TAED: Topic-Aware Event Detection

Yan Liang
School of Computer Science
University of Oklahoma
Norman, USA
yliang@ou.edu

Christan Grant
School of Computer Science
University of Oklahoma
Norman, USA
cgrant@ou.edu

Abstract—**Identifying event trigger words and classifying event types known as the event detection task is a fundamental step for extracting event-related knowledge from textual sources. Examples of the topics within documents include “military conflict,” “earthquake,” “concert tour,” “wrestling,” and others. Topical information embedded within documents where the events are extracted from is rarely explored. Rich topic information could be a helpful indicator of the event’s type. Semantically similar topics share similar event types, while event types are quite different between distinguishable document topics. In this paper, we explored a novel method of integrating document topic information to complete the event detection task. We summarized our contribution as the following: we used the topic information of the documents to generate topic comprehensive sentence representations. We adopted a multi-task deep neural network, trained with event detection and topic classification tasks. We evaluated our method with two datasets that are designed for more diverse and general event types event detection MAVEN [1] and RAMS [2]. We demonstrated that the topic-aware model outperformed the baseline model F_1 score on both MAVEN and RAMS datasets. An analysis in the few-shot event types scenario showed that topic-aware model can beat the baseline by up to 13.34% on the F_1 score for the rare event types.**

I. Introduction

Event detection is an important task of information retrieval in the natural language processing. Event detection and monitoring are the focus of public affair management for governments and researchers as timely understanding of social outbursts and evolution of popular social events [3], [4], [5]. Structured events are used in the construction domain for the development of knowledge bases [6], [7], [8]. In the business and financial domain, event detection helps companies quickly discover market responses of their products and influencing signals for risk analysis and suggestions [9], [10]. Despite its promising applications, event detection is still a rather challenging task. As events are diverse, they come with different structures and components. Further, natural languages are often with semantic ambiguities and discourse styles.

Events are typically created manually or developed with a seed of manually labeled patterns. Event detection aims to find the *event triggers* — the main word that most clearly expresses an event occurrence, typically a verb or a noun. Event detection techniques then use the triggers to classify the *event type* into a predefined set.

978-1-6654-8045-1/22/$31.00 ©2022 IEEE

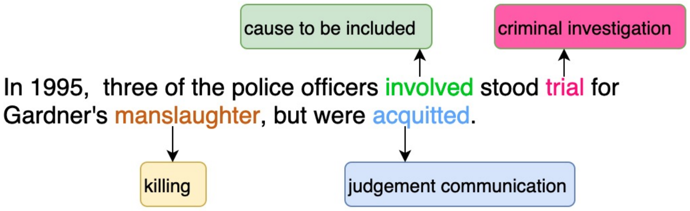
cause to be included
criminal investigation
In 1995, three of the police officers involved stood trial for Gardner's manslaughter, but were acquitted.
killing
judgement communication

Fig. 1. The single sentence contains four event-trigger words that belong to different event types.

Figure 1 shows events types and triggers described in the MAVEN dataset [1]. In the sentence, “In 1995, three of the police officers involved stood trial for Gardner’s manslaughter, but were acquitted.” where “involved” triggers a **cause to be included** event, “trial” triggers a **criminal investigation** event, “manslaughter” triggers a **killing** event, and “acquitted” triggers a **judgment communication** event.

Early approaches for event detection rely on pattern matching [11]. Later, researchers started testing machine learning-based approaches such as support vector machines to address event detection [12]. More recently, deep learning has been successfully applied to various NLP tasks including event detection. The current approach to build a deep neural network is to first take word embeddings as input and output a classification result for each word. Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) , and Graph Neural Network (GNN) have been explored and applied [13], [14], [15], [16], [17], [18].

However, none of the previous efforts take event topic information into consideration for the event detection task [14], [15], [19], [16]. Examples of topics that the documents belong to include “terrorist attack”, “horse race”, “earthquake” as shown in Table I. Intuitively, topic information is important for event detection tasks as documents belonging to different topics naturally have different event type distributions. There are several existing event detection datasets, for example, ACE05 [20], ERE series [21], [22], TERRIER [5], [23], [24], [25]. In order to validate our assumption, we did analysis with MAVEN [1] dataset. We chose MAVEN because it has a large range of event types compared to others. For example, Maven has 168 event types while ACE05 contains 8 event types and 33 specific subtypes. We gathered all the 168 event

TABLE I

An example of event type distribution (top 5) for three topics: Earthquake, Horse Race, and Terrorist Attack.

earthquake	event types	catastrophe	causation	damaging	coming to be	destroying
	event types distribution	0.255	0.076	0.072	0.043	0.033
horse race	event types	competition	process start	process end	causation	hold
	event types distribution	0.201	0.104	0.058	0.047	0.036
terrorist attack	event types	attack	killing	terrorism	bodily harm	causation
	event types distribution	0.145	0.074	0.058	0.049	0.035
civilian attack	event types	killing	attack	statement	causation	bodily harm
	event types distribution	0.094	0.068	0.041	0.035	0.032

types in MAVEN. Then, for each topic, we normalized the event type occurrence count to a 168 dimensional vector, of which all the 168 elements in the vector summarize to 1.. We used this vector to represent the event distribution of the current topic. For each pair of the topics, we performed a two sample Kolmogorov–Smirnov test and report the P-Value as a heat map in Figure 2. When the P-Value is bigger than 0.05, it means we can not reject the null hypothesis that the two topics follow the same event type distribution. When the P-Value is less than or equal to 0.05, we can reject the null hypothesis.

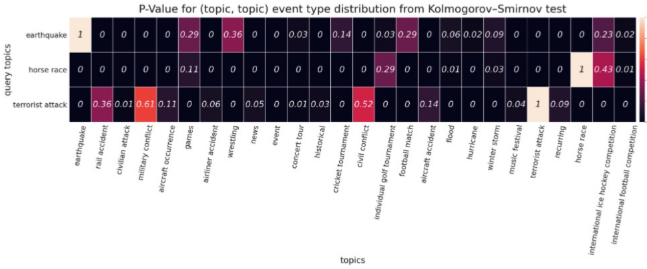

Fig. 2. P-Value from Kolmogorov–Smirnov test on distribution of event types across topics. The smaller the P-Value in the cell is, the bigger the difference of event type distributions between two topics. (Partial version. Full version in Figure 3)

Based on Figure 2 we can see when talking about event type distribution, topic “terrorist attack” is most similar to “military conflict” and “civil conflict” topics with corresponding large P-Value 0.61 and 0.52, respectively. Topic “horse race” is most similar to “international ice hockey competition” and “individual golf tournament” topics with corresponding P-Value 0.43 and 0.29, respectively. In Table I, we show the top-5 event types for the 3 topics: “earthquake”, “horse race” and “terrorist attack”, from which we can clearly see that event type distributions are affected by different topics. Semantically similar topics share similar event type distributions, while semantically different topics have heterogeneous distributions of event types. This inspires us to explore effective ways of using topic information in the event detection task to improve its performance.

