Article:

Reinforcement Learning-based Dialogue Guided Event Extraction to Exploit Argument Relations

Qian Li, Hao Peng, Jianxin Li, *Member, IEEE*, Jia Wu, *Senior Member, IEEE*, Yuanxing Ning, Lihong Wang, Philip S. Yu, *Fellow, IEEE*, Zheng Wang, *Member, IEEE*

Abstract—Event extraction is a fundamental task for natural language processing. Finding the roles of event arguments like event participants is essential for event extraction. However, doing so for real-life event descriptions is challenging because an argument's role often varies in different contexts. While the relationship and interactions between multiple arguments are useful for settling the argument roles, such information is largely ignored by existing approaches. This paper presents a better approach for event extraction by explicitly utilizing the relationships of event arguments. We achieve this through a carefully designed task-oriented dialogue system. To model the argument relation, we employ reinforcement learning and incremental learning to extract multiple arguments via a multi-turned, iterative process. Our approach leverages knowledge of the already extracted arguments of the same sentence to determine the role of arguments that would be difficult to decide individually. It then uses the newly obtained information to improve the decisions of previously extracted arguments. This two-way feedback process allows us to exploit the argument relations to effectively settle argument roles, leading to better sentence understanding and event extraction. Experimental results show that our approach consistently outperforms seven state-of-the-art event extraction methods for the classification of events and argument role and argument identification.

Index Terms—Event extraction, reinforcement learning, incremental learning, multi-turned.

I. INTRODUCTION

E VENT extraction aims to detect, from the text, the occurrence of events of specific types and to extract arguments (e.g., typed event participants or other attributes) that are associated with an event [1]. It is a fundamental technique underpinning many Natural Language Processing

Qian Li, Hao Peng, Jianxin Li and Yuanxing Ning are with Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing 100191, China. E-mail: {liqian, penghao, lijx, ningyx}@act.buaa.edu.cn.

Jia Wu is with the Department of Computing, Macquarie University, Sydney, Australia. E-mail: jia.wu@mq.edu.au.

Lihong Wang is with the National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China. Email: wlh@isc.org.cn.

Philip S. Yu is with the Department of Computer Science, University of Illinois at Chicago, Chicago 60607, USA. E-mail: psyu@uic.edu.

Zheng Wang is with the School of Computing, University of Leeds, Leeds LS2 9JT, UK. E-mail: z.wang5@leeds.ac.uk.
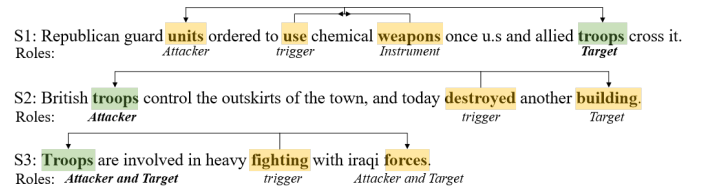
S1: Republican guard **units** ordered to **use** chemical **weapons** once u.s and allied **troops** cross it.
Roles: *Attacker*   *trigger*   *Instrument*   ***Target***

S2: British **troops** control the outskirts of the town, and today **destroyed** another **building**.
Roles: ***Attacker***   *trigger*   *Target*

S3: **Troops** are involved in heavy **fighting** with iraqi **forces**.
Roles: ***Attacker and Target***   *trigger*   *Attacker and Target*

Fig. 1: Three example sentences belong to the "Conflict: Attack" event type from the ACE 2005 dataset.

(NLP) tasks like knowledge reasoning [2], text summarization [3], and event prediction [4].

Event extraction requires extracting all arguments and their roles corresponding to each event. Doing so is challenging because an event is often associated with more than one argument, whose role can vary in different contexts. For example, the argument "troops" has different roles in multiple sentences, as shown in Fig. 1. This argument has the role of being the "*target*" in sentence $S1$, while in sentence $S2$, its role is the "*attacker*". In sentence $S3$, the argument "troops" could be either the "*target*" or the "*attacker*". To extract an event, we need to identify the argument role correctly. Failing to do so can lead to erroneous information propagation, affecting the recognition of other event arguments and sentence understanding. For example, incorrectly associating the "troops" argument in the sentence $S2$ as the *target* will lead to misunderstanding of the sentence. Unfortunately, argument role detection remains an open problem because an argument can be associated with multiple roles.

Our work aims to find new ways to identify event argument roles for event extraction. **Our key insight is that multiple arguments associated with an event are typically strongly correlated**. Their correlations can provide useful information for determining the role of an event argument. Consider again our example given earlier in Fig. 1. To determine the role of argument "troops" in the sentence $S1$, we can consider its relevant arguments of "weapons" and "use". The roles and appearance order of "weapons" and "use" suggest "troops" is the target in the context. Although the argument "use" has many subtle roles, we can still attribute it with the *"Conflict: Attack"* event type in sentence $S1$ by using "weapons" as a hint. Moreover, the role of argument "troops" can be recognized to role by looking at the argument "weapons" and "use". If the "troops" is detected firstly, it may misidentify its argument role and other arguments. As can be seen from this representative example, the relation among arguments can help in inferring the argument roles that are essential for event extraction. However, prior work largely ignores such relations, leaving
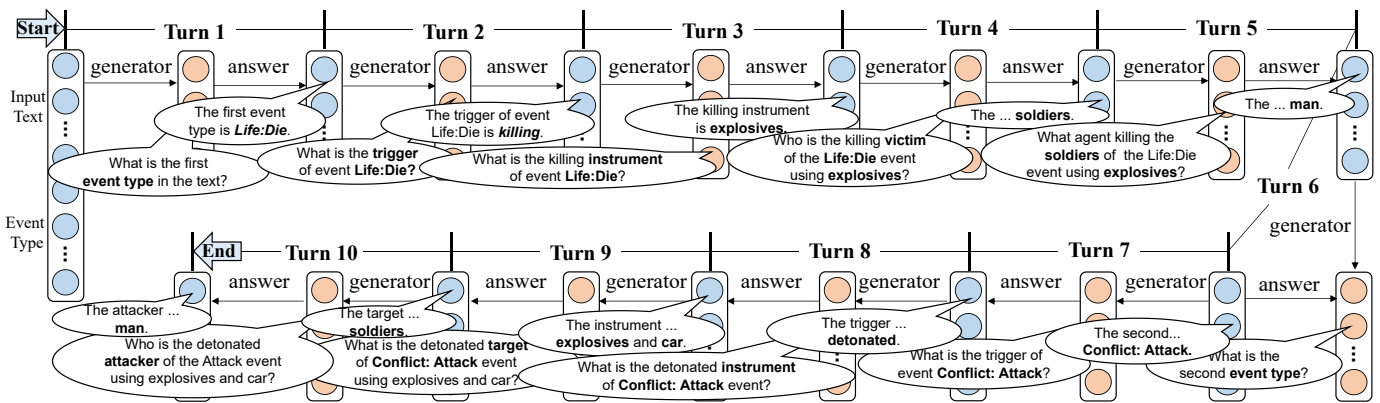
Fig. 2: An example of our dialogue guided event extraction. The input sentence is: "As the soldiers approached, the man detonated explosives in the car killing all four of the soldiers". It needs 10 turns to complete event extraction on the sentence.

much room for improvement. Most existing methods extract all arguments simultaneously [5, 6] or individual arguments sequentially [7], all of which do not consider the effect of argument extraction order. Our work seeks to close this gap by explicitly modeling the argument relation for event extraction.

Our approach is enabled by the recent advance in task-oriented dialogue systems [8, 9] that are shown to be highly effective in entity-relation extraction [10]. A task-oriented dialogue system uses domain knowledge, e.g., knowledge structures of intentions extracted from sentences, to complete a specific task. The natural language understanding task is structured as many slots are to be filled, where each slot can take a set of possible values. For example, a travel query could be translated into a structure consisting of slots like *original_city*, *destination_city*, *departure_time* and *arrival_time*. The goal of the dialogue system in this context is thus to extract the right values from the user sentence to fill the slots. The recent progress in a task-oriented dialogue system allows one to effectively exploit the dialogue historical information to optimize slot filling with the right order [11–13].

In this work, we formulate the event extraction task within a task-oriented dialogue system, as shown in Fig. 2. We consider the problem of event extraction as filling slots of relevant arguments and their roles extracted from the input sentence. To this end, we develop a multi-turn dialogue system [9] with two agents to iteratively solve the slot filling problem. During each turn, one agent selects an argument role and generates a query through a dialogue generator. For example, the query for the role of "instrument" could be "What is the killing instrument of event *Life: Die*?" in Turn 3 of Fig. 2. The other agent answers the query by identifying the right argument or event type from the sentence. This iterative generation and answering paradigm enable us to introduce knowledge obtained from previous turns when extracting a current argument. For instance, it can exploit the argument relation like "weapon", "use" and "troops" in sentence $S1$ in Fig. 1 to improve the quality of event extraction. Our dialogue-based method extracts the argument of victim *"soldiers"* according to the argument of trigger *killing* and instrument *explosives* for event *Life: Die* in Fig. 2. This multi-turn process also enables us to leverage additional information about the newly extracted

argument to update and correct the argument roles identified in the previous turn. As we repeat the process, we will obtain more information about event arguments and better understand the sentence over time. This richer information helps us extract argument roles more precisely towards the end of the process.

While our multi-turn dialogue system provides a potentially powerful event extraction capability, its potential can only be fully unlocked if the arguments are processed in the right order. Since we extract arguments and determine their roles in sequential order by utilizing the knowledge obtained from previously extracted arguments, the order of argument extraction is crucial. Ideally, we would like to start from event arguments whose argument roles are likely to be accurately decided using already extracted information and leave the more challenging ones later once we have obtained sufficient information from others. For instance, we may wish to extract argument "weapon" before "use" of sentence $S1$ given in Fig. 1 because determining the former's role is more straightforward than doing that for the latter.

