Received January 20, 2019, accepted February 10, 2019, date of publication February 18, 2019, date of current version March 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2900124

Joint Event Extraction Based on Hierarchical Event Schemas From FrameNet

WEI LI[1,2], DEZHI CHENG[1,2], LEI HE[3], YUANZHUO WANG[1,2], AND XIAOLONG JIN[1,2], (Member, IEEE)

[1]CAS Key Laboratory of Network Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[2]School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100049, China
[3]Engineering and Applied Science School, Aston University, Birmingham B4 7ET, U.K.

Corresponding author: Yuanzhuo Wang (wangyuanzhuo@ict.ac.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1000902, and in part by the National Natural Science Foundation of China under Grant U1836206, Grant 61572469, Grant 91646120, Grant 61772501, and Grant 61572473.

ABSTRACT Event extraction is useful for many practical applications, such as news summarization and information retrieval. However, the popular automatic context extraction (ACE) event extraction program only defines very limited and coarse event schemas, which may not be suitable for practical applications. FrameNet is a linguistic corpus that defines complete semantic frames and frame-to-frame relations. As frames in FrameNet share highly similar structures with event schemas in ACE and many frames actually express events, we propose to redefine the event schemas based on FrameNet. Specifically, we extract frames expressing event information from FrameNet and leverage the frame-to-frame relations to build a hierarchy of event schemas that are more fine-grained and have much wider coverage than ACE. Based on the new event schemas, we propose a joint event extraction approach that leverages the hierarchical structure of event schemas and frame-to-frame relations in FrameNet. The extensive experiments have verified the advantages of our hierarchical event schemas and the effectiveness of our event extraction model. We further apply the results of our event extraction model on news summarization. The results show that the summarization approach based on our event extraction model achieves significant better performance than several state-of-the-art summarization approaches, which also demonstrates that the hierarchical event schemas and event extraction model are promising to be used in the practical applications.

INDEX TERMS Event extraction, event schema definition, information extraction, joint inference.

I. INTRODUCTION

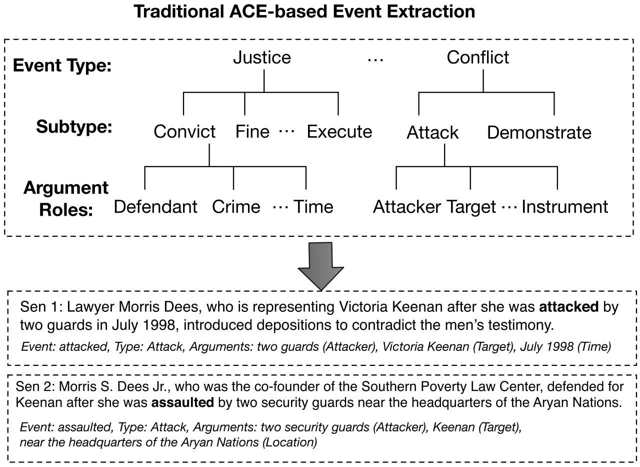
Events are things that happen or occur, and usually involve entities (people, time, place, etc.) as their properties. Understanding events based on their descriptions in text is essential for machine reading systems. It is also useful in many practical applications such as news summarization, information retrieval, question answering, knowledge base construction, etc.

As one of the main tasks in information extraction area, event extraction is to extract event information from text according to predefined event schemas. The event extraction programs in ACE [1] (Automatic Context Extraction) and ERE [2](Entities, Relations and Events) are the main event extraction programs in this area. An event in ACE is represented by an event trigger, an event type and a set of arguments with different roles. The goal of event extraction is to identify event triggers with specific types and arguments with specific roles. However, the event schemas defined in ACE may not be suitable for practical applications because of two major drawbacks: (1) Due to the limited number of pre-defined event schemas (only 8 types with 33 subtypes), the extraction results would miss much salient event information. For example, in Fig. 1, the extraction results on the two example sentences only contain "*Attack*" events. Other salient events, such as events evoked by "*representing*", "*contradict*" and "*defend*", cannot be extracted, as there are no definitions and annotations for these event types; (2) Event schemas defined in ACE are quite coarse, which cannot distinguish different semantic phenomena. For example, all kinds of violent acts, such as street fights, rapes and wars, are treated as a single event type "*Attack*" in ACE.

The associate editor coordinating the review of this manuscript and approving it for publication was Biju Issac.

[1] https://www.ldc.upenn.edu/collaborations/past-projects/ace

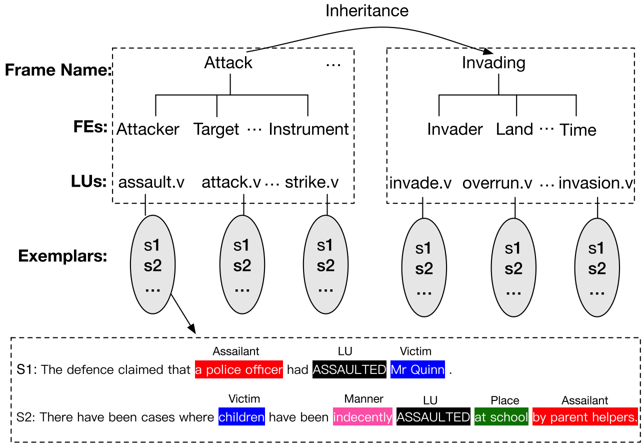
[2] ERE is a lighter-weight version of ACE with similar event schema definition. We only discuss about the characteristic of ACE in the following, which also applies to ERE.



Sen 1: Lawyer Morris Dees, who is representing Victoria Keenan after she was **attacked** by two guards in July 1998, introduced depositions to contradict the men's testimony.
Event: attacked, Type: Attack, Arguments: two guards (Attacker), Victoria Keenan (Target), July 1998 (Time)

Sen 2: Morris S. Dees Jr., who was the co-founder of the Southern Poverty Law Center, defended for Keenan after she was **assaulted** by two security guards near the headquarters of the Aryan Nations.
Event: assaulted, Type: Attack, Arguments: two security guards (Attacker), Keenan (Target), near the headquarters of the Aryan Nations (Location)

FIGURE 1. The event schemas defined in ACE and the results of event extraction for two example sentences. The results show that ACE-based event extraction can only detect and extract few specific types of events, and miss other types of events in text (e.g. the underlined event information).



S1: The defence claimed that a police officer had ASSAULTED Mr Quinn .

S2: There have been cases where children have been indecently ASSAULTED at school by parent helpers.

FIGURE 2. An illustration of the hierarchical structure in FrameNet. S_k under each LU is an exemplar annotated for it. *Inheritance* is a semantic relation between the frames *Invading* and *Attack*. This figure shows that frames in FrameNet share highly similar structures with events in ACE, and many frames actually express certain types of events.

TABLE 1. Examples of mappings from the frames in FrameNet to the events in ACE.

Frame	Event	Sample in FrameNet
Invading	Attack	*Kuwait was* ***invaded*** *by Iraq in August.*
Rape	Attack	*Roy Fisher* ***raped*** *two women in Oxford last year*
Hit_target	Attack	*He* ***hit*** *the bull's-eye with his first arrow*
Quitting	End-Posit.	*Riddell* ***quits*** *Capital House post*
Fining	Fine	*The court* ***fined*** *her $4000.*

FrameNet [1], [2] is a linguistic corpus containing considerable information about lexical and predicate-argument semantics in the form of frames. A frame in FrameNet is defined as a composition of a frame name, a set of Frame Elements (FEs) and a list of Lexical Units (LUs). A LU is a word or phrase that evokes the corresponding frame. FEs indicate the set of semantic roles associated with the frame. Most frames contain a set of exemplars with annotated LUs and FEs. Moreover, there are a set of labeled relations between frames and FEs, such as "*Inheritance*" (see Fig. 2 and Section II-B for details).

From the above definitions of events and frames, we can find that frames defined in FrameNet share highly similar structures with events in the ACE event extraction program. As shown in Fig. 2, the LU of a frame plays a similar role as the trigger of an event. ACE defines the trigger of an event as the word or phrase that most clearly expresses the occurrence of an event. Analogously, the LU of a frame is also the word or phrase that is capable of indicating the occurrence of the expressed frame. Furthermore, the FEs of a frame also play similar roles as the argument roles of an event. Both FEs and argument roles indicate the semantic roles of participants involved in the corresponding frame or event. Besides having similar structures as events, many frames in FrameNet actually express certain types of events in ACE. Table 1 shows some examples of mappings from frames in FrameNet to events in ACE. Most importantly, the FrameNet corpus defines complete semantic frames and thus has wider coverage compared to ACE.

Those observations motivate us to explore: (1) *Can we build a hierarchy of event schemas that is more fine-grained and has wider coverage than ACE based on frames in FrameNet?* (2) *How to leverage the FrameNet structure, such as frame-to-frame relations, to build a more powerful event extraction model?* (3) *Whether the event extraction results based on the new event schemas can be used in practical applications like news summarization?*

For the first question, we extract all frames in FrameNet that express events, and leverage the frame-to-frame relations to build a hierarchy of event schemas. Specifically, three types of frame-to-frame relations, including "*Inheritance*", "*Using*" and "*Subframe*" (defined in Section II-B), are used to determine the hierarchical relations between different event types, such as "*Invading*" is a subtype of "*Attack*". The frame-to-frame relations are also reserved as event-to-event relations. Moreover, the annotated dataset in FrameNet is used as training dataset for event extraction models.