We summarize our contributions as the following: (1) We perform detailed analysis explaining why topic information helps on event detection task. (2) We introduce topic name enhanced sentence representation for event detection and explore different ways to embed the topic name information including: using attention-based versus concatenation-based interaction, `[CLS]` versus token average based attribute embedding and using topic keywords to generate topic embedding versus using topic names. (3) We introduce topic classification and event detection as a multi-task learning setup, which further improves the performance and conduct experiments with two event detection datasets that have a variety of event types. We achieve up to $+1.8\%$ on the F_1 score compared to the baseline. (4) Furthermore, we show the topic-aware model proposed can improve the few-shot event types scenario by a large margin $+13.34\%$ on the F_1 score and provide heuristic explanations in the case study.

The rest of this paper is organized as follows. Section II gives the definition of NLP event detection task. Section III describes the system encoders, representations, and decoders used for event detection and topic classification training. Section IV describes experimental results. Section V conducts analysis of the results and Section VI discusses related work. Finally, Section VII concludes and suggests future work.

II. Event Detection Definition

An event is a specific occurrence of something that happens in a certain time and a certain place, which can frequently be described as a change of state [26]. An *event structure* is defined as follows in ACE05 terminology:

- Event Mention: a phrase or sentence describing an event, including a trigger and several arguments.
- Event Trigger: the main word that most clearly expresses an event occurrence, typically a verb or a noun.

The event detection tasks are defined as follows:

- Trigger Identification: aims to identify the most important word that characterizes an event.
- Trigger Classification: aims to classify the event trigger into predefined, fine-grained categories.

Recent neural network methods typically formulate event detection task as a token-level multi-class classification task [27], [15] or a sequence labeling task [28], and only report the trigger classification results [1], [29]. An additional type N/A is introduced and classified at the same time to indicate the candidate is not a trigger. We adopt the above settings and evaluate the performance with precision, recall and F_1 on a micro level.

III. Methodology

TAED leverages the document topics for event detection. The underlying intuition is that event type distributions are

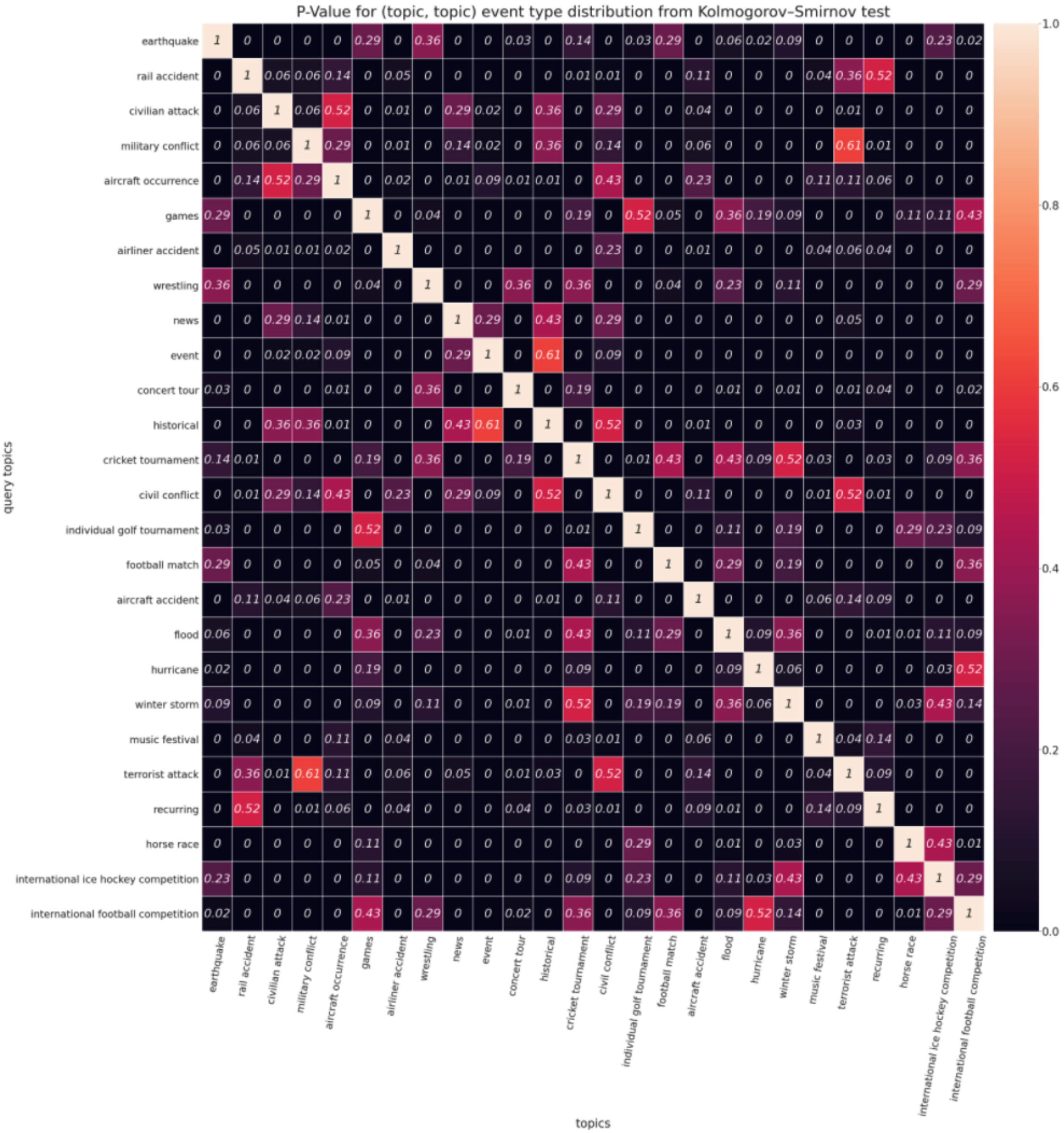

Fig. 3. P-Value from Kolmogorov–Smirnov test on distribution of Event Types across Topics. The smaller the P-Value in the cell is, the bigger the difference of event type distributions between two topics. (Full version)

different across the topics. Our model uses the topic name embedding to enhance the sentence representation. Furthermore, we have modeled topic classification and event detection as a multi-tasking learning setup.

