



DAFS: a domain aware few shot generative model for event detection

Nan Xia¹ · Hang Yu^{1,2} · Yin Wang¹ · Junyu Xuan¹ · Xiangfeng Luo¹

Received: 1 November 2021 / Revised: 23 April 2022 / Accepted: 26 May 2022 /

Published online: 4 September 2022

© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2022

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 depends heavily on labeled training data. Unfortunately, there is not enough such 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-Shot resource. Therefore, we propose a Domain-Aware Few-Shot (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

Editors: Bo Han, Tongliang Liu, Quanming Yao, Mingming Gong, Gang Niu, Ivor W. Tsang, Masashi Sugiyama.

✉ Hang Yu
yuhang@shu.edu.cn

Extended author information available on the last page of the article

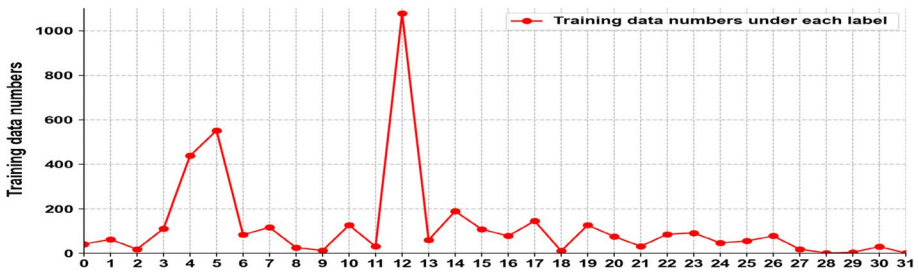


Fig. 1 Training data distribution of ACE2005 corpus under 33 categories. The number ranged from 2 to 1078 training data

“*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 Writee*” triggered by “*called*”.

Most of the ED methods before 2018 applied a word-wise classification paradigm, which has achieved significant progress (Lin et al., 2018). Afterward, with the rise of the new pre-trained model BERT (Devlin et al., 2018), the method of representation learning can obtain semantic information in a sentence more precisely, as it is known that word-wise ED models suffer from the trigger word ambiguity and semantic loss problems (Lin et al., 2018). For instance, we can’t directly detect the event of “*bankrupt*” in the 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 a fine-tuning mechanism, but it needs more data of this type.

Furthermore, we summarize the similarities and differences between training data and test data in real data in Table 1. As shown the first line in Table 1, the classifier can easily recognize the event type because of similar trigger words in both training and test corpus. In addition, 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*. The third line in Table 1 represents the test data with no repetition triggers but with similar semantics to the training corpus. “*Typhon*” in TD is the triggered word that has 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 incur high labor costs. At this time, the generation model becomes a suitable research method. VAE(Shao et al., 2020) and GAN (Liu et al., 2020) are committed to generating highly simulated data, but their training itself requires thousands of data to make losses converge. However, in the field of ED, the number of data for one class ranges from 2 to 1000 (As shown 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 (Yang et al., 2019). 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

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 <i>fire accident</i> , BYD's share price plunged yesterday.	8	A <i>fire</i> broke out in Haijia Technology Co., Ltd., resulting in shutdown	8
2) Repetition triggers but special semantic	The <i>fire</i> of Hai Pujia Technology Co., Ltd. led to the shutdown	8	Shangxi Co., Ltd. denied the <i>fire</i> incident	0
3) No repetition triggers but have similar semantic	<i>Fire</i> at CICC gold subsidiary caused shutdown.	8	<i>Typhoon</i> caused Dragon Group to stop production	8
4) No repetition triggers and no similar semantic	Suzhou Solid Technetium: serious <i>fire</i> .	8	The <i>explosion</i> in Shi Zuishan coal mine has killed 19 people	8

the data that most models can hardly predict because neither similar trigger words nor similar semantic sentences appeared in training data. Embedding prior knowledge is a good method (Tong et al., 2020), 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 shot (DAFS) generative model which can generate diverse and effective labeled data using the few-shot resource. Firstly, we construct the domain to prepare for training data, and then we apply the 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 in the classification model. Meanwhile, we develop a simple data filter process 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 high performance in a real financial corpus. Experiments show that our method is effective on multilingual corpora (English & Chinese) and alleviates 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-Shot 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 the 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 increases 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 (Li et al., 2013). As neural networks become popular in NLP (Cao et al., 2018), data-driven methods use various superior DMCNN, DLRNN and PLMEE models (Duan et al., 2017; Nguyen & Grishman, 2018; Yang et al., 2019) for end-to-end event detection. FBMA (Mehta et al., 2019) attends to different aspects of text while constructing its representation. Recently, weakly-supervised methods (Huang et al., 2018; Zeng et al., 2017; Yang et al., 2018) have been proposed to generate more labeled data. Wang et al. (Wang et al., 2018) uses complementary information between domains to improve event detection. (Ferguson et al., 2018) relies on sophisticated pre-defined rules to bootstrap from the parallel-ing news streams. (Wang et al., 2019) limit the data range of adversarial learning to trigger words appearing in labeled data. (Cao et al., 2021) propose an Incremental Heterogeneous