For the second question, both global information such as event-to-event relations, and local features which are essential for event extraction (such as part-of-speech tags and dependency labels) should be properly utilized. A straightforward way is to represent all of them as features and feed them into a classifier. However, it is impossible to encode some global information (such as event-to-event relations) as simple features. To resolve this problem, we propose to encode global information as first-order logic formulas and model them using Markov Logic Network (MLN) [3], [4]. However, it is difficult to model those sophisticated local features using MLN, as they are typically extremely high dimensional [5], [6]. Therefore, we propose a joint event extraction approach which consists of two parts: the local part and the global part, as shown in Fig. 3. Specifically, in the

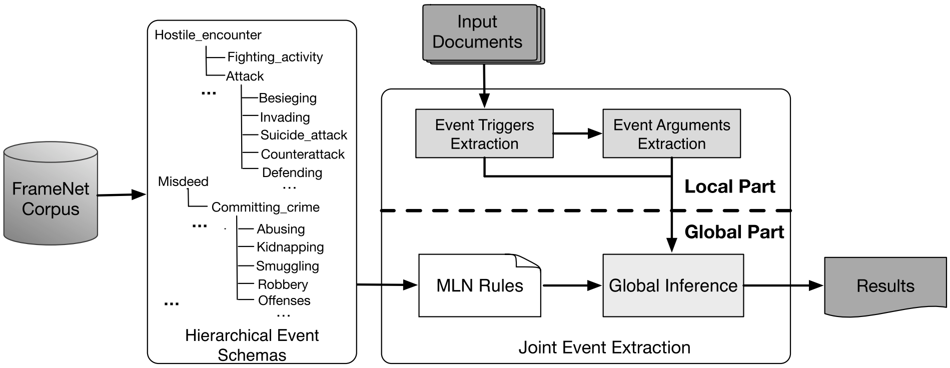


FIGURE 3. The framework of the event extraction system. It consists of two components: (1) hierarchical event schema construction, (2) joint event extraction. The joint event extraction model contains two parts: a local event extraction model and a MLN-based global inference model.

local part, we learn two classifiers that employ predominantly local features to generate initial judgments for event trigger extraction and argument extraction. Then, in the global part, we construct an MLN to encode global information and generate the final results through MLN-based inference.

For the third question, we implement a simple graph-ranking based unsupervised extractive approach to Multi-Document Summarization (MDS) by leveraging the event extraction results. Experiments show that the simple method can achieve significantly better performance compared with several state-of-the-art unsupervised extractive MDS methods. However, the same method with ACE-based event schemas and event extraction models achieves very poor performance, which demonstrates the advantages of our new hierarchical event schemas and event extraction model.

To sum up, the main contributions of this paper include: (1) This is the first work, to the best of our knowledge, to redefine event schemas that are much more fine-grained and have wider coverage based on the FrameNet corpus with considerations for practical applications (all the definitions for event schemas, event-to-event relations and the hierarchical structure of event schemas are released for future use by the NLP community[3]); (2) We propose an effective joint event extraction approach based on MLN, which consists of the local part and the global part, to combine local (from texts) and global (from frameNet structure and texts) information for event extraction. (3) We verify that the event extraction results based on our event schemas and event extraction model can be used in news summarization. They are also promising to be used in other practical applications.

The rest of the paper is organized as follows. The background of the work is described in Section II. In Section III, we present the characteristics of our hierarchical event schemas in detail. Then our proposed local event extraction model and MLN-based global inference model are described in Sections IV and V, respectively. Section VI describes the experiments and the results. The application of our event schemas and event extraction model on news summarization is presented in Section VII. Finally, Section VIII concludes the paper.

II. BACKGROUND

A. EVENT EXTRACTION TASK DESCRIPTION

In this paper, the event extraction task is the same as in ACE evaluation, where an event is defined as a specific occurrence involving participants. First, we introduce some terminology based on ACE to facilitate the understanding of this task:

- **Entity**: an object or a set of objects in one of the semantic categories of interests.
- **Entity mention**: a reference to an entity (typically, a noun phrase).
- **Event trigger**: the main word that most clearly expresses an event occurrence (typically, a verb or a noun).
- **Event argument**: an entity mention, temporal expression or value that is involved in an event (participants).
- **Argument role**: the relationship between an argument and the event in which it participates.
- **Event type**: the semantic type of an event, which has its own set of potential argument roles.
- **Event mention**: a phrase or sentence within which an event is described, including a trigger and arguments.

Given a text document, an event extraction system should identify event triggers with specific event types and their arguments with specific argument roles. Fig. 1 has shown several examples of event extraction under the ACE evaluation.

B. FRAMENET STRUCTURE

The FrameNet corpus is a taxonomy of manually identified semantic frames for English. Fig. 2 shows the hierarchical structure of the FrameNet corpus. FrameNet[4] in total contains more than 1200 various frames and 13500 LUs with 202000 manually annotated exemplars. Eight types of relations are defined between frames in FrameNet, but we only use the following three of them because the others can't express hierarchical relations between frames:

[3] https://github.com/weili-ict/EventSchemasBasedOnFrameNet

[4] We use the FrameNet 1.7 version.

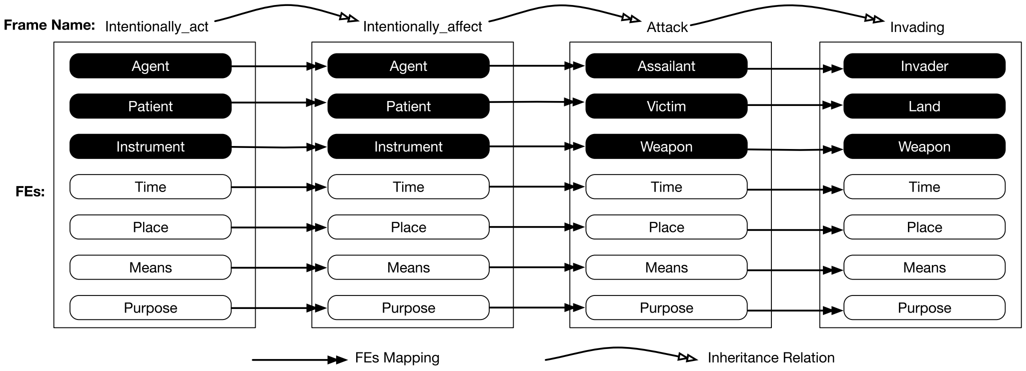


FIGURE 4. Partial illustration of the relations between frames, and the mappings between FEs of related frames. Core FEs are filled in black. Non-core FEs (such as Time and Place) are unfilled.

- **Inheritance**. Frame A inherits from frame B indicating that A corresponds to an equal or more specific fact about B. For example, "invading" inherits from "attack".
- **SubFrame**. Frame A is a subFrame of frame B indicating that B is a complex frame that refers to sequences of separate states and transitions, and A is one of them. For example, "Committing_crime" is a subFrame of "Crime_scenario".
- **Using**. Frame A and frame B are connected by this relation indicating that a part of the scene evoked by A refers to B. It is a directional relation, where B is usually more abstract than A (i.e., "Abusing" using "Cause_harm").

Besides the relations between frames, there are also mapping relations between FEs of related frames. Fig. 4 shows the relations between frames "Intentionally_act", "Intentionally_affect", "Attack" and "Invading", and the mappings relations between their FEs (such as FE "Assailant" of frame "Attack" is mapped with both FE "Invader" of frame "Invading" and FE "Agent" of frame "Intentionally_affect").

C. RELATED WORK

1) EVENT EXTRACTION BASED ON ACE

Most of existing researches on event extraction are based on ACE-defined event schemas. Nearly all of them use supervised paradigm, which can be divided into two categories: feature-based methods and representation-based methods. The feature-based methods model the problem of event extraction as a classification task by using large volumes of features [7], such as lexical features, syntactical features, external knowledges, etc. A set of global features, such as global evidence from related documents [8], global features from other events and entities [9]–[11], and clues between event triggers and arguments [12], have been utilized to design joint event extraction models. Some global inference approaches have also been proposed to leverage both latent local and global information to improve the performance of event detection [13].

