x_2 stands for main content in a sentence. Sentence 1:$\{x_1, x_2, x_3\}$, Sentence 2: $\{x_4, x_5, x_6\}$ are in Domain1. Sentence 3:$\{x_7, x_8, x_9\}$, Sentence4: $\{x_{10}, x_{11}, x_{12}\}$ are in Domain2.

to remove duplication and guarantee sample balance by recognizing trigger words. Finally, we integrate DAFS and BERT into an active learning workflow to solve regarding one shot learning issues.

We evaluate our model on the ACE2005 benchmark and real financial corpus. Our method surpasses the baselines of ACE2005 and achieves a great performance in real financial corpus. Experiments show that our method is effective on multilingual corpora (English & Chinese) and alleviate the zero-shot, few-shot classification problems from a novel perspective. Our contributions can be summarized as:

1) We propose a novel Domain Aware Few Shots Generative Model which can learn from existing few shot labeled corpus to generate more annotation data.

2) We propose a domain-based joint algorithm in our DAFS to maintain the diversity and effectiveness of generated training data. And it is approved to be effective in experiments.

3) After integrating active learning mechanism, DAFS can systematically alleviate the one shot, few shot regarding issues in ED.

4) Experiments on benchmark ACE2005-Chinese (ACE2005-CH) show that our method improves the states of arts by 3.8 (4.6%), 9.3 (10.7%), 12.3 (14.7%) in Precision, Recall & F1-score respectively. On ACE2005-English (ACE2005-EN) corpus, our Recall increase by 6.7 (8.6%). Additionally, we get an increment of 7.0 (7.7%), 10.2 (11.4%), 9.5 (10.6%) on real financial data.

2 Related work

2.1 Event Detection

Traditional feature-based methods exploit both lexical and global features to detect events [7]. As neural networks become popular in NLP [8], data-driven methods use various superior DMCNN, DLRNN and PLMEE model [5, 9, 10]

for end-to-end event detection. FBMA [11] attends to different aspects of text while constructing its representation. Recently, weakly-supervised methods [12–14] have been proposed to generate more labeled data. Wang et al. [15] uses complementary information between domains to improve event detection. Ferguson [16] relies on sophisticated pre-defined rules to bootstrap from the paralleling news streams. Wang et al. [17] limits the data range of adversarial learning to trigger words appearing in labeled data. Cao et al.[18] propose an Incremental Heterogeneous Graph Neural Network for incremental social event detection. Zheng et al.[19] propose TaLeM: a novel taxonomy-aware learning model which can deal with the low-resources problem in ED.

2.2 Event Generation

As the neural network architecture encounters bottlenecks, more and more attention is paid to data-driven methods, and event generation is one of the main application areas. External resources such as Freebase, Frame-Net and WordNet are commonly employed to generate event and enrich the training data. Several previous event generation approaches [13, 20] are based on a strong assumption in distant supervision to label events in unsupervised corpus. In fact, co-occurring entities could have none expected relationship. In addition, Huang et al. [21] incorporates abstract meaning representation and distribution semantics to extract events. While Liu et al. [22] manages to mine additional events from the frames in FrameNet. Tong et al. [6] leverages external open-domain trigger knowledge to reduce the inherent bias of frequent triggers in annotations.

3 Methodology

In this section, we introduce DAFS to generate even and diverse data to improve ED. In general, our workflow mainly divides into three parts. Firstly, we introduce our process of domain-construction and architecture of the DAFS about how to use self-supervised information to train the generation model. Secondly, we illustrate our joint algorithm which can combine prior and domain transition probability to generate more diverse annotation data. Finally, we describe the whole workflow from data generation to data classification.

3.1 Domain Construct

Domain means all the data sets of an event. As we have 33 event types in the ACE-2005 corpus, we will automatically build 33 domains at initialization. Moreover, for our real financial data,we have 10 domains. Formally, we denote x_i as the word in each sentence then we have $S_i = \{x_1, x_2, \cdots, x_i\}$. Meanwhile, for each labeled sentence, we have a set $H_W = \{(S_i, Y_i)\}_{i=1}^{W}$. W stands for the number of sentences for whole training dataset and Y_i stands for the supervision label of the event. Then we assign data to different domains which its label belongs to, so that we can construct a domain-based corpus