We address the challenge of argument extraction order by employing Reinforcement Learning (RL) to rank arguments to best utilize the argument relation. To allow RL to navigate the potentially large problem space, we need to find the right representation of each word in the target sentence and use the representation to predict the start and end position of each argument. To that end, **we use both a lexicon-based graph attention network [14] and an event-based BERT model [15] for learning the word representation from semantics and context two perspectives**. We then utilize the learned representation to determine which argument to extract and in what order. We go further by designing an incremental learning strategy to iteratively incorporate the argument relation into the multi-turned event extraction process by continually updating the event representation across turns. By doing so, the representation becomes increasingly more accurate as the argument extraction process proceeds, which, in turn, enhances the quality of the resulting argument and event extraction.

We evaluate our approach[1] by applying it to sentence-level

---

[1]Code and data are available at: https://github.com/xiaoqian19940510/TASLP-EAREE

event extraction performed on the ACE 2005 dataset [16]. We compare our approach to 7 recently proposed event extraction approaches [5, 6, 17–19, 7, 20]. Experimental results show that our approach can effectively utilize the argument relation to identify the argument roles, leading to better event extraction performance. We show that our incremental event learning strategy is particularly useful when the amount of labeled data is limited.

This paper makes the following contributions. It is the first to:

- develop a multi-turned, task-oriented dialogue guided event extraction framework aiming to fill arguments extracted from input text for specific arguments roles (Section III);
- employ reinforcement learning to rank argument extraction order to utilize argument correlation for event extraction (Section III-B);
- leverage a lexicon-based graph attention network and event-based BERT, under an incremental learning framework to learn word representation for even extraction (Sections III-A);

II. RELATED WORK

Event extraction [21–23] is a form of information representation to extract what users are interested in massive data and present it to users in a certain way. For event extraction task, it can be divided into four subtasks: event classification, trigger identification, argument identification, and argument role classification. Most recent event extraction works are based on a neural network architecture like Convolutional Neural Network (CNN) [24, 25], Recurrent Neural Network (RNN) [26, 5], Graph Neural Network (GNN) [6, 27], Transformer [19, 28], or other networks [29, 30]. The method of event extraction based on deep learning first adopts pipeline. The pipeline-based method [24, 19, 31] is the earliest event extraction based on neural networks and extracts event arguments by utilizing triggers. It realizes event trigger identification, event classification, event argument identification, and argument role classification tasks successively [32] and takes the results of previous tasks as prior knowledge. The first two tasks are usually called event detection and the last two tasks are called argument extraction. Chen et al. [24] and Nguyen et al. [26, 33] use the CNN model to capture sentence-level clues and overcome complex feature engineering compared with traditional feature-based approaches. DBRNN [5] have been proved to be influential in introducing graph information into event extraction tasks. JMEE [6] is proposed with an attention-based GCN, learning syntactic contextual node representations through first-order neighbors of the graph. As we all know, an argument usually plays different roles in different events, enhancing the difficulty of the event extraction task. Yang et al. [19] propose a pre-trained language model-based event extractor [34] to learn contextualized representations proven helpful for event extraction. It separates argument predictions according to the roles and overcomes the argument overlap problem. However, this requires a high accuracy of trigger identification. A wrong trigger will seriously affect the accuracy of argument identification and argument role classification. Therefore, the pipeline based method considers the event trigger as the core of an event [5, 35, 36] and requires high accuracy of event trigger identification, avoiding an adverse effect on event argument extraction. In order to overcome the propagation of error information caused by event detection, joint-based event extraction methods are proposed. The joint event extraction method avoids the influence of trigger identification error on event argument extraction. It reduces the propagation of error information by combining trigger identification and argument extraction tasks.

The existing event extraction corpus has a few labeled data and hard to expand, such as ACE 2005 with only 599 annotated documents [16]. Existing deep learning-based approaches usually require lots of manually annotated training data. Consequently, except for the difficulty of event extraction itself, inadequate training data also hinders. The zero-shot learning method is the right choice, which has been widely applied in NLP tasks [37]. Based on this, Huang et al. [38] design a transferable neural architecture and stipulate a graph structure to transfer knowledge from the existing types to the extraction of unseen types. It finds the event types graph structure, which learns representation almost matching representations from the parsed AMR structure [39]. Existing event extraction systems, which usually adopt a supervised learning paradigm, have to rely on labelled training data, but the scarcity of high-quality training data is a common problem [40, 31].

Machine Reading Comprehension (MRC) tasks extract a span from text [41], is a basic task of question answering. MRC based event extraction [28, 7] designs questions for each argument, helps capture semantic relationship between question and input sentence. Question answering methods are emerging as a new way for extracting important keywords from a sentence [28, 7]. By incorporating domain-knowledge to the question set, one can guide the extraction framework to focus on essential semantics to be extracted from a sentence. Existing approaches do not utilize the relations among multiple arguments, leaving much room for improvement. Our work aims to close this gap by using the argument relations to infer roles of arguments that are hard to settle in isolation, leading to better performance for argument and event classification.

Task-oriented dialogue systems aim to assist the user to complete specific tasks according to existing corpora [9, 42]. The typical task-oriented dialogue system has four components: natural language understanding, dialogue state tracker, dialogue policy learning, and natural language generation [43]. Ramadan et al. [44] introduce an approach utilizing semantic correlation in ontology terminologies and dialogue utterances. It easily utilizes history knowledge for current dialogue [8, 9], aiming to build the connection among dialogue. Therefore, there have been some explorations on formulating NLP tasks as a dialogue task. It can overcome the inadequacy problem of historical information utilization in MRC. Therefore, there have been some explorations on formulating NLP tasks as a dialogue task. We treat event extraction as a dialogue for capturing relationships among arguments. Our work leverages the recent development in task-oriented dialogue systems [9, 42]. Such a system decouples the problem to be solved
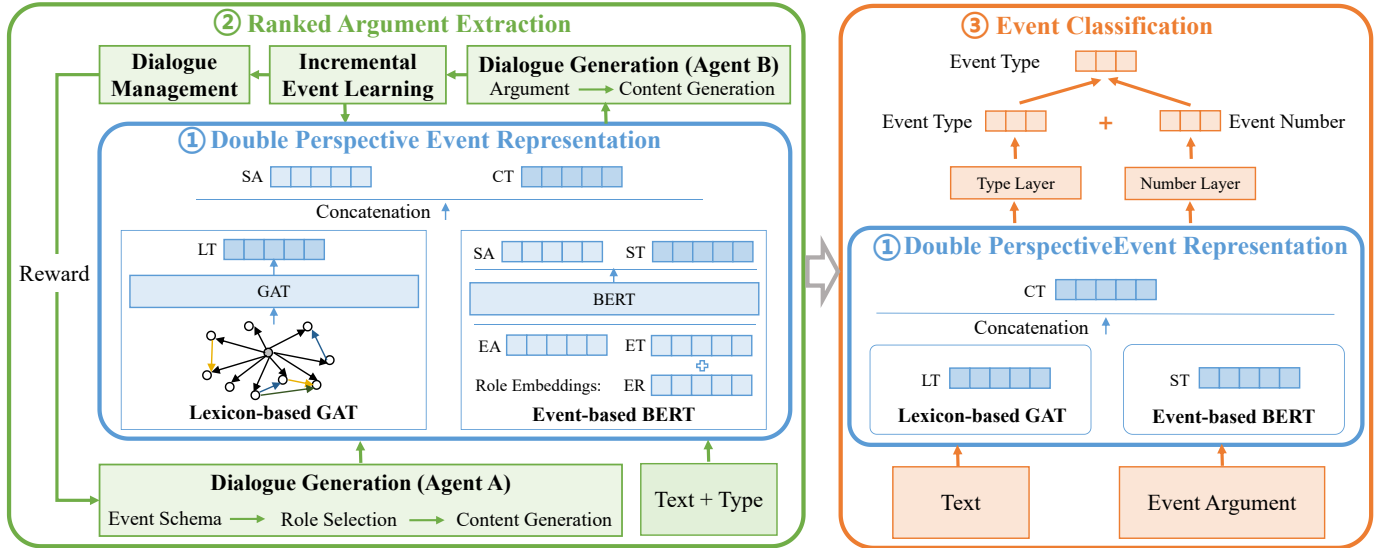
Fig. 3: Overview of our event extraction framework. The training flow is: ② → ③; the testing flow during deployment is: ③ → ② → ③. LT is the lexicon-based word representation. EA, ER, and ET are the dialogue content embedding of *Agent A*, argument role embedding, and event type embedding. SA and ST are the dialogue content representation of *Agent A* and representation of ET after BERT. CT is the final representation of input text.

through a multi-turn dialogue, allowing the system to connect and exploit information obtained during multiple conversations to solve a new task [8, 9].

III. EVENT EXTRACTION FRAMEWORK

Fig. 3 gives an overview of our framework that consists of three components for *(1) double perspective event representation, (2) ranked argument extraction, and (3) event classification*. The argument extraction module automatically generates the dialogue based on the event type and the selected predicted arguments. The selected arguments are produced by incremental event learning method to add the pseudo label as the training data and append the pseudo relation to the lexicon-based graph. The pseudo label is the arguments predicted by our ranked argument extraction model. The pseudo relation is the argument role of the predicted argument.