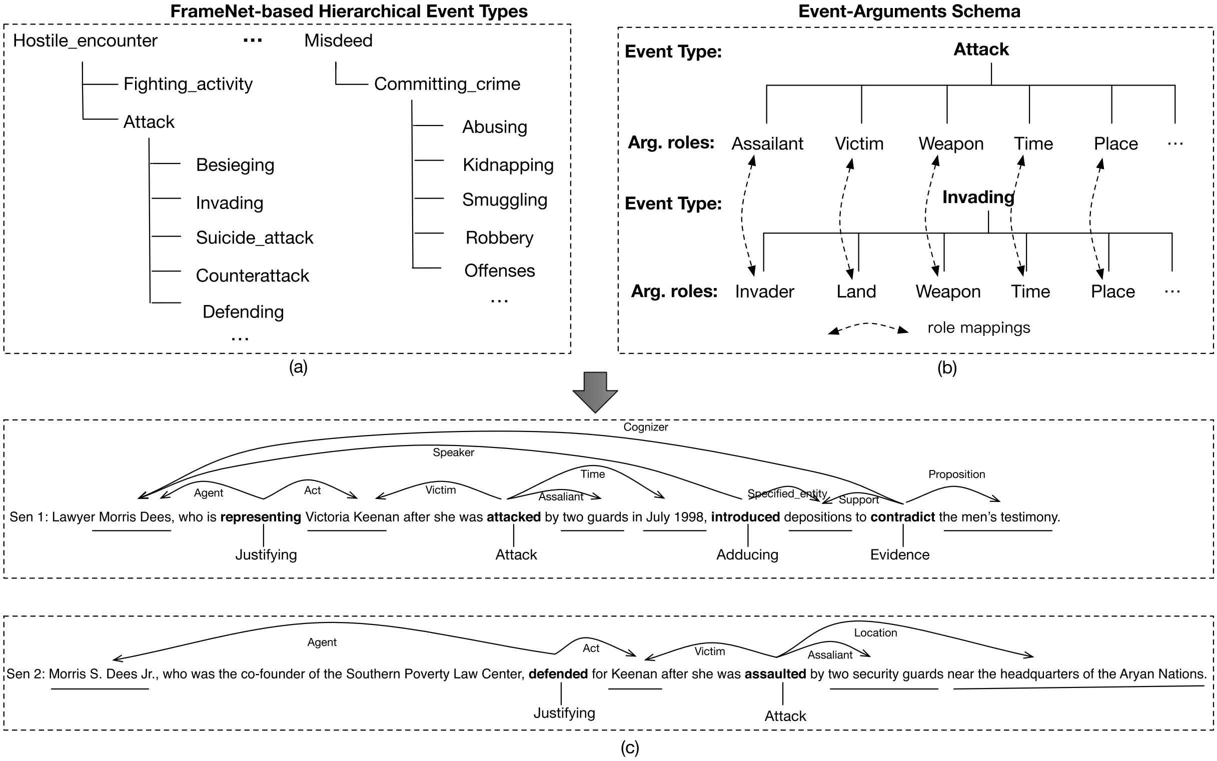
The representation-based methods are more and more popular in these years. Both event mentions and arguments are first represented as low-dimensional embeddings, then fed into neural networks for classification. A dynamic multi-pooling convolution neural network model was first proposed to use distributed word representations for event extraction [14]. Event detection methods based on convolution neural network have been further studied by importing argument information [15] and domain information [16]. Later, event extraction methods based on Recurrent Neural Network (RNN) were further studied [17]. Recently, several joint neural models with attention mechanism were proposed to do jointly multiple events detection [18] and jointly multiple events extraction [19]. A novel dependency bridge recurrent neural network model was proposed for jointly event triggers and event arguments extraction [20].

2) EVENT SCHEMA INDUCTION

Some open information extraction methods without predefined event schemas have been proposed. Unspecified event detection from social media has attracted great attention, such as event detection from Flickr data or twitter corpus through wavelet analysis [21], [22]. Ritter *et al.* [23] propose an open domain event extraction method for twitter corpus based on topic inference model. However, the event definition is just a phrase or cluster of phrases, which cannot capture the argument semantics of an event mention. Huang *et al.* [24] propose a liberal event extraction framework to extract events and discover event schemas from input corpus automatically. However, the event schemas discovered automatically are unavoidably noisy and hard to be named correctly.

3) EVENT EXTRACTION BASED ON FRAMENET

Aguilar *et al.* [25] pointed out that all events, relations, and attributes that represented by ACE/ERE and TAC-KBP standards can be mapped to FrameNet representations through some adjustments. Liu *et al.* [26] propose to leverage FrameNet to improve automatic event detection. They proposed a PSL (Probabilistic Soft Logic) based global inference approach based on three hypotheses between frames and events. Furthermore, they also analyze the mappings from frames/LUs to event types. Liu *et al.* [27] also propose to extract salient events from documents based on event-evoking



Sen 1: Lawyer Morris Dees, who is **representing** Victoria Keenan after she was **attacked** by two guards in July 1998, **introduced** depositions to **contradict** the men's testimony.

Sen 2: Morris S. Dees Jr., who was the co-founder of the Southern Poverty Law Center, **defended** for Keenan after she was **assaulted** by two security guards near the headquarters of the Aryan Nations.

FIGURE 5. The structure of the new event schemas. (a) shows the hierarchical event schemas defined based on FrameNet; (b) shows the event types, argument roles and the role mappings between arguments roles. (c) shows two event extraction examples based on the new event schema.

frames in FrameNet. Our work is inspired by the results of these work.

III. HIERARCHICAL EVENT SCHEMA CONSTRUCTION

As many frames in FrameNet actually express events, and the structure of frame in FrameNet is similar with event schema in ACE, we propose to build new event schemas based on FrameNet. Specifically, all frames expressing event information (in total 655) are extracted from FrameNet corpus, and the frame-to-frame relations *"Inheritance"*, *"Subframe"* and *"Using"* are leveraged to build hierarchical event schemas, shown as in Fig. 5 (a). The frame name are directly used as event type and their FEs are used as arguments roles, shown as in Fig. 5 (b). For each frame in FrameNet, there are core FEs and non-core FEs, as shown in Fig. 4. Core FEs are conceptually or syntactically necessary to the central meaning of the frame (analogous to the core arguments ARG0-ARG5 in PropBank). By contrast, non-core FEs loosely correspond to syntactic adjuncts and carry broadly applicable information such as time and place. In our event schema definition, we reserve all the core FEs and only some of the common non-core FEs such as *Time* and *Place*, as argument roles.

The frame-to-frame relations are also directly reserved as event-to-event relations. Moreover, the mapping relations between FEs are also used to build role mappings between event argument roles. We define the terminology of role mapping as following:

- **Role mapping**: the argument roles of related events are associated with each other. Mapped argument roles describe similar type of relations between arguments and events in which they participate.

The mappings between argument roles are directly taken from mappings between FEs in FrameNet.

Among the 655 event schemas, there are 51 schemas expressing specific event scenarios, such as *"Employment_scenario"*, *"Crime_scenario"* and *"Commerce_scenario"*. Each event scenario contains tens of sub-events, which are obtained from the hierarchical structure of event schemas. Fig. 6 illustrates parts of the event scenarios. We define the terminology of event scenario as following:

- **Event scenario**: each event scenario expresses an abstract event which consists of tens of sub-events describing related event information.

In our event schema definition, the event scenarios are directly taken from specific frames (with *"_scenario"* in frame name) in FrameNet. We define that two events belong to the same event scenario if they are both sub-events of the same event scenario. Events belonging to the same event scenario all describe related information.

The advantages of our hierarchical event schemas[5] include:

1) It in total contains 655 well-defined event schemas. Based on our event schemas, we can extract more rich event information from text. For example, in part (c) of

[5] The new event schemas, event scenarios, event-to-event relations and role mappings are available at https://github.com/weili-ict/EventSchemasBasedOnFrameNet

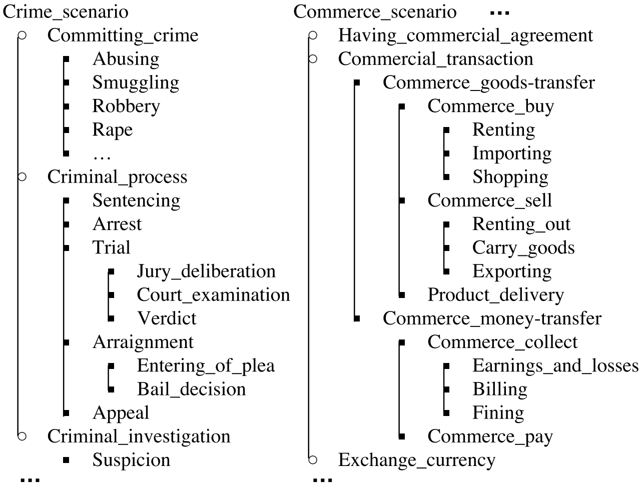


FIGURE 6. Partial illustration of the event scenarios. The sub-events of each root event scenario type belong to the same event scenario.

Fig. 5, 4 events are extracted from the first example sentence and 2 events are extracted from the second example sentence. However, only 1 event can be extracted from both sentences by ACE evaluation.

2) The event types in our definition are more fine-grained while the event types in ACE are too coarse. For example, all kinds of violent acts, such as street fights and wars, are treated as a single type *"Attack"*, however, there are more rich definitions in our event schemas (e.g. *"Attack"* has 5 subtypes shown in Fig. 5(a)). Moreover, the hierarchical structure can be used for global inference in event extraction (Section V-B).
3) The relations between frames are directly used as relations between events. Also, the mappings between FEs are extracted to build role mappings between event argument roles (as shown in Fig. 5 (b)). Both the event-to-event relations, the argument role mappings and the structure of event scenarios can be used to improve the performance of event extraction through global inference (Section V-B.1 and V-B.2).
4) The LUs of frames are also used as LUs of events, which are helpful for event trigger identification (see Section IV-A). The large manually annotated dataset in FrameNet can be directly used as training and evaluation data for event extraction. For example, the exemplar sentences shown in Fig. 2 can be directly used as annotations for event extraction.

IV. LOCAL EVENT EXTRACTION MODEL

The goal of event extraction is to detect event triggers, identify their corresponding event types, extract arguments for each event and identify their corresponding argument roles. Our local event extraction model consists of two parts: (1) event trigger extraction; (2) event argument extraction. We model them by two discriminative log-linear models because of their ability in handling high-dimensional sparse features.

A. EVENT TRIGGER EXTRACTION

The task of event trigger extraction needs to identify all event triggers and classify their event types. The processes of trigger identification and classification are performed in a unified manner, which have been proved to be superior to being handled separately [28]. Specifically, a log-linear classification model is used to classify each trigger candidate into one of candidate event types (including a *None* class to differentiate non-event triggers).