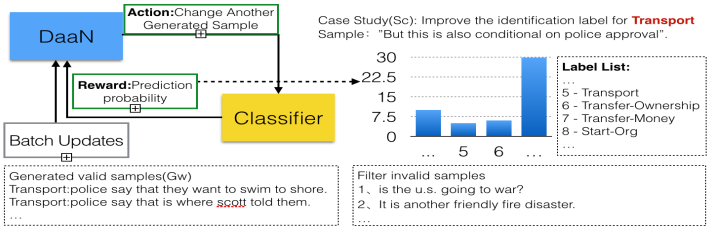


Fig. 3 The active learning workflow (AL) of our integration of DAFS and classifier to achieve incremental learning. For event "Transport", DAFS generate valid samples as well as invalid ones, AL pickup the right ones through its prediction probability, if the generated samples do positive effects to classifier then we collect it up to a certain amount and use it to evolve DAFS.

$H_D = \{(S_j, Y_j)\}_{j=1}^{D}$. D stands for the number of sentences for special domain. With the above supervision data, we can get the transition matrix of each word for specific domain $\mathbb{M}^D$ and the whole data $\mathbb{M}^W$ by calculating the word frequency. Based on the matrix $\mathbb{M}^D$, $\mathbb{M}^W$, we can get the transition probability of the top 10 tokens which are $E_d = [\mathbb{M}^D_{i,top1}, \mathbb{M}^D_{i,top2}, \cdots, \mathbb{M}^D_{i,top10}]$, $E_w = [\mathbb{M}^W_{i,top1}, \mathbb{M}^W_{i,top2}, \cdots, \mathbb{M}^W_{i,top10}]$. And i stands for the given word. Given a chain of words S_i, our goal is to jointly calculate the generation probability of the next word:

$$\max_{G} P\left(G \mid E_d, E_w, E_m\right) \tag{1}$$

E_m represents the transition matrix of each word according to the context. As in Fig.2(a), "Start-Position" and "End-Position" are two examples for domain building. And the preparation of $\mathbb{M}_D$ and $\mathbb{M}_W$ adjacency matrices are essential for the following chapters. E_m is obtained through the generation model in section 3.2.

3.2 Event Generation

In order to make the information flows across domains in either the forward and backward pass, we employ Transformer-XL [23] as our feature extractor. As in Transformer-XL, we define the length of each segment as L. Each segment contains several sentences, for the consecutive segments we have $S_t = [x_{t1}, \cdots, x_{tL}]$ and $St + 1 = [x_{tL+1}, \cdots, x_{t2L}]$ respectively. So the n-th hidden states of the t-th segment is expressed as $h_t^n \in \mathbb{R}^{L \times d}$, where d is the hidden dimension. In order to obtain a longer dependency, we combine two consecutive segments and get

$$\widetilde{\mathbf{h}}_{t+1}^{n-1} = \left[N_{BP}\left(\mathbf{h}_t^{n-1}\right) \circ \mathbf{h}_{t+1}^{n-1}\right] \tag{2}$$

Then applied with the self attention mechanism, we can have n-th layer hidden state as follows:

$$\mathbf{h}_{t+1}^{n} = TL\left(\mathbf{q}_{t+1}^{n}, \mathbf{k}_{t+1}^{n}, \mathbf{v}_{t+1}^{n}\right) \tag{3}$$

where N_{BP} represents the hidden state s_t no longer propagates backward and TL stands for transformer-layer. $\mathbf{q}_{t+1}^{n}, \mathbf{k}_{t+1}^{n}, \mathbf{v}_{t+1}^{n}$ represent the query, key,

value from the training sentences at time $t+1$. Furthermore, each domain contains several segments. As in Fig.2b, the hidden state of each position, except itself, depends on the token of first $(L-1)$ position in the next layer. So the length of dependency will increase L-1 with each layer going down. Therefore, the longest dependency length is $n(L-1)$, and n is the number of layers in the model. Context aware distance of dependency can be approximately $O(N \times L)$, so the number of sentences in each domain of training corpus should be more than $N \times L/N_a$, while N_a is the average length of each sentence. In particular, the characteristics of the initial and trigger words of each domain can be well learned, because they repeatedly appear in the domain as $S_l = \{B-Tags, x_{n1}, \cdots, x_{n_l}, E-Tags\}$, where "$B-tags$" and "$E-Tags$" are represented as the special domain label as visualized in Fig.2a. For completeness, we adopt Masked LM task [2] *Masked-Softmax* and relative positional encoding mechanism[23] *Positionwise-Feed-Forward* to exploit surrounding words to learn the specific semantics of each character and the expression of transfer probability from context-based attention features $\mathbf{A}_t^n$. Then we get final output $\mathbf{h}_t^n$ as:

