TAED: Topic-Aware Event Detection

Anonymous authors

Abstract

Identifying event trigger words and classifying event types known as event detection task is the most important and fundamental step for extracting event-related knowledge from raw text. However, the topic information of the document where the events are extracted from is rarely explored. Examples of the topics of the documents can be military conflict, earthquake, concert tour, wrestling etc. Topic information is important event detection, since event type distribution could be different from topic to topic. Semantically similar topics share similar event types while event types are quite different between distinguishable topics. In this paper, we explored methods of how to use topic information of the document in event detection. To the best of our knowledge, we are the first one to explore how to use the structured topic knowledge in the event detection task. We summarized our contribution as the following: We use the topic information as prior knowledge in a novel multi-task neural network to enrich the sentence contextual representations. We evaluated our method with MAVEN dataset [Wang et al., 2020] which is designed for event detection task with more general event types introduced, and showed that our topic-aware model performs better than the baseline model by +1.8% on the $F_1$ score on a topic label balanced dataset.

1. Introduction

Event detection is an important task of information retrieval in the natural language processing, which has lots of applications in different domains. For example, event detection and monitoring have long been the focus of public affair management for governments, as timely knowing the outbursts and evolution of popular social events helps the authorities to respond promptly [Conlon et al., 2015, Atkinson et al., 2009]. The structured events can be directly used in constructing or expanding knowledge base [Rosspicher et al., 2016, Li et al., 2018] In the business and financial domain, event detection can also help companies quickly discover market responses of their products and influencing signals for risks analysis and suggestions [Nuij et al., 2013, Capet et al., 2008].

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. As in Figure 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 use pattern matching technique [Riloff, 1993]. Later on, people start to use traditional machine learning algorithms other than neural networks to address the event detection problem. Techniques include using support vector machine (SVM) [Li et al., 2012]. More recently, deep learning has been successfully applied to different NLP tasks, and the same story applies to event detection. The general process to build a neural network is to take word embeddings as input and output a classification result for each word. Convolutional Neural Network (CNN) [Kim, 2014], Recurrent Neural Network (RNN) [Nguyen et al., 2016, Ghaeini et al., 2016], and Graph Neural Network (GNN) [Liu et al., 2018] have been all explored and applied to event detection task.

However, none of the previous efforts [Kim, 2014, Nguyen et al., 2016, Ghaeini et al., 2016, Liu et al., 2018] take event topic information into consideration for the event detection task. Examples of topics that the documents belong to include be “Terrorist Attack”, “Horse Race”, “Earthquake” etc as shown in Table 1. Topic information is important for event detection tasks, since intuitively, documents belonging to different topics have different event type distributions. In order to validate this assumption, we did analysis with MAVEN dataset [Wang et al., 2020]. We chose MAVEN since (1) it has a large range of event topics, and (2) the dataset itself comes with the topic labels. First of all, we gather all the 168 event types in MAVEN. Then, for each topic, we normalized the event type occurrence count to a 168 dimension vector, of which all the 168 elements in the vector summarize to 1. We use 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 P-Value is bigger than 0.05, it means we can’t reject the null hypothesis that the two topics follow the same event type distribution. When P-Value is less than or equal to 0.05, it means we can accept the null hypothesis, i.e., the two topics probably follow the same event type distribution.

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 recurring is most similar to rail accident
Figure 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 5, Appendix F)

and music festival topics with corresponding P-Value 0.52 and 0.14, respectively. In Table 1, we show the top-5 event types for 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 event detection task in order to improve its performance. To the best of our knowledge, we are the first one to explore how to use topic information in event detection task. We summarize our contributions as the following: (1) We performed detailed analysis explaining why topic information can help on event detection task. (2) We introduced topic name enhanced sentence representation for event detection and explored the effectiveness of using topic keywords instead of topic name. (3) Furthermore, we introduced topic classification and event detection as a multi-task learning setup, which further improved the performance. The rest of this paper is organized as follows. Section 2 gives the definition of NLP event detection task. Section 3 described the system encoders, representations, and decoders used for event detection training and