In our event schemas, both verbs and nouns can evoke events. Instead of treating all words as trigger candidates, we build a valid event lexical unit (LU) set by extracting all LUs that evoke event frames from FrameNet, including annotations in exemplars and full texts[6]. Then the event LU set is expanded by using synsets in WordNet and including all their morphological variants. When given a sentence, a word or phrase is considered as a trigger candidate if it is contained by the event LU set. For each trigger candidate, its candidate event types are restricted to those events that include the trigger candidate as a lexical unit. The simple method is able to cover 90.94% of the gold-standard event triggers in the test set. The trigger candidate extraction method not only reduces the label bias of the training data (most are non-event triggers), but also largely reduce the number of candidate event types for each trigger candidate (from 656 to about 10 on average).

For a given sentence $X = \langle x_1, \cdots, x_n\rangle$ with candidate event triggers $t = \langle t_1, \cdots, t_m\rangle$, t_i denotes the i_{th} trigger word and t_i^l denotes its lemma. Let $\mathcal{L}$ denotes the event LU set, $\mathcal{L}_f$ indicates the subset of event LU set which evokes a particular event type f. Let $\mathcal{L}^l$ and $\mathcal{L}_f^l$ denote the lemmatized versions of $\mathcal{L}$ and $\mathcal{L}_f$. The set of candidate event types for t_i is defined as $\mathcal{F}_i = \{f|\forall f, t_i^l \in \mathcal{L}_f^l\} \cup \{None\}$ (*None* indicates a non-event class). We seek a list of event types $f = \langle f_1, \cdots, f_m\rangle$ for all target trigger words $t = \langle t_1, \cdots, t_m\rangle$.

For each trigger candidate t_i, we aim to find the best event type from candidate event types $\mathcal{F}_i$ by:

$$f_i = argmax_{f\in\mathcal{F}_i} P_\theta\left(f|t_i, X\right) \tag{1}$$

We use a log-linear model for the event type classification model:

$$P_\theta\left(f|t_i, X\right) = \frac{exp\left(\theta^T g\left(f, t_i, X\right)\right)}{\sum_{f'\in\mathcal{F}_i} exp\left(\theta^T g\left(f', t_i, X\right)\right)} \tag{2}$$

where g indicates the feature vector for event type classification (described in Table 2) and θ denotes the corresponding feature weights.

We discriminatively train the event type classification model by maximizing the following log-likelihood for training datasets $< X^{(j)}, t^{(j)}, f^{(j)} >$:

$$max_\theta \sum_{j=1}^{d}\sum_{i=1}^{m_j} logP_\theta\left(f_i^{(j)}|t_i^{(j)}, X^{(j)}\right) \tag{3}$$

[6] Excluding the test set in Table 6

TABLE 2. The features for event type identification.

Lexical features	Word and lemma of the trigger word, its parent and children in dependency tree
POS-tag features	part-of-speech tags of the trigger word, its parent and children in dependency tree
Syntactic features	the set of syntactic dependencies of the trigger word; the set of dependency labels of its children; the dependency label connecting it and its parent
Word vector features	Pre-trained 100-dimensional GloVe word vector [29] of the trigger word

All incorporate with the event type being scored.

TABLE 3. The features for event argument extraction.

Entity types	the fine-grained entity types of entity mention s
Ordering features	the relative position of entity mention s with respect to trigger t: before, after or overlap; the distance between nearest word of s and t;
Lexical features	word, lemma, and POS-tag of the first, last and head word of s and trigger word t; the number of words in s;
Syntactic features	The syntactic dependency label between head word of s and trigger t; the syntactic dependency of the first word of s with respect to its head; all syntactic dependents and labels of the head word of s and trigger t; sequence of labeled, directed edges from the head word of s to head word of t.
Word vector features	Pre-trained 100-dimensional GloVe word vector [29] of head word of s and t; the summation and average of word vectors of all words in s; the summation and average of word vectors of all words in t.

All incorporate with the event type f and argument role r being scored.

where d denotes the total number of annotated sentences in training dataset and m_j indicates the number of targets in the sentence $X^{(j)}$.

B. EVENT ARGUMENT EXTRACTION

Event arguments are entity mentions which play different roles in an event. Event argument extraction is to identify entity mentions that act as arguments, and classify their argument roles. In this work, we first extract all entity mentions from a sentence as argument candidates, then utilize a unified log-linear classification model to identify event arguments and classify their argument roles.

Given a sentence $X = \langle x_1, \cdots, x_n \rangle$, the set of target event triggers are denoted as $t = \langle t_1, \cdots, t_m \rangle$ and the corresponding event types are denoted as $f = \langle f_1, \cdots, f_m \rangle$. For each event type f_i, let $\mathcal{R}_{f_i} = \{r_1, \cdots, r_{|\mathcal{R}_{f_i}|}\}$ denotes its argument roles. We identify a set of entity mentions S that are argument candidates for filling any role $r \in \mathcal{R}_{f_i}$. All noun phrases extracted from the sentence X are regarded as argument candidates. Texts are preprocessed by Stanford CoreNLP pipeline [31]. Entity mentions are extracted based on dependency parsing tree. For further details of the entity mention extraction, see our previous work [32]. The valid candidate arguments are restricted as the extracted entity mentions. It is able to cover 98.4% of the gold-standard argument mentions in the test set.

As fine-grained entity types play an important role in event extraction [11], we further cluster all extracted entity mentions to obtain their fine-grained entity types (such as "Army" or "President"). Similar with [13], we first use WordNet [33] to generate descriptions of entity mentions. For a given entity mention, its related words, hypernyms and synonyms in WordNet are all used to describe it. Then we perform K-means clustering algorithm based on the generated descriptions for entity mentions. The number of clusters is set the same as in [13]. Finally, the indexes of clusters are used as the entity types of entity mentions. The fine-grained entity types are used as features for both the local event argument extraction model (see Table 3) and the global inference model (see Section V).

For each event trigger t_i with event type f_i and argument roles $\mathcal{R}_{f_i} = \{r_1, \cdots, r_{|\mathcal{R}_{f_i}|}\}$, let A_i denotes the mapping of entity mentions in S to argument roles in $\mathcal{R}'_{f_i} = \mathcal{R}_{f_i} \cup None$ (*None* indicates a non-argument role, used to differentiate non-argument entity mentions). We identify the argument role of each entity mention $s \in S$ by:

$$A_i(s) = argmax_{r_i \in \mathcal{R}'_{f_i}} P_\vartheta (r_i | s, f_i, t_i, X) \tag{4}$$

where $A_i(s)$ denotes the argument role of entity mention s.

We use a conditional log-linear model over candidate roles for each entity mention:

$$P_\vartheta (A(s) = r_k | s, f_i, t_i, X) = \frac{exp\left(\vartheta^T h(s, r_k, f_i, t_i, X)\right)}{\sum\limits_{r' \in \mathcal{R}'_{f_i}} exp\left(\vartheta^T h(s, r', f_i, t_i, X)\right)} \tag{5}$$

where h indicates the feature vector for argument role identification (described in Table 3) and ϑ denotes the corresponding feature weights.

We train the argument identification model by maximizing the log-likelihood of training dataset $\langle X^{(j)}, t^{(j)}, f^{(j)}, A^{(j)} \rangle$ as following:

$$max_\vartheta \sum_{j=1}^{d} \sum_{i=1}^{m_j} \sum_{s \in S} logP_\vartheta \left(A_i^{(j)}(s) | f_i^{(j)}, t_i^{(j)}, \mathcal{R}'_{f_i}, X^{(j)}\right) \tag{6}$$

where $A_i^{(j)}(s)$ indicates the annotated argument role for entity mention s.

V. GLOBAL INFERENCE MODEL

Our new event schemas contain many useful structural properties, including the hierarchical structure of event schemas, event-to-event relations, argument role mappings and event scenarios. These structural properties are useful global information for both event trigger identification and event argument identification. However, they are hard to be represented as simple features in the local event extraction model. Furthermore, the local model which extracts event triggers and arguments in two stages will unavoidably suffers from error propagation. So, we propose a global inference model based on MLN [4] to jointly learn event trigger identification and argument identification by encoding the global information across multi-event, multi-argument and event-argument.

A. MARKOV LOGIC NETWORK

Markov Logic Network (MLN) [3], [4] is a Statistical Relational Learning language based on First Order Logic and Markov Networks, tending to unify logic and probability. It can be seen as a formalism that extends First Order Logic to formulae that can be violated with some penalty. A MLN is actually a set of weighted first-order logic formulae $\{(l_i, w_i)\}$, where w_i is the weight associated with formula l_i. These weighted first-order logic formulae define a probability distribution over sets of grounded predicates:

$$P(y) = \frac{1}{Z} exp\left(\sum\nolimits_{(l_i,w_i)\in\mathcal{M}} w_i N\left(l_i, y\right)\right) \tag{7}$$

where y is a grounding world (assignment on every predicate), $N(l_i, y)$ is the number of groundings of l_i that evaluates to True in y, and Z is a normalization constant.

The key inference task over MLN is to compute the most probable explanation given evidence, which can be formulated as:

$$\hat{y} = argmax_y P\left(y\right) = argmax_y \sum_{(l_i,w_i)\in\mathcal{M}} w_i N\left(l_i, y\right) \tag{8}$$

B. GLOBAL CONSTRAINTS

Our global inference approach is based on three types of global constraints: multi-event joint learning constraints, multi-argument joint learning constraints and event-argument joint learning constraints. All constraints are represented in the form of first-order logic formulas for MLN-based global inference. Predicate $eventType(t, f)$ is defined to indicate that trigger candidate t evokes an event of type f. Predicate $ArgumentRole(s, t, f, r)$ is also defined to indicate that entity mention s acts as the argument role r of event trigger t with event type f. They are the only two target predicates in our model, whose assignments are not given during inference and thus need to be predicted. All other predicates are observed predicates whose truth values are always known during inference. Table 4 lists all the predicates used for designing the following three types of global constraints.