$$\mathbf{a}_t^n = \mathit{Masked\text{-}Softmax}\ (\mathbf{A}_t^n)\,\mathbf{v}_t^n \tag{4}$$

$$\mathbf{o}_t^n = \mathit{LayerNorm}\ \left(\mathit{Linear}\,(\mathbf{a}_t^n) + \mathbf{h}_t^{n-1}\right) \tag{5}$$

$$\mathbf{h}_t^n = \mathit{Positionwise\text{-}Feed\text{-}Forward}\,(\mathbf{o}_t^n) \tag{6}$$

As a result, the effective context can be transferred in and out of the domain, which makes the generated labeled semantics more diverse.

3.3 Domain-Base Joint Algorithm

Although we employ domain-aware generation model for considering context information, when it comes to predict and generate new labeled data, we believe embedding prior knowledge is also one of the important factors. However, the extra annotation information will make our model appear to be meaningless in practice, because our original intention is to save the cost of human annotation. We turn to use the self-supervised information and take into account diversity and effectiveness to generate labeled data. $\mathbb{M}_D$ and $\mathbb{M}_W$ we mentioned in Section 3.1 are considered to be effective supervision information because they not only contain domain-specific knowledge, but also the possibility of global transition probability. Formally, for given the input S_i, generation model will generate next word x_{i+1} which considers context information. However, as different domains are adjacent to each other, part of the generated data may undergo domain transfer, that is, other types of generated data appear in the current domain. To alleviate this problem, $\mathbb{M}_D$ is extremely important, because once the probability of words in a particular domain increases, it is possible to maintain the key features of the domain. Meanwhile, $\mathbb{M}_W$ gives us the possibility of more words appearing in the generated sentence, because there will be more choices for the next word in the

global probability. All in all, in order to ensure the effectiveness and diversity of the generated data, the global information (that is, prior knowledge), the transfer information in the domain, and the context information must be considered comprehensively. Formally, a joint probability can be described as:

$$J(\theta) = \alpha E_m + \lambda E_d + (1 - \lambda) E_w \tag{7}$$

For E_m, E_d, E_w, we have illustrated in Formula (1). As E_m, E_d has been calculated before training, E_m is trained and generated by generative model according to self-supervised information. Therefore, our lightweight generation model will not encounter the problem of loss convergence. α is the only hyper parameter in this formula to adjust the smoothness of generated words. To alleviate long tail issues, λ is used to increase the transfer weight of the probability of words in small sample events and it's inversely to the proportion of domain in the total sentence.

$$\lambda = \frac{e^{\phi_{\mathbf{d}}}}{\sum_{k=1}^{D} e^{\phi_{\mathbf{d}}}} \tag{8}$$

N_{domain} stands for labeled training data in specific domain, while N_{total} stands for total number of sentences.

$$\phi_d = \sqrt{\frac{N_{domain}}{N_{total}}} \tag{9}$$

With λ, the weight of key word for few shot data is increased which alleviate domain shift caused by long tail issue. In the meantime, the variety of domain will be more abundant due to the introduction of E_w.

3.4 Event Detection

BERT has achieved SOTA performance on a wide range of tasks and has been proved very effective on ED secnario [17]. We apply BERT as our classifier. It could obtain semantics level information which overcomes the mismatch problem between words and event triggers [1]. Following to the mechanism of BERT fine-tuning in dealing with classification tasks, our event type classifier directly uses the sub-types of the event, which ignores the hierarchical relationship of event types and the direct impact of event trigger words on event detection.