We add a pseudo edge by appending an edge among predicted arguments of an event type to update word representation using existing prediction results. It takes the sentence and event type as input. Event classification module detects whether the input sentence is an event and classifies the event type to which the text belongs. We design a multi-task learning module to calculate the combined loss of the two tasks to overcome the event type imbalance results in a low recall. For different event type, different event schema are designed for extracting different arguments according to the schema. Our framework is first trained offline using a small amount of labeled data. To expand the training data, we design a dialogue generation module, generating multiple question-answering pairs for each trigger or argument for data enhance. The trained models can then be applied to extract event types and associated event arguments.

During the training phase, the reinforcement learning-based, dialogue-guided argument extraction model learns how to extract event arguments by taking as input the target sentence

and a label of the event type. Our framework will first learn several rounds of conversational argument extraction according to event types and sentences, and train the event classification model according to event arguments. In each turn, the predicted argument is provided as a pseudo relation in the lexicon-based graph and a pseudo label in role embeddings of event-based BERT used by the incremental event learning method. It updates the textual representation by adding the pseudo argument knowledge. The event classification model is then trained to predict the event type using the pseudo relation knowledge provided by the event extraction module.

During deployment, we used the trained models in an iterative process to perform event extraction. We will first predict the event type with the event classification model, and then implement argument extraction according to the predicted event type. So the model ends up running through all the predicted event types. To do so, we first use the event classification model to predict the event type without a pseudo label and relation. Next, we use the argument extraction model to identify all arguments associated with the predicted event type. We then go back to ask the event classification module again to update the event type using the pseudo label and argument relations extracted by the argument extraction model. This 2-stage iterative process uses the predicted event type to extract arguments, and the extracted information helps the event classification model improve its prediction.

A. Double Perspective Event Representation

The first step of our argument extraction pipeline is to learn the representation (or embeddings) to be used for argument selection. We do so by first constructing a lexicon-based graph, from which we learn the lexicon-based representation of individual words. We then learn the context representation at the sentence-level across multiple words. To that end,
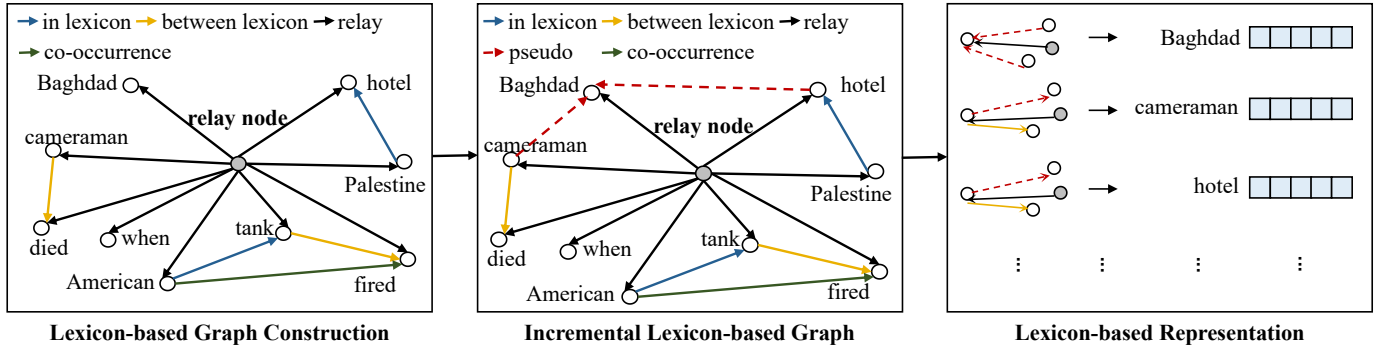
Fig. 4: The flowchart of lexicon-based representation. The input sentence is "In Baghdad, a cameraman died when an American tank fired on the Palestine hotel.". Before constructing the graph, we remove punctuation marks and prepositions from the sentence, which can reduce the complexity of the graph calculation. The graph construction module has five edges: the black edge connects the relay node and all words; the orange edge is the phase's connection; the blue is the association between phrases; the green connects words with high co-occurrence probability; the red is the pseudo argument relation.

we use both a lexicon-based graph attention network [14] and an event-based BERT model [15] for learning the word representation from semantics and context two perspectives.

1) Lexicon-based Representation: Lexicon-based graph neural network has been designed for the node classification task [14], proved to be an effective way to learn global semantics. Thus, we use lexical knowledge to concatenate characters capturing the local composition and a global relay node to capture long-range dependency.

We convert the sentence to a directed graph (as shown in Fig. 4) where each word is represented as a graph node, and a graph edge represents one of the five relations: words in a lexicon; lexicon to lexicon; a relay node connecting to all nodes; co-occurrence words; and pseudo relation among arguments of an event. The first connects words in the phrase sequentially until the last word. The second is to create a line between phrases that the last word of the current phrase is connected with the latter phrase, and each edge represents the potential characteristics of the word that may exist. We also use a relay node as a virtual hub, which is connected to all other nodes. It gathers all the edges and nodes' information, eliminating the boundary ambiguity between words, and learning long-range dependency. Therefore, the representation of the relay node can be regarded as the representation of the sentence. The fourth edge represents pseudo argument relation by connecting the predicted arguments in an event. The last one is calculating the co-occurrence probability of words within sliding windows in the corpus. The edge weights are measured by pointwise mutual information (PMI):

$$\text{PMI}(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} = \log \frac{N_{w_i, w_j} N_s}{N_{w_i} N_{w_j}}, \quad (1)$$

where $N_{w_i}, N_{w_j}, N_{w_i, w_j}$ are the number of sliding windows containing word w_i, w_j and both w_i, w_j, and $i, j \in [1, N]$. N_s is the total number of sliding windows in the corpus.

To learn the word-level representation, we extend the lexicon-based graph attention network (LGAT) that is designed for learning global semantics for node classification [14]. We extend the LGAT by adding a pseudo edge, i.e., by appending an edge among predicted arguments of an event type. Our goal

is to update word representation using existing prediction results. Given an N-word text $T = \{T_1, T_2, \ldots, T_N\}$, the model embeds the initial input text $ET = \{ET_1, ET_2, \ldots, ET_N\}$ through the pre-trained embedding matrix. The embedding of predicted event role $R = \{R_1, R_2, \ldots, R_N\}$ is represented as $ER = \{ER_1, ER_2, \ldots, ER_N\}$. The model takes as input $EI = ET \oplus ER$ where $\oplus$ means concatenation and produce a hidden representation $H^n = \{h_1^n, h_2^n, \ldots, h_N^n\}$, for text T. Here, h_i^n represents the text feature of the i-th word on the n-th hidden layer. Therefore, the final node representation is:

$$LT_i = f(H_i^N), i = 0, 1, 2, \ldots, N, \quad (2)$$

where LT_0 is the relay node used to predict the event type. It is then used to obtain the optimal decision through the conditional random field (CRF). The probability of a label sequence $\hat{y} = \hat{c}_1, \hat{c}_2, \ldots, \hat{c}_k$ can be defined as follows:

$$p(\hat{y} \mid s) = \frac{\exp\left(\sum_{i=1}^N \phi(\hat{c}_{i-1}, \hat{c}_i, \mathbf{LT})\right)}{\sum_{y' \in L(s)} \exp\left(\sum_{i=1}^N \phi(c'_{i-1}, c'_i, \mathbf{LT})\right)}, \quad (3)$$

where $L(s)$ is the set of all arbitrary label sequences.

$$\phi(\hat{c}_{i-1}, \hat{c}_i, \mathbf{h}) = \mathbf{W}_{(c_{i-1}, c_i)} \mathbf{c}_i^T + \mathbf{b}_{(c_{i-1}, c_i)}, \quad (4)$$

where $\mathbf{W}_{(c_{i-1}, c_i)} \in \mathbb{R}^{k \times N}$ and $\mathbf{b}_{(c_{i-1}, c_i)} \in \mathbb{R}^{k \times N}$ are the weight and bias parameters specific to the labels c_{i-1} and c_i, k is the number of event types, and N is the input length of text.

2) Event-based Context Representation: BERT is a multi-layer bidirectional Transformer [15], achieving significant performance improvement on event extraction task [19]. We use a BERT model [19] to learn the context representation. Specifically, we feed the sentence text into an event-based BERT model to encode the input text T, the predicted event role R, and the embedding of *Agent A* (used for dialogue generation described in Section III-B1) $EA = \{EA_1, EA_2, \ldots, EA_M\}$. We extend BERT by adding a self-attention mechanism to learn new contextual representations of *Agent A* (denoted as SA,) and input text (denoted as ST).

3) Final Representation: We concatenate the lexicon and event-based representation to produce the final representation CT_i, to be used for argument extraction.

$$CT_i = LT_i \oplus ST_i. \tag{5}$$

B. Ranked Argument Extraction

Given an event type, our argument extraction component aims to generate high-quality dialogue content by ranking the argument extraction order and utilizing the historical dialogue content. It consists of four main modules: *dialogue generation, argument extraction, incremental event learning, and reinforcement learning based dialogue management*. The dialogue guided argument extraction model automatically extracts the arguments by inputting the actual event type and text. The incremental event learning module then adds the pseudo label as the training data and appends the pseudo-relationship to the lexicon-based graph. After generating the target sentence's event representation, we follow a dialogue-guided strategy for argument extraction. Specifically, *Agent A* uses the question set from [7] to generate a query (or question) about an event or a chosen argument (e.g., "What is the trigger of event X?"). Then, *Agent B* answers the query by predicting an argument role or event type. Based on the answer, *Agent A* will then generate a new query in the next turn of argument extraction. This iterative process is driven by an RL-based dialogue management system described in Section III-B4, aiming to optimize the order of argument extraction. The answer given by *Agent B* will also be fed into an incremental event learning module described later to update the previous answers to be used for the next turn of argument extraction. Later in Table III, we give an example dialogue generated by our approach. The automatically generated dialogue produces extra information to be used during the next-turn argument extraction. Our approach is highly flexible, allowing one to tailor the event extraction framework to a specific domain by populating the question set with domain knowledge.