1) MULTI-EVENT JOINT LEARNING CONSTRAINTS

Events in the same sentence or document tend to be related with each other according to the *One Sense Per Discourse* theory [8], [34]. The hierarchical structure of event schemas, the event-to-event relations and the structure of event scenarios all describe different kinds of relations between events, which are helpful for event trigger extraction. For example, the following sentence contains two event mentions "left" and "go":

*Example 1: He **left** the company, and he planned to **go** home directly.*

The ambiguous word *"left"* can trigger events with several different event types, such as *"Quitting"* (an employee voluntarily left the service of an employer) and *"Departing"* (a person left a place). It is difficult to tell which event *"left"* triggers if only consider the first clause *"He left the company"*. Since the triggers *"left"* and *"go"* are in the same sentence, which should convey related and consistent information. Based on the inter-dependency, it is easy to determine that *"left"* evokes a *"Departing"* event and *"go"* evokes a *"Motion"* event, since *"Departing"* and *"Motion"* are more relevant in semantics (they are related by *"Using"* relation in the event schemas hierarchy).

To model the dependencies between events with different relations, we define the following observed predicates:

- *Inheritance*(f_1, f_2) is true when event types f_1 and f_2 are connected by *Inheritance* relation;
- *Subframe*(f_1, f_2) is true when event types f_1 and f_2 are connected by *SubFrame* relation;
- *Using*(f_1, f_2) is true when event types f_1 and f_2 are connected by *Using* relation;
- *SameParent*(f_1, f_2) is true when event types f_1 and f_2 have the same parent in the event schema hierarchy;
- *SameScenario*(f_1, f_2) is true when event types f_1 and f_2 belong to the same event scenario.

Besides the above relations, some events usually co-occur with each other, which also presents semantic relations between them. For example, an *"Attack"* event is very likely to co-occur with *"Death"* events and *"Cause_harm"* events.

We design six different first-order logic formulas (i.e. $F_1 \ldots F_6$) to capture these kinds of global dependencies between events, as shown in Table 5. Our training instances for multi-event global inference consists of all pairs of trigger candidates that co-occur in the same sentence or in different sentences that are connected by a coreferent subject/object.

2) MULTI-ARGUMENT JOINT LEARNING CONSTRAINTS

The role mappings in our event schemas describe the relations between arguments of related events, as shown in Fig. 4. The arguments with role mappings normally have consistent entity types and share a set of features in common. In other words, entity mentions that have consistent entity types normally participate in related events as similar roles. For example, considering the following sentences:

*Example 2: Then Jon-O's forces **ambushed** them on the left flank from a line of low hills.*

*Example 3: A powerful Muslim force was **besieging** his city of Tiberias.*

TABLE 4. Description of predicates in our MLN.

Type	Predicate	Description
Target predicates	$eventType(t, f)$	the event trigger t has an unique event type f
	$ArgumentRole(s, t, f, r)$	the entity mention s acts as the argument role r of event trigger t with event type f
Observed predicates	$EntityType(s, c)$	the entity mention s has fine-grained entity type c
	$Inheritance(f_1, f_2)$	event types f_1 and f_2 are connected by *Inheritance* relation
	$Subframe(f_1, f_2)$	event types f_1 and f_2 are connected by *SubFrame* relation
	$Using(f_1, f_2)$	event types f_1 and f_2 are connected by *Using* relation
	$SameParent(f_1, f_2)$	event types f_1 and f_2 have the same parent in the event schema hierarchy
	$SameScenario(f_1, f_2)$	event types f_1 and f_2 belong to the same event scenario
	$Mapping(r_1, f_1, r_2, f_2)$	event types f_1 and f_2 are related and their argument roles r_1 and r_2 are mapped with each other
	$Overlap(s_1, s_2)$	entity mentions s_1 and s_2 are overlapped (i.e. share common words in a sentence) with each other

TABLE 5. First-order logic formulas in our MLN.

	ID	Formula	Weight
Multi-event	F_1	$eventType(t_1, +f_1) \wedge Inheritance(f_1, f_2) \wedge eventType(t_2, +f_2)$	w_1
	F_2	$eventType(t_1, +f_1) \wedge Subframe(f_1, f_2) \wedge eventType(t_2, +f_2)$	w_2
	F_3	$eventType(t_1, +f_1) \wedge Using(f_1, f_2) \wedge eventType(t_2, +f_2)$	w_3
	F_4	$eventType(t_1, +f_1) \wedge SameParent(f_1, f_2) \wedge eventType(t_2, +f_2)$	w_4
	F_5	$eventType(t_1, +f_1) \wedge SameScenario(f_1, f_2) \wedge eventType(t_2, +f_2)$	w_5
	F_6	$eventType(t_1, +f_1) \wedge eventType(t_2, +f_2)$	w_6
Multi-argument	F_7	$EntityType(s_1, +c_1) \wedge EntityType(s_2, +c_2) \wedge Mapping(r_1, f_1, r_2, f_2)$ $\wedge ArgumentRole(s_1, t_1, f_1, +r_1) \wedge ArgumentRole(s_2, t_2, f_2, +r_2)$	w_7
	C_1	$Overlap(s_1, s_2) \wedge ArgumentRole(s_1, t, f, r_1) \Rightarrow \neg\exists r_k ArgumentRole(s_2, t, f, r_k)$	$+\infty$
	C_2	$ArgumentRole(s_1, t, f, r) \Rightarrow \neg\exists s_2 ArgumentRole(s_2, t, f, r)$	$+\infty$
Event-argument	F_8	$eventType(t_1, +f_1) \wedge ArgumentRole(s, t_1, f_1, +r_1) \wedge EntityType(s, +c)$	w_8

The "+" symbol indicates that different groundings of the formula with respect to the corresponding variable may have different weights.

Example 4: The People 's Liberation Army ***stormed*** *central Peking with the loss of hundreds of lives.*

The target word *"ambushed"* triggers *"Attack"* event, *"besieging"* triggers *"Besieging"* event and *"stormed"* triggers *"Attack"* event. Their arguments (underlined phrases) with the same role *"Assailant"* have the same fine-grained entity type (all about *"Army"*). With the inter-dependencies between arguments of related events, the prediction of argument roles can be jointly learned cross document-level. We model it by the rule F_7 shown in Table 5, where $Mapping(r_1, f_1, r_2, f_2)$ is an observed predicate defined to indicate whether the argument role r_1 of event f_1 and argument role r_2 of event f_2 are mapped with each other. And $EntityType(s, c)$ is defined to indicate that the entity mention s has fine-grained entity type c.

Besides the above joint inference rules, we further introduce two consistency constraints to improve the performance of event argument extraction. Observed predicate $Overlap(s_1, s_2)$ is defined to indicate whether entity mentions s_1 and s_2 overlap (i.e. share common words in a sentence) with each other. The first constraint (C_1) restricts that two entity mentions overlapping with each other cannot be arguments of the same event, which is consistent with 97.3% of the role instances in the FrameNet 1.7 full text annotations. The second constraint (C_2) restricts that each argument role of an event can have at most one overt argument, which is consistent with 96.7% of the role instances in the FrameNet 1.7 full text annotations. Both the two constraints are set as hard constraints by setting their weights as infinite, as shown in Table 5.

3) JOINT EVENT-ARGUMENT LEARNING CONSTRAINTS

The fine-grained entity types are helpful for determining both the event type of a trigger candidate and the argument roles of entity mentions. For example, considering the following sentences with an ambiguous word *"fired"*:

Example 5: He has ***fired*** *his air defense chief.*

Example 6: In Baghdad, a cameraman died when an American tank ***fired*** *on the Palestine Hotel.*

The above two example sentences show two different types of events triggered by *"fire.v"*. Since *"fire.v"* can trigger *"Firing"* event (be dismissed from a job) and *"Shoot_projectiles"* event (shooting with a weapon), so we can only determine its type by its context information. In Example (5), the argument *"air defense chief"* means a job-title, which indicates that *"fired"* is more likely to trigger a *"Firing"* event. In Example (6), the argument *"an American tank"* describes a weapon, so the corresponding trigger word is more likely to trigger a *"Shoot_projectiles"* event.

To capture the inter-dependencies between event type, argument role and entity type, we import a first-order logic formula F_8 as shown in Table 5.

C. INFERENCE

The local event extraction model leverages sophisticated local features to generate initial judgments about event trigger identification and argument identification. Based on the initial judgments, the MLN-based global inference model further conducts global inference to make the final judgments on events' types and arguments' roles. Our global

TABLE 6. The annotated dataset for event extraction.

	Exemplar annotations	Exemplar_training_set
#sentences	86962	77508
Full text annotations	Training set	Test set
#Sentences	3402	717
#Events	8079	1755

TABLE 7. Comparison of the coverage of event schema definition with ACE and ERE.

Data	Our System	ACE	ERE
# of Event Types	655	33	38
# of Argument Roles	2050	159	185
# of Distinct Arg. Roles	604	28	26

inference model encodes the output of the local model as prior knowledge in the form of soft formulas, which can be represented as: $eventType(t, f)$ with weight $P_\theta\,(f|t, X)$, and $ArgumentRole(s, t, f, r)$ with weight $P_\vartheta\,(A(s)=r|s, f, t, X)$ (X indicates the sentence containing the corresponding event mention).