Formally, given the token features of the input S, firstly we get the hidden representation H for each sentence through BERT, then a fully connected layer and softmax will be applied to calculate the score assigned to each event sub-type:

$$H = BERT(S) \tag{10}$$

$$c = HW_f + b_f \tag{11}$$

$$P(y \mid x) = softmax(c) \tag{12}$$

3.5 Active learning workflow

The most difficult part of test data to predict are the pentagram ones in Fig.1. For this part of data, there are two main difficulties. Firstly, as there are no obvious trigger words or semantics supervision information in corresponding training domain, it's hard to fit the distribution in test data. Secondly, when adding new trigger words that are similar to existing ones, it's hard for deep model to perfectly learn it and overcome the catastrophic forgetting issues in the incremental learning process. To alleviate these problems, we apply active learning mechanism to directly evaluate correct and wrong labels of the generated data . As in Fig.3, given the wrong sentences list $S_c = \{S_w, Y_w\}\,|_1^w$, our DAFS will continue to generate data $G_w = \{G_x, G_y\}$ until our classifier achieves the highest scores when predict $[S_w, Y_w]$. Formally, For DAFS:

$$g = \begin{cases} r = 1 \;\; add\ to\ G^+ \\ r = 0 \;\; turn\ to\ G_{x+1} \end{cases} \tag{13}$$

For Classifier:

$$r = \begin{cases} 1 \;\; P(Y_w) > Tep\ \ when\ predict\ S_w \\ 0 \quad otherwise \end{cases} \tag{14}$$

where S_y stands for score from classifier and Tep is the critical point of our probability value in the multi-label classification task. G^+ stands for new collection from valid generated data. G_{x+1} stands for next generated data. Finally, when we collect and generate a certain amount of data-N_g to G^+, we will train our classifiers in batches. Typically, we set N_g to one-tenth of the total data-W.

On the other hand, in the learning of new trigger words, DAFS's domain adjacency matrix solves the catastrophic forgetting problem of incremental learning very well. Suppose we have a domain dictionary with d dimension and we have probability transition matrix $\mathbb{M}_D^{d\times d}$, we face two situations: The first one is that the domain dictionary matrix contains new trigger word while the other does not. As shown in the Fig.4, we can get the maximum in-degree and out-degree probability and their corresponding token of the original word. We define them as $\mathbb{V}_{in}^{1\times 10}$ and $\mathbb{V}_{out}^{1\times 10}$. If the new trigger word is similar to the original one, we just need to modify the 20 relative positions in transition matrix $\mathbb{M}_D^{d\times d}$. In addition, if the new trigger word is out of dictionary of $\mathbb{M}_D^{d\times d}$, we have to update the original matrix to $\mathbb{M}_D^{(d+1)\times(d+1)}$ and do same thing as above.

Through the above two methods, we can generate a large number of sentences containing new trigger words, thereby improving the classifier's ability to fit zero shot and one shot samples.

Table 2 Ant Financial Event Detection(AFED) corpus

Event Type	Train	Dev	Test
Cooperation	793	74	463
Business/asset arrangement	820	93	580
Provide false certification	676	48	10
Actual controller breaks law	395	38	42
Actual controller arbitration	321	15	100
Guarantee liability	160	31	35
Bankruptcy liquidation	349	44	170
Stop production	516	43	198
Serious safety accident	911	102	200
Other	4087	406	3098

Table 3 The influence of special semantics on ED. ACE2005 is not sensitive to the above special semantics, but in real scenes, these semantics are more important to trigger events.

DataSet	Adversative	Negation	Interrogative	Hypothesis	Uncertainty
ACE(EN, CH)	✖	✖	✖	✖	✖
AFED	✔	✔	✔	✔	✔

4 Experiments

4.1 Experiment Settings

Datasets

We conducted experiments on three corpora: ACE2005-EN corpus, ACE2005-CH corpus and our real financial corpus, Ant Financial Event Detection(AFED). For ACE2005-CH corpus, we use the same setup as [24], [25] and [26], in which 521/64/64 documents are used as training/development/test set. Due to the different definitions of trigger events, we build AFED to show the robustness of our model in dealing with different data. AFED corpus has more complex evaluation criteria, which embodies in the following three aspects:1) Trigger words are not the only criteria for triggering an event. For instance, if the special case semantics in Table 3 occurs around the trigger word, it may mean that the event is not triggered. 2) In addition to the trigger words, there are many implicit features in the sentence. Only when the key features and trigger words appear at the same time can the event be truly triggered. For example, "actual controller breaks law", only when "controlling shareholder" and "actual controller" appear in the event "violation of the law" can the event be regarded as triggered. 3) The "Other" class is very complex, and there will be interference items with similar semantics. For example, the negative sample of bankruptcy liquidation - "CIMC Group intends to purchase the bankrupt company". This belongs to "Other" category, because "bankrupt" is not to describe the subject. Data distribution for AFED can be seen in Table 2. All AFED data are obtained from real-time news and will be released on GitHub in the future.