<table>
<thead>
<tr>
<th>event types</th>
<th>catastrophe</th>
<th>causation</th>
<th>damaging</th>
<th>coming to be</th>
<th>destroying</th>
</tr>
</thead>
<tbody>
<tr>
<td>earthquake topic distribution</td>
<td>0.255</td>
<td>0.076</td>
<td>0.072</td>
<td>0.043</td>
<td>0.033</td>
</tr>
<tr>
<td>horse race topic distribution</td>
<td>0.201</td>
<td>0.104</td>
<td>0.058</td>
<td>0.047</td>
<td>0.036</td>
</tr>
<tr>
<td>terrorist attack topic distribution</td>
<td>0.145</td>
<td>0.074</td>
<td>0.058</td>
<td>0.049</td>
<td>0.035</td>
</tr>
</tbody>
</table>

Table 1: An example of event type distribution (top 5) for three topics: Earthquake, Horse Race, and Terrorist Attack.
topic classification training. Section 4 describes experimental results and Section 6 discusses related work. Finally, Section 7 concludes and suggests future work.

2. 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 [ACE, 2005]. 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 [Chen et al., 2015, Nguyen et al., 2016] or a sequence labeling task [Chen et al., 2018], and only report the trigger classification results. [Wang et al., 2020, Zeng et al., 2018] 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.

3. Methodology

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

3.1 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 [Devlin et al., 2018] whose underlying layers use the self-attention mechanism to mitigate the long-range dependencies issue,

$$h_1, h_2, ..., h_T = Encoder(x_1, x_2, ..., x_T)$$

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

4
3.2 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 tokens in Figure 3) returned from BERT encoder to represent the entire information carried by the topic keywords.

\[ h_{\text{topic}} = Encoder(\text{topicword}_1, \ldots, \text{topicword}_N), \] (2)

where \( h_{\text{topic}} \in \mathbb{R}^d \)

3.3 Topic-Aware Sentence Representation

To associate the sentence representation with its document’s topic, we append the topic vector representation \( h_{\text{topic}} \) to each token vector representation \( (h_1, h_2, \ldots, h_T) \) in the sentence,
shown in Figure 3. Then we get the topic-aware contextualized vector representations of the sentence tokens as

\[ \tilde{h} = (\tilde{h}_1, ..., \tilde{h}_T) = (h_1; h_{\text{topic}}, ..., h_T; h_{\text{topic}}) \]  

(3)

where \( h_{\text{new}} \in \mathbb{R}^{2d} \) and ; operator represents concatenation.

### 3.4 Event Detection CRF Decoder

We feed the topic-aware contextualized token representations \((\tilde{h}_1, \tilde{h}_2, ..., \tilde{h}_T)\) to CRFs [Lafferty et al., 2001] to get the sequence of BIOE tags with the highest probability:

\[ (y_1, y_2, ..., y_T) = \text{CRF}(\tilde{h}_1, \tilde{h}_2, ..., \tilde{h}_T), \]  

(4)

CRF decoder [Huang et al., 2015] can enforce the tagging consistency that captures dependency between the output tags. 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:

\[
\text{score}(X, \text{topic}, Y) = \sum_{i=1}^{T-1} T_{y_i, y_{i+1}} + \sum_{i=1}^{T} E_{i, y_i},
\]  

(5)

where \( T \in \mathbb{R}^{m \times m} \) is the transition matrix, \( T_{ij} \) is the transition score of i-th tag to the j-th tag. \( E \in \mathbb{R}^{T \times m} \), \( E_{ij} \) represents the i-th token is assigned j-th tag in the tagset. \( m \) is the number of tags in the tagset which includes B, I, E tag for each event type, for example: B-Killing, I-Killing, E-Killing and an O tag.

### 3.5 Event Detection Training

The event detection task is trained to maximize the log likelihood of \((X, \text{topic}, Y)\) triplets in the training set, the score of given tokens, and topic that has predicted tags \( Y \) is given in equation (5), and the log likelihood to maximize is defined as:

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

(6)

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

\[
\text{Loss}_{\text{event extraction}} = -\sum_{i=1}^{N} \log p(\hat{Y}_i | X_i, \text{topic}_i),
\]  

(7)

where \( \hat{Y}_i \) is the ground truth label for sentence \( i \).
3.6 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 \( \hat{h} \) from equation (3) to classify the topics, we directly use the \([\text{CLS}]\) token representation denoted as \( h \) from the sentence encoder (the purple token in Figure 3) to classify the topic.

\[
(p_1, ..., p_{|C|}) = \text{softmax}(W_t h + b_t)
\]

\[
\text{Loss}_{\text{topic}} = -\sum_{j=1}^{N} \sum_{i=1}^{|C|} y_{ij} \log(p_{ij}),
\]

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.