Then our MLN defines the following probability distribution over sets of grounded predicates:

$$P(y) = \frac{1}{Z} exp\left(\sum\nolimits_{(l_i,w_i)\in\mathcal{M}} w_i N\,(l_i, y) + \Phi\,(y)\right) \tag{9}$$

where $\Phi\,(y)$ represents the prior knowledge on the set of target predicates and is given by: $\Phi(y) = \sum\{\mathbb{I}_y(eventType(t, f))P_\theta\,(f|t, X) + \mathbb{I}_y(ArgumentRole(s, t, f, r))P_\vartheta\,(A(s)=r|s, f, t, X)\}$. $\mathbb{I}_y(x)$ is an indicator function that equals to 1 if x is true in y and 0 otherwise.

To learn the weights of all first-order logic formulas in our MLN structure, we formulate the global inference process as a structured prediction problem and estimate the weights with structured hinge loss:

$$-S(y, x) + max_{y'}\left(S\left(y', x\right) + L\left(y, y'\right)\right) \tag{10}$$

where $S(y, x) = \sum_{(l_i,w_i)\in\mathcal{M}} (w_i N\,(l_i, y) + \Phi\,(y))$ (defined in (9)), y denotes the grounding world of all predicates based on gold event annotations for input document x, y' denotes the corresponding predictions for x, and $L(y, y')$ denotes the number of false event type and argument role predictions by y'. We solve the above problem by transforming it into an integer linear programing. Each predicate is transformed into a binary variable. Each first-order logic formula is transformed into a set of linear constraints. For further details of about transforming a MLN into ILP, see [36], [37]. The weights of our model are learned by training on a corpus of documents paired with annotated events.

VI. EXPERIMENTAL EVALUATION

A. DATASET AND EXPERIMENTAL SETTINGS

Both the annotated exemplars and full text annotations in FrameNet 1.7 corpus are transformed into annotated datasets for event extraction by filtering non-event frames. The exemplar sentences only contain annotations for a single event, while the full text sentences contain annotations for all events. The full text annotations are split into training set and test set. The Exemplar_training_set dataset is constructed by filtering sentences which are also contained by the full text annotations.

The local event extraction model is trained on the combination of the full text training set and the Exemplar_training_set. Note that only the annotated trigger target in each exemplar sentence is used to train the local event extraction model. However, all trigger candidates extracted from each sentence in the full text training set are used for training the local event extraction model. The MLN-based global inference model is trained on the full text training set in order to capture more global features for global inference. The model performance is evaluated on the test set. The details of the datasets are shown in Table 6.

We also conduct experiments on the ACE 2005 corpus. To compare with state-of-the-art event extraction systems on the ACE 2005 corpus, we follow the same evaluation settings in previous work [8], [10], [11] and use 40 newswire documents of ACE as our test set.

B. EVENT SCHEMA DEFINITION

We compare the event schema definition in our work to both predefined ACE and ERE event schemas. ERE was designed as a lighter-weight version of ACE and a simple approach to entity, relation and event annotation. As shown in Table 7, our event schema definition contains 655 event types with 2050 argument roles (604 distinct roles) in total, which is a few orders of magnitude larger than the definitions in ACE and ERE. More definition of event types will make our event extraction system extract more rich event information from text, which are promising to be used in many following NLP tasks, such as summarization, question answering, etc.

We further compare the coverage of event extraction results of our system with several ACE-based state-of-the-art systems on both our test set and the ACE 2005 test set:

- **Structure-Joint**[7]: A structured perception joint model based on symbolic global semantic features [12].
- **JRNN**[8]: A joint event extraction model via recurrent neural network [35].

As our system and the ACE-based models use different event schema definition, we cannot compare the accuracy of event type classification and argument role classification directly. In our experiment, we mainly evaluate the performance of event trigger identification and event argument identification. An event trigger is correctly identified if its offsets match those of a gold-standard trigger; an event argument is correctly identified if its offsets and event trigger match those of any of the reference argument mentions in the document.

The results in Table 8 show that our system can extract most of the human annotated events and arguments in the

[7] https://github.com/Aureliu/BIU-RPI-Event-Extraction-Project
[8] https://github.com/bishanyang/EventEntityExtractor

TABLE 8. Comparison of the coverage of event extraction results on our test set.

Data	#Human	Our MLN-Joint System				Structure-Joint [12]				JRNN [35]			
		#Extracted	Precision	Recall	F1	#Extracted	Precision	Recall	F1	#Extracted	Precision	Recall	F1
Trigger Identification	1755	**1468**	92.51	77.38	**84.27**	245	77.96	10.88	19.10	310	67.74	11.97	20.34
Argument Identification	3281	**2367**	80.99	58.43	**67.88**	220	37.27	2.50	4.68	381	35.43	4.11	7.37

TABLE 9. Comparison of the coverage of event extraction results on the ACE2005 test set.

Data	MLN-Joint	Structure-Joint [12]	JRNN [35]	DMCNN [14]	Liberal [24]
	Recall	Recall	Recall	Recall	Recall
Trigger Identification	**85.38**	79.48	78.30	63.6	50.1
Argument Identification	**47.94**	32.70	23.92	46.9	39.4

Note: As ACE annotations have only annotated parts of events in text, so only the recalls are compared between each system on the ACE 2005 test set.

test set, however, the ACE-based models can only detect very few of them. The results of our system not only obtains much better recall, but also significantly higher precision. The results verify that our event extraction system is able to extract much more rich event information from text.

We also directly test the coverage of event extraction results of our trained model on the ACE 2005 test set. As ACE annotations have only annotated parts of events in text, so only the recalls are compared between each system. The recalls of Structure-Joint and JRNN are computed by the public released models. We also compare our system with one popular neural network-based supervised model and one state-of-the-art unsupervised model:

- **DMCNN**: A dynamic multi-pooling convolutional neural network for event trigger and argument extraction [14].
- **Liberal**: A liberal event extraction framework to extract events and discover event schemas from input corpus automatically [24].

The recalls of DMCNN and Liberal are both taken from [15]. The results in Table 9 show that our model can extract most of the human annotated events and arguments in ACE annotations, which demonstrate that our event schema definition covers the ACE annotations. The recall of our system is even significantly higher than the ACE-based models on the ACE 2005 test set. We analyze the main reason is that our annotated dataset is much larger than ACE annotations, and more effective structure features, such as the hierarchical structure of event schemas and event-to-event relations have been used in our model.

Moreover, our event schema definition is much more fine-grained than ACE and ERE definition. Table 10 shows some examples, including the definition of "Transfer-Money", "Transfer-ownership", "Start-Position" and "End-Position" in ACE, ERE and our system. Obviously, our event definition with hierarchical structure is more fine-grained and semantic richer. Actually, all event types defined in ACE and ERE can be mapped as event types in our event schemas. Furthermore, for each coarse event type, several more fine-grained subtypes are also defined to capture more rich semantic information in our event schemas.

TABLE 10. Examples showing the fine-grained property of our event schema definition.

ACE	ERE	Our System
Transfer-Money	Transfermoney	Commerce_money-transfer -Commerce_collect –Billing –Fining -Commerce_pay –Earnings_and_losses
Transfer-Ownership	Transferowners.	Commerce_goods-transfer -Commerce_buy –Renting –Importing –Shopping -Commerce_sell –Exporting –Renting_out
Start-Position End-Position	startposition endposition	Employee_scenario -Quitting -Get_a_job -Being_employed Employer_scenario -Hiring -Firing -Employing

C. ANALYSIS OF MLN-BASED GLOBAL INFERENCE

To evaluate the performance of event trigger classification and argument role classification, we adopt the evaluation metrics for events as defined in [12]. An event trigger is correctly classified if its offsets match those of a gold-standard trigger and its event type also matches the type of the gold-standard trigger. An event argument is correctly classified if its offsets and event type match those of any of the reference argument mentions in the document and its argument role is also correct.

The comparison results of the event extraction performance of the MLN-based joint model and the local pipeline model on the test are shown in Table 11. The results show that our MLN-based joint model achieves significantly better performance than the local pipeline model, which demonstrates the effectiveness of our proposed joint event extraction model. Based on the prior output of the local pipeline model, the MLN-based global inference is able to improve the event extraction performance through combining the hierarchical

TABLE 11. Overall performance of event extraction on the test set.

		Precision	Recall	F1-score
Trigger Classification	Local Pipeline	82.96	69.39	75.57
	MLN-Joint	**85.69**	**71.68**	**78.06**
Argument Classification	Local Pipeline	72.72	52.46	60.95
	MLN-Joint	**74.86**	**54.01**	**62.75**

TABLE 12. Impact of joint learning rules.

Method	Trigger Classification (%)			Argument Classification (%)		
	P	R	F_1	P	R	F_1
MLN-Joint	**85.69**	**71.68**	**78.06**	**74.86**	**54.01**	**62.75**
w/o F_1	84.89	71.01	77.33	74.39	53.67	62.35
w/o F_2	84.13	70.37	76.64	74.48	53.73	62.43
w/o F_3	85.11	71.19	77.53	74.51	53.75	62.45
w/o F_4	84.92	71.03	77.36	74.53	53.77	62.47
w/o F_5	85.01	71.11	77.44	74.62	53.83	62.54
w/o F_6	83.24	69.63	75.83	74.37	53.65	62.33
w/o F_7	84.02	70.28	76.54	73.17	52.79	61.33
w/o F_8	83.45	69.80	76.02	73.12	52.75	61.29
w/o C_1	85.11	71.14	77.65	74.26	53.31	62.27
w/o C_2	85.12	71.32	77.35	74.37	53.42	62.13

structure of our event schema definition and the event-to-event relations in FrameNet corpus.