A. Sentence Encoder

The sentence encoder represents the text tokens of the sentence $(x_1, x_2..., x_T)$ as low-dimensional, real-valued vectors. To effectively capture the long-range dependencies between the input tokens, we use BERT [30] whose underlying layers use the self-attention mechanism to mitigate the long-range

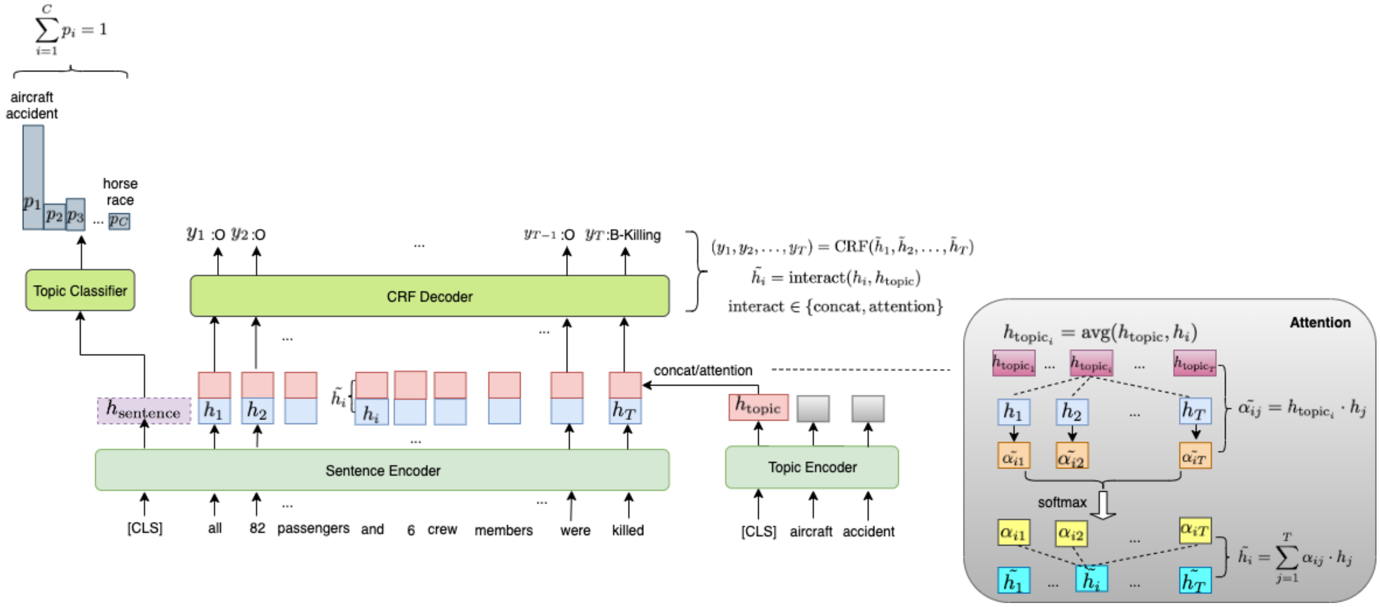
$\sum_{i=1}^{C} p_i = 1$
aircraft accident
horse race
p_1 p_2 p_3 ... p_C
Topic Classifier
y_1 :O y_2 :O
...
y_{T-1} :O y_T :B-Killing
$(y_1, y_2, \dots, y_T) = \mathrm{CRF}(\tilde{h}_1, \tilde{h}_2, \dots, \tilde{h}_T)$
$\tilde{h}_i = \mathrm{interact}(h_i, h_{\mathrm{topic}})$
interact ∈ {concat, attention}
CRF Decoder
concat/attention
h_{sentence}
h_1 h_2 $\tilde{h}_i$ h_i h_T
h_{topic}
Sentence Encoder
Topic Encoder
[CLS] all 82 passengers and 6 crew members were killed
[CLS] aircraft accident
Attention
$h_{\mathrm{topic}_i} = \mathrm{avg}(h_{\mathrm{topic}}, h_i)$
h_{topic_1} ... h_{topic_i} ... h_{topic_T}
h_1 h_2 ... h_T
$\tilde{\alpha_{ij}} = h_{\mathrm{topic}_i} \cdot h_j$
$\tilde{\alpha_{i1}}$ $\tilde{\alpha_{i2}}$ $\tilde{\alpha_{iT}}$
softmax
α_{i1} α_{i2} ... α_{iT}
$\tilde{h}_i = \sum_{j=1}^{T} \alpha_{ij} \cdot h_j$
$\tilde{h}_1$ ... $\tilde{h}_i$ ... $\tilde{h}_T$

Fig. 4. TAED Architecture: Using topic name embedding along with the sentence tokens embedding as the enhanced representation of the sentence. Utilizing multi-task learning of event detection and topic classification together to further improve the performance.

dependencies issue,

$$h_1, h_2 ..., h_T = \mathrm{Encoder}(x_1, x_2, ..., x_T) \tag{1}$$

where $h_i \in \mathbb{R}^d$.

B. Topic Encoder

Our topic encoder encodes the topic information by using the topic name or topic representative vocabulary that is mined by using the ranked tf-idf features from each topic. For example the top-5 representative vocabularies for “civilian attack” topic is: “massacre”, “attack”, “kill”, “police”, “people”. Similarly we use BERT as encoder to encode the topic information. Different from the sentence encoder, here we use the [CLS] token (red, solid-border token in Figure 4) returned from BERT encoder to represent the entire information carried by the topic keywords.

$$h_{\mathrm{topic}} = \mathrm{TopicEncoder}(\mathrm{topicword}_1, ..., \mathrm{topicword}_N), \tag{2}$$

where $h_{\mathrm{topic}} \in \mathbb{R}^d$

C. Topic-Aware Sentence Representation

To associate the sentence representation with its document's topic, we append the topic vector representation h_{topic} to each token vector representation $h = (h_1, h_2, ..., h_T)$ in the sentence, shown in Figure 4. Then we get the topic-aware contextualized vector representations of the sentence tokens

$$\tilde{h} = (\tilde{h}_1, ..., \tilde{h}_T) = (h_1; h_{\mathrm{topic}}, .., h_T; h_{\mathrm{topic}}) \tag{3}$$

where $\tilde{h}_i \in \mathbb{R}^{2d}$ and ; operator represents concatenation. The drawback of using concatenation is the topic information contributes to each token in the sentence evenly. In order to address this issue, we proposed an attention based interaction method to obtain a topic-aware comprehensive sentence representation. The idea of attention was first used in Neural Machine Translation (NMT) [31]. Instead of paying attention to everything, the attention mechanism is designed to highlight the important information in a sequence. In order to calculate the attention we first need to define query, key, value used in our scenario. For the query, we obtain a d-dimensional vector h_{topic_i} by taking average of h_{topic} and h_i. We use $(h_1, h_2, \dots, h_T)$ for both key and value. Given index i, a dot product operation is applied on h_{topic_i} and h_j, where j ranges from 1 to T. Each of the dot product operation generates a weight α_{ij}, calculated as:

$$\tilde{\alpha_{ij}} = h_{\mathrm{topic}_i} \cdot h_j, \text{ and} \tag{4}$$

$$(\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{iT}) = \mathrm{softmax}(\tilde{\alpha_{i1}}, \tilde{\alpha_{i2}}, \dots, \tilde{\alpha_{iT}}). \tag{5}$$

In this way, given a query vector h_{topic_i}, we obtain a sequence of weights $(\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{iT})$. The weights are used to measure the importance of tokens in the sentence when talking about the topics. A higher weight indicates a higher importance. Afterwards, a topic-comprehensive representation for the query h_{topic_i} can be obtained by calculating:

$$\tilde{h_i} = \sum_{j=1}^{T} \alpha_{ij} \cdot h_j. \tag{6}$$

Similarly, we can obtain $\tilde{h_i}$ for each query vector h_{topic_i}, where i ranges from 1 to T. Eventually, a sequence of topic weighted topic-aware hidden vectors $(\tilde{h_1}, \tilde{h_2}, \dots, \tilde{h_T})$ are obtained. Each $\tilde{h_i}$ is a d-dimensional vector.