3.7 Multi-Task Training

We jointly train TAED for event detection and Topic Classification in a multi-task learning setting [Caruana, 1997, Martínez Alonso and Plank, 2017, Yang et al., 2017] by combining the loss of the two tasks:

\[
\text{Loss} = \text{Loss}_{\text{event extraction}} + \gamma \cdot \text{Loss}_{\text{topic}}
\]

where \( \gamma \) is a non-negative tunable 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.

4. Experiments

In order to validate our hypothesis that the event topic information can help event detection, we used the MAVEN [Wang et al., 2020] dataset which has a large range of event topics and also comes with the topic labels to conduct our experiments. The 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 first combined the training and validation dataset, then separated the merged dataset further into a 70%/15%/15% distribution on the documents level. This gave us the distribution of the documents in our training/validation/test dataset as: 2913/710/857, and the corresponding number of sentences as 32431/8042/9400 (since each document might has multiple sentences). The topic labels come from MAVEN dataset which are annotated by humans.

4.1 Ablation Study

BiLSTM-CRF is using bidirectional LSTMs as the encoder and CRF as the decoder for the task [Huang et al., 2015]. BERT-CRF is using BERT as the encoder and CRF as the decoder. BERT-CRF-TOPIC is our TAEE architecture as shown in Figure 3. The topic-classification-weight is the weight set on the topic classification task when setting the event detection task weight as 1.
**Topic Name Encoding.** As shown in Table 2, 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 of the entity level micro $F_1$, which is a commonly adopted metric for event detection task [Wang et al., 2020], by 0.76% from 63.66% to 64.42%.

<table>
<thead>
<tr>
<th>Model Type</th>
<th>topic-classification weight</th>
<th>general event word removed</th>
<th>P(%)</th>
<th>R(%)</th>
<th>$F_1$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiLSTM-CRF</td>
<td>NA</td>
<td>NA</td>
<td>88.60</td>
<td>47.44</td>
<td>61.79</td>
</tr>
<tr>
<td>BERT-CRF</td>
<td>NA</td>
<td>NA</td>
<td>66.91</td>
<td>60.71</td>
<td>64.66</td>
</tr>
<tr>
<td>BERT-CRF-TOPIC</td>
<td>1</td>
<td>True</td>
<td>64.44</td>
<td>66.52</td>
<td><strong>65.46</strong></td>
</tr>
<tr>
<td>BERT-CRF-TOPIC</td>
<td>0.1</td>
<td>True</td>
<td>62.37</td>
<td>66.66</td>
<td>64.44</td>
</tr>
<tr>
<td>BERT-CRF-TOPIC</td>
<td>0.5</td>
<td>True</td>
<td>64.17</td>
<td>64.92</td>
<td>64.53</td>
</tr>
<tr>
<td>BERT-CRF-TOPIC</td>
<td>2</td>
<td>True</td>
<td>63.76</td>
<td>65.33</td>
<td>64.51</td>
</tr>
<tr>
<td>BERT-CRF-TOPIC</td>
<td>10</td>
<td>True</td>
<td>64.26</td>
<td>58.73</td>
<td>61.38</td>
</tr>
<tr>
<td>BERT-CRF-TOPIC</td>
<td>25</td>
<td>True</td>
<td>63.8</td>
<td>47.34</td>
<td>54.33</td>
</tr>
<tr>
<td>BERT-CRF-TOPIC</td>
<td>50</td>
<td>True</td>
<td>60.36</td>
<td>33.06</td>
<td>42.65</td>
</tr>
<tr>
<td>BERT-CRF-TOPIC</td>
<td>75</td>
<td>True</td>
<td>55.98</td>
<td>25.69</td>
<td>34.99</td>
</tr>
<tr>
<td>BERT-CRF-TOPIC</td>
<td>100</td>
<td>True</td>
<td>49.58</td>
<td>19.38</td>
<td>27.79</td>
</tr>
<tr>
<td>BERT-CRF-TOPIC</td>
<td>1</td>
<td>False</td>
<td>65.59</td>
<td>64.29</td>
<td>64.93</td>
</tr>
<tr>
<td>BERT-CRF-TOPIC (w/ vocab)</td>
<td>1</td>
<td>True</td>
<td>64.97</td>
<td>65.08</td>
<td>65.02</td>
</tr>
</tbody>
</table>

Table 2: 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.