1) Dialogue Generation: Our dialogue generation module uses two agents (A and B) to assist event extraction through a sequence of question-answer conversations. Here, *Agent A* generates dialogue content according to the currently processed role. For each current role, it generates a question set [7] to create more training data for argument extraction. For example, when the goal of *Agent A* is to generate dialogue for the argument role "Instrument", we select one of the pre-designed question set templates to generate dialogue. All we need to do is filling the argument roles for the given template. *Agent A* produces a hybrid content consisting of both the current argument role and arguments already extracted. *Agent B* then generates the content, including the predicted argument given by the argument extraction (described in the next paragraph). The predicted argument is then fed into *Agent A* to generate a new dialogue for the next turn of argument extraction. Similar to *Agent A*, *Agent B* also provides a dialog content generation template and only needs to fill the template with predicted arguments. If the predicted argument meets the confidence conditions, it will be part of the content of *Agent A*. The content of *Agent B* will also be fed into the incremental

event learning module described later to add high confidence results for the next turn of argument extraction.

2) Argument Extraction: *Agent B* responds to a query by filling the answer slots of a simple answer template designed for each specific question template. It does so by using the learned representation, CT_i, to locate the start (i_s) and end (i_e) position of an argument within the target sentence. Specifically, we obtain the word probability of a chosen argument as:

$$P_{start}(r, t, k) = \frac{\exp(W^{rs} CT_k)}{\sum_{i=1}^{i=N} \exp(W^r CT_i)}, \tag{6}$$

$$P_{end}(r, t, k) = \frac{\exp(W^{re} CT_k)}{\sum_{i=1}^{i=N} \exp(W^r CT_i)}, \tag{7}$$

where W^{rs} and W^{re} are vectors to map CT_k to a scalar. Each event type t has a type-specific W^{rs} and W^{re}. The probability, $P_{span}(r, t, a_{i_s, i_e})$, of an argument being the description (or answer) for argument role r and event type t is given as follows:

$$P_{span}(r, t, a_{i_s, i_e}) = P_{start}(r, t, i_s) \times P_{end}(r, t, i_e). \tag{8}$$

3) Incremental Event Learning: Our incremental argument learning module tries to incorporate the information obtained at the current argument extraction turn to extract a new argument in the next turn. We do so by adding the extracted argument roles (i.e., pseudo labels) whose reward (evaluated by RL) is greater than a configurable threshold to the input text to provide additional information for extracting new arguments. We also add a new edge (i.e., pseudo relation) to connect the extracted arguments in the lexicon-based graph for an event, so that we can update the lexicon representation to be used for the dialogue in the next turn.

4) Reinforcement Learning-based Dialogue Management: We use RL to optimize the argument extraction order during our iterative, dialogue-guided argument extraction process.

Dialogue action. The dialogue-guided event extraction method defines the action as the set of event schema and argument. It indicates that the reinforcement learning algorithm requires determining the argument roles to go from the current argument to the following argument. Different from the previous reinforcement learning-based method, we design two agents with varying spaces of action. For *Agent A*, action a_A is a role from the event schema, which is the action of *Agent A*. For *Agent B*, action a_B is the argument, which is the action of *Agent B*. While it is essential to determine if the event type of current dialogue turn needs to be converted to the next event type. It means our method can determine well of event type changing.

Dialogue state. Defined at time step t, state $S_t \epsilon S$, is characterized through $S_t = (R_t, c_t^A, c_t^B, Q_0, H^C)$. Where R_t is the argument role selected through the reinforcement learning-based dialogue management, c_t^A, c_t^B are the current embedding of *Agent A* and B, Q_0 is the initial question of *Agent A* about event type, H^C represents the history of the conversation. Different historical conversation H^C contributes differently to the target argument extraction. States s^A and s^B are history-dependent, encoding the historical dialogue and T. s^A and s^B are the concatenation of the state s_{t-1}^A, s_{t-1}^B from

the last dialogue content and the current content embedding c_t^A, c_t^B, represented as:

$$s_t^A = s_{t-1}^B \oplus c_t^A, \qquad (9)$$

$$s_t^B = s_t^A \oplus c_t^B. \qquad (10)$$

Dialogue policy network. The policy is choosing the right action for role selection. The policy network is a parameterized probability map in action space and confidence degree, which aims to maximize the expected accumulated reward.

$$
\begin{aligned}
\pi_\theta \left(a^A, a^B \mid s^A, s^B\right) &= \pi \left(a^A, a^B \mid (s^A, s^B); \theta\right) \\
&= \mathbb{P}\left(a_t^A = a^A, a_t^B = a^B \mid s_t^A = s^A, s_t^B = s^B, \theta_t = \theta\right),
\end{aligned}
$$
$$(11)$$

where θ is the learnable parameter representing the weight on our dialogue policy network. The dialogue policy network decides that actions are chosen of T. It consists of two networks. The first network is the feed-forward network for encoding the dialogue histories is implemented using a softmax function. The second network is the BiLSTM for encoding the dialogue histories $H_t^C = (H_{t-1}^C, a_{t-1}, s_t^A, s_t^B)$ being a continuous vector h_t^C, and H_t^C is an observation, and a_{t-1} is an action sequence, updated through BiLSTM.

$$e_T = \text{softmax}\left(W_T F\left(s^A, s^B\right) + b_T\right), \qquad (12)$$

$$h_t^C = BiLSTM\left(h_{t-1}^C, \left[a_{t-1}; s_t^A, s_t^B\right]\right), \qquad (13)$$

where W_T and b_T are the parameters, $F\left(s_t\right)$ is the state feature vector, e_T is the vector of the input sentence T, h_{t-1}^C is the representation in $t-1$ conversation, a_{t-1}, s_t^A and s_t^B are the action representation and the current state representation of *Agent A* and *B* respectively.

Dialogue reward. The dialogue management module saves the historical dialogues. For the specific event type, the search space is limit. We design a reward function to evaluate all actions. The reward $R(s^B, a^A, a^B)$ is defined as the relatedness of the argument extraction part.

$$R(s^B, a^A, a^B) = \sum_i P_{span}(r_i, t, a_{i_s, i_e}). \qquad (14)$$

The policy agent can be effectively optimized using the reward signal. Note that we simultaneously identify all remaining arguments whose reward is less than a threshold in the final turn to avoid error propagation. The threshold in our model is 0.75.

C. Event Classification

Event classification detects whether the input sentence is an event and classifies the event type to which the sentence belongs. Each sentence is fed into a lexicon-based graph neural network model and an event-based BERT [34] model to learn the global knowledge and context knowledge of the sentence, respectively. The event classification model detects what kinds of events the sentence contains by adding pseudo argument relation knowledge. If the sentence does not contain an event, NULL is output, and the subsequent modules are not executed. enabling it to distinguish the prediction error.

We use a fully connected layer to compute the context-aware utterance representation y_i as follows:

$$y_i = ReLU(\boldsymbol{W}(LT_i \oplus ST_i) + b), \qquad (15)$$

where $\boldsymbol{W}$ and b are trainable parameters, and $\oplus$ denotes vector concatenation operation. For event classification sub-task, an $ReLU$ activation is used to enforce sparsity. To improve event classification performance, we design an extra task - predicting the number of event types. Our multi-task event classification model calculates the combined loss of the two tasks to overcome the event type imbalance results in a low recall.

1) Trigger Classification: The existing event classification is based on the trigger identification to identify the event type, but our method is directly according to the input sentence to identify the event type. Therefore, when we evaluate the performance of trigger classification, we use the trigger predicted in the previous Section III-B of ranked argument extraction. We concatenate the predicted trigger and the input sentence to classify the event type.

2) Multi-task Joint Loss for Event Classification: The multi-task joint loss function estimates the difference in the predicted result and ground-truth value. We design two tasks to learn the distinction between prediction error. For the event classification task, we employ the cross-entropy loss function defined as:

$$\mathcal{L}_T = -\sum_i yt_i \log\left(\hat{yt_i}\right) + (1 - yt_i)\log\left(1 - \hat{yt_i}\right), \quad (16)$$

where yt_i and $\hat{yt_i}$ are the i-th real label and the predicted label on the event classification task. For the type number prediction task, we adopt the mean square loss with the L_2-norm:

$$\mathcal{L}_N = \sum_i \left\|\hat{yl_i} - yl_i\right\|^2 + \eta\|\Theta\|_2, \qquad (17)$$

where yl_i and $\hat{yl_i}$ are the i-th label and prediction on the second task. Θ is model parameters, and η is a regularization factor. The overall loss function for optimizing the whole event classification model is:

$$L = \lambda_1 L_T + \lambda_2 L_N, \qquad (18)$$

where λ_1, λ_2 are hyperparameters to balance the two loss. The λ_1 is 0.65 and λ_2 is 0.35 in our model.