To analyze the effectiveness of the first-order logic rules in our MLN, we do several ablation experiments to evaluate and compare the performance of event extraction by removing them during MLN global inference. Results in Table 12 show that the performance of event extraction decreases consistently while removing each first-order logic rule in our MLN. The results demonstrate that each first-order logic rule in our MLN is able to capture useful global features to improve the performance of event extraction.

Table 13 shows several examples that the MLN-based joint model corrects the errors made by the local pipeline model. For example, in example (1), the local pipeline model incorrectly classifies *"renewing"* as an instance of *"Rejuvenation"* and entity mention *"a defense treaty with the U.S"* as an *"Entity"* argument. However, the joint model can incorporate the entity type information of *"a defense treaty with the U.S"* by joint event-argument inference, and obtain the correct event type and argument role predictions. In example (2), the event trigger *"hired.v"* has been incorrectly identified as an instance of event *"Renting"* by the local pipeline model, however, the joint model predicts the correct event type by incorporating global features from context, such as co-occurring event *"work"*, based on multi-event inference. The example (3) also shows the effectiveness of multi-argument learning of our joint model, which correctly identifies entity mentions *"U.S. military aid"* and *"U.S. troops"* as the same argument role of event *"Respond_to_proposal"*. The results demonstrate that our global inference model is able to improve the performance of event extraction by considering more global context information.

VII. APPLICATION ON TEXT SUMMARIZATION

Text summarization is to generate a condensed version of the original documents. The major issues for text summarization are identifying important information and eliminating redundant information. Most of previous methods are extractive, which select several salient sentences to form a summary. Graph-ranking based methods are the most popular unsupervised extractive methods, which build a graph where vertexes are sentences and edges are the similarity between sentences [38]–[40]. However, events are the most basic information units in news text, so we propose to summarize documents based on event extraction results.

A. EVENT RELATEDNESS GRAPH

Based on our hierarchical event schema and event extraction model, we propose an event-ranking based unsupervised extractive MDS methods named **EventRank**. EventRank builds an event relation graph where all events extracted from news text are vertexes (denoted as $E = \{e_i\}$) and the relatedness between events are edges (denoted as $R = \{r_{i,j}\}$, where $r_{i,j}$ indicates the similarity between event e_i and e_j).

The similarity $r_{i,j} \in [0, 1]$ between event e_i and e_j is computed by:

$$r_{i,j} = \lambda_1 sim(t_i, t_j) + \frac{\lambda_2}{K} \sum_{k=1}^{K} \max_{a_j^l \in Args(e_j)} \left(sim(a_i^k, a_j^l) \right) \tag{11}$$

where t_i denotes the event trigger of event e_i, a_i^k denotes the k-th arguments of event e_i, and K denotes the total number of arguments in event e_i. $sim(t_i, t_j)$ denotes the similarity between the event triggers t_i and t_j, which is set as 1 if e_i and e_j have the same event type, otherwise the cosine similarity between pre-trained word embeddings [29] of trigger words. $sim(a_i^k, a_j^l)$ denotes the similarity between arguments a_i^k and a_j^l, which is set as 1 if corresponding entity mentions are coreferent, otherwise the cosine similarity between the average pre-trained word embeddings of argument words. λ_1 and λ_2 are hyper-parameters to tune the relative weights between event trigger and event arguments.

B. EVENT RANKING

Then graph ranking-based methods [30] are used to obtain the saliency scores of events $score(e_i)$ by:

$$score(e_i) = (1-d)\frac{1}{|E|} + d \sum_{e_j \in In(e_i)} \frac{r_{i,j} * score(e_j)}{\sum_{e_k \in Out(e_j)} r_{j,k}} \tag{12}$$

where d represents the damping factor (set as 0.85 as in [38]), $|E|$ denotes the total number of events extracted from text, $In(e_i)$ is the set of vertexes connected to e_i, $Out(e_j)$ is the set of vertexes connected from e_j. The convergence threshold is set as 10^{-5}. Experiments show that (12) usually converges in $20 - 30$ iterations.

The saliency score of a sentence s_t is computed by accumulating the saliency scores of events occurred in that sentence as: $score(s_t) = \sum_{e \in s_t} score(e)$. Then sentences are sorted according to their saliency scores.

TABLE 13. Examples showing the effectiveness of MLN-based global inference.

Example (1)	Mr. Gonzalez also has split with the left in **renewing** *a defense treaty with the U.S* .
Pipeline	a) Event: renewing, Type: **Rejuvenation**, Arguments: *a defense treaty with the U.S*(Entity), *Mr. Gonzalez*(Agent)
MLN-Joint	a)Event: renewing, Type:**Activity_resume**, Arguments: *a defense treaty with the U.S*(Activity), *Mr. Gonzalez*(Agent)
Example (2)	Former Soviet bioweaponeers have been **hired** by Iran to specifically work on its BW arsenal .
Pipeline	a) Event: hired, Type: **Renting**, Arguments: *former Soviet bioweaponeers*(Goods), *Iran*(Lessee); b) Event: work, Type: Being_employed, Arguments: *former Soviet bioweaponeers*(Employee), *its BW arsenal*(Place_of_employment)
MLN-Joint	a)Event: hired, Type:**Hiring**, Arguments: *former Soviet bioweaponeers*(Employee), *Iran*(Employer); b) Event: work, Type: Being_employed, Arguments: *former Soviet bioweaponeers*(Employee), *its BW arsenal*(Place_of_employment)
Example (3)	Mr . Barco has refused *U.S. troops* or advisers but has **accepted** *U.S. military aid* .
Pipeline	a) Event: accepted, Type: **Receiving**, Arguments: *Mr . Barco*(Recipient), *U.S. military aid*(Theme); b) Event: refused, Type: Respond_to_proposal, Arguments: *Mr . Barco*(Speaker), *U.S. troops*(Proposal), *advisers*(Proposal)
MLN-Joint	a) Event: accepted, Type: **Respond_to_proposal**, Arguments: *Mr . Barco*(Speaker), *U.S. military aid*(Proposal); b) Event: refused, Type: Respond_to_proposal, Arguments: *Mr . Barco*(Speaker), *U.S. troops*(Proposal), *advisers*(Proposal)

TABLE 14. Summarization performance on the DUC 2004 dataset in terms of ROUGE.

System	ROUGE-1	ROUGE-2	ROUGE-L
Lead	31.42	6.33	28.12
Coverage	34.48	7.74	29.90
Centroid	35.69	8.62	30.50
TextRank [38]	35.97	8.67	30.68
LexPageRank [39]	34.12	7.09	29.34
ClusterCMRW [40]	38.01	9.36	33.36
Submodular_Lin2010 [42]	38.12	9.28	33.38
Submodular_Li2012 [43]	38.67	9.35	33.91
EventRank(Str.-Joint)	32.08	6.27	29.41
EventRank-part(Str.-Joint)	31.01	5.97	28.11
EventRank(JRNN)	33.17	7.22	30.41
EventRank-part(JRNN)	31.08	6.01	28.15
EventRank(MLN-Joint)	39.12*	9.74*	34.38*
EventRank-part(MLN-Joint)	**39.43***	**10.05***	**35.27***

* indicates that our model achieves significantly (p<0.01) better performance compared with previous unsupervised MDS models.

C. SUMMARIZATION

Finally, a MMR (Maximum Marginal Relevance)-based greedy method is used to extract sentences with salient events to form a summary. Denoting the candidate sentences set as $S = \{s_t\}$, for each step we greedily select a sentence until reaching the length limit:

$$s = \arg\max_{s_t \in S} \left(\gamma score(s_t) - (1-\gamma) \max_{s_h \in P} sim(s_t, s_h) \right) \quad (13)$$

where P denotes the partial selected sentence set in current step, γ is a coefficient parameter to balance between saliency and diversity, $sim(s_t, s_h)$ denotes the similarity between sentence s_t and s_h. We simply set the similarity between sentences as the maximum similarities between events in them.

Instead of selecting the entire sentence, another method **EventRank-part** is designed to extract only parts of sentences which cover the target event mentions. The event selection process is the same as the sentence selection process. For each step, an event with the largest score is selected as in Equation 13, and the part of sentence which covers the event mention is selected into the summary. The selection process stops until the summary reaching the length limit.

D. RESULTS

We evaluate the above simple unsupervised method on the DUC 2004 dataset by the ROUGE1.5.5 toolkit [41]. After tuning on the DUC 2003 dataset, the hyper-parameters λ_1 and λ_2 are both set as 0.5, and γ is set as 0.7 for **EventRank** and 0.65 for **EventRank-part**. We compare our model with most of previous unsupervised methods, including graph-ranking based methods [38]–[40], submodular-based methods [42], [43], as well as several widely used unsupervised baselines (Lead, Coverage and Centroid) [44]. We also set other four baselines (i.e. EventRank(Str.-Joint), EventRank-part(Str.-Joint), EventRank(JRNN) and EventRank-part(JRNN)) which use the same method as EventRank or EventRank-part but use the event extraction results by ACE-based models **Structure-Joint** [12] and **JRNN** [35], respectively. The experimental results in Table 14 show that the simple method EventRank(MLN-Joint) and EventRank-part(MLN-Joint) achieve significantly (p<0.01) better performance than previous state-of-the-art unsupervised extractive MDS methods. In particular, EventRank-part(MLN-Joint) achieves the best performance, which demonstrate that events are better information units than sentences for document summarization. However, the same method with ACE-based event extraction results achieves very poor performance, and the systems which extract only parts of sentences covering the selected events are even worse than their counterparts which select entire sentences. The reason is that the ACE-based models can only extract very limited events from text and drop many salient events which should be included in the summaries. The results demonstrate that our event extraction model is able to extract much more rich event information from news text, which is helpful for document summarization. Our event extraction results can also be used for abstractive document summarization [32].