Graph Neural Network for incremental social event detection. (Zheng et al., 2021) propose TaLeM: a novel taxonomy-aware learning model which can deal with the low-resources problem in ED. (Wang et al., 2020) propose a survey on Few-Shot Learning.

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 events and enrich the training data. Several previous event generation approaches (Chen et al., 2017; Zeng et al., 2017) are based on a strong assumption in distant supervision to label events in an unsupervised corpus. In fact, co-occurring entities could have none expected relationship. In addition, (Huang et al., 2016) incorporates abstract meaning representation and distribution semantics to extract events. While (Liu et al., 2017) manages to mine additional events from the frames in FrameNet. (Tong et al., 2020) leverages external open-domain trigger knowledge to reduce the inherent bias of frequent triggers in annotations. (Han et al., 2018) propose structure-aware probabilistic model incorporating a structure prior by mask mechanism which inspire us to use self-supervised information to support a Few-Shot generative model.

2.3 Pre-trained model

The Pre-trained model greatly improves the semantic generalization ability of classification model through transfer learning. (McCann et al., 2017) exploits language model pre-trained on supervised translation corpus in the target task. ELMO (Peters et al., 2019) gets context sensitive embeddings by encoding characters with stacked bidirectional LSTM (Hochreiter & Schmidhuber, 1997) and residual structure (He et al., 2016). GPT (Radford et al., 2018) improves the state of the art in 9 of 12 tasks. BERT (Devlin et al., 2018) breaks records of 11 NLP tasks and receives a lot of attention. GPT-2 (Radford et al., 2019) is on the basis of GPT, focusing on solving Zero-Shot problem expanding by the training corpus. XLNet (Yang et al., 2019) applies the transformer-xl mechanism and outperforms BERT on 20 tasks.

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 that is 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

Definition 1 - Domain Domain is a semantic block that contains the event type, the event trigger and related event semantic sentences, which is integrated through preprocessing. Event

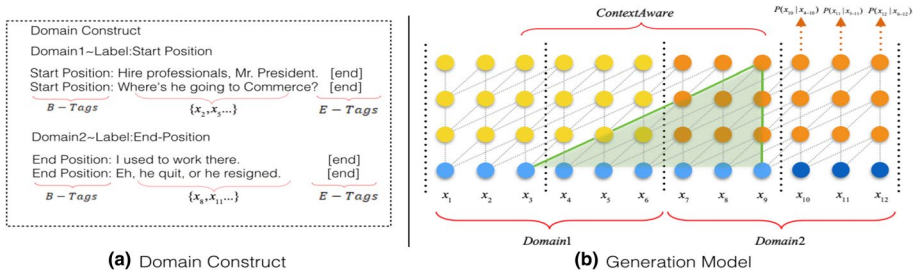


Fig. 2 Sample of Domain construction on the left and training process of DAFS model on the right. DAFS not only guarantees the learning of the potential relationship of key features in a domain, but also generates more abundant annotated corpus by combining the transfer probability of words outside the 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

definition focuses on the label and trigger-words, but domain is the concept of integrating all event-related information in a region. Formally, we can define as

$$\{x, y, z\} \in Z_{dn}, \{x, y\} \in Z_{en}, Z_{en} \neq \emptyset, Z_{dn} \neq \emptyset, Z_{en} \subset Z_w \tag{1}$$

where x, y, z denote event type, event trigger and domain context semantics. Z_{dn}, Z_{en}, Z_w stand for domain set, event set, corpus set respectively.

Definition 2 - Domain-aware Domain aware means that word considers the relevant information of the current domain in the process of transferring to the next word. Specifically, it can be divided into two phases: 1) During the training phrase, through the preprocessing of domain construct, the relationship between domain label and domain data can be strengthened, and the diversity and domain correlation can be balanced through the algorithm. 2) During the testing phase, by outputting labels, we can automatically generate data in related domains.