IV. EXPERIMENTS AND RESULTS

A. Tasks

We test our approach on the ACE 2005 [16] dataset, which is the most widely-used dataset in event extraction. It contains 599 documents, annotated with 8 coarse-grained event types, 33 event subtypes, and 36 argument roles. The part that we use for evaluation is fully annotated with 5,272 event triggers and 9,612 arguments. To make our results directly comparable, we keep the same data split as previous work [6, 19, 7, 20]. The number of documents for the training set, validation set, and test set is 529, 30, and 40, respectively. It contains a complete set of training data in English, Arabic, and Chinese for the ACE 2005 technology evaluation. For the

TABLE I: The P (precision), R (recall), and F1 score on event classification, argument identification and argument role classification results performed on the ACE 2005 test set. Best results are highlighted in bold and "–" means results are not available. We use the two perspective event representation module and multi-turn dialogue module in our dialogue guided model. The difference between our full model and dialogue-guided model is whether using reinforcement learning.

Task	Trigger Classification			Trigger Identification			Argument Identification			Argument Role Classification			Runtime
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
DBRNN [5]	74.10	69.80	71.90	-	-	-	71.30	64.50	67.70	66.20	52.80	58.70	-
JMEE [6]	76.30	71.30	73.70	80.20	72.10	75.90	71.40	**65.60**	68.40	66.80	54.90	60.30	-
Joint3EE [17]	68.00	71.80	69.80	70.50	74.50	72.50	59.90	59.80	69.90	52.10	52.10	52.10	-
GAIL-ELMo [18]	74.80	69.40	72.0	76.80	71.20	73.90	63.30	48.70	55.10	61.60	45.70	52.40	-
PLMEE [19]	81.00	80.40	80.70	84.80	83.70	84.20	71.40	60.10	65.30	62.30	54.20	58.00	23.8h
Chen et al. [45]	66.70	74.70	70.50	68.90	77.30	72.90	44.90	41.20	43.00	44.30	40.70	42.40	20.3h
Du et al. [7]	71.12	73.70	72.39	74.29	77.42	75.82	58.9	52.08	55.29	56.77	50.24	53.31	24.6h
MQAEE [20]	-	-	71.70	-	-	74.50	-	-	55.20	-	-	53.40	31.5h
Our fine-tuned BERT	75.34	76.15	75.93	85.51	84.24	86.02	70.42	60.24	64.12	62.34	53.45	57.80	32.9h
Our dialogue guided model	78.24	**80.44**	79.62	86.34	85.21	86.91	72.75	63.53	67.71	67.67	**54.92**	59.42	36.1h
Our full model	**81.23**	80.00	**80.71**	**87.94**	**87.22**	**87.73**	**73.43**	65.30	**69.97**	**69.82**	54.43	**61.42**	38.1h

dialogue content generation of *Agent A*, we generate content for the argument role selected by the reinforcement learning-based dialogue management module. For example, if the event type is "Life:Die" and the argument role is "instrument" for example in Fig. 2, the generated content is "What is the killing instrument of event Life:Die?". In dialogue, we add predicted arguments from *Agent B*. To keep the sentence grammatically correct, we add fixed compositions, such as adding "using" before the role of "instrument". For the dialogue content generation of *Agent B*, we generate content according to the predicted argument. Here, as with *Agent A*, the template is pre-designed. We evaluate the performance of our model and comparison models for trigger classification (TC), trigger identification (TI), argument identification (AI), and argument role classification (ARC) sub-tasks. The evaluation metrics include precision (P), recall (R), and F1.

B. Parameters

We implement our model based on BERT [34]. We use BERT [34] as sequence encoding for queries and the hyperparameters of the decoder are the same as for the encoder. It has 12 layers, 768-dimensional hidden embeddings, 12 attention heads, and 110 million parameters. The dialogue generation module generates multiple questions for the same trigger and argument. The final number of sentences is added to 2,4000 in training, validation, and test sets of 19,200, 2,400, and 2,400 sentences, at an 8:1:1 ratio. The maximum sequence length is 512-word pieces, the learning rate is 3×10^5 with an Adam optimizer, the maximum gradient norm for gradient clipping is 1.0. The model is trained for 10 epochs and the batch size is 8, where we set the max question length to 128 and the max answer length to 64. The optimal hyperparameters are tuned on the validation set by grid search, and we tried each hyperparameter five times. The dialogue policy network contains a 128 unit bidirectional LSTM and a softmax layer. The dimension of BERT-based word embeddings is 512 dimensions. The mini-batch size is 128 in training. We use Adam as optimization algorithm with the gradient clipping being 5.

C. Comparisons.

We compare our extraction method with eight event extraction methods: **DBRNN** [5] leverages the dependency graph

information to extract event triggers and argument roles. **JMEE** [6], a jointly event extraction framework, introduces attention-based GCN to model graph information. **Joint3EE** [17] is a multi-task model that performs entity recognition, trigger detection and argument role assignment by shared Bi-GRU hidden representations. **GAIL-ELMO** [18] is an ELMo-based model that utilizes generative adversarial network to focus on harder-to-detect events. **PLMEE** [19] is a BERT-based pipeline event extraction method and employs event classification depending on trigger.

Chen et al. [45] use bleached statements giving models acquire to information included in annotation manuals. **Du et al.** [7] apply machine reading comprehension method employs event extraction and enhance data by constructing multiple question for each argument. **MQAEE** [20] is a multi-turn question answering method expediently utilizing history answer to implement event extraction.

D. Main Results

Table I shows the overall results of each approach per evaluation task with six GAT layers. In the trigger classification task, in order to make our model and baseline model adopt the same evaluation index, we supplement the trigger classification experiment. We use the predicted triggers together with the text as the input of the event classification model to predict the event type corresponding to the current triggers. In our dialogue guided model, we use our event representation module and multi-turn dialogue model without reinforcement learning. In our full model, we introduce reinforcement learning-based dialogue management. The difference between full model and dialogue-guided model is whether using reinforcement learning to learn the order of argument extraction. Our model consistently outperforms all other approaches on F1 and precision. Compare to Du et al. [7] and MQAEE [20], two recent machine reading comprehension models, our dialogue-guided model respectively achieves 7.23% and 7.92% improvements on the F1-score on the TC sub-task. For TI and AI, our dialogue-guided model improves the F1-score by at least 10% through reinforcement learning to guide the argument extraction order. It achieves 6.11% and 6.02% improvements on the ARC sub-task. It shows that our method is significantly
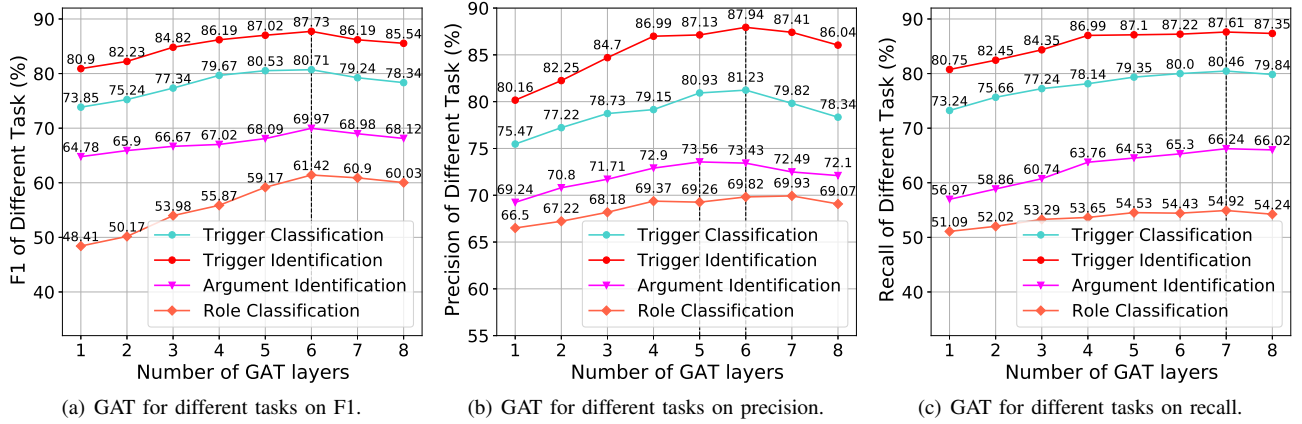
(a) GAT for different tasks on F1.

(b) GAT for different tasks on precision.

(c) GAT for different tasks on recall.

Fig. 5: The change of Precision, Recall and F1 for our model with different GAT layers under different tasks.

(a) GAT for different label rate on TC.

(b) GAT for different label rate on AI.

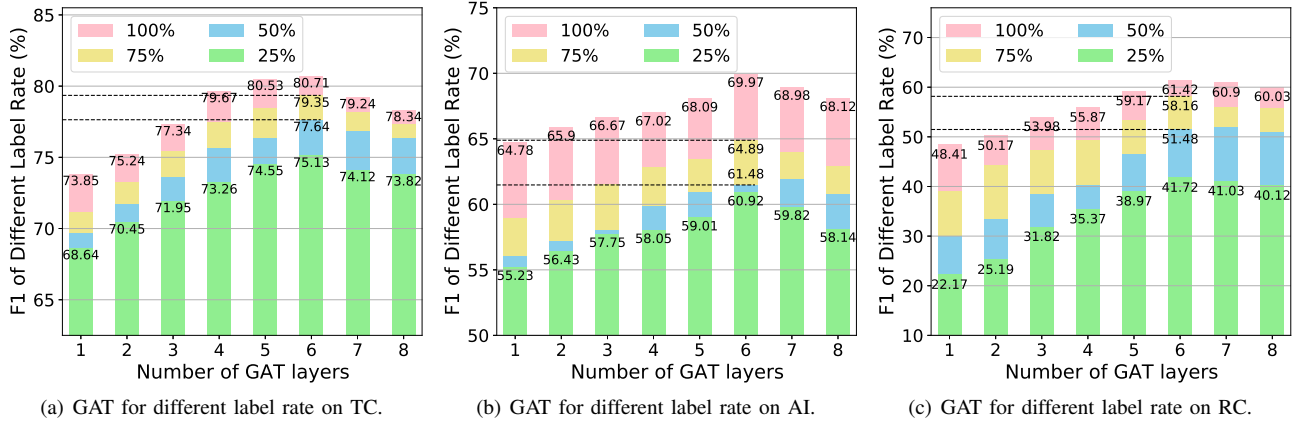(c) GAT for different label rate on RC.