D. Event Detection CRF Decoder

We adopt the BIOE tagging scheme. “B”/“E” indicates the corresponding word is the beginning/ending of an entity value, “I” means the word is inside an entity value, and “O” means the word is outside any entity value. Table II shows an example of identifying “took place” as a trigger of event type **process start**.

We feed the topic-aware contextualized token representations $(\tilde{h}_1, \tilde{h}_2, ..., \tilde{h}_T)$ to CRFs [32] to get the sequence of BIOE tags with the highest probability:

$$(y_1, y_2, ..., y_T) = \text{CRF}(\tilde{h}_1, \tilde{h}_2, ..., \tilde{h}_T), \tag{7}$$

CRF decoder [33] can enforce the tagging consistency that captures dependency between the output tags. For example, if we already know the starting boundary of an attribute (B), this increases the likelihood of the next token to be an intermediate (I) or end of boundary (E), rather than being outside of boundary (O). CRF contains a linear layer and a transition matrix, which are used to calculate the emission and transition scores for the tag predictions respectively. The score for an input text sequence X which belongs to a specific topic to be assigned with a tag sequence Y can be calculated as:

$$\text{score}(X, \text{topic}, Y) = \sum_{i=1}^{T-1} \mathbf{T}_{y_i, y_{i+1}} + \sum_{i=1}^{T} \mathbf{E}_{i, y_i}, \tag{8}$$

where $\mathbf{T} \in \mathbb{R}^{m \times m}$ is the transition matrix, $\mathbf{T}_{ij}$ is the transition score of i-th tag to the j-th tag. $\mathbf{E} \in \mathbb{R}^{T \times m}$, $\mathbf{E}_{ij}$ represents the i-th token is assigned j-th tag in the tagset. m is the number of tags in the tagset which includes different B, I, E tags for each event type and a shared O tag. For example, given two event types **Killing** and **Cause to be included**, there will be 7 tags including B-**Killing**, I-**Killing**, E-**Killing**, B-**Cause to be included**, I-**Cause to be included**, E-**Cause to be included** and an O tag. Let a be the number of event types, then $m = 3a + 1$.

E. Event Detection Training

The event detection task is trained to maximize the log likelihood of (X,topic,Y) triplets in the training set, the score of given tokens, and topic that has predicted tags Y is given in equation (8), and the log likelihood to maximize is defined as:

$$\log p(Y|X, \text{topic}) = \log \frac{\text{score}(X, \text{topic}, Y)}{\sum_{Y' \in tagset^T} \text{score}(X, \text{topic}, Y')}. \tag{9}$$

Assuming we have N samples in the training set, then the loss (ℓ) to minimize for the event detection task is defined as:

$$\ell_{\text{event\_detection}} = -\sum_{i=1}^{N} \log p(\hat{Y}_i | X_i, \text{topic}_i), \tag{10}$$

where $\hat{Y}_i$ is the ground truth label for sentence i.

F. Topic Classification Training

Our topic classifier classifies each sentence into its corresponding topic. In order to avoid information leakage, instead of using the topic-aware contextualized token embeddings $\tilde{h}$ from equation (3) to classify the topics, we directly use the [CLS] token representation denoted as h_{sentence} from the sentence encoder (the purple, dashed-border token in Figure 4) to classify the topic.

$$\begin{gathered}(p_1, ..., p_C) = \text{softmax}(W_t h_{\text{sentence}} + b_t) \\ Loss_{\text{topic}} = -\sum_{j=1}^{N} \sum_{i=1}^{C} y_{ij} \log(p_{ij}),\end{gathered} \tag{11}$$

where $W_t \in \mathbb{R}^{C \times d}$, $b_t \in \mathbb{R}^C$ and C is the number of the topics in the training dataset.

G. Multi-Task Training

We jointly train TAED for event detection and topic classification in a multi-task learning [34], [35], [36] setting, by combining the loss of the two tasks:

$$\ell = \ell_{\text{event\_detection}} + \gamma \cdot \ell_{\text{topic}} \tag{12}$$

where γ is a non-negative hyper-parameter. By training the model in a multi-tasking setting, both of the event detection and topic classification tasks will contribute to the contextualized vector representation learning for the sentence and topic tokens.

IV. Experiments

In order to validate our hypothesis that the event topic information can help event detection, we used the MAVEN [1] dataset which has a large range of event types (168) and also comes with the topic labels that was annotated by humans to conduct our experiments. Furthermore, we also test our model with the RAMS [2] dataset which also comes with many event types (139). Though it does not come with topic labels for each document, we use LDA [37] to generate pseudo topic labels. For the RAMS dataset topic generation, we choose the topic number from the range of [10, 15, 25, 30, 35] and manually judge the quality of the topics returned and end up using the topic number of 25 as the best fit for the RAMS dataset. The pseudo topic name is from the outputs of the LDA, which is the combination of the top 5 important words for the specific topic. The pseudo topic names are listed below in Table III for RAMS dataset. We first tested our work on full MAVEN dataset. Further, due to the MAVEN dataset only releases the gold labels for training and validation dataset instead of the gold labels for test dataset, in order to speed up our experiments with the data that has gold labels, we combined the training and validation dataset, then separated the merged dataset further into a 70%/15%/15% distribution. Additionally, the topic occurrence in the original dataset is extremely skewed, with the highest topic occurrence as 984 and lowest topic occurrence as 1. We further sampled several topic balanced datasets to test the effectiveness of our proposed method in a topic balanced scenario.