VIII. CONCLUSIONS

In this work, we proposed to construct a hierarchy of event schemas that is fine-grained and has much wider coverage based on the FrameNet corpus with considerations for practical applications. Based on the defined hierarchical event schemas, an MLN-based joint inference model was proposed to leverage the structure of event schema hierarchy, frame-to-frame relations and several global features to extract events from text effectively. Extensive experiments have verified the effectiveness of both the hierarchical event schemas and the joint event extraction model. The successful application on news summarization demonstrates that our event schemas and event extraction model can extract very rich event information from text, which can be used in many practical applications.

REFERENCES

[1] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The Berkeley framenet project," in *Proc. 17th Int. Conf. Comput. Linguistics*, vol. 1, 1998, pp. 86–90.

[2] C. J. Fillmore, C. R. Johnson, M. R. L. Petruck, "Background to framenet," *Int. J. lexicogr.*, vol. 16, no. 3, pp. 235–250, 2003.

[3] S. Kok and P. Domingos, "Learning the structure of Markov logic networks," in *Proc. ACM 22nd Int. Conf. Mach. Learn.*, 2005, pp. 441–448.

[4] M. Richardson and P. Domingos, "Markov logic networks," *Mach. Learn.*, vol. 62, nos. 1–2, pp. 107–136, 2006.

[5] H. Poon and P. Domingos, "Joint inference in information extraction," in *Proc. AAAI* 2007, pp. 913–918.

[6] D. Venugopal, C. Chen, V. Gogate, and V. Ng, "Relieving the computational bottleneck: Joint inference for event extraction with high-dimensional features," in *Proc. EMNLP*, 2014, pp. 831–843.

[7] D. Ahn, "The stages of event extraction," in *Proc. Workshop Annotating Reasoning Time Events*, 2006, pp. 1–8.

[8] H. Ji and R. Grishman, "Refining event extraction through cross-document inference," in *Proc. ACL-HLT*, 2008, pp. 254–262.

[9] P. Gupta and H. Ji, "Predicting unknown time arguments based on cross-event propagation," in *Proc. ACL-IJCNLP Conf. Short Papers*, 2009, pp. 369–372.

[10] S. Liao and R. Grishman, "Using document level cross-event inference to improve event extraction," in *Proc. 48th Annu. Meeting Assoc. Comput. Linguistics*, 2010, pp. 789–797.

[11] Y. Hong, J. Zhang, B. Ma, J. Yao, G. Zhou, and Q. Zhu, "Using cross-entity inference to improve event extraction," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2011, pp. 1127–1136.

[12] Q. Li, H. Ji, and L. Huang, "Joint event extraction via structured prediction with global features," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2013, pp. 73–82.

[13] S. Liu, K. Liu, A. He, and J. Zhao, "A probabilistic soft logic based approach to exploiting latent and global information in event classification," in *Proc. AAAI*, 2016, pp. 2993–2999.

[14] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, "Event extraction via dynamic multi-pooling convolutional neural networks," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Language Process.*, vol. 1, 2015, pp. 167–176.

[15] T. H. Nguyen and R. Grishman, "Graph convolutional networks with argument-aware pooling for event detection," in *Proc. AAAI*, 2018, pp. 5900–5907.

[16] T. H. Nguyen and R. Grishman, "Event detection and domain adaptation with convolutional neural networks," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Language Process.*, vol. 2, 2015, pp. 365–371.

[17] Y. Chen, S. Liu, S. He, K. Liu, and J. Zhao, "Event extraction via bidirectional long short-term memory tensor neural networks," in *Proc. Chin. Comput. Linguistics Natural Lang. Process. Based Naturally Annotated Big Data*, 2016, pp. 190–203.

[18] Y. Chen, H. Yang, K. Liu, J. Zhao, and Y. Jia, "Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, vol. 2018, pp. 1267–1276.

[19] X. Liu, Z. Luo, and H. Huang. (2018). "Jointly multiple events extraction via attention-based graph information aggregation." [Online]. Available: https://arxiv.org/abs/1809.09078

[20] L. Sha, F. Qian, B. Chang, Z. Sui, "Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 5916–5923.

[21] L. Chen and A. Roy, "Event detection from flickr data through wavelet-based spatial analysis," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 523–532.

[22] M. Cordeiro, "Twitter event detection: Combining wavelet analysis and topic inference summarization," in *Proc. Doctoral Symp. Inform. Eng.*, 2012, pp. 11–16.

[23] A. Ritter, O. Etzioni, and S. Clark, "Open domain event extraction from Twitter," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 1104–1112.

[24] L. Huang *et al.*, "Liberal event extraction and event schema induction," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2016, pp. 258–268.

[25] J. Aguilar *et al.*, "A comparison of the events and relations across ace, ere, TAC-KBP, and framenet annotation standards," in *Proc. 2nd Workshop EVENTS, Definition, Detection, Coreference, Represent.*, 2014, pp. 45–53.

[26] S. Liu, Y. Chen, S. He, K. Liu, and J. Zhao, "Leveraging framenet to improve automatic event detection," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2016, pp. 2134–2143.

[27] Z. Liu, C. Xiong, T. Mitamura, and E. Hovy. (2018). "Automatic event salience identification." [Online]. Available: https://arxiv.org/abs/1809.00647

[28] C. Chen and V. Ng, "Joint modeling for chinese event extraction with rich linguistic features," in *Proc. COLING*, 2012, pp. 529–544.

[29] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. EMNLP*, 2014, pp. 1532–1543.

[30] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the Web," Dept. Comput. Sci., Stanford Univ., Stanford, CA USA, Tech. Rep. 1999–0120, 1999.

[31] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford CoreNLP natural language processing toolkit," in *Proc. ACL, Syst. Demonstrations*, 2014, pp. 55–60.

[32] W. Li, L. He, and H. Zhuge, "Abstractive news summarization based on event semantic link network," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers (COLING)*, 2016, pp. 236–246.

[33] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[34] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proc. 33rd Annu. Meeting Assoc. Comput. Linguistics*, 1995, pp. 189–196.

[35] T. H. Nguyen, K. Cho, and R. Grishman, "Joint event extraction via recurrent neural networks," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 300–309.

[36] L. G. Mojica and V. Ng, "Markov logic networks for text mining: A qualitative and empirical comparison with integer linear programming," in *Proc. LREC*, 2016, pp. 1–8.

[37] S. Sarkhel, D. Venugopal, P. Singla, and V. G. Gogate, "An integer polynomial programming based framework for lifted MAP inference," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3302–3310.

[38] R. Mihalcea and P. Tarau, "TextRank: Bringing order into text," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2004, pp. 1–8.

[39] G. Erkan and D. R. Radev, "LexPageRank: Prestige in multi-document text summarization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2004, pp. 1–7.

[40] X. Wan and J. Yang, "Multi-document summarization using cluster-based link analysis," in *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2008, pp. 299–306.

[41] C. Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proc. Text Summarization Branches Out*, 2004, pp. 74–81.

[42] H. Lin and J. Bilmes, "Multi-document summarization via budgeted maximization of submodular functions," in *Proc. Hum. Lang. Technol., Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2010, pp. 912–920.

[43] J. Li, L. Li, and T. Li, "Multi-document summarization via submodularity," *Appl. Intell.*, vol. 37, no. 3, pp. 420–430, 2012.

[44] J. Zhang, T. Wang, and X. Wan, "PKUSUMSUM: A Java platform for multilingual document summarization," in *Proc. 26th Int. Conf. Comput. Linguistics, Syst. Demonstrations (COLING)*, 2016, pp. 287–291.

WEI LI is currently pursuing the Ph.D. degree with the Institute of Computing Technology, Chinese Academy of Sciences, University of Chinese Academy of Sciences, Beijing, China. He has published several papers on these topics at international conferences and journals, such as EMNLP and COLING. His current research interests include text summarization, information extraction, and natural language processing. He has also severed as a Reviewer for several international conferences, including NAACL, EMNLP, and COLING, and international journals, including the IEEE Access and *Frontiers of Computer Science*.

DEZHI CHENG is currently pursuing the master's degree with the Institute of Computing Technology, Chinese Academy of Sciences, University of Chinese Academy of Sciences, Beijing, China. His research interests include information extraction and natural language processing.

LEI HE is currently pursuing the Ph.D. degree with the Engineering and Applied Science School, Aston University. Her current research interests include information extraction and natural language processing.

YUANZHUO WANG received the Ph.D. degree in computer science from Tsinghua University, Beijing, in 2008. He is currently a Professor with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing. His current research interests include social computing and open knowledge network.

XIAOLONG JIN (M'07) received the Ph.D. degree in computer science from Hong Kong Baptist University, in 2005, and the M.Eng. degree from CAS, in 2001. He is currently a Professor and a Ph.D. Advisor with the Institute of Computing Technology, Chinese Academy of Sciences. He is also the Deputy Secretary-General of the Task Force on Big Data, China Computer Federation (CCF TFBD). His current research interests include knowledge graph, knowledge engineering, and social computing. He is also a Senior Member of CCF and a member ACM.