Fig. 6: The change of sub-tasks for our model with different GAT layers under different label rates.

superior to the MRC methods, which only utilize relationships among arguments.

Our dialogue guided model also consistently outperforms PLMEE [19], the best-performing baseline model. The results show the importance of exploiting argument knowledge and relation for event classification and argument extraction. Our full model boosts the F1-score by 9.01%, 13.23%, 14.77%, and 8.02% on TC, TI, AI, and RC, respectively, when compared to the MQAEE, the best-performing alternative on F1-score. Our approach delivers higher precision than other approaches. On some tasks, our approach gives a lower Recall compared to the best-performing baseline model, but the resulting Recall is not far from the best model JMEE (0.47%) and our dialogue guided model exceeds it. By utilizing argument extraction order, our approach delivers the best overall results.

We compare the runtime of our model with BERT-based baseline models, including the QA-based model. In our model, event extraction is realized through multiple rounds of our dialogue system, and reinforcement learning is introduced to optimize the argument extraction order. It can accomplish event extraction more accurately but increases training complexity. The original intention of our model is to improve the precision, recall and F1 of event extraction by making full use of the dependence among arguments under limited data. We will consider improving the accuracy without increasing the model complexity and training difficulty in future work.

E. Impact of GAT Layer Number

A GAT model in a semi-supervised task with a lower label rate, which means the proportion of labeled data to the ACE dataset, requires more graph attention layers to maintain the best performance. Fig 5 presents some empirical evidence to demonstrate the layer effect on our lexicon-based graph for all sub-tasks. We test the performance on TC, TI, AI, and RC four tasks and observe that the F1-score enhances when the GAT layer increases until the sixth layer reach the maximum. With respect to Precision and Recall, GAT achieves the best performance on the sixth and seventh layer.

As shown in Fig 6, we also test our model's performance with different layers in special label rates for TC, AI and RC sub-tasks. It is apparent to note that the model gets the best performance under different label rates when the GAT layer is 6, and the number of layers under the best performance exhibits an increasing trend as the label rate increases. It demonstrates that the best GAT layer gets a stable performance under different tasks and changing the label rate. As can be seen from Fig. 5 and Fig. 6, using more GAT layers can improve the performance, judging by the Precision, Recall and F1-score. However, the improvement reaches a peak when using 6 GAT layers across all three evaluated tasks, and a further increase in the number of layers does not give improved performance. Therefore, we choose to use 6 GAT layers.
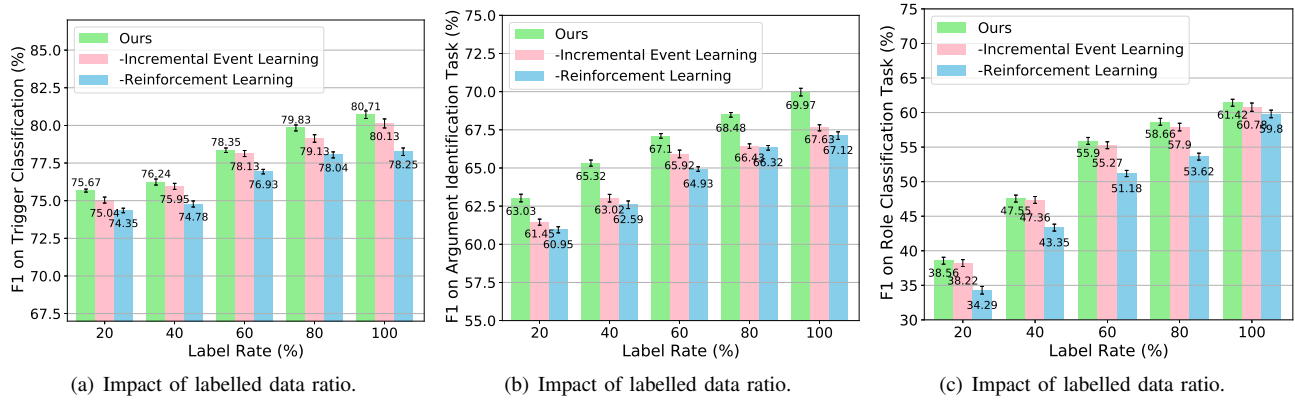
(a) Impact of labelled data ratio.

(b) Impact of labelled data ratio.

(c) Impact of labelled data ratio.

Fig. 7: The resulting F1 score of our approach as the ratio of labeled data changes for different tasks on different tasks.
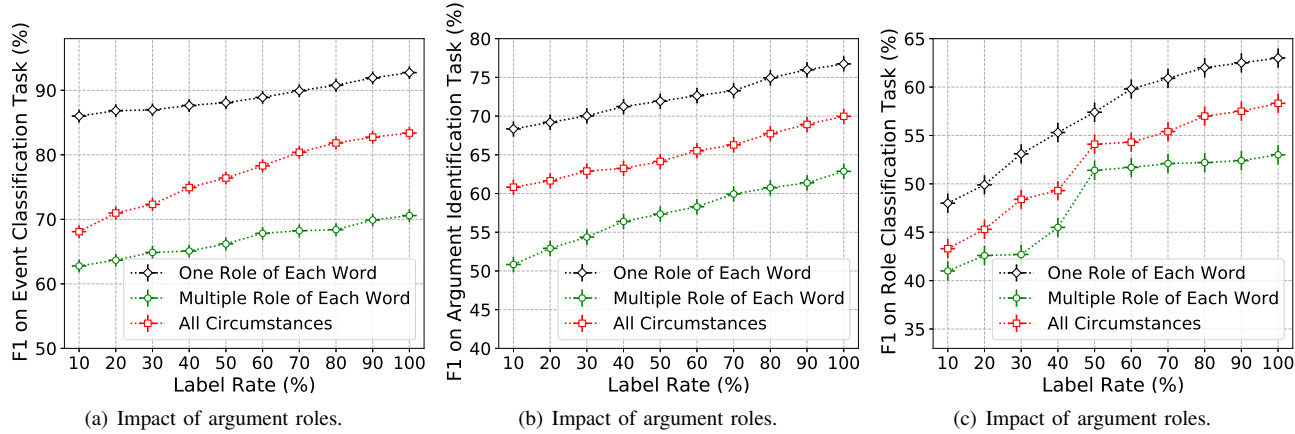

(a) Impact of argument roles.

(b) Impact of argument roles.

(c) Impact of argument roles.

Fig. 8: The F1-score of our approach as the ratio of labeled data changes for multiple type test data on different tasks.

(a) Average loss on different iteration.

(b) Average reward on different iteration.
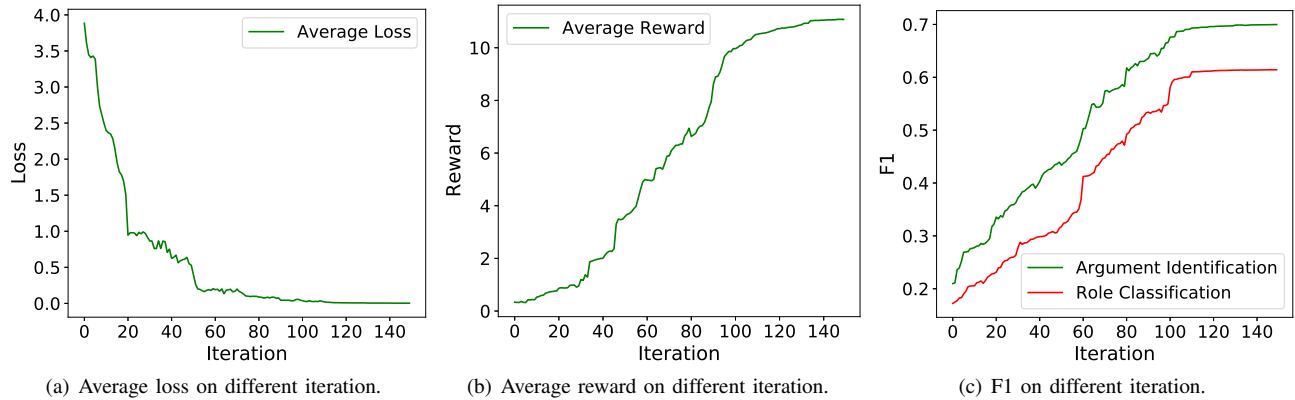
(c) F1 on different iteration.

Fig. 9: The influence of reinforcement learning on argument extraction task.

F. Impact of Data Settings

To verify that our model can achieve significant performance improvement even with little labeled data, we test the model's performance by changing the training data amount. We evaluate the impact of training data on three subtasks in Fig .7. Compared with the other two sub-tasks, trigger classification is the least affected by the labelled data ratio, which can basically reach 75% F1-score. It shows that our model can achieve relatively stable results under different data scales. Fig. 7c shows how the F1-score changes as the ratio of labeled data available to our scheme changes on argument role classification task. As expected, using more labelled data thus improves the performance of our models. Our reinforcement learning module can achieve good performance, even when the amount of labeled data is small. The combination of reinforcement and incremental event learning can improve the robustness of our approach, leading to better overall results than individual techniques. Still, when we remove the incremental event learning and reinforcement learning module, respectively, the former changes dramatically than the latter. It indicates that the incremental event learning module can make our model insensitive to the data scale change.