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 $S_i = \{x_1, x_2, \dots, x_i\}$. 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 the whole training dataset and Y_i stands for the supervision label of the event. Then we assign data to different domains to which its label belongs, so that we can construct a domain-based corpus $H_D = \{(S_j, Y_j)\}_{j=1}^D$. D stands for the number of sentences for a special domain. With the above supervision data, we can get the transition matrix of each word for specific domain M^D and the whole data M^W by calculating the word frequency. Based on the matrix M^D, M^W , we can get the transition probability of the top 10 tokens which are $E_d = [M^D_{i,top1}, M^D_{i,top2}, \dots, M^D_{i,top10}]$, $E_w = [M^W_{i,top1}, M^W_{i,top2}, \dots, M^W_{i,top10}]$. And i stands for the given word. Given a chain of words S_i , our goal is to jointly calculate the generation probability G of the next word:

$$\max_G P(G | E_d, E_w, E_m) \tag{2}$$

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 of domain building. And the

preparation of \mathbb{M}_D and \mathbb{M}_W adjacency matrices is essential for the following chapters. E_m is obtained through the generation model in Sect. 3.2. E_w, E_d, E_m is embedded in the data and we can train and use it without additional labels. This is the self-supervised information mainly used in the DAFS.

3.2 Event generation

In order to make the information flow across domains in either the forward or backward pass, we employ Transformer-XL (Dai et al., 2019) 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 $S_t = [x_{t1}, \dots, x_{tL}]$ and $S_{t+1} = [x_{(t+1)L+1}, \dots, x_{(t+1)2L}]$ respectively. So the n -th hidden states of the t -th segment is expressed as $\mathbf{h}_t^n \in \mathbb{R}^{L \times d}$, where d is the hidden dimension. To obtain a longer dependency, we combine two consecutive segments and get

$$\tilde{\mathbf{h}}_{t+1}^{n-1} = [N_{BP}(\mathbf{h}_t^{n-1}) \circ \mathbf{h}_{t+1}^{n-1}] \quad (3)$$

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

$$\mathbf{h}_{t+1}^{n-1} \mathbf{W}_q^T, \tilde{\mathbf{h}}_{t+1}^{n-1} \mathbf{W}_k^T, \mathbf{h}_{t+1}^{n-1} \mathbf{W}_v^T = \mathbf{q}_{t+1}^n, \mathbf{k}_{t+1}^n, \mathbf{v}_{t+1}^n \quad (4)$$

$$\mathbf{h}_{t+1}^n = TL(\mathbf{q}_{t+1}^n, \mathbf{k}_{t+1}^n, \mathbf{v}_{t+1}^n) \quad (5)$$

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 $t+1$. $\tilde{\mathbf{h}}_{t+1}^{n-1}$ stands for the extended context and W_* denotes model parameters. 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 the 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 appear repeatedly in the domain as $S_t = \{B - Tags, x_{n1}, \dots, x_{n_t}, E - Tags\}$, 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 (Devlin et al., 2018) *Masked-Softmax* and relative positional encoding mechanism (Dai et al., 2019) *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 the final output \mathbf{h}_t^n as:

$$\mathbf{a}_t^n = \text{Masked-Softmax}(\mathbf{A}_t^n) \mathbf{v}_t^n \quad (6)$$

$$\mathbf{o}_t^n = \text{LayerNorm}(\text{Linear}(\mathbf{a}_t^n) + \mathbf{h}_t^{n-1}) \quad (7)$$

$$\mathbf{h}_t^n = \text{Positionwise-Feed-Forward}(\mathbf{o}_t^n) \quad (8)$$

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-based joint algorithm

Although we employ a domain-aware generation model for considering context information, when it comes to predicting and generating new labeled data, we believe embedding prior knowledge is also an important factor. 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 (From which we get E_m , as described it in Sect. 3.1.) and take into account of diversity and effectiveness to generate labeled data. \mathbb{M}_D and \mathbb{M}_W mentioned in Sect. 3.1 are considered to be effective supervision information because they contain not only domain-specific knowledge, but also the possibility of global transition probability. Formally, for given the input S_i , the generation model will generate the next word x_{i+1} which considers context information. However, as different domains are adjacent to each other, part of the generated data might 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, 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:

$$J(\theta) = \alpha E_m + (1 - \lambda)E_d + \lambda E_w \quad (9)$$

For E_m , E_d , E_w , we have illustrated in Formula (1). E_w , E_d has been calculated before training and we get E_m through generative model. 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. E_m uses a masking mechanism to make the predicted sentences more like the original distribution, but the sentences generated by E_m alone can generate a lot of repeated sentences like data sources. At this time, the larger α makes the data more like source data, while the smaller α makes the word pay more attention to domain information E_d and global information E_w in the process of transfer to the next word. It should be noted that the E_w information contains more possibilities for each word, because it is the global transfer matrix, which is the source of making generated sentence diversity. The higher weight of E_w , the more choices for next word. To alleviate long tail issues, weight parameter λ is used to increase the transfer weight of the probability of words in small sample events and it's inversely related to the proportion of domain in the total sentence.