Comparison Methods

In order to demonstrate the robustness of our approach on Multilingual and real data sets, We applied different optimal models to Chinese and English corpora:

ACE2005 Chinese . We include classic papers such as Convolutional Bi-LSTM model (C-BiLSTM) proposed by [26], Forward-backward Recurrent Neural Networks (FBRNN) by [27], word-based DMCNN and Hybrid Neural Network proposed by [25], which incorporates CNN with Bi-LSTM and achieves the SOTA NN based result on ACE2005. Rich-C [28] developed several handcraft Chinese-specific features, which improve the effect of Chinese ED. In addition, we adopt NPNs [1] which can solve the word-trigger mismatch problem by directly proposing entire trigger nuggets centered at each character. Hybrid Character Representation(HCR) for ED [29] employs BERT-base model as its trigger classifier and achieve a relatively good score. It is the state of arts on ACE2005-CH corpus.

ACE2005 English. We compare our methods with other six state-of-the-art data enhancement models, including: GCN-ED deeply excavates the structural information from labeled data with dependency syntax tree and uses GCN for classification [10]. Lu's DISTILL proposes a learning approach which applied effective separation, incremental learning, and finally adaptive synthesis of different event feature representation [30]. TS-DISTILL exploits the entity ground-truth and uses an adversarial imitation based knowledge distillation approach for ED [31]. AD-DMBERT adopts a confrontational simulation model to continuously train the discriminator's resistance to noise [17]. DRMM employs an alternating dual attention to select informative features for mutual enhancements to ED [32]. EKD leverages external open-domain trigger knowledge to reduce the inherent bias of frequent triggers in annotations [6] The last three baselines both use BERT as the feature extractor.

AFED. In order to reflect the effective of our model DAFS, we only use the original BERT [2] model which is the best classifier in the real data set for comparison.

4.2 Overall Performance

Table 4, 5, 6 show the results on ACE2005-CH & EN and AFED respectively. From the results, we can draw the following observations:

1) DAFS achieve significant improvement of the precision, recall and F1-score by 3.8, 9.3, 12.3 on ACE2005-CH and 7, 12, 9.5 on AFED respectively. This is mainly benefiting from the effective data enhancement and the large scale pre-training information of BERT. Our method expand the training data to further enhance BERT, which achieve better performance and demonstrate the effectiveness of our model. HCR also uses BERT as its feature extractor. It uses word vector splicing. Experiments show that compared with the whole

Table 4 Results on ACE2005-CH Corpus for Event Detection

Method	Precision	Recall	F1-Score
FBPNN(Char)	57.5	42.8	49.1
DMCNN(Char)	57.1	58.5	57.8
C-BiLSTM	60	60.9	60.4
FBRNN(Word)	59.9	59.6	59.7
DMCNN(Word)	61.6	58.8	60.2
HNN*	77.1	53.1	63.0
Rich-C*	58.9	68.1	63.2
NPN(Task-specific)	60.9	69.3	64.8
HCR	66.6	77.0	71.2
BERT	78.1	80.5	79.2
DAFS+BERT	**80.9**	**86.3**	**83.5**

sentence vector produced by original BERT Finetune, it will cause a loss in precision.

2) For English Corpus as shown in Table 5, BERT contributes 4.9 of recall enhancement compared with none-BERT-base model TS-DISTILL. Since we expect to generate more realistic words, we retain tense, plural and other forms in the process of word segmentation, making our English vocabulary up to 10355. In the meantime, the vocabulary of ACE2005-CH is only 3305. This brings some difficulties to the generation of sparse features, but our data enhancement based on DAFS still keeps the growth of 4, 6.6 and 4 compared with original BERT. DAFS+BERT improves state of arts by 6.7 in recall. EKD introduces data from the outside, which improves precision a lot, indicating that it introduces a lot of additional constraints. Due to the increase of positive samples from DAFS, Recall is greatly improved. However, due to the similar combination from internal dictionary, the boundary of each event is not obvious, and the improvement of precision is limited.