TABLE II: Ablation study on global constraints on F1-score (%).

Tasks	Trigger Classification		Trigger Identification		Argument Identification		Argument Role Classification	
	P	F1	P	F1	P	F1	P	F1
Ours	**81.23**	**80.71**	**87.94**	**87.73**	**73.43**	**69.97**	**69.82**	**61.42**
-RLD	79.34	78.25	86.21	86.33	71.82	67.12	67.92	59.80
-IEL	79.62	80.13	87.01	86.85	71.92	67.63	67.99	60.78
-LGAT	80.52	80.46	86.21	86.10	72.03	69.45	68.51	61.01
-MTL	80.02	79.49	86.93	86.81	71.89	67.72	67.20	60.22

TABLE III: A dialogue generated by our RL-based dialogue system for sentence: "*As the **soldiers** approached, the **man** detonated explosives in the car, **killing** all four of the soldiers.*".

Agent A query	**Agent B** answer
A: What is the first **event type**?	**B:** The first event type is **Life: Die**.
A: What is the trigger of event **Life:Die**?	**B:** The **trigger** of event **Life:Die** is **killing.**
A: What is the **killing instrument** of event **Life:Die**?	**B:** The **killing instrument** of event **Life:Die** is **explosives**.
A: Who is the **killing victim** of event **Life:Die** using **explosives**?	**B:** The **killing victim** of event **Life:Die** is the **soldiers**.
A: What **agent** killing the **soldiers** of event **Life:Die** using **explosives**?	**B:** The **agent** of event **Life:Die** is the **man**.
A: What is the second **event type**?	**B:** The second event type is **Conflict: Attack.**
A: What is the **trigger** of event **Conflict: Attack**?	**B:** The **trigger** of event **Conflict: Attack** is **detonated.**
A: What is the **detonated instrument** of event **Conflict: Attack**?	**B:** The **detonated instrument** of event **Conflict: Attack** is **explosives** and **car**.
A: What is the **detonated target** of event **Conflict: Attack** using **explosives** and **car**?	**B:** The **detonated target** of event **Conflict: Attack** is **soldiers**.
A: Who is the **detonated attacker** of event **Conflict: Attack** using **explosives** and **car**?	**B:** The **detonated attacker** of event **Conflict: Attack** is **man**.

Fig. 8 shows the impact of argument compositions. In this experiment, we divide the test data into three parts where each word has (1) one role (*One Role of Each Word*), (2) more than one role (*Multiple Role of Each Word*) and (3) a mixture of both (*All Circumstances*). Our model achieves 92.74%, 76.74% and 63.00% F1-score on the part of one role of each word. As expected, our approach gives better results when the argument has just one role than other data compositions. Nonetheless, our framework still delivers good performance for other data settings and can use the increased labeled data to improve performance. For sentences having multiple roles of words, our model is less affected by the label rate in the event classification task, but more affected by the label rate in the argument extraction sub-task. In general, our model can have relatively stable performance in both one role and multiple roles for each word.

G. The influence on RLD module

In order to verify the reinforcement learning helping for argument extraction, and the model performance is improved through iterative training. We demonstrate this from three aspects: average loss, average reward, and F1 score. Our model converges in the $110th$ iteration, with a loss value of 0.0282. The average reward of reinforcement learning converges in the $120th$ iteration and is 10.7256. The F1 score or argument identification and role classification converge in the $120th$ iteration. The F1 argument identification subtask is 0.6142, and the F1 of role classification is 0.6997. Therefore, our model converges on $120th$ iterations when we add the RLD module.

H. Ablation Study

We evaluate four variants of our approach, given in Table II. We remove the **R**einforcement **L**earning-based **D**ialogue

(**RLD**) module, which is the most key module. The descending on TI, AI and RC sub-tasks is 1.40%, 2.85% and 1.62% F1-score, leading to performance changing significantly. When it comes to the **I**ncremental **E**vent **L**earning (**IEL**) module, F1-score decreases 0.58%, 0.88%, 2.34% and 0.64%, respectively. It may prove that our IEL module can provide useful pseudo labels and relations to model the arguments relation. We remove the **L**exicon-based **G**raph **At**tention network (**LGAT**) module. F1-score decreases on all four sub-tasks, which shows that the LGAT module positively affects learning lexicon-based knowledge. It may prove that our lexicon-based graph attention network and event-based BERT model can learn better of the event representation. Moreover, the **M**ulti-**T**ask **L**earning (**MTL**) module is employed for TC. It improves 1.22% F1 score by distinguishing the error types and contributes F1 score of TI in terms of 0.92%, AI in terms of 2.25% and ARC in terms of 1.20%. It suggests that the MTL module can accurately identify the sentence with multiple same event types. The results suggest that all variants are useful, and the RLD is the most important, as removing it can result in the most drastic performance degradation. By utilizing historical dialogue knowledge, our approach achieves about 6.55%, 11.83%, 11.92% and 6.40% F1-score gains than the best-reported question answering based method MQAEE [20] on the four subtasks. It may demonstrate the effectiveness of reinforcement learning-based dialogue generation. For argument role classification, the precision enhances 7.52% compared with the best-reported model PLMEE [19]. It may prove that our lexicon-based graph attention network and event-based BERT model can learn better of the event representation.

Sentence: **Some 70 people** were arrested *Time-With* **Saturday** as **demonstrators clashed** with **police** at the end of a major peace **rally here**, as at least 200,000

Artifact *Trigger* *Destination*
anti war **protesters** **took** to the **streets** across the United States and Canada.

Event 1	Event 2	Event 3	Event 4
Type: Justice: Arrest-Jail	**Type:** Conflict: Attack	**Type:** Conflict: Demonstrate	**Type:** Movement: Transport
Trigger: arrested	**Trigger:** clashed	**Trigger:** rally	**Trigger:** took
Time-Within: Saturday	**Time-Within:** Saturday	**Time-Within:** Saturday	**Time-Within:** Saturday
Place: here	**Place:** here	**Place:** here	**Artifact:** protesters
Person: people	**Attacker:** demonstrators, police	**Entity:** protesters	**Destination:** streets

Fig. 10: An example on ACE 2005 with a standard answer table for event extraction. The words in bold are the event arguments.

I. Case Study

Table III gives a dialogue conversation generated by our approach (Section III-B). Our model can solve arguments with multiple roles, including words with different roles in different contexts. It can recognize the argument more completely by learning the relationship between arguments and the order of argument extraction. For example, in the sentence "As the soldiers approached [...]", the word "soldiers" can play different roles. When the event type is "Life: Die", its role is "Victim", while when the event type is "Conflict: Attack", its role becomes "Target". By contrast, the word "explosives" plays the same role ("Instrument") in different events. Both cases can be correctly recognized by our approach.

In some scenarios where the text contains multiple events, our approach may fail to identify some arguments of scattered distribution. For example, as shown in Fig 10, the sentence "Some 70 people were arrested Saturday [...]" contains four event types: "Justice: Arrest-Jail", "Conflict: Attack", "Conflict: Demonstrate" and "Movement: Transport". For the latest event type, our model does not recognize "Saturday" although it has the same role in multiple events. This is because this argument is further away from other arguments, and our model does not explicitly capture such relation when determining the event extraction order. Our future work will look into this.

V. Conclusion

We have presented a new approach for event extraction by utilizing event arguments' relationships. We tackle the problem within a task-oriented dialogue guided framework designed for event extraction. Our framework is driven by reinforcement learning. We use RL to decide the order for extracting arguments of a sentence, aiming to maximize the likelihood of successfully inferring the argument role. We then leverage the already extracted arguments to help resolve arguments whose roles would be difficult to settle by considering the argument in isolation. Our multi-turn event extraction process also uses the newly obtained argument information to update decisions of the previously extracted arguments. This dual-way feedback process enables us to exploit the relation among event arguments to classify the argument's role in different text contexts. We evaluated our approach on the ACE 2005 dataset and compared to 7 prior event extraction methods. Experimental results show that our approach can enhance event extraction, outperforming competing methods in the majorities of the tasks. In the future, we plan to improve the multi-semantic representation of the dialogue guided event extraction by introducing commonsense knowledge.

References

[1] J. R. Cowie and W. G. Lehnert, "Information extraction," *Commun. ACM*, vol. 39, no. 1, pp. 80–91, 1996.

[2] C. Baral, *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University Press, 2010.

[3] E. D. Liddy, "Advances in automatic text summarization," *Inf. Retr.*, vol. 4, no. 1, pp. 82–83, 2001.

[4] S. Wasserkrug, "Event prediction," in *Encyclopedia of Database Systems*, pp. 1048–1052, 2009.

[5] L. Sha, F. Qian, B. Chang, and Z. Sui, "Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 5916–5923, 2018.

[6] X. Liu, Z. Luo, and H. Huang, "Jointly multiple events extraction via attention-based graph information aggregation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 1247–1256, 2018.

[7] X. Du and C. Cardie, "Event extraction by answering (almost) natural questions," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 671–683, 2020.

[8] Z. C. Lipton, X. Li, J. Gao, L. Li, F. Ahmed, and L. Deng, "Bbq-networks: Efficient exploration in deep reinforcement learning for task-oriented dialogue systems," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 5237–5244, 2018.

[9] Z. Zhang, X. Li, J. Gao, and E. Chen, "Budgeted policy learning for task-oriented dialogue systems," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 3742–3751, 2019.

[10] X. Li, F. Yin, Z. Sun, X. Li, A. Yuan, D. Chai, M. Zhou, and J. Li, "Entity-relation extraction as multi-turn question answering," in *Proceedings of the 57th Conference of the Association for Computational*

Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pp. 1340–1350, 2019.