TABLE II
AN EXAMPLE OF THE TAG SEQUENCE FOR EVENT TYPE "PROCESS START" ANNOTATED WITH THE BIOE SCHEME.

the	Total	Nonestop	Action	Wresting	(	TNA	)	promotion	that	took	place	on	October	23
O	O	O	O	O	O	O	O	O	O	B-Process Start	E-Process Start	O	O	O

TABLE III
PSEUDO TOPIC LABELS GENERATED BY LDA FOR RAMS DATASET

Topic index	Pseudo Topic Label
0	photo caption image hide force
1	oil sanction export year price
2	inform secure intelligence government email
3	police attack shoot fan man
4	email investigate fbi hack department
5	attack bomb build kill force
6	world war think agreement
7	president state country nato unit
8	nuclear bank missile system weapons
9	attack report investigate news told
10	force rebel group military air
11	women attack share right day
12	isra aid human intern right
13	support crimea muslim use
14	campaign republican president donald democrat
15	president support campaign former bush
16	million tax percent wall pay
17	foundation million report government department
18	vote voter elect party poll
19	women sexual rape extradite year
20	senate gop bill seek ryan
21	white house democrat party polite
22	question ask debate cnn death
23	polite think president attempt thing
24	state unit war foreign world

Furthermore, to validate that topic can be used as a "bridge" to transfer knowledge from high resource event types to low resource event types, we grouped event types based on their occurrences in the training dataset defined in Table V. Examples of low resource event types can be from *Rare* group, with the occurrence of the event type less than 20 times. While high resource event types can be from *High* group, with the occurrence of the event type more than 500 times. We generated the Macro average group precision, recall and F_1 score accordingly.

A. Performance

Altogether, we tested our topic-aware event detection framework on 5 settings of MAVEN which include the following: full MAVEN dataset, two new splits of the train and validation dataset which has the gold labels and two new splits of the train and validation datasets sampled in a topic balanced way. On each of the setting, we included 3 random seeds and report the means and variations of each metric in Table IV. For the generation of the topic balanced dataset, we first removed the tail topics whose topic occurrence is less than a threshold, and then down sampling the head topics to the median occurrence of the topics. BERT-CRF is using BERT as the encoder and CRF as the decoder. BERT-CRF-TOPIC is our TAED architecture as shown in Figure 4. The topic-classification-weight is the weight set on the topic classification task when setting the event detection task weight as 1. The performance of topic-aware and non-topic-aware model are shown in Table IV. We saw that on full MAVEN and two generated dataset settings from splits of train and validation dataset, topic-aware model improves the baseline around 0.5% on the entity level micro F_1 score, which is a commonly adopted metric for event detection task [1]. For the two topic balanced datasets, the improvement on the F_1 score is around 1.8%. On the RAMS dataset, we also saw improvement on the F_1 score around 0.6%. We observed that the performance of event detection model on RAMS is relative low. By further investigation on the training data, 25% of event types have less than 27 labeled instance which explains this.

The group performance based on high and low resources event types classification is shown in Figure 5. We observed that topic-aware model is doing much better on the low resource event types like *Rare* and *Low* group, with up to 13.34% improvement on F_1 score compared to non topic-aware model.

B. Ablation Study

In the ablation study, we first evaluate different ways to generate the topic embeddings. Further, we conduct experiments to show the effectiveness of different ways the topic information interacting with the main contextual sentence. Lastly, we carried out experiments to show that auxiliary topic-classification task is effective.

a) Topic Name Encoding.: As shown in Table VI, we observed that using the topic information to generate the context embedding without using a multi-task learning (set weight = 0) is effective. We got improvement on the F_1 by 1.07% from 63.66% to 64.73%.

We further tried different ways to generate the topic name embedding. Instead of using the [CLS] token hidden representation, we tried to use the averaged topic name tokens embeddings. Furthermore, aside from concatenating the topic name embedding on top of the context token embedding, we also experimented to use topic name attend to the context sentence tokens. We see similar F_1 performance for the above variations as show in Table VII.

b) Topic Name Variations.: The column "general event word removed" shown in Table VI indicates whether or not we remove the very general word "event" from the topic name. Since the original topic name could be like: "recurring event", "historical event", "wrestling event". After removing the word "event", the topic name should look like "recurring",

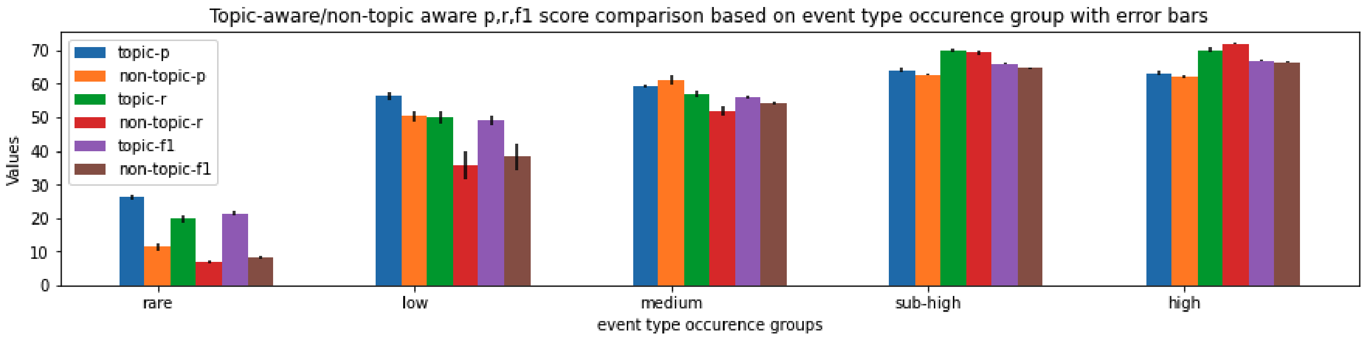

Fig. 5. Topic-aware/non-topic-aware model Macro P, R, F1 performance with error bars on different event type occurrence groups defined in Table V

TABLE IV
PRECISION, RECALL, AND F_1 PERFORMANCE ACROSS SIX DIFFERENT DATASETS WITH AND WITHOUT TOPIC INFORMATION

Model Type	**Dataset**	**P(%)**	**R(%)**	**F_1(%)**
BERT-CRF	Full MAVEN Data	66.15 ± 0.24	69.64 ± 0.43	67.85 ± 0.07
BERT-CRF-TOPIC	Full MAVEN Data	66.28 ± 0.38	70.39 ± 0.40	$\mathbf{68.27 \pm 0.06}$
BERT-CRF	Generated Data 1	65.18 ± 1.14	70.32 ± 2.96	67.63 ± 0.05
BERT-CRF-TOPIC	Generated Data 1	66.21 ± 0.16	70.23 ± 0.16	$\mathbf{68.16 \pm 0.03}$
BERT-CRF	Generated Data 2	65.65 ± 0.30	69.74 ± 0.38	67.63 ± 0.08
BERT-CRF-TOPIC	Generated Data 2	66.35 ± 0.12	70.14 ± 0.34	$\mathbf{68.19 \pm 0.10}$
BERT-CRF	Generated Data Topic Balanced 1	64.09 ± 1.67	62.68 ± 1.72	63.33 ± 0.05
BERT-CRF-TOPIC	Generated Data Topic Balanced 1	63.9 ± 0.3	65.17 ± 0.18	$\mathbf{64.52 \pm 0.09}$
BERT-CRF	Generated Data Topic Balanced 2	63.93 ± 1.51	63.21 ± 1.28	63.53 ± 0.11
BERT-CRF-TOPIC	Generated Data Topic Balanced 2	64.41 ± 0.28	66.50 ± 0.24	$\mathbf{65.44 \pm 0.02}$
BERT-CRF	RAMS	34.05 ± 0.14	33.83 ± 0.03	33.94 ± 0.05
BERT-CRF-TOPIC	RAMS	36.67 ± 0.12	32.69 ± 0.04	$\mathbf{34.56 \pm 0.04}$