3) As analyzed in Section 4.1, AFED has complex interference and class boundary complexity. As shown in Table 8, Experiments show that DAFS contributed a lot of effective data to original corpus, making significant improvement of the Precision,Recall and F1-score by 7, 10.2, 9.5 respectively. This proves that our model is also effective in generating corpus with fuzzy boundaries, negative and questionable semantics problems in actual scene.

4) Fig.5 show that, our model has obvious improvement in alleviating the long tail problem. The F1-Scores of Chinese and English training data between 10 and 30 were improved by 0.2 and 0.16 respectively. In addition to the number of 0-10, the amount of other train data phases have increased by about 0.05-0.1 due to its original high score. It's worth noting that we increased "0-10" phase from 0, 0 to 0.1, 0.2 for ACE-EN & ACE-CH respectively.

4.3 Domain-Base Joint Algorithm of Generative Model

In order to prove the effectiveness of our joint algorithm, we do the following ablation experiments. Firstly, we define zero shot as test data has no trigger

Table 5 Results on ACE2005 English Corpus for Event Detection

Method	Precision	Recall	F1-Score
GCN-ED	77.9	68.8	73.1
Lu's DISTILL	76.3	71.9	74.0
TS-DISTILL	76.8	72.9	74.8
AD-DMBERT	77.9	72.5	75.1
DRMM	77.9	74.8	76.3
EKD	**79.1**	78.0	78.6
BERT	70.1	77.4	74.5
DAFS+BERT	74.1	**84.1**	**78.8**

Table 6 Results on AFED for Event Detection

Method	Precision	Recall	F1-Score
BERT	83.4	79.4	80.4
DAFS+BERT	**90.4**	**89.6**	**89.9**

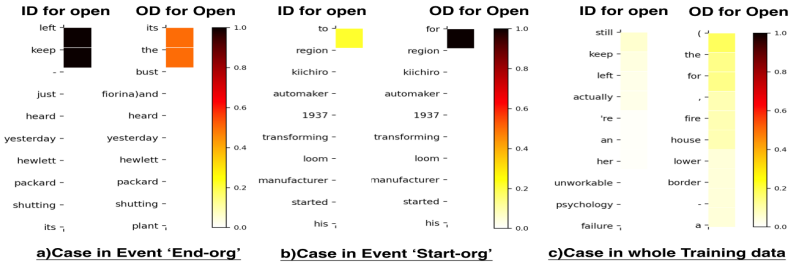


Fig. 4 The Global and specific domain transition probability. Example of the transition probability for the word "open" in Event "Start-org", "End-Org" and train data. OD is short for Out of Degree and ID is short for In Degree.

word appearing in training data for classifier. Secondly, we define "Few Shot" as the number of data in the training corpus does not exceed 50. In the meantime, as shown in Table 7, "Normal" means the number of training data for generative model is around 200. We choose "Meet" event as our "Normal" case, it has data of 190 in training data. To be fair, we choose "End-Org" event as our "zero shot" and "few shot" case. It has 31 records of training data. "Dismantling", "dissolved", "crumbled" are the trigger words that appears in the test set but not in the training set. Experiments in Table 7 show that DTP matrix is helpful to maintain the stability of data generation, especially in the case of zero shot situation. It improves 0 to 5 of when take into consideration of DTP. Meanwhile, GTP increases the diversity of generated text. But the out degree of probability in GTP used to be very small (around 8% for ID in Fig.4c) and the probability of DTP is usually large (Figure 4b &c). So we introduce λ to adjust its weight and calculate the joint probability which can achieve the best relative effect as visualized in Table 7.

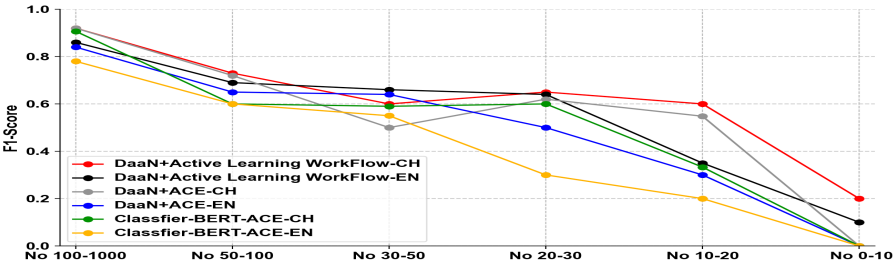


Fig. 5 The average F1-Score for different amount of training data on ACE multilingual dataset. X-axis represents the range of training data.