[11] N. Mrksic, D. O. Séaghdha, T. Wen, B. Thomson, and S. J. Young, "Neural belief tracker: Data-driven dialogue state tracking," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 1777–1788, 2017.

[12] W. Zhang, K. Song, Y. Kang, Z. Wang, C. Sun, X. Liu, S. Li, M. Zhang, and L. Si, "Multi-turn dialogue generation in e-commerce platform with the context of historical dialogue," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pp. 1981–1990, 2020.

[13] Y. Sun, Y. Hu, L. Xing, J. Yu, and Y. Xie, "History-adaption knowledge incorporation mechanism for multi-turn dialogue system," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 8944–8951, 2020.

[14] T. Gui, Y. Zou, Q. Zhang, M. Peng, J. Fu, Z. Wei, and X. Huang, "A lexicon-based graph neural network for chinese NER," 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 2019, Hong Kong, China, November 3-7, 2019*, pp. 1040–1050, 2019.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017.

[16] G. R. Doddington, A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. M. Strassel, and R. M. Weischedel, "The automatic content extraction (ACE) program - tasks, data, and evaluation," in *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, May 26-28, 2004, Lisbon, Portugal*, 2004.

[17] T. M. Nguyen and T. H. Nguyen, "One for all: Neural joint modeling of entities and events," in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 6851–6858, 2019.

[18] T. Zhang, H. Ji, and A. Sil, "Joint entity and event extraction with generative adversarial imitation learning," *Data Intell. 2019*, 2019.

[19] S. Yang, D. Feng, L. Qiao, Z. Kan, and D. Li, "Exploring pre-trained language models for event extraction and generation," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 5284–5294, 2019.

[20] 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, EMNLP 2020, Online Event, 16-20 November 2020*, pp. 829–838, 2020.

[21] L. Huang, T. Cassidy, X. Feng, H. Ji, C. R. Voss, J. Han, and A. Sil, "Liberal event extraction and event schema induction," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

[22] B. Yang and T. M. Mitchell, "Joint extraction of events and entities within a document context," in *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pp. 289–299, 2016.

[23] J. Ferguson, C. Lockard, D. S. Weld, and H. Hajishirzi, "Semi-supervised event extraction with paraphrase clusters," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pp. 359–364, 2018.

[24] 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 of the Asian Federation of Natural Language*

Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers, pp. 167–176, 2015.

[25] Z. Zhang, W. Xu, and Q. Chen, "Joint event extraction based on skip-window convolutional neural networks," in *Natural Language Understanding and Intelligent Applications - 5th CCF Conference on Natural Language Processing and Chinese Computing, NLPCC 2016, and 24th International Conference on Computer Processing of Oriental Languages, ICCPOL 2016, Kunming, China, December 2-6, 2016, Proceedings*, pp. 324–334, 2016.

[26] T. H. Nguyen and R. Grishman, "Modeling skip-grams for event detection with convolutional neural networks," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pp. 886–891, 2016.

[27] K. Huang, M. Yang, and N. Peng, "Biomedical event extraction on graph edge-conditioned attention networks with hierarchical knowledge graphs," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pp. 1277–1285, 2020.

[28] J. Liu, Y. Chen, K. Liu, W. Bi, and X. Liu, "Event extraction as machine reading comprehension," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 1641–1651, 2020.

[29] J. Zhang, Y. Qin, Y. Zhang, M. Liu, and D. Ji, "Extracting entities and events as a single task using a transition-based neural model," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 5422–5428, 2019.

[30] P. Huang, X. Zhao, R. Takanobu, Z. Tan, and W. Xiao, "Joint event extraction with hierarchical policy network," in *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pp. 2653–2664, 2020.

[31] 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 Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 6045–6052, 2018.

[32] W. Xiang and B. Wang, "A survey of event extraction from text," *IEEE Access*, vol. 7, pp. 173111–173137, 2019.

[33] T. H. Nguyen and R. Grishman, "Event detection and domain adaptation with 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 of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, 2015.

[34] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

[35] M. Mejri and J. Akaichi, "A survey of textual event extraction from social networks," in *Proceedings of the First Conference on Language Processing and Knowledge Management, LPKM 2017, Kerkennah (Sfax), Tunisia, September 8-10, 2017*, 2017.

[36] Y. Chen, S. Liu, S. He, K. Liu, and J. Zhao, "Event extraction via bidirectional long short-term memory tensor neural networks," in *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data - 15th China National Conference, CCL 2016, and 4th International Symposium, NLP-NABD 2016, Yantai, China, October 15-16, 2016, Proceedings*, pp. 190–203, 2016.

[37] B. Romera-Paredes and P. H. S. Torr, "An embarrassingly simple approach to zero-shot learning," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pp. 2152–2161, 2015.

[38] L. Huang, H. Ji, K. Cho, I. Dagan, S. Riedel, and C. R. Voss, "Zero-shot transfer learning for event extraction," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pp. 2160–2170, 2018.

[39] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider, "Abstract meaning representation for sembanking," in *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, LAW-ID@ACL 2013, August 8-9, 2013, Sofia, Bulgaria*, pp. 178–186,

2013.

[40] T. Mitamura, Y. Yamakawa, S. Holm, Z. Song, A. Bies, S. Kulick, and S. M. Strassel, "Event nugget annotation: Processes and issues," in *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation, EVENTS@HLP-NAACL 2015, Denver, Colorado, USA, June 4, 2015*, pp. 66–76, 2015.

[41] T. Zhao, Z. Yan, Y. Cao, and Z. Li, "Asking effective and diverse questions: A machine reading comprehension based framework for joint entity-relation extraction," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pp. 3948–3954, 2020.

[42] F. Mi, M. Huang, J. Zhang, and B. Faltings, "Meta-learning for low-resource natural language generation in task-oriented dialogue systems," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 3151–3157, 2019.

[43] H. Chen, X. Liu, D. Yin, and J. Tang, "A survey on dialogue systems: Recent advances and new frontiers," *SIGKDD Explor.*, vol. 19, no. 2, pp. 25–35, 2017.

[44] O. Ramadan, P. Budzianowski, and M. Gasic, "Large-scale multi-domain belief tracking with knowledge sharing," in *ACL, 2018*, 2018.

[45] Y. Chen, T. Chen, S. Ebner, A. S. White, and B. V. Durme, "Reading the manual: Event extraction as definition comprehension," in *Proceedings of the Fourth Workshop on Structured Prediction for NLP@EMNLP 2020, Online, November 20, 2020*, pp. 74–83, 2020.

Jia Wu received the Ph.D. degree in computer science from the University of Technology Sydney, Ultimo, NSW, Australia. Dr Wu is currently an ARC DECRA Fellow in the Department of Computing, Macquarie University, Sydney, Australia. His current research interests include data mining and machine learning. Since 2009, he has published 100+ refereed journal and conference papers, including IEEE TPAMI, IEEE TKDE, IEEE TNNLS, IEEE TMM, ACM TKDD, NIPS, WWW, and ACM SIGKDD. Dr. Wu was the recipient of SDM'18 Best Paper Award in Data Science Track, IJCNN'17 Best Student Paper Award, and ICDM'14 Best Paper Candidate Award. He is the Associate Editor of the ACM Transactions on Knowledge Discovery from Data (TKDD) and Neural Networks (NN). He is a Senior Member of the IEEE.

Yanxing Ning is currently pursuing the master degree with the Department of Computer Science and Engineering, Beihang University (BUAA), Beijing, China. Her research interests include text mining, representation learning, and event extraction.

Qian Li is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Beihang University (BUAA), Beijing, China. Her research interests include text mining, representation learning, and event extraction.

Lihong Wang is currently a Professor with the National Computer Network Emergency Response Technical Team/Coordination Center of China. Her current research interests include information security, cloud computing, big data mining and analytics, information retrieval, and data mining.

Hao Peng is currently an Assistant Professor at the School of Cyber Science and Technology, and Beijing Advanced Innovation Center for Big Data and Brain Computing in Beihang University. His research interests include representation learning, social network mining and reinforcement learning. To date, Dr Peng has published over 70 research papers in top-tier journals and conferences, including the IEEE TKDE, TC, TPDS, TITS, ACM TOIS, TKDD, TIST, TSAS, AAAI, IJCAI, SIGIR, Web Conference, CIKM, ICDM, DASFAA, etc.

Philip S. Yu is a Distinguished Professor and the Wexler Chair in Information Technology at the Department of Computer Science, University of Illinois at Chicago, and also holds the Wexler Chair in Information Technology. Before joining UIC, he was at the IBM Watson Research Center, where he built a world-renowned data mining and database department. He is a Fellow of the ACM and IEEE. Dr. Yu was the Editor-in-Chiefs of ACM Transactions on Knowledge Discovery from Data (2011-2017) and IEEE Transactions on Knowledge and Data Engineering (2001-2004). He has received several IBM honors including 2 IBM Outstanding Innovation Awards, an Outstanding Technical Achievement Award, 2 Research Division Awards and the 94th plateau of Invention Achievement Awards. His research interest is in big data, including data mining, data stream, database and privacy. He has published more than 780 papers in refereed journals and conferences. He holds or has applied for more than 250 US patents.

Jianxin Li is currently a Professor with the State Key Laboratory of Software Development Environment, and Beijing Advanced Innovation Center for Big Data and Brain Computing in Beihang University. His current research interests include social network, machine learning, big data and trustworthy computing.

Zheng Wang is an Associate Professor with the University of Leeds, UK. His research cuts across the boundaries of parallel program optimization, systems security, and applied machine learning.