TABLE V
EVENT TYPE GROUPS BASED ON ITS OCCURRENCE FREQUENCY IN TRAINING DATA

Groups	**Occurrence**	**Event Type Count**	**Event Type Examples**	**Macro P(%)**		**Macro R(%)**		**Macro F_1(%)**	
				topic	non-topic	topic	non-topic	topic	non-topic
Rare	(0,20]	38	besieging, ratification	26.12 ± 0.49	11.29 ± 1.09	19.63 ± 1.03	6.76 ± 0.42	$\mathbf{21.49 \pm 0.54}$	8.15 ± 0.36
Low	(20, 50]	35	warning, rescuing	56.39 ± 1.17	50.25 ± 1.50	49.99 ± 1.71	35.35 ± 4.25	$\mathbf{50.66 \pm 1.46}$	38.25 ± 3.82
Medium	(50, 100]	35	assistance, escaping	59.12 ± 0.42	61.16 ± 1.54	56.78 ± 0.89	51.72 ± 1.44	$\mathbf{56.51 \pm 0.47}$	54.07 ± 0.57
Sub-high	(100, 500]	53	damaging, destroying	64.06 ± 0.67	62.60 ± 0.23	70.00 ± 0.38	69.26 ± 0.51	$\mathbf{66.03 \pm 0.31}$	64.61 ± 0.38
High	$(500,\infty)$	7	catastrophe, causation	63.13 ± 0.75	61.93 ± 0.38	70.08 ± 0.60	71.78 ± 0.22	66.34 ± 0.38	$\mathbf{66.39 \pm 0.16}$

“historical”, “wresting” etc. This is going to help make our topic embedding more discriminatory from each other. By adding this pre-processing step for the topic name, we can see that the performance gets improved, as shown in Table VI. We further explored to add the most important keywords of the topic along with the topic name to enrich the topic contextual embedding. We aggregated the documents that belong to one topic, and ranked the words in each topic by their tf-idf features. We used the top-5 keywords as the representatives and appended them to the topic name. Examples of added keywords are shown in Table VIII.

However, after adding the keywords to the topic names, the performance got a little bit worse, which could be caused by the noise brought in by the keywords. For example, the keyword “new” was added for the winter storm topic and “1930” was added for the war topic.

c) Multi-Task Learning.: We have conducted experiments by using different weights on topic classification task, where γ in equation (12) ranges from 0 to 100 where 0 means we ignore the topic classification loss during back propagation. We saw that the sweet spot to achieve the best performance is to set the classification weight as 1 shown in Figure 6.

By setting equal loss weight on event detection and topic classification tasks, we further improved the F_1 score by another 0.73% on top of the topic name embedding contribution.

C. Hyperparameter Settings

We implement Topic Aware Event Detection framework by using functionality provided by PyTorch and Transformers package. We adopt bert-base-cased version of BERT model and use the default AdamW optimizer with the Learning rate as $5*10^{-5}$ and Adam Epsilon as $1*10^{-8}$. An dropout layer is introduced after the BERT encoder layer with a dropout rate 0.3. And the training batch size for the model is 16.

TABLE VI
TAED PERFORMANCE WITH DIFFERENT TOPIC-CLASSIFICATION WEIGHTS, PERFORMANCE OF GENERAL EVENT WORDS KEPT/REMOVED AND PERFORMANCE OF EXTRA TOPIC KEYWORDS ADDED ON FOR A SPECIFIC TOPIC.

Model Type	**topic-classification weight**	**general event word removed**	**P(%)**	**R(%)**	**F_1(%)**
BERT-CRF	NA	NA	66.91	60.71	63.66
BERT-CRF-TOPIC	1	True	64.44	66.52	**65.46**
BERT-CRF-TOPIC	0	True	63.52	66.04	64.73
BERT-CRF-TOPIC	0.1	True	62.37	66.66	64.44
BERT-CRF-TOPIC	0.5	True	64.17	64.92	64.53
BERT-CRF-TOPIC	2	True	63.76	65.33	64.51
BERT-CRF-TOPIC	10	True	64.26	58.73	61.38
BERT-CRF-TOPIC	25	True	63.8	47.34	54.33
BERT-CRF-TOPIC	50	True	60.36	33.06	42.65
BERT-CRF-TOPIC	75	True	55.98	25.69	34.99
BERT-CRF-TOPIC	100	True	49.58	19.38	27.79
BERT-CRF-TOPIC	1	False	65.59	64.29	64.93
BERT-CRF-TOPIC (with vocab)	1	True	64.97	65.08	65.02

TABLE VII
PERFORMANCE OF USING DIFFERENT WAYS TO GENERATE AND USE TOPIC NAME EMBEDDING.

Model Type	**Topic Embedding Type**	**P(%)**	**R(%)**	**F_1(%)**
BERT-CRF-TOPIC	[CLS]	67	63.78	65.35
BERT-CRF-TOPIC	Average Token Embedding	64.53	66.44	65.47
BERT-CRF-TOPIC	[CLS] freeze	64.57	66.57	**65.56**
BERT-CRF-TOPIC	Average Token Embedding freeze	65	65.64	65.32
BERT-CRF-TOPIC	[CLS] (topic as attention)	63.93	67.02	65.44

V. RESULT ANALYSIS

The performance showed in Table VI combines both the performance of trigger identification and trigger classification. We further get the performance only for trigger identification shown in Table IX. From which we can see that the topic-aware event detection model gets better performance on both trigger identification and trigger classification. The error cases for event identification could come from two sources: 1. The triggers are not correctly identified. 2. The triggers are correctly identified, but the classification of the identified triggers is wrong. We conducted case studies for both of the error sources. "Flight 821 is the deadliest **accident** involving a Boeing 737-500, surpassing the 1993 crash of Asiana Airlines Flight 733, and was the second-deadliest aviation **incident** in 2008, behind Spanair Flight 5022." The topic of the sentence is "aircraft accident" and the top event type for this topic is **catastrophe**, **causation**, **motions**. The gold labels for the triggers, "accident" and "incident", are both **B-catastrophe**. The non topic-aware model failed to identify the triggers in the first place, while the topic-aware model identifies the triggers and classifies them correctly into a **catastrophe** event. Another example: "This was the first southern stadium rock show since ZZ TOP **played** to 80,000 people at UT Austin on September 1, 1974 and tore up the field." Both of the method identified "played" as the trigger. However, the topic-aware model predicted "played" as **B-competition** while non topic-aware model predicted it as **B-participation**. The gold label for "played" is **B-competition**. The topic of the sentence is "music festival" and the top event type for this topic includes **social event**, **process start**, **arranging**, **competition**. We can see that in both of the error cases, topic information plays an important role for event detection task.