Table 7 Data generation results on different training set scales. DTP is short for Domain transfer probability. GTP is short for global transfer probability. M represents DTP+GTP, N represents DTP+GTP+λ.

Method	Zero Shot	Few Shot	Normal
DAFS	0	11	45
DAFS+GTP	0	10	43
DAFS+DTP	5	7	35
DAFS+M	7	8	37
DAFS+N	**9**	**17**	**59**

Table 8 DAFS-W represents the result of introducing incremental learning

Method	ACE-CH	ACE-EN	AFED
DAFS	83.5	78.5	89.9
DAFS-W	85.4	80.6	91.4

4.4 Incremental Learning

As shown in Table 8, when incremental learning workflow is applied in our model, the improvement for F1-score on ACE2005-CH, ACE2005-EN, AFED for F1-score is 1.9, 2.1, and 1.5 respectively. The workflow based on active learning technology can choose better generated data to support the incremental evolution of classification model. In real production environment, we often need models which have the ability to learn the relevant features through a sample quickly, and our joint algorithm based on domain transfer possibility could quickly generate data to fit new samples from the perspective of training data to realize incremental learning.

5 Conclusions and Future Work

By utilizing the potential supervisory information in the limited corpus, DAFS and the proposed domain-based algorithm generate more diverse and effective training data sets to solve the zero shot and the few shot problems, thus significantly improving the robustness and accuracy of the classification model. Based on the framework of DaaN and the active learning mechanism, our workflow effectively solve the problems related to one shot learning. Experiments

demonstrate that our method surpasses other 15 strong baselines through multilingual data sets. Our method is based on the comprehensive calculation of context probability, global transition probability and domain transition probability. We are going to try the above methods in knowledge inference, QA and other tasks in the future.

References

[1] Lin, H., Lu, Y., Han, X., Sun, L.: Nugget proposal networks for chinese event detection. arXiv preprint arXiv:1805.00249 (2018)

[2] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

[3] Shao, H., Yao, S., Sun, D., Zhang, A., Liu, S., Liu, D., Wang, J., Abdelzaher, T.: Controlvae: Controllable variational autoencoder. Proceedings of the 37th International Conference on Machine Learning (ICML) (2020)

[4] Liu, Z., Wang, J., Liang, Z.: Catgan: Category-aware generative adversarial networks with hierarchical evolutionary learning for category text generation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 8425–8432 (2020)

[5] Yang, S., Feng, D., Qiao, L., Kan, Z., Li, D.: Exploring pre-trained language models for event extraction and generation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5284–5294 (2019)

[6] Tong, M., Xu, B., Wang, S., Cao, Y., Hou, L., Li, J., Xie, J.: Improving event detection via open-domain trigger knowledge. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 5887–5897. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.acl-main.522. https://www.aclweb.org/anthology/2020.acl-main.522

[7] Li, Q., Ji, H., Huang, L.: Joint event extraction via structured prediction with global features. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 73–82 (2013)

[8] Cao, Y., Hou, L., Li, J., Liu, Z.: Neural collective entity linking. arXiv preprint arXiv:1811.08603 (2018)

[9] Duan, S., He, R., Zhao, W.: Exploiting document level information to improve event detection via recurrent neural networks. In: Proceedings of the Eighth International Joint Conference on Natural Language

Processing (Volume 1: Long Papers), pp. 352–361 (2017)

[10] Nguyen, T.H., Grishman, R.: Graph convolutional networks with argument-aware pooling for event detection. In: AAAI, vol. 18, pp. 5900–5907 (2018)

[11] Mehta, S., Islam, M.R., Rangwala, H., Ramakrishnan, N.: Event detection using hierarchical multi-aspect attention. In: The World Wide Web Conference, pp. 3079–3085 (2019)

[12] Huang, M., You, Y., Chen, Z., Qian, Y., Yu, K.: Knowledge distillation for sequence model. In: Interspeech, pp. 3703–3707 (2018)

[13] Zeng, Y., Feng, Y., Ma, R., Wang, Z., Yan, R., Shi, C., Zhao, D.: Scale up event extraction learning via automatic training data generation. arXiv preprint arXiv:1712.03665 (2017)