$$\lambda = \frac{e^{\Phi_d}}{\sum_{k=1}^D e^{\Phi_k}} \quad (10)$$

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

$$\phi = \sqrt{\frac{N_{domain}}{N_{total}}} \quad (11)$$

Therefore, we can see from formulas (8) and (9) that the smaller the proportion between the number of sentences in a domain and the number of all sentences, the larger weight of the domain transition probability E_d and the smaller weight of the global transition probability E_w . With λ , the weight of the key word for Few-Shot data is increased which alleviates domain shift caused by any long-tail issue. At the same time, due to the introduction of E_w , the vocabulary diversity in the field will be richer.

3.4 Event detection

BERT has achieved SOTA performance on a wide range of tasks and has been proved very effective on ED scenarios (Wang et al., 2019). We apply BERT as our classifier. It could obtain semantics level information, overcoming the mismatch problem between words and event triggers (Lin et al., 2018). Following 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, after which a fully connected layer and softmax will be applied to calculate the score assigned to each event sub-type:

$$H = BERT(S) \quad (12)$$

$$c = HW_f + b_f \quad (13)$$

$$P(y | x) = softmax(c) \quad (14)$$

3.5 Active learning workflow

The most difficult part of test data to predict is 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 a deep model to perfectly learn it and overcome the catastrophic forgetting issues in the incremental learning process. To alleviate these problems, we apply an active learning mechanism to directly evaluate correct and wrong labels of the generated data. As in Fig. 3, if a poor amount of data is generated $S_c = \{S_w, Y_w\} |_1^w$, the effect of the classification model will be reduced, and we will abandon this batch of data. In the meantime, DAFS will continue to generate new data $G_w = \{G_x, G_y\}$ until our classifier achieves the relative higher scores when predicting $[S_w, Y_w]$. Formally, For DAFS:

$$g = \begin{cases} r = 1 & \text{add to } G^+ \\ r = 0 & \text{turn to } G_{x+1} \end{cases} \quad (15)$$

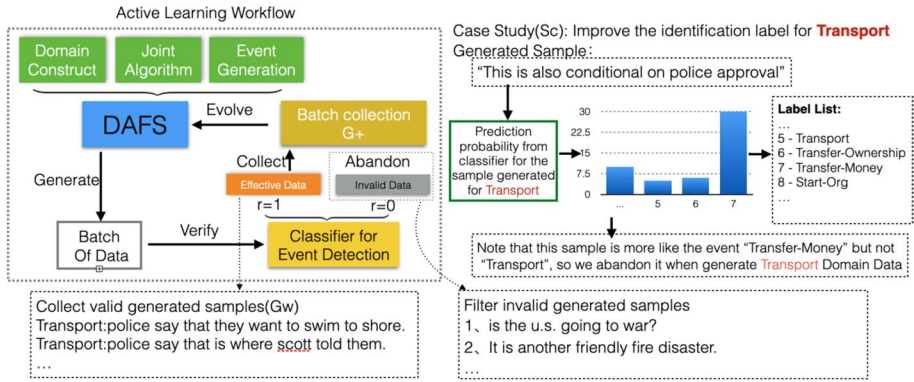


Fig. 3 The active learning workflow (AL) of our integration of DAFS and classifier to achieve incremental learning. For the event “Transport”, DAFS generate valid samples as well as invalid ones, AL picks up the right ones through its prediction probability, if the generated samples do positive effects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

For Classifier:

$$r = \begin{cases} 1 & P(Y_w) > Tep \text{ when predict } S_w \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

where S_y stands for the 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 the data generated next. 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 . The overall process is shown in algorithm 1.

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 $M_D^{d \times d}$, 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 $V_{in}^{1 \times 10}$ and $V_{out}^{1 \times 10}$. If the new trigger word is similar to the original one, we just need to modify the 20 relative positions in transition matrix $M_D^{d \times d}$. In addition, if the new trigger word is out of the dictionary of $M_D^{d \times d}$, we have to update the original matrix to $M_D^{(d+1) \times (d+1)}$ and do the 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.