**Topic Name Variations.** The column “general event word removed” shown in Table 2 indicates whether 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”, “historical”, “wrestling” etc. This is going to help make our topic embedding more discriminated from each other. Based on the performance we can see that the performance gets enhanced by adding this pre-processing step for the topic name. 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 3.

After adding the keywords to the topic names, the performance got a little bit worse though, 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.

**Multi-Task Learning.** We have conducted experiments by using different weights on topic classification task, where $\gamma$ ranges from 0 to 100. 0 means we ignore the topic classification loss during backpropagation. We saw that the sweet spot to achieve the best performance is to set the classification weight as 1 shown in Figure 4.

By setting equal loss weight on event detection and topic classification tasks, we further improved the $F_1$ score by another 1.04% on top of the topic name embedding contribution. Altogether TAED showed improvement of the task on $F_1$ score by 1.8%.
<table>
<thead>
<tr>
<th>Topic</th>
<th>Topic Vocabulary</th>
</tr>
</thead>
<tbody>
<tr>
<td>earthquake</td>
<td>magnitude, occurred, quake, intensity, damage</td>
</tr>
<tr>
<td>winter storm</td>
<td>snow, blizzard, snowfall, new, winds</td>
</tr>
<tr>
<td>tennis event</td>
<td>open, doubles, slam, singles, djokovic</td>
</tr>
<tr>
<td>rugby match</td>
<td>chiefs, brumbies, sharks, final, crusaders</td>
</tr>
<tr>
<td>university boat race</td>
<td>oxford, cambridge, lengths, crews, goldie</td>
</tr>
<tr>
<td>war</td>
<td>paulo, vargas, 1930, presets, garais</td>
</tr>
<tr>
<td>military operation</td>
<td>bomb, manchester, ira, bombing, embassy</td>
</tr>
<tr>
<td>swimming event</td>
<td>golds, medals, bronze, silver, freestyle pool</td>
</tr>
<tr>
<td>cricket series</td>
<td>ashes, england, australia, test, wickets</td>
</tr>
<tr>
<td>civilian attack</td>
<td>massacre, attack, kill, police, people</td>
</tr>
</tbody>
</table>

Table 3: Sample of 10 topic vocabulary terms and top-5 representative keywords.

Figure 4: $F_1$ performance vs. $\gamma$ (Each $\gamma$ 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 $\gamma$.)

5. Result Analysis

The performance showed in Table 2 combines both the performance of trigger identification and trigger classification. We further get the performance only for trigger identification shown in table 7. From which we can see that the topic-aware event detection model get better performance on both trigger identification and trigger classification. The error cases for event identification can come from two sources: 1. We failed to identify triggers. 2. We identify the triggers correctly, 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, motionso". The gold values for the triggers, "accident" and "incident", are both
"B-catastrophe". The non topic-aware model failed to identify the triggers at the first place, while the topic-aware model identifies the triggers and classifies them correctly into a catastrophe event. "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". The topic-aware model predicted "played" as "B-competition" while non topic-aware model predicted it as "B-participation". The gold value 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.

6. Related work

Ji and Grishman [2008] employs an approach to propagate consistent event arguments across sentences and documents. By combining global evidences from related documents and local decisions, a cross-document method is created to improve event detection task. [Ji and Grishman, 2008] 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 et al., 2015] 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 et al., 2018] 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 et al., 2019] uses a dependency tree based on graph convolutional network with aggregative attention to explicitly model and aggregate multi-order syntactic representations in sentences. 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.