TABLE VIII
SAMPLE OF 10 TOPIC VOCABULARY TERMS AND TOP-5 REPRESENTATIVE KEYWORDS.

Topic	**Topic Vocabulary**
earthquake	magnitude, occurred, quake, intensity, damage
winter storm	snow, blizzard, snowfall, new, winds
tennis event	open, doubles, slam, singles, djokovic
rugby match	chiefs, brumbies, sharks, final, crusaders
university boat race	oxford, cambridge, lengths, crews, goldie
war	paulo, vargas, 1930, presets, garais
military operation	bomb, manchester, ira, bombing, embassy
swimming event	golds, medals, bronze, silver, freestyle pool
cricket series	ashes, england, australia, test, wickets
civilian attack	massacre, attack, kill, police, people

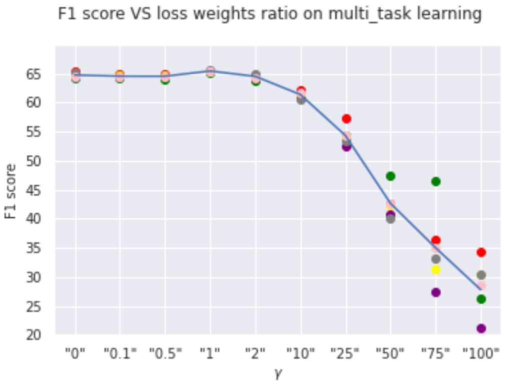

Fig. 6. F_1 performance vs. γ (Each γ on X-axis has been run 5 times with different random seeds represented by points with different colors. The curve is the average performance of the 5 runs for each γ.)

In addition, we did case studies to understand the "bridge" behavior of document topic on transferring knowledge from high resource event types to low resource event types. For example, **besieging** is a rare event type. The most frequent topic that the event type belongs to is "military conflict". The "military conflict" further has frequent event types like: **hostile encounter**, **attack** etc, which are semantically related to the low resource event type **besieging**. By using the topic name as prior knowledge along with the introduction of the topic classification task, we reinforce the topic information in the hidden layer sent to the decoder. The hidden layer thus carries the information of high resource event types that belongs to the given topic. This further leads to information transformation to low resource event types that are semantically related to high resource event types.

VI. RELATED WORK

Ji [38] employs an approach to propagate consistent event arguments across sentences and documents. By combining

TABLE IX
PERFORMANCE OF BERT-CRF AND BERT-CRF-TOPIC ONLY ON TRIGGER IDENTIFICATION

Model Type	**P(%)**	**R(%)**	**F$_1$(%)**
BERT-CRF	77.3	77.9	77.6
BERT-CRF-TOPIC	77.93	78.59	**78.26**

global evidences from related documents and local decisions, a cross-document method is created to improve event detection task. Li [39] proposes a joint framework which extracts triggers and arguments together to alleviate the problem of error propagation caused by event triggers and arguments are predicted in isolation. Chen [27] proposes a dynamic multi-pooling convolutional neural network according to event triggers and arguments in order to reserve more crucial information for event detection. Zhao [40] first learns event detection oriented embedding of documents through a hierarchical and supervised attention based RNN, then further uses the learned document embedding to identify event triggers. Yan [41] uses a dependency tree based on graph convolutional network with aggregative attention to explicitly model and aggregate multi-order syntactic representations in sentences. Du [42] formulates the event extraction task as a question answering task that extracts the event arguments in an end-to-end manner. Li [43] casts the event extraction task into a series of reading comprehension problems, by which it extracts triggers and arguments successively from a given sentence. Yi [44] introduces a general framework for several event extraction tasks that share span representations using dynamic constructed span graph. The dynamic span graph refines the span representations by allowing the co-reference and relation type confidences to propagate through the graph. Different from their work, we used the topic information to enhance the sentence representation and further utilized the topic classification task as a facilitator for event detection task by having a multitask setup.

VII. CONCLUSION

In this study, we proposed a topic-aware event detection method by using the topic name embedding to enrich the contextual representations of the sentences along with the multi-task setup of event detection and topic classification task. We showed effectiveness of this method by testing on different datasets and conducting ablation studies. We explored different ways to generate topic embedding and different interaction methods between topic embeddings and sentence embeddings. A further analysis showed the topic-aware model architecture beats the non-topic-aware model with a large margin in a few-shot event type scenario. Furthermore, we analyzed the event type distribution based on topics which fundamentally explains why the sentence topic information can help the event detection task.

REFERENCES

[1] X. Wang, Z. Wang, X. Han, W. Jiang, R. Han, Z. Liu, J. Li, P. Li, Y. Lin, and J. Zhou, "MAVEN: A Massive General Domain Event Detection Dataset," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 1652–1671. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-main.129

[2] S. Ebner, P. Xia, R. Culkin, K. Rawlins, and B. Van Durme, "Multi-sentence argument linking," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 8057–8077. [Online]. Available: https://aclanthology.org/2020.acl-main.718

[3] M. Atkinson, J. Piskorski, H. Tanev, E. van der Goot, R. Yangarber, and V. Zavarella, "Automated event extraction in the domain of border security," in *International Conference on User Centric Media*. Springer, 2009, pp. 321–326. [Online]. Available: https://doi.org/10.1007/978-3-642-12630-7_40

[4] S. J. Conlon, A. S. Abrahams, and L. L. Simmons, "Terrorism information extraction from online reports," *Journal of Computer Information Systems*, vol. 55, no. 3, pp. 20–28, 2015. [Online]. Available: https://doi.org/10.1080/08874417.2015.11645768

[5] Y. Liang, K. Jabr, C. Grant, J. Irvine, and A. Halterman, "New techniques for coding political events across languages," in *2018 IEEE International Conference on Information Reuse and Integration (IRI)*. IEEE, 2018, pp. 88–93.

[6] M. Rospocher, M. van Erp, P. Vossen, A. Fokkens, I. Aldabe, G. Rigau, A. Soroa, T. Ploeger, and T. Bogaard, "Building event-centric knowledge graphs from news," *Journal of Web Semantics*, vol. 37, pp. 132–151, 2016. [Online]. Available: https://doi.org/10.1016/j.websem.2015.12.004

[7] Z. Li, X. Ding, and T. Liu, "Constructing narrative event evolutionary graph for script event prediction," p. 4201–4207, 2018. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/3304222.3304354

[8] A. K. Gunasekaran, M. B. Imani, L. Khan, C. Grant, P. T. Brandt, and J. S. Holmes, "Sperg: Scalable political event report geoparsing in big data," in *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2018, pp. 187–192.