Algorithm 1 Active Learning Workflow

Input: *Classifier*, *DAFS*, N_g , G^+ , L_d
 L_d stands for domain name, *TGD* stands for total generative data for L_d

```

1: for  $N = 0$  in  $D_t$  do
2:    $TGD = DAFS(L_d)$ 
3:   for Batch in  $TGD$  do
4:      $P(Y_w) = \text{Classifier}(\text{Batch})$ 
5:     if  $P(Y_w) > T_{ep}$  then
6:       Bod add to  $G^+$ ,  $L_{effective} = \text{Len}(G^+)$ 
7:       if  $L_{effective} \geq N_{stop}$  then
8:          $Update(DAFS)$ 
9:          $N = N + 1$ 
10:        break
11:      end if
12:    else
13:      continue
14:    end if
15:  end for
16: end for

```

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 (Chen et al., 2009), (Feng et al., 2018) and [35], 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 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 the “Other” category, because “bankrupt” is not to

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	✓	✓	✓	✓	✓

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 (Zeng et al., [yyy](#)), Forward-backward Recurrent Neural Networks (FBRNN) as proposed by (Ghaeini et al., [2018](#)), word-based DMCNN and Hybrid Neural Network proposed by (Feng et al., [2018](#)), incorporate CNN with Bi-LSTM and achieves the SOTA NN based result on ACE2005. Rich-C (Chen & Ng, [2012](#)) developed several handcraft Chinese-specific features, which improve the effect of Chinese ED. In addition, we adopt NPNs (Lin et al., [2018](#)) which can solve the word-trigger mismatch problem by directly proposing entire trigger nuggets centered at each character. Hybrid Character Representation(HCR) for ED (Xiangyu et al., [2019](#)) employs BERT-base model as its trigger classifier and achieve a relatively good score. It is the-state-of-the-arts for an 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 (Nguyen & Grishman, [2018](#)). Lu's DISTILL proposes a learning approach that applied effective separation, incremental learning, and finally adaptive synthesis of different event feature representation (Lu et al., [2019](#)). TS-DISTILL exploits the entity ground-truth and uses an adversarial imitation-based knowledge distillation approach for ED (Liu et al., [2019](#)). AD-DMBERT adopts a confrontational simulation model to continuously train the discriminator's resistance to noise (Wang et al., [2019](#)). DRMM employs an alternating dual attention to select informative features for mutual enhancements to ED (Tong et al.,

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

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 Table caption

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

2020). EKD leverages external open-domain trigger knowledge to reduce the inherent bias of frequent triggers in annotations (Tong et al., 2020) The last three baselines both use BERT as the feature extractor.

AFED To reflect the effectiveness of our model DAFS, we use only the original BERT (Devlin et al., 2018) model which is the best classifier in the real data set for comparison.

4.2 Overall performance

Tables 4, 5 and 6 show the results on ACE2005-CH & EN and AFED respectively. From the results, we can make 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 benefiting mainly from the effective data enhancement and the large-scale pre-training information of BERT. Our method expands the training data to further enhance BERT, which achieve better performance and demonstrates 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 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 the state of the arts by 6.7 in Recall. EKD introduces data from the outside, which improves precision considerably, indicating 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 Sect. 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 the original corpus, significantly improving the Precision, Recall and F1-score by 7, 10.2, 9.5 respectively. This proves that our model is also effective in generating a corpus with fuzzy boundaries, negative and questionable semantics problems in actual scene.
- (4) Figure 5 show that, our model delivers 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 amounts 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 and ACE-CH respectively.

4.3 Domain-based joint algorithm of generative model

To prove the effectiveness of our joint algorithm, we perform the following ablation experiments. Firstly, we define zero-shot as test data with no trigger word appearing in training data for the 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 the "Meet" event as our "Normal" case with its 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 the DTP matrix is helpful to maintain the stability of data generation, especially under the case of zero shot condition. It improves from 0 to 5 of when DTP is taken into consideration. Meanwhile, GTP increases the diversity of generated text. As shown in Table 7, DAFS with GTP could provide more data than DAFS with DTP in Few shot and normal region. And with GTP,

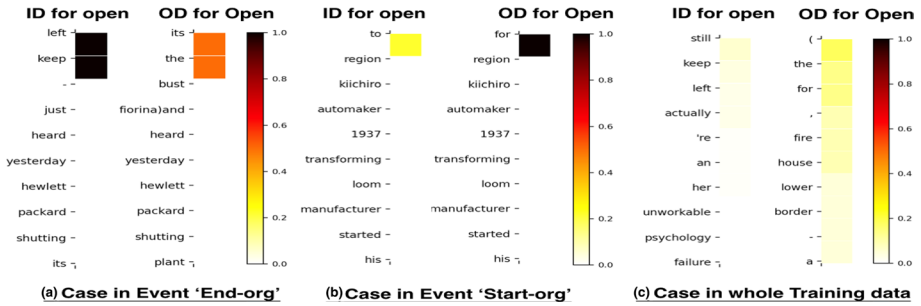


Fig. 4 The Global and specific domain transition probability. Example of the transition probability for the word “open” in Event “Start-org”, “End-Org”and train data. OD is short for Out of Degree and ID is short for In Degree

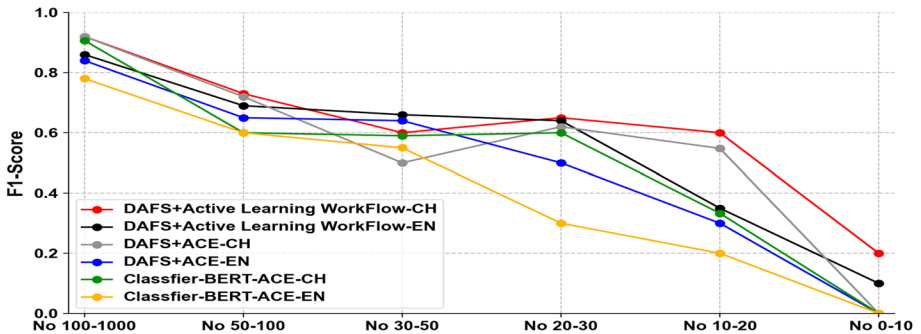


Fig. 5 The average F1-Score for different amount of training data on ACE multilingual dataset. X-axis represents the range of training data

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

DTP works better in Zero Shot conditions improving the generated number from 5 to 7. Here, we define diversity as the generation of richer and non duplicate data. Formally, when $\text{duplicate data}/\text{generate data} < 0.75$, we consider it as diversity. Here, duplicate data means the generated data, which is same as in the training corpus. Experiments show that the global transition matrix provides more choices than the domain matrix. However, 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 and c). So, we introduce weight parameter λ^1 to adjust its weight and calculate the joint probability that can achieve the best relative effect as visualized in Table 7.

4.4 Case study

In this section, we use some generated corpus as case studies to show the generation model under Zero Shot, Few Shot and Normal conditions². Note that, the generated data is not the fact, but the re combination of key trigger words and tags under the domain joint probability. As shown in Table 9, the generated corpora are positive samples conducive to ED. For the Zero-Shot data, its trigger only appears in test data but not in training data. We change the transition probability of each word in the transition matrix to realize synonym replacement. For example, in ACE2005, the trigger word “ordered” does not appear in the training corpus, but it has similar semantics to “buy” and “purchase”, which appear in the training corpus. And we generate “ordered” related data in the form of synonym substitution. The sample data generated by DAFS are all translated from Chinese, but it can be seen from the examples that the trigger words are complex in AFED, usually a combination of multiple words. For example, the triggered word is “Guarantee liability” in AFED for the domain Guarantee Liability, but in the test corpus showed the new word transposition combinations - “liability guarantee” and a new similar phrase - “liability for refund”. This phenomenon is more obvious in Chinese. Usually, a trigger word is composed of 4-5 words on average. However, our model can still generate more effective data to improve the classifier, which also proves the robustness for DAFS. In addition, as shown in the example, for the original data with normal level, DAFS can generate data that takes diversity and effectiveness into account.

4.5 Active learning workflow

As shown in Table 8, when active 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 suitable generated data to improve support for the incremental evolution of the classification model. In a real production environment, we often need models which can 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. Note that, under this workflow, we use test data to verify the generation data and then iterate to create both the DAFS and classifier evolution. This is more like offline learning, and so we discuss it separately in this chapter.

¹ This has been introduced in Sect. 3.3.

² The definitions of Zero-Shot, Few-Shot, Normal have been described in Sect. 4.3.

Table 9 case study for generated corpus

Generated Corpus	Domain	Volume	Corpus
Yao Qizhi, born in Shanghai, went to Tai Wan to study in the 1950s	Be-Born	Few Shot	ACE2005
Toshiba of Japan ordered 60 aircraft at this air show	Transfer Ownership	Zero Shot	ACE2005
The second in command of Putin's Government served as chairman	Start Position	Normal	ACE2005
In addition, Minsheng Bank added Li Qing's joint and several liability guarantee	Guarantee liability	Few Shot	AFED
Harbin Administration for Industry and Commerce issued the administrative guidance, requiring Harbin Pharmaceutical Co., Ltd. to bear joint and several liability for refund	Guarantee liability	Zero Shot	AFED
Saab admits that the bankruptcy risk is huge, and the performance of the group may be severely damaged	Bankruptcy liquidation	Normal	AFED

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 DAFS and the active learning mechanism, our workflow effectively solves the problems related to One-Shot learning. Experiments demonstrate that our method surpasses the 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.

Author Contributions Conceptualization: Nan Xia, Hang Yu and Xiangfeng Luo Methodology: Nan Xia, Hang Yu and Yin Wang Software: Nan Xia and Yin Wang Validation: Hang Yu and Yin Wang Formal analysis: Nan Xia and Yin Wang Investigation: Nan Xia and Yin Wang Resources: Xiangfeng Luo and Hang Yu Original draft preparation: Nan Xia and Yin Wang Writing: Nan Xia and Yin Wang Review and editing: Hang Yu and Nan Xia Supervision: Hang Yu and Xiangfeng Luo Project administration: Hang Yu and Nan Xia Funding acquisition: Xiangfeng Luo All authors have read and agreed to the published version of the manuscript.

Funding This project is supported by the following projects: 1) the Natural Science Foundation of China under the Grant No.91746203; Implementation period: January 2018 to December 2021. 2) Shanghai excellent academic leader project "construction method of dynamic knowledge map of tens of millions of small, medium and micro enterprises" under the Grant No.20XD1401700; Implementation period: August 2020 to September 2023.

Availability of data and material Most data has appeared in this paper.

Code availability The code will be released later in GitHub.

Declarations

Conflict of interest The authors have not disclosed any competing interests..

Ethics approval Not Applicable.

Consent to participate Not Applicable.

Consent for publication All authors appeared in this paper agree to publication in this journal.

References

- Cao, Y., Peng, H., Wu, J., Dou, Y., Li, J., Yu, P.S. (2021). Knowledge-preserving incremental social event detection via heterogeneous gnn. 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, <https://doi.org/10.1145/3442381.3449834> timestamp = Sun, 25 Jul 2021 11:46:32 +0200, <https://dblp.org/rec/conf/www/CaoPWDLY21>. sourcedblp computer science bibliography, <https://dblp.org>.
- Cao, Y., Hou, L., Li, J., Liu, Z. (2018). Neural collective entity linking. arXiv preprint [arXiv:1811.08603](https://arxiv.org/abs/1811.08603).


- Chen, Y., Liu, S., Zhang, X., Liu, K., Zhao, J. (2017). 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.
- Chen, Z., Ji, H. (2009). 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.
- Chen, C., Ng, V. (2012). Joint modeling for chinese event extraction with rich linguistic features. In: Coling. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., Salakhutdinov, R. (2019). Transformer-xl: Attentive language models beyond a fixed-length context. arXiv preprint [arXiv:1901.02860](https://arxiv.org/abs/1901.02860).
- Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805).
- Duan, S., He, R., Zhao, W. (2017). 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.
- Feng, X., Qin, B., & Liu, T. (2018). A language-independent neural network for event detection. *Science China Information Sciences*, 61(9), 092106.
- Ferguson, J., Lockard, C., Weld, D.S., Hajishirzi, H. (2018). Semi-supervised event extraction with paraphrase clusters. arXiv preprint [arXiv:1808.08622](https://arxiv.org/abs/1808.08622).
- Ghaeini, R., Fern, X.Z., Huang, L., Tadepalli, P. (2018). Event nugget detection with forward-backward recurrent neural networks. arXiv preprint [arXiv:1802.05672](https://arxiv.org/abs/1802.05672).
- Han, B., Yao, J., Niu, G., Zhou, M., Tsang, I., Zhang, Y., Sugiyama, M. (2018). Masking: A new perspective of noisy supervision. arXiv preprint [arXiv:1805.08193](https://arxiv.org/abs/1805.08193).
- He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- Huang, M., You, Y., Chen, Z., Qian, Y., Yu, K. (2018). Knowledge distillation for sequence model. In: Interspeech, pp. 3703–3707.
- Huang, L., Cassidy, T., Feng, X., Ji, H., Voss, C., Han, J., Sil, A. (2016). 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.
- Li, Q., Ji, H., Huang, L. (2013). 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.
- Lin, H., Lu, Y., Han, X., Sun, L. (2018). Nugget proposal networks for chinese event detection. arXiv preprint [arXiv:1805.00249](https://arxiv.org/abs/1805.00249).
- Liu, J., Chen, Y., & Liu, K. (2019). Exploiting the ground-truth: An adversarial imitation based knowledge distillation approach for event detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 6754–6761.
- Liu, Z., Wang, J., & Liang, Z. (2020). Catgan: Category-aware generative adversarial networks with hierarchical evolutionary learning for category text generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34, 8425–8432.
- Liu, S., Chen, Y., Liu, K., Zhao, J. (2017). 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.
- Lu, Y., Lin, H., Han, X., Sun, L. (2019). 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.
- McCann, B., Bradbury, J., Xiong, C., Socher, R. (2017). Learned in translation: Contextualized word vectors. In Advances in Neural Information Processing Systems, pp. 6294–6305.
- Mehta, S., Islam, M.R., Rangwala, H., Ramakrishnan, N. (2019). Event detection using hierarchical multi-aspect attention. In: The World Wide Web Conference, pp. 3079–3085.
- Nguyen, T.H., Grishman, R. (2018). Graph convolutional networks with argument-aware pooling for event detection. In: AAAI, vol. 18, pp. 5900–5907.
- Peters, M.E., Neumann, M., Logan IV, R.L., Schwartz, R., Joshi, V., Singh, S., Smith, N.A. (2019). Knowledge enhanced contextual word representations. arXiv preprint [arXiv:1909.04164](https://arxiv.org/abs/1909.04164).
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 9.
- Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. (2018). Improving language understanding by generative pre-training.