7. 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 conducting ablation studies. We further used different ways to encode the topic information by removing general words in the topic name and adding topic keywords into topic name. Furthermore, we analyzed the event type distribution prior to topics which fundamentally explains why the sentence topic information can help with the event detection task. In future efforts, we want to generate the topic name information from the raw text as an end-to-end task along with event detection task, instead of using the topic labels from human annotations or semi-supervised generated labels from the MAVEN dataset.
References


Ellen Riloff. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, AAAI’93, page


Figure 5: 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)
Appendix B. Performance of BERT-CRF and BERT-CRF-TOPIC on two more random splits of MAVEN

We supplemented the existing experiments by adding two more random splits. We sampled data to maintain the topic label balance. In doing so, we see a similar trend that the topic-aware model surpassed the performance of the baseline.

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Split</th>
<th>P(%)</th>
<th>R(%)</th>
<th>F₁(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-CRF</td>
<td>split 1</td>
<td>63.7</td>
<td>65.66</td>
<td>64.67</td>
</tr>
<tr>
<td>BERT-CRF-TOPIC</td>
<td>split 1</td>
<td>64.3</td>
<td>65.66</td>
<td>65.4</td>
</tr>
<tr>
<td>BERT-CRF</td>
<td>split 2</td>
<td>61.68</td>
<td>68.02</td>
<td>64.7</td>
</tr>
<tr>
<td>BERT-CRF-TOPIC</td>
<td>split 2</td>
<td>64.3</td>
<td>66.61</td>
<td>65.44</td>
</tr>
</tbody>
</table>

Table 4: Performance of BERT-CRF and BERT-CRF-TOPIC on two more random splits of MAVEN maintaining the topic label balance.

Appendix C. Performance of BERT-CRF and BERT-CRF-TOPIC on MAVEN

We found that the topic-aware model get similar performance as the non topic-aware model, which could be caused by very skewed topic distribution in MAVEN dataset. Our current model has limitation to address the skewed topic distribution data.

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Dataset</th>
<th>P(%)</th>
<th>R(%)</th>
<th>F₁(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-CRF</td>
<td>Full MAVEN</td>
<td>65</td>
<td>70.9</td>
<td>67.8</td>
</tr>
<tr>
<td>BERT-CRF-TOPIC</td>
<td>Full MAVEN</td>
<td>65.66</td>
<td>70.28</td>
<td>67.89</td>
</tr>
</tbody>
</table>

Table 5: Performance of BERT-CRF and BERT-CRF-TOPIC on MAVEN.

Appendix D. Performance of using different ways to generate topic name embedding

We supplemented the existing experiments by exploring different ways to get the topic embedding, which includes using freeze/unfreeze version of the [CLS] embedding, freeze/unfreeze version of the average topic name token embedding. We see similar performances across different variations.

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Topic Embedding Type</th>
<th>P(%)</th>
<th>R(%)</th>
<th>F₁(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-CRF-TOPIC</td>
<td>[CLS]</td>
<td>67</td>
<td>63.78</td>
<td>65.35</td>
</tr>
<tr>
<td>BERT-CRF-TOPIC</td>
<td>Average Token Embedding</td>
<td>64.53</td>
<td>66.44</td>
<td>65.47</td>
</tr>
<tr>
<td>BERT-CRF-TOPIC</td>
<td>[CLS] freeze</td>
<td>64.57</td>
<td>66.57</td>
<td>65.56</td>
</tr>
<tr>
<td>BERT-CRF-TOPIC</td>
<td>Average Token Embedding freeze</td>
<td>65</td>
<td>65.64</td>
<td>65.32</td>
</tr>
</tbody>
</table>

Table 6: Performance of using different ways to generate topic name embedding.
Appendix E. Hyperparameter Settings

The hyperparameters used for the experiments are shown as the following, Learning Rate: \(5 \times 10^{-5}\), Bert Model: bert-base-cased, Adam Epsilon: \(1 \times 10^{-8}\), Dropout Rate: 0.1, Batch Size: 16.

Appendix F. Performance of BERT-CRF and BERT-CRF-TOPIC only on Trigger Identification

<table>
<thead>
<tr>
<th>Model Type</th>
<th>P(%)</th>
<th>R(%)</th>
<th>F₁(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-CRF</td>
<td>77.3</td>
<td>77.9</td>
<td>77.6</td>
</tr>
<tr>
<td>BERT-CRF-TOPIC</td>
<td>77.93</td>
<td>78.59</td>
<td>78.26</td>
</tr>
</tbody>
</table>

Table 7: Performance of BERT-CRF and BERT-CRF-TOPIC only on Trigger Identification