[9] W. Nuij, V. Milea, F. Hogenboom, F. Frasincar, and U. Kaymak, "An automated framework for incorporating news into stock trading strategies," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 4, pp. 823–835, 2013. [Online]. Available: https://doi.org/10.1109/TKDE.2013.133

[10] P. Capet, T. Delavallade, T. Nakamura, A. Sandor, C. Tarsitano, and S. Voyatzi, "A risk assessment system with automatic extraction of event types," in *International Conference on Intelligent Information Processing*. Springer, 2008, pp. 220–229. [Online]. Available: https://doi.org/10.1007/978-0-387-87685-6_27

[11] E. Riloff, "Automatically constructing a dictionary for information extraction tasks," in *Proceedings of the Eleventh National Conference on Artificial Intelligence*, ser. AAAI'93. AAAI Press, 1993, p. 811–816. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/1867270.1867391

[12] P. Li, G. Zhou, Q. Zhu, and L. Hou, "Employing compositional semantics and discourse consistency in chinese event extraction," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012, pp. 1006–1016. [Online]. Available: https://www.aclweb.org/anthology/D12-1092

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[14] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751. [Online]. Available: http://dx.doi.org/10.3115/v1/D14-1181

[15] T. H. Nguyen, K. Cho, and R. Grishman, "Joint event extraction via recurrent neural networks," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 300–309. [Online]. Available: https://www.aclweb.org/anthology/N16-1034

[16] X. Liu, Z. Luo, and H. Huang, "Jointly multiple events extraction via attention-based graph information aggregation," pp. 1247–1256, Oct.-Nov. 2018. [Online]. Available: http://dx.doi.org/10.18653/v1/D18-1156

[17] J. Yan, N. Zalmout, Y. Liang, C. Grant, X. Ren, and X. L. Dong, "AdaTag: Multi-attribute value extraction from product profiles

with adaptive decoding," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 4694–4705. [Online]. Available: https://aclanthology.org/2021.acl-long.362

[18] Y. Ding, Y. Liang, N. Zalmout, X. Li, C. Grant, and T. Weninger, "Ask-and-verify: Span candidate generation and verification for attribute value extraction," 2022.

[19] R. Ghaeini, X. Fern, L. Huang, and P. Tadepalli, "Event nugget detection with forward-backward recurrent neural networks," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2016, pp. 369–373. [Online]. Available: http://dx.doi.org/10.18653/v1/P16-2060

[20] C. Walker, S. Strassel, J. Medero, and K. Maeda, "Ace 2005 multilingual training corpus," 2006.

[21] J. Ellis, J. Getman, and S. M. Strassel, "Overview of linguistic resources for the tac kbp 2014 evaluations: Planning, execution, and results," in *Proceedings of TAC KBP 2014 Workshop, National Institute of Standards and Technology*, 2014, pp. 17–18.

[22] J. Ellis, J. Getman, D. Fore, N. Kuster, Z. Song, A. Bies, and S. M. Strassel, "Overview of linguistic resources for the tac kbp 2015 evaluations: Methodologies and results." in *TAC*, 2015.

[23] A. Halterman, J. Irvine, M. Landis, P. Jalla, Y. Liang, C. Grant, and M. Solaimani, "Adaptive scalable pipelines for political event data generation," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 2879–2883.

[24] A. Halterman, J. Irvine, C. Grant, K. Jabr, and Y. Liang, "Creating an automated event data system for arabic text," in *ISA Annual Meeting San Francisco*, 2018.

[25] Y. Liang, "Scaling up labeling, mining, and inferencing on event extraction," 2022.

[26] "Ace english annotation guidelines for events," *Linguistic Data Consortium Philadelphia*, 2005. [Online]. Available: https://catalog.ldc.upenn.edu/LDC2006T06

[27] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, "Event extraction via dynamic multi-pooling convolutional neural networks," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 167–176. [Online]. Available: https://www.aclweb.org/anthology/P15-1017

[28] Y. Chen, H. Yang, K. Liu, J. Zhao, and Y. Jia, "Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 1267–1276. [Online]. Available: https://www.aclweb.org/anthology/D18-1158

[29] Y. Zeng, Y. Feng, R. Ma, Z. Wang, R. Yan, C. Shi, and D. Zhao, "Scale up event extraction learning via automatic training data generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/12030

[30] J. Devlin, M.-W. Chang, K. Lee, and K. N. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018. [Online]. Available: https://arxiv.org/abs/1810.04805

[31] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[32] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, p. 282–289. [Online]. Available: https://dl.acm.org/doi/10.5555/645530.655813

[33] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," 2015. [Online]. Available: https://arxiv.org/abs/1508.01991

[34] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997. [Online]. Available: https://doi.org/10.1023/A:1007379606734

[35] H. Martínez Alonso and B. Plank, "When is multitask learning effective? semantic sequence prediction under varying data conditions," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 44–53. [Online]. Available: https://www.aclweb.org/anthology/E17-1005

[36] Z. Yang, R. Salakhutdinov, and W. W. Cohen, "Transfer learning for sequence tagging with hierarchical recurrent networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=ByxpMd9lx

[37] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, no. null, p. 993–1022, mar 2003.

[38] H. Ji and R. Grishman, "Refining event extraction through cross-document inference," in *Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics, Jun. 2008, pp. 254–262. [Online]. Available: https://www.aclweb.org/anthology/P08-1030

[39] Q. Li, H. Ji, and L. Huang, "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)*, 2013, pp. 73–82.

[40] Y. Zhao, X. Jin, Y. Wang, and X. Cheng, "Document embedding enhanced event detection with hierarchical and supervised attention," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 414–419. [Online]. Available: https://www.aclweb.org/anthology/P18-2066

[41] H. Yan, X. Jin, X. Meng, J. Guo, and X. Cheng, "Event detection with multi-order graph convolution and aggregated attention," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 5766–5770. [Online]. Available: https://www.aclweb.org/anthology/D19-1582

[42] X. Du and C. Cardie, "Event extraction by answering (almost) natural questions," *arXiv preprint arXiv:2004.13625*, 2020.

[43] F. Li, W. Peng, Y. Chen, Q. Wang, L. Pan, Y. Lyu, and Y. Zhu, "Event extraction as multi-turn question answering," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, 2020, pp. 829–838.

[44] Y. Luan, D. Wadden, L. He, A. Shah, M. Ostendorf, and H. Hajishirzi, "A general framework for information extraction using dynamic span graphs," *arXiv preprint arXiv:1904.03296*, 2019.