- Shao, H., Yao, S., Sun, D., Zhang, A., Liu, S., Liu, D., Wang, J., Abdelzaher, T. (2020). Controlvae: Controllable variational autoencoder. Proceedings of the 37th International Conference on Machine Learning (ICML).
- Tong, M., Wang, S., Cao, Y., Xu, B., Li, J., Hou, L., Chua, T.-S. (2020). Image enhanced event detection in news articles. In AAAI, pp. 9040–9047.
- Tong, M., Xu, B., Wang, S., Cao, Y., Hou, L., Li, J., Xie, J. (2020). 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). 10.18653/v1/2020.acl-main.522. <https://www.aclweb.org/anthology/2020.acl-main.522>.
- Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. *ACM Computer Survey*, 53(3), 63–16334. <https://doi.org/10.1145/3386252>.
- Wang, J., Zhao, L. (2018). Multi-instance domain adaptation for vaccine adverse event detection. In Proceedings of the 2018 World Wide Web Conference, pp. 97–106.
- Wang, X., Han, X., Liu, Z., Sun, M., Li, P. (2019). Adversarial training for weakly supervised event detection. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 998–1008.
- Xiangyu, X., Tong, Z., Wei, Y., Jinglei, Z., Rui, X., Shikun, Z. (2019). A hybrid character representation for chinese event detection. In 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–8.
- Yang, H., Chen, Y., Liu, K., Xiao, Y., Zhao, J. (2018). Dcfec: A document-level chinese financial event extraction system based on automatically labeled training data. In Proceedings of ACL 2018, System Demonstrations, pp. 50–55.
- Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In Advances in Neural Information Processing Systems, pp. 5753–5763.
- Yang, S., Feng, D., Qiao, L., Kan, Z., Li, D. (2019). 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.
- Zeng, Y., Feng, Y., Ma, R., Wang, Z., Yan, R., Shi, C., Zhao, D. (2017). Scale up event extraction learning via automatic training data generation. arXiv preprint [arXiv:1712.03665](https://arxiv.org/abs/1712.03665).
- 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.
- Zheng, J., Cai, F., Chen, W., Lei, W., Chen, H. (2021). 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, <https://doi.org/10.1145/3442381.3449949>, timestamp = Mon, 07 Jun 2021 14:20:06 +0200, <https://dblp.org/rec/conf/www/ZhengCCLC21>. sourcedb computer science bibliography, <https://dblp.org>.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Nan Xia¹ · Hang Yu^{1,2}  · Yin Wang¹ · Junyu Xuan¹ · Xiangfeng Luo¹

Nan Xia
shklt@shu.edu.cn

Yin Wang
wangyin2018@shu.edu.cn

Junyu Xuan
junyu.xuan@uts.edu.au

Xiangfeng Luo
luoxf@shu.edu.cn

- 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