[14] Yang, H., Chen, Y., Liu, K., Xiao, Y., Zhao, J.: Dcfee: A document-level chinese financial event extraction system based on automatically labeled training data. In: Proceedings of ACL 2018, System Demonstrations, pp. 50–55 (2018)

[15] Wang, J., Zhao, L.: Multi-instance domain adaptation for vaccine adverse event detection. In: Proceedings of the 2018 World Wide Web Conference, pp. 97–106 (2018)

[16] Ferguson, J., Lockard, C., Weld, D.S., Hajishirzi, H.: Semi-supervised event extraction with paraphrase clusters. arXiv preprint arXiv:1808.08622 (2018)

[17] Wang, X., Han, X., Liu, Z., Sun, M., Li, P.: 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, Volume 1 (Long and Short Papers), pp. 998–1008 (2019)

[18] Cao, Y., Peng, H., Wu, J., Dou, Y., Li, J., Yu, P.S.: Knowledge-preserving incremental social event detection via heterogeneous gnns. In: Leskovec, J., Grobelnik, M., Najork, M., Tang, J., Zia, L. (eds.) WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021, pp. 3383–3395. ACM/IW3C2, year=2021, url = https://doi.org/10.1145/3442381.3449834, doi = 10.1145/3442381.3449834, timestamp = Sun, 25 Jul 2021 11:46:32 +0200, biburl = https://dblp.org/rec/conf/www/CaoPWDLY21.bib, bibsource = dblp computer science bibliography, https://dblp.org

[19] Zheng, J., Cai, F., Chen, W., Lei, W., Chen, H.: Taxonomy-aware learning for few-shot event detection. In: Leskovec, J., Grobelnik, M., Najork, M., Tang, J., Zia, L. (eds.) WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021, pp. 3546–3557. ACM/IW3C2, year = 2021, url = https://doi.org/10.1145/3442381.3449949, doi = 10.1145/3442381.3449949, timestamp = Mon, 07 Jun 2021 14:20:06 +0200, biburl = https://dblp.org/rec/conf/www/ZhengCCLC21.bib, bibsource = dblp computer science bibliography, https://dblp.org

[20] Chen, Y., Liu, S., Zhang, X., Liu, K., Zhao, J.: Automatically labeled data generation for large scale event extraction. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 409–419 (2017)

[21] Huang, L., Cassidy, T., Feng, X., Ji, H., Voss, C., Han, J., Sil, A.: Liberal event extraction and event schema induction. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 258–268 (2016)

[22] Liu, S., Chen, Y., Liu, K., Zhao, J.: Exploiting argument information to improve event detection via supervised attention mechanisms. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1789–1798 (2017)

[23] Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., Salakhutdinov, R.: Transformer-xl: Attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860 (2019)

[24] Chen, Z., Ji, H.: Language specific issue and feature exploration in chinese event extraction. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, pp. 209–212 (2009)

[25] Feng, X., Qin, B., Liu, T.: A language-independent neural network for event detection. Science China Information Sciences **61**(9), 092106 (2018)

[26] Zeng, Y., Yang, H., Feng, Y., Wang, Z., Zhao, D.: A convolution bilstm neural network model for chinese event extraction. In: Natural Language Understanding and Intelligent Applications, pp. 275–287. Springer

[27] Ghaeini, R., Fern, X.Z., Huang, L., Tadepalli, P.: Event nugget detection with forward-backward recurrent neural networks. arXiv preprint arXiv:1802.05672 (2018)

[28] Chen, C., Ng, V.: Joint modeling for chinese event extraction with rich

linguistic features. In: Coling (2012)

[29] Xiangyu, X., Tong, Z., Wei, Y., Jinglei, Z., Rui, X., Shikun, Z.: A hybrid character representation for chinese event detection. In: 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2019)

[30] Lu, Y., Lin, H., Han, X., Sun, L.: Distilling discrimination and generalization knowledge for event detection via delta-representation learning. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4366–4376 (2019)

[31] Liu, J., Chen, Y., Liu, K.: Exploiting the ground-truth: An adversarial imitation based knowledge distillation approach for event detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6754–6761 (2019)

[32] Tong, M., Wang, S., Cao, Y., Xu, B., Li, J., Hou, L., Chua, T.-S.: Image enhanced event detection in news articles. In: AAAI, pp. 9040–9047